Applications of Bayesian mixture models and self-exciting processes to retail analytics

James Pitkin

A dissertation submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

of the

University of London.

Department of Statistical Science
University College London

September 6, 2018
Declaration of authorship

I, James Pitkin, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.
Abstract

Retail analytics has been transformed by big data, which has led to many retailers using detailed analytics to improve performance at a range of operational levels. This is the case with the collaborator of this research, dunnhumby, who have large amounts of retailer data derived from the numerous activities that retailers operate at. This thesis focuses on two challenges retailers face; the analysis of products through their price elasticity coefficients and demand forecasting of products known as slow-moving inventory.

The analysis of products in terms of their price elasticity coefficients is well studied. Existing approaches are hampered by the challenging nature of cross-elasticity data, as cross-elasticity coefficients typically vary in dimension and exhibit an inherent censoring. We address these problems by developing a systematic model-based approach by reinterpreting the cross-elasticity coefficients as realisations of variable length order statistics sequences, and develop a nonparametric Bayesian methodology to cluster these sequences. Our approach uses the Dirichlet process mixture model that allows data to dictate the appropriate number of clusters and provides interpretable parameters characterising the decay of the leading entries.

Slow-moving inventory are characterised by having intermittent demand, in that the demand is populated with an abundance of zero sales and that, when a sale does a occur, it is often followed by a quick succession of sales. This demand intermittency inhibits the use of traditional analytics which crucially affects optimal inventory management. To combat this, we represent intermittent demand as a structured multivariate point process which allows for auto- and cross-correlation frequently observed in sparse sales data. Our approach uses a hurdle component to cope with zero sales inflation, the Hawkes process to capture the temporal clustering and a hierarchal structure to pool information across products.

We illustrate our methods on real retailer data, from access granted by dunnhumby.
Impact statement

The UK retail sector is a significant one; during 2017, consumers in the UK spent around £406 billion on retail purchases, with 39p of every £1 being spent in food stores [?]. This scale, along with the proliferation of data sources derived from the services and products that retailers offer to consumers, has meant responsible data science is increasingly being used to identify inefficiencies and opportunities, as well as helping to provide a higher degree of personalisation to consumers than ever before. A company at the forefront of consumer data science is the collaborator of this research, dunnhumby.

This research, supported by the EPSRC, dunnhumby and the Alan Turing Institute, looks to explore the applications that Bayesian nonparametric mixture modelling, excitation processes and hierarchical modelling have to retail analytics. We focused on two specific problems that retailers face: product clustering and intermittent demand forecasting. The first output of this research was a product clustering methodology that used a Dirichlet process mixture model to capture the nuanced structure exhibited by the elasticity coefficients outputted from demand models traditionally used by retailers. The second output was a forecasting methodology, where we demonstrated the effectiveness that information pooling, a discretised Hawkes process and regression covariates had on the issue of time series forecasting of intermittent demand. These methodologies provide refreshing reference points from which other product clustering and intermittent demand forecasting models could be benchmarked. In addition, each of these approaches may have fruitful applications to fields beyond retail analytics that strive to cluster strictly decreasing or increasing censored data, or to time series forecasting where intermittency inhibits the use of traditional methodologies. These investigations led to two paper submissions to leading statistics journals: the Annals of Applied Statistics and Journal of the Royal Statistical Society: Series C.

The insights of this research provide important implications to professional analysts within the retail analytics industry. Our product clustering methodology casts light over the structural
differences among products' sales sensitivities as exhibited through their cross-elasticity coefficients, and how these sales sensitivity differences are split across brands and food categories. These findings are valuable to retailers, as information on product differences can be used to improve promotional activities and help retailers to differentiate themselves from competitors, by using such features to improve customer loyalty campaigns. The forecasting methodology developed during this research afforded greater transparency into the challenging dynamics exhibited by intermittent demand. Our approach allows retailers to more clearly understand the predictive benefits that hierarchical modelling, seasonality, price and temporal excitation have in intermittent demand. This benefits retailers by allowing them to manage their supply of inventory more optimally by improving the short-term demand forecasts of products, which in turn enables retailers to reduce operational costs associated with stockpiling. Furthermore, it gives retailers the ability to make accurate assessments of the effects that promotions, price changes and marketing campaigns would have on aspects such as profit and revenue, that inaccurate forecasting methodologies are unable to do.
Acknowledgements

I would like to thank the EPSRC, the Centre for Doctoral Training in Financial Computing & Analytics and the Alan Turing Institute for the financial support that made my PhD possible. I am especially grateful to Professor Treleaven, who believed in the project from the outset and whose advice and contacts were valuable throughout my work. I am also grateful to our industrial collaborator dunnhumby, both for the further financial support received and for the range of exciting projects made available to us through this partnership. Particular mention goes to Rosie Prior, Dr. Marco Brambilla, Dr. David Hoyle, Piers Stobbs, and Giles Pavey, all of whom provided valuable insights into our work and who facilitated an innovative research environment. Furthermore I would like to thank my supervisors, Dr. Ioanna Manolopoulou and Dr. Gordon Ross, who have been hugely influential on this work and who have both been a pleasure to work with. I have learned so much from you both.

On the social front, I want to thank Tristan, Phil and Hugh for the lively debates and, importantly, people to play squash with. Special thanks goes to Anthony & Jean Levy, who were both a home away from home and on numerous occasions listened to me ramble at length about many topics. Needless to say, so much thanks goes to Emma, who has been so supportive, caring, and amusing throughout my endeavours. I only hope I can be the person who you were for me one day. Finally, Mum & Dad - thanks for everything.
##Contents

1 Introduction
  1.1 Clustering methodologies .................................................. 11
    1.1.1 Clustering cross-elasticity coefficients ........................... 11
  1.2 Demand forecasting .......................................................... 12
    1.2.1 Forecasting intermittent demand of slow-moving-inventory ....... 13

2 Bayesian inference
  2.1 The statistical model paradigm ............................................ 15
    2.1.1 The Bayesian method .................................................. 16
  2.2 MCMC algorithms ............................................................. 17
    2.2.1 Metropolis Hastings .................................................. 18
    2.2.2 Gibbs sampling ....................................................... 19
    2.2.3 Hamiltonian Monte Carlo ............................................. 20
  2.3 Implementation details .................................................... 21

3 Bayesian Nonparametrics
  3.1 Bayesian Nonparametric mixture models ................................. 23
  3.2 Notation & mathematical background .................................... 24
  3.3 Dirichlet distribution ...................................................... 25
  3.4 Dirichlet process ............................................................ 28
    3.4.1 Blackwell-MacQueen urn scheme .................................. 30
    3.4.2 Sethuraman stick-breaking construction ........................... 31
  3.5 Dirichlet process applications .......................................... 31
    3.5.1 Dirichlet process mixture model .................................. 32
    3.5.2 Density estimation ................................................... 33
  3.6 Posterior inference ......................................................... 35
4 Elasticity clustering & related methodologies

4.1 Elasticity clustering background

4.1.1 PriceStrat and cross-elasticity output

4.2 Analysis of cross-elasticity coefficients

4.2.1 Current approaches

4.2.2 Shortcomings of existing analyses

5 Elasticity clustering using Dirichlet process mixtures

5.1 Relevant distributions

5.1.1 Order statistics of continuous distributions

5.1.2 Exponentiated Weibull distribution

5.1.3 EW distribution application to order statistics

5.2 Model

5.2.1 Nonparametric mixture model of variable length order statistic sequences

5.3 Posterior inference

5.3.1 Sample from \( p(\theta_i | \theta_{-i}, \nu, x_i) \)

5.3.2 Sample from \( p(\alpha^*, \beta^*, \lambda^*, w^* | \nu, x_{(i\in C_i=k)}) \)

5.3.3 Sample from \( p(\nu | \theta_1, ..., \theta_N, N^*, X) \)

5.4 Results

5.4.1 Simulated data 1

5.4.2 Simulated data 2

5.4.3 Prior distributions and posterior sampling

5.4.4 Retail analytics dataset

5.5 Summary & future work

6 Slow-moving-inventory & related methodologies

6.1 Slow-moving inventory background

6.1.1 Challenges of slow-moving inventory forecasting

6.2 Analysis of slowing-moving-inventory forecasting

6.3 Zero-modified distributions

6.3.1 Hurdle models

6.4 Temporal point processes

6.4.1 Point processes background

6.4.2 Hawkes process

6.4.3 Multivariate Hawkes Process

6.4.4 Discretised Hawkes process
List of Figures

3.1 Simulated DP measures produced from the urn process of (3.7) for 50 iterations, along with the corresponding densities produced from a mixtures of normal kernels $f(\cdot | \theta, 0.5)$ when these measures are used as priors over $\theta$. These measures and mixture of densities are produced over various $\nu$ values. The R code producing these plots is included in A.1. 34

4.1 Various plots of $\eta_i$ data 46

4.2 Various plots of $\eta_i$ data 47

5.1 EW density for $(\alpha, \beta, \lambda) = (1.2, 0.8, 1.0)$ [black solid], $(1.55, 0.8, 1.0)$ [blue dashed], $(0.24, 5.0, 1.0)$ [red dotted] and $(1.8, 1.4, 0.5)$ [green dashed-dotted lines] respectively. 56

5.2 Left panel: realisations of order statistics sequences with $EW(\alpha, \beta, \lambda)$ kernel for combinations $(\alpha, \beta, \lambda) = (0.2, 0.6, 0.7)$ [black solid], $(0.5, 1.5, 1.5)$ [blue dashed], $(4, 5, 1.5)$ [red dotted] respectively. Right panel: Density plots of $f(k|x)$ with $EW(0.5, 1.5, 1.5)$ kernel for orders $k=10$ [dotted], $9$ [dashed] and $8$ [solid]. 56

5.3 Posterior probability cluster co-membership probability heatmap, histograms and density estimates for $N^*, l$ and a few order statistics of simulated data (5.19). 67

5.4 Posterior probability cluster co-membership probability heatmap, histograms and density estimates for $N^*, l$ and a few order statistics of simulated data (5.20). These density plots demonstrate the EW kernel successfully describing the mixture of decay sequences. Although subfigure 5.4(b) indicates that the sampling procedure is often producing four occupied clusters, seemingly counter to subfigure 5.4(a)’s three clusters, we note that DP inference can produce partitions consisting of a few singleton clusters consisting of one datapoint. This is the case here, on the simulated mixture of Gamma distributions (the challenging study), where the inferential procedure largely achieves its task of good density estimates and overall partitions, but spends some time producing singleton clusters. 69
List of Figures

5.5 Histograms of the number of observed entries in each cross-elasticity vector, as well as the top two entries of the cross-elasticity vectors, with corresponding density estimates from our model. The censored entries (corresponding to 0 elasticities) have been omitted from the histograms. 70

5.6 Heatmaps of pairwise posterior distributions of the parameters. The scales have been specifically selected here to produce interpretable heatmaps. It should be noted, that although three distinct clusters have been produced as a result of our inferential procedure, the smallest cluster (≈ 6% of snacks category) is only vaguely noticeable at this current scale. Even at scales where the smallest cluster could in theory be visible, it is difficult by-eye to pickup on a ≈ 6% cluster through a heatmap due to its low density nature. 73

5.7 Left panel: histogram of $N^*$. Right panel: heatmap of cluster co-membership probabilities (re-grouped with respect to $C_{LG}^*$). 73

5.8 Histograms of AC samples with the 2.5%, 97.5% quantiles (red-dashed lines) and AC (solid black line) for each cluster induced by $C_{LG}^*$. 74

5.9 Trace plots of MCMC samples for unique atoms of $(\sqrt{\alpha}, \sqrt{\beta}, \sqrt{\lambda}, w)$ parameters and $N^*$ on dunnhumby’s cross elasticity data of the snack category. As discussed in section 5.3.2, for good overall mixing across all unique clusters induced by the DP($\nu G_0$), it is important to have good mixing during the Metropolis Hastings phase for each of unique location updates of $(\alpha_k^*, \beta_k^*, \lambda_k^*, w_k^*)$ for $k = 1, \ldots, N^*$ (the second step of the inferential procedure outlined in 5.3). As a consequence, special attention should be paid to the Metropolis Hastings acceptance rate from sampling $p (\alpha^*, \beta^*, \lambda^*, w^* \mid \nu, x_{\{i : C_i = k\}})$ across each of the unique clusters $k = 1, \ldots, N^*$. Depending on the modelling context, more sophisticated MCMC methodologies, such as adaptive Metropolis Hastings may be appropriate to produce optimal mixing rates across each of the unique clusters $k = 1, \ldots, N^*$. 75

6.1 Sales plots (solid black line) for four tablets with their respective log prices in £ (dashed blue) over a subset of 364 days of demand data. The shaded region is the 30 days prior to the 25th of December - a seasonal period typically associated with higher demand. The first three panels are of low volume tablets (i.e. lower number of sales) and the final forth panel is a high volume tablet (i.e. higher number of sales). 81
6.2 Plots of the count process $N(t)$ (as indicated by the solid blacks dots) and temporal point process $\{t_1, t_2, \ldots\}$ (as indicated by the sequence of vertical blue dotted lines) induced from the homogeneous Poisson process with conditional intensity function $\lambda(t) = \lambda_0 = 0.05$. ................................................................. 92

6.3 Plots of the count process $N(t)$ (as indicated by the solid blacks dots) and temporal point process $\{t_1, t_2, \ldots\}$ (as indicated by the sequence of vertical blue dotted lines) induced from the Hawkes process with conditional intensity function $\lambda(t) = 0.05 + 0.3 \sum_{t_j < t} e^{-(t-t_j)}$. ................................................................. 92

6.4 Simulated example. Two series of samples are generated from $E_t \sim \text{Bernoulli}(p_t)$, with logit$(p_t) = \theta + \kappa \sum_{j < t} E_j g(t - j \mid \mu, \tau)$ for $t = 1, \ldots, 364$ where $g(\cdot \mid \mu, \tau)$ is the negative binomial density on the positive integers with mean and scale $\mu, \tau$. The blue dots are $E_t$ samples generated from $(\theta, \kappa, \mu, \tau) = (-3.2, 3.1, 1.0, 5.0)$ and the solid blue line is the corresponding $p_t$. The maroon dots are $E_t$ samples generated from $(\theta, \kappa, \mu, \tau) = (-2.5, 5.5, 60)$ and the solid maroon line is the corresponding $p_t$. We observe how the differing $(\theta, \kappa, \mu, \tau)$ lead to different clustering patterns and the underlying shape of the probability of seeing events. ............. 95

7.1 Plots of demand series (solid black) for two tablets with their respective log(prices) in £ (dashed blue) over 364 days of training data. The left and right panel demand is a high volume and low volume tablet respectively. The shaded region is the month prior to Christmas. ................................................. 98

7.2 Plots of tablet sales across two brands over proportions of the training set. The left plot corresponds to the GADGET brand and the right plot to the TECHY brand. For each plot, the differing colours correspond to the sales of a particular product within the given brand. The model proposed in subsequent sections (in-particular section 7.2.2), will incorporate the cross-excitation dynamic that the sales of products within the GADGET brand can trigger further sales of differing products within the same GADGET brand (and likewise for the TECHY brand). 99

7.3 Plots of the predictive models HB$^+$ (left plot) and HBE$^+$ (right plot) for product $i = 13$ over a portion of the test set. The blue and magenta dots represent self and cross event days respectively (i.e. $E_{it}$ and $\tilde{E}_{it}$). The black line is the estimated posterior mean of an event day observation (i.e. $p_{it}$) and the shaded region is the 95% credible interval of these estimates. ................................................. 111
7.4 Plots of the predictive models BE\textsuperscript{c} (left plot) and HBE\textsuperscript{c} (right plot) for product $i = 11$ over a portion of the test set. The blue and magenta dots represent self and cross event days respectively (i.e. $E_{it}$ and $\tilde{E}_{it}$). The black line is the estimated posterior mean of an event day observation (i.e. $p_{it}$) and the shaded region is the 95% credible interval of these estimates. 

7.5 Histograms of $\sum_{t: y_{it} = k} y_{it}$ with corresponding 95% credible intervals of the posterior predictive distributions for models BE\textsuperscript{c} (left plot) and HBE\textsuperscript{c} (right plot) for product $i = 12$. The lower of 2.5% credible interval (the lower bound of the whisker bars) for $\sum_{t: y_{it} = 1} \tilde{y}_{it}$ will at best be $\sum_{t: y_{it} = 1} 1$. This is since the count distribution is lower bounded by 1. 

7.6 Histograms of $\sum_{t: y_{it} = k} y_{it}$ with corresponding 95% credible intervals of the log of the posterior predictive distributions of $\sum_{t: y_{it} = k} \tilde{y}_{it}$ (sale counts) for models HB\textsuperscript{c} (left plot) and HBE\textsuperscript{c} (right plot) for product $i = 2$. 

7.7 Plots of the combined models HBE\textsuperscript{c} and HBE\textsuperscript{c} for product $i = 4$ (left plot) and product $i = 12$ (right plot) over the entire training and test sets. The solid blue lines represent the sales of the respective touchscreen tablets and the black dashed lines are the 95% credible interval of the predictive posterior distribution of the sales counts. The dashed-dotted vertical black line at $t = 365$ represents the end and start of the training and test sets respectively.

B.1 Box plots of count process parameters $\beta_{81}, \beta_{82}$ from models BE\textsuperscript{c}, HBE\textsuperscript{c} and FBE\textsuperscript{c} with histograms of $\sum_{t: y_{8t} = k} y_{8t}$ with 95% credible intervals of the posterior predictive distributions for models HBE\textsuperscript{c} and BE\textsuperscript{c}. These demonstrate model BE\textsuperscript{c} being penalised compared to models FBE\textsuperscript{c} and HBE\textsuperscript{c} that allow information pooling. The box plot of $\beta_{81}, \beta_{82}$ estimates for model BE\textsuperscript{c} are different to those of models HBE\textsuperscript{c} and FBE\textsuperscript{c}, which leads to poor predictions for product $i = 8$ in model BE\textsuperscript{c}. These discrepancies arise from sale counts of product $i = 8$ being $y_{8t} < 3$ over the training set. However, over the test interval a sale count 3 is observed. This shows models HBE\textsuperscript{c} and FBE\textsuperscript{c} benefit from having ‘seen’ sale counts $> 2$ from other intermittent demand series that non-hierarchical models cannot.

B.2 Box plots of $\mu_{ci}$ across all products for model HBE\textsuperscript{c}. The $\mu_{ci}$ estimates being greater than 2 indicates the temporal excitation exhibited in that data typically occurs at lags greater than 1.
B.3 Box plots of various count process parameters. This figure shows that parameters across the models HBE\(c\) and FBE\(c\) to be almost identical except for parameter \(\kappa_{11}^c\). The root of this discrepancy derives from the significant excitation exhibited in product \(i = 2\), as demonstrated from the HBE\(c\) estimate of \(\kappa_2^c\) in plot B.3(e). This creates a skew in the shared parameter of \(\kappa^c\) in model FBE\(c\), a skew that does not exist in the \(\kappa_{11}^c\) of HBE\(c\). This lack of heterogeneity in model FBE\(c\) reduces the predictive accuracy on the test set compared to hierarchical equivalent model of HBE\(c\) .

B.4 Trace plots of selected zero process parameters. .................................................. 136

B.5 Trace plots of selected count process parameters. .................................................. 137
List of Tables

4.1 Ordered elasticity output $\varphi$ and $\eta$ for two fictional products, Bobby’s puffs and Lucan’s Salted crisps. For each product we have columns of order elasticity coefficients $\varphi_i$, $\varphi_i \eta_j$ along with the respective sequences of $\eta_j$, which demonstrates the decreasing nature of data from model (4.4). The number of potential cross competitors is set to $n_i = 6$, although the number of terms censored to 0 differs. Importantly, the set of competitors can differ for each of the products and in instances where there is a shared competitor (as with Supermarket puffs in this case), the value of $\varphi_i \eta_j$, as well as its position in the ordering, need not be consistent across products. ................................................................. 45

5.1 EW density behaviours for various combinations of $\left(\alpha, \beta, \lambda\right)$ ........................................ 55

5.2 Posterior means and (2.5%,97.5%) credible intervals for the parameters $(\alpha, \beta, \lambda, w)$ of each cluster partitions induced by $C_{HC}^*$ from simulated data (5.19). The top and bottom rows show the true parameters and number of observations assigned to each cluster respectively. ......................................................... 68

5.3 Posterior means and (2.5%,97.5%) credible intervals for the parameters $(\alpha, \beta, w)$ of each cluster partitions induced by $C_{HC}^*$ from simulated data (5.20). The top and bottom rows show the true parameters and the number of observations assigned to each cluster respectively ................................. 68

5.4 Retail analytics cluster-wise inference. Posterior means and (2.5%,97.5%) credible intervals for each of the four parameters $(\alpha, \beta, \lambda, w)$ along with other breakdown statistics for each the clusters induced by $C_{LG}^*$. ......................................................... 77

7.1 Summary statistics of SMI demand within tablet category on the training set. The brands have been anonymised with fictitious names for privacy purposes. . . . 98

4.2 lppd$^z_{i,\text{test}}$ and lppd$^z_{i,\text{train}}$ scores of the zero process fits for the models Base$^z_1$, HB$^z$, BE$^z$, HBE$^z$, FBE$^z$BEC$^z$, HBEC$^z$ and FBEC$^z$ and each product. ...................... 110
7.3 \( \text{lppd}^c \) scores of the count process fits for the models Base, HB, BE, HBE and FBE for each product and fitted model. ................................. 115

B.1 Prior formulation of models Base, HB, BE, HBE, FBE, BEC, HBEC and FBEC ................................................................. 130

B.2 Prior formulation of models Base, HB, BE, HBE and FBE. ......................... 135
Chapter 1

Introduction

The key aim of this work is to develop a range of novel statistical methodologies that have applications to problems arising in retail analytics. The retail sector, as defined by the UK Office of National statistics, is an industry where ‘sales of products by retailers are made directly to end consumers, including spending on goods (in store and online) and spending on services’. The British retail industrial sector is a significant one; during 2016 the retail sector contributed £192 billion to UK economic output (11.4% of the total) as measured by Gross Value Added [?] and the value of retail sales at current prices (non-seasonally adjusted) on predominantly food stores grew from 151 billion to 154 billion (1.5% increase) [?]. Within 2016, for every pound spent in the retail industry, 40 pence was spent in food stores.

Many companies comprise the British retail industry as a whole, but especially in the food sector, a large percentage of the supply of food to British households is through a small number of companies who operate numerous supermarket chains throughout multiple regions in the United Kingdom. To illustrate this point, 60% of the market share of grocery stores in the UK between January 2015 to March 2017 was composed of the ‘big four’: Tesco, Sainsbury’s, Asda and Morrisons [?]. To put this scale into context: Tesco during the 2015-2016 financial year reported approximately 79 million shopping trips per week across their over 6700 stores globally [?]. Unsurprisingly, as a consequence of operating at such a transactional and regional scale, the volume of data these companies now collect is large, and increasing. Whether from the increase of retail purchases done online in the UK [?], or the fact that globally 66% of shoppers report to be a member of one or more loyalty programs [?], it generally illustrates the vast resource of information to which retailers now have access.

In recent years, this growth of data that British retailers now collect is part of a larger trend known as ‘big data’. For retailers, the value and insights that data can provide is well established,
as retailers are increasingly deploying teams of analysts and modelling specialists who devote themselves to better understanding how data can be used to find solutions and insights that previously were unobtainable without such a mass of available information. On the cutting edge of this development of retail analytics is the collaborator of this work, dunnhumby, who have used analytics to save £16 million in one year by optimally managing their supply chain during seasonal periods and to improve repeat business by building effective loyalty schemes [??]. In the increasingly competitive landscape of retail, innovative companies are now using ‘big data’ to identify inefficiencies and create a distinct advantage that otherwise would be difficult to achieve.

During this work, we focus on the supermarket sector of the retail industry and explore how clustering and forecasting methodologies can be used to create a competitive advantage. Although it should be noted, that all data this work is based on has been fully anonymised for general research purposes so that no individual shoppers, or any other sensitive data could be identified.

1.1 Clustering methodologies

The scale and complexity of data that retailers now manage has, in instances, led to many conventional methodologies of processing, analysing and interpreting data needing to adapt. One such class of approach used to better understand and interpret data is clustering methodologies, which have now been widely applied to the field of retail analytics. Clustering approaches have been extensively used to improve retailer processes; for example, boost revenue by improving product recommendations [?], improve the stock replenishment of inventory systems [??], reduce computational runtime [?], target marketing efforts more effectively [?] and represent a neat and automated approach of classifying large and complex retailer data that allows for clear, interpretable and actionable results [?]. In short, clustering approaches are widely applied in retail, where they demonstrate utility in terms of predictive performance and allowing data to be more easily interpreted, both of which create an advantage to retailers who have the data to support such initiatives.

1.1.1 Clustering cross-elasticity coefficients

One such area in which retail analytics could benefit from a clustering methodology is the analysis of cross-elasticity coefficients. Retailers frequently implement demand modelling methodologies for a range of purposes, one reason being to better understand how the underlying phenomena at play in the environment, such as a product’s price, its competitors’ prices and seasonal dynamics, impact a product’s demand. One set of phenomena retailers are particularly interested in is how a product’s demand is affected by changes of its own price as well as changes
of its competitors’ prices. This relationship between change in demand and price changes is often expressed through direct and cross-elasticity coefficients. These coefficients link the rate of change in the demand of a product to a change of its own price, and the rate of change in the demand of a product to changes of its competitor’s price, respectively. Broadly speaking, the motivation of analysing products in terms of their direct and cross-elasticity coefficients is two-fold. Firstly, these coefficients are analysed to derive possible insights aimed at aiding public policy with findings often striving to improve dietary trends. Secondly, they are analysed from the retailers’ perspective, where these coefficients are used to form a strategic understanding of how their products demand is affected by competitors’ prices, measure the responsiveness to marketing efforts or generally inform profit maximisation strategies. However, in spite of the analysis of cross-elasticity coefficients being widespread in the literature, there is surprisingly no formal methodology that automates the comparison of the cross-elasticity coefficients in a systematic way. Instead, economists and retail analytics professionals manually analyse the cross-elasticity output in an ad-hoc fashion, where they assess similarity between products by studying the absolute differences between cross-elasticity coefficients, and ignore much of the structurally interesting aspects that cross-elasticity coefficients can exhibit.

The first motivation of this work is the following: we look to develop a methodology that aims to non-parametrically cluster products in terms of their cross-elasticity coefficients in an automated fashion, requiring minimal experimenter intervention as possible. In particular, we use novel Bayesian nonparametric methods that are able to accommodate the complex heterogeneity and inherent structure exhibited in cross elasticity data.

1.2 Demand forecasting

Forecasting the demand of products that retailers offer to their market is well established. The aim of demand forecasting is rooted in retailers trying to balance supply and demand. By being able to anticipate demand inflows, organisations are able to appropriately manage their supply of a product, which allows retailers to avoid opportunity costs of understocking products (loss of potential revenue and customer dissatisfaction), or overstocking products (incurring inventory costs, stock depreciation and the likelihood of making sharp price reductions to shift stock). Consequently, accurate demand forecasting is closely related to effective supply chain and inventory management and consequently, retailers often make concerted efforts to develop methodological approaches that reduce the downsides of inaccurate demand forecasts. Finally, retailers’ interest in demand forecasting stems from a desire to understand the effect that numerous drivers have on the demand of their products. By understanding what drives a product’s
Chapter 1. Introduction

demand, retailers can better ascertain the motivations behind consumer choices, marketing effects and seasonal aspects which crucially allows them to make confident and assured decisions in light of the available information, and invest resources accordingly [???]. Unsurprisingly, demand forecasting is especially important for large retailers who operate many large regional outlets, as the impact of incremental improvements of efficiencies translates to significant gains or losses in revenue. Thus, retailers that use their data to anticipate demand place themselves at distinct advantage compared to retailers who do not.

1.2.1 Forecasting intermittent demand of slow-moving-inventory

Many demand forecasting approaches are successful at achieving their objectives of profit maximisation, improving inventory management or gaining a clearer understanding of the factors impacting a product’s demand. However, forecasting the demand of some products is more difficult than it is for others. One such class of products that are traditionally difficult to forecast the demand for are known as slow-moving-inventory.

The demand patterns of slow-moving-inventory products are generally characterised by being very intermittent, in that there is typically an inflation of zero sales and that, when a sale does occur, it is often followed by a quick succession of sales. The inflation of zeros and burstiness in the intermittent demand of slow-moving inventory make demand forecasting challenging, as it can obfuscate an understanding how the environment, and more specifically the covariate information, impacts the demand signal. Consequently, forecasting models not fully incorporating the nuanced covariate and temporal dynamics of intermittent demand of slow-moving inventory frequently lead to inaccurate forecasts, which in turn inhibits a retailer’s ability to balance supply with demand. Methodologies from fields such as machine learning and statistics have sought to handle the issues arising from intermittent demand forecasting. However, there does not seem to be a unified forecasting methodology that handles the zero-inflation, temporal burstiness and provides an explanation of the underlying phenomena existing in the intermittent demand of slow-moving inventory in a simultaneous fashion.

The second motivation of this work is the following: we strive to develop a forecasting methodology for the intermittent demand of slow-moving inventory that unifies the structural artefacts of hierarchy, auto-correlation, cross-correlation and temporal clustering across multiple intermittent demand series whilst still offering explanatory power. We do this by representing the intermittent demand as a structured multivariate point process which includes a hurdle component for the abundance of zero demand, a Hawkes process to cope with temporal clustering within and across products, and a hierarchal structure to pool information across a large number
of products.

The rest of this work is structured as follows: Chapter 2 provides a background of the Bayesian method and outlines the inferential procedures relevant to this work. Chapter 3 presents the work of Bayesian nonparametric (BNP) models, focusing on Dirichlet process mixture models. Chapter 4 describes in deeper detail retailers’ interest in the analysis of cross-elasticity coefficients and outlines existing work in cross-elasticity analysis. Chapter 5 presents a clustering methodology for the coefficients of cross-elasticity demand models which allows retail analysts to characterise supermarket products in terms of their sales sensitivity. Chapter 6 introduces the challenges related to forecasting the intermittent demand of slow-moving inventory. This chapter goes on to outline the existing work in intermittent demand forecasting, and further describes hurdle regression models and a class of point process known as the Hawkes process. Chapter 7 presents a novel regression model able to forecast the intermittent demand for slow-moving-inventory that uses a Bayesian hierarchical hurdle model with excitation components described by a Hawkes process that captures the temporal dynamics exhibited in the intermittent demand of slow-moving-inventory data. Chapter 8 concludes by reiterating the contribution of this work and further describes the scope for future applications of the Bayesian approach to retail analytics.
Chapter 2

Bayesian inference

This Chapter gives a brief overview of the statistical modelling paradigm known as Bayesian statistics. Section 2.1 describes a general statistical framework relevant to this work along with descriptions and benefits of the Bayesian method. Section 2.2 provides a background on Markov chain Monte Carlo (MCMC) methods as the primary route to statistical inference under the Bayesian framework. The objective of this chapter is to give the reader a practical understanding of the framework underpinning Bayesian statistics and MCMC theory as well as providing a description of the relevant MCMC algorithms employed in this work.

2.1 The statistical model paradigm

The broad aim of statistical inference is to describe a set of observations from some process as draws from some probability model that is itself a representation of the original process. More concretely, suppose we observe a sequence $y_1, y_2, ..., y_n$ of instances which are assumed to be drawn randomly from a sequence of random variables $Y_1, Y_2, ..., Y_n$ with respect to the sample set $\Omega$. We then introduce the notion of probability models, $P_\theta$ (i.e. processes of identical distributional form with respect to some $\theta$), indexed by $\theta \in \Theta$ - where $\theta$ are parameters with respect to some parameter space $\Theta$. Assuming that these $y_1, y_2, ..., y_n$ observations were generated independently and identically from the models $P_\theta$, we can then write the following:

$$Y_i \overset{iid}{\sim} P_\theta, \text{ for } i = 1, ..., n$$ (2.1)

for some $\theta \in \Theta$ [?]. The key objective to statistical modelling is making inferences based on the observed data $D = \{y_1, y_2, ..., y_n\}$ with the constraints of the probability models specified by $P_\theta$. There are two main approaches to statistical inference; the classical and Bayesian approaches. Broadly speaking, the classical approach treats all parameter values $\theta$ as unobserved fixed constants, whereas the Bayesian approach treats $\theta$ as another random variable. During this work, we operate under the Bayesian approach.
2.1.1 The Bayesian method

Under the Bayesian paradigm, the model of (2.1) can be re-expressed as the following:

\[ y_i \overset{i.i.d}{\sim} F(y \mid \theta), \text{ for } i = 1, \ldots, n \]
\[ \theta \sim \pi \]

where \( F(y \mid \theta) \) is the distributional form of \( P_\theta \) and \( \pi \) is the prior distribution of \( \theta \). The key to the Bayesian approach is the specification of the prior \( \pi \) over parameter space of \( \Theta \). The prior \( \pi \) is aimed to represent all prior knowledge of the \( \theta \) values, and supposed to reflect expertise before the outset of an experiment. Consequently, Bayesian statistics is interested in characterising the entire distribution of \( p(\theta \mid D) \), referred to as the posterior distribution, and then deriving relevant statistics from this distribution. The derivation of relevant statistics from \( p(\theta \mid D) \) is known as posterior inference.

Bayesian methodologies have been widely applied to a range of disciplines from spatial weather modelling to the temporal modelling of coal mining disasters \[??\], and are perceived to have useful properties when used to model data. However, for the purposes of this work, the benefits of the Bayesian method are the following:

- **Expressing prior beliefs**: Through the specification of prior distribution \( \pi \), experimenters are able to express their prior beliefs of the likely values of \( \theta \). This allows experimenters to penalise the complexity of fitted statistical models and therefore, can be used as a mechanism to reduce overfitting data \[??\]. The process of overfitting is the situation where a statistical model too closely fits a limited set of data. In addition to penalising complexity, priors can be used to handle issues related to parameter estimation in instances of small sample sizes \[?\].

- **Hierarchical borrowing**: In the situation of hierarchical or multilevel modelling, the Bayesian framework naturally allows information pooling between parameters across the various levels of the model hierarchy. Under the i.i.d assumption, a Bayesian hierarchical model can be expressed as:

\[ y_i \overset{i.i.d}{\sim} F(y \mid \theta), \text{ for } i = 1, \ldots, n \]
\[ \theta \sim \pi(\omega) \]
\[ \omega \sim \Pi \]

where \( \Pi \) is the prior of the hyper-parameters \( \omega \) that parametrise \( \pi \). In this instance, the
hyper-parameters $\omega$ of the prior distribution $\pi$ are themselves random quantities. This information pooling across parameters has been found to offer improvements to model fit and predictive performance in instances where a hierarchical structure exists [??].

- **Expressing uncertainty of experimental quantities:** As quantities in the Bayesian framework are themselves random processes, expressing the uncertainty around parameter inferences is automatically inherited from the Bayesian approach [??]. Consequently, the posterior predictive distribution $p(y^* \mid D)$, where $y^*$ is new predictive data point, is naturally accompanied with uncertainty which crucially allows prediction intervals to be constructed around quantities of interest. This is particularly valuable in contexts such as ours, where experimenters are interested in the likely distribution of outcomes. In these situations, the Bayesian approach has been shown to offer particular utility [??].

Although these points are by no means exhaustive, they are a stylised list of benefits relevant to this work. However, analytical expressions and direct sampling from $P(\theta \mid D)$ is often intractable, which therefore presents challenges. A class of sampling algorithms known as Markov chain Monte Carlo (MCMC) have been developed that can accommodate posterior inference.

### 2.2 MCMC algorithms

The key idea behind MCMC is to simulate a sequence of random variables \{\(\theta_0, \theta_1, \theta_2, \ldots\)\} such that these samples are samples drawn from $P(\theta \mid D)$. One route to generating a sequence of random variables \{\(\theta_0, \theta_1, \theta_2, \ldots\)\} equivalent to samples drawn from $P(\theta \mid D)$ is by constructing a Markov chain whose stationary distribution $\pi$ is precisely the target distribution $P(\theta \mid D)$.

We now denote the distribution $P(\theta \mid D)$ as $\phi(\cdotp)$, and refer to this as the target distribution. Before constructing the appropriate Markov chain whose stationary distribution $\pi$ is $\phi(\cdotp)$, we outline the sufficient conditions that such a Markov chain $T$ needs to satisfy. These conditions are as follows:

**Definition 1.** $T$ is irreducible

A Markov chain $T_{ij}$ is irreducible if for every $i, j$ there exists a finite integer $m_{ij}$ such that $p(T_{m_{ij}} = j \mid T_0 = i) > 0$.

**Definition 2.** $T$ is aperiodic

A Markov chain $T_{ij}$ is aperiodic if for every $i$, $\gcd\{n > 0 : p(T_n = i \mid T_0 = i) > 0\} = 1$ is satisfied.

**Definition 3.** $T$ is positive recurrent
Let \( t_i = \min \{ n \geq 0 \mid T_n = i \} \), then a Markov chain \( T_{ij} \) is positive recurrent if for every \( i \),
\[
E(t_i \mid T_0 = i) < \infty.
\]

Given such a Markov chain \( T \), it will have a unique stationary distribution \( \pi(\cdot) \), and is such that for every \( i,j \in S \)
\[
\lim_{n \to \infty} T^n_{i,j} = \pi_j
\]
where \( T^n_{i,j} = p(T_n = j \mid T_0 = i) \), and \( \pi_j \) is the density of state \( j \) under the distribution \( \pi \) [?]. Importantly, the right hand side of this limit is independent of the initial state \( i \), which therefore indicates the unique stationary distribution is independent of the starting state of the Markov chain.

Having generated \( \{ \theta_0, \theta_1, \theta_2, \ldots \} \) samples from the target distribution \( \phi(\cdot) \), quantities of interest such as posterior mean and variance associated with \( \phi(\cdot) \) can be estimated using Monte Carlo integration. Monte Carlo integration evaluates integrals of the form \( E[f(\theta)] = \int f(\theta) \phi(\theta) d\theta \) by drawing samples \( \{ \theta_t, t = 1, \ldots, n \} \) from \( \phi \), and approximating \( E[f(\theta)] \) as:
\[
E[f(\theta)] \approx \frac{1}{n} \sum_{t=1}^{n} f(\theta_t).
\]

We now present a few MCMC methodologies that allows one to generate samples from the target distribution \( \phi \).

### 2.2.1 Metropolis Hastings

The Metropolis-Hastings (MH) algorithm [?] is a method used to generate samples from some target distribution \( \phi(\cdot) \), and does so by constructing a Markov chain whose stationary distribution \( \pi(\cdot) \) is exactly \( \phi(\cdot) \). The algorithm is as follows:

**Metropolis-Hastings algorithm:**

Given a current state \( \theta_t \) at iteration \( t \), the next state \( \theta_{t+1} \) is selected by first sampling a candidate point \( \theta^* \sim q(\cdot \mid \theta_t) \), where \( q(\cdot \mid \theta_t) \) is some proposal distribution. The candidate point \( \theta^* \) is accepted as the new state \( \theta_{t+1} \) with probability \( \alpha(\theta_t, \theta^*) \) which is given by:
\[
\alpha(\theta_t, \theta^*) = \min \left( 1, \frac{\phi(\theta^*)q(\theta_t \mid \theta^*)}{\phi(\theta_t)q(\theta_t^* \mid \theta_t)} \right)
\]

If the candidate point \( \theta^* \) is accepted, then \( \theta_{t+1} = \theta^* \) and otherwise \( \theta_{t+1} = \theta_t \). The MH algorithm is guaranteed to converge to the stationary distribution \( \pi(\cdot) \) subject to conditions on \( q(\cdot \mid \cdot) \) [?]. This algorithm produces a sequence of \( \{ \theta_0, \theta_1, \theta_2, \ldots \} \) samples such that, once the
first instance $\theta_t$ is sampled from $\pi(\cdot)$, then all subsequent samples $\theta_k$ for $k > t$ will also be from $\pi(\cdot)$, as guaranteed by the positive recurrent condition [?].

The MH algorithm is equivalent to constructing a Markov chain with transition matrix $T_{t,t+1} = p(\theta_{t+1} | \theta_t)$ given by:

$$T_{t,t+1} = q(\theta_{t+1} | \theta_t) \alpha(\theta_t, \theta_{t+1}) + \mathbb{I}_{(\theta_{t+1} = \theta_t)} \left[ 1 - \int q(\theta | \theta_t) \alpha(\theta_t, \theta) d\theta \right].$$

This Markov transition matrix satisfies the conditions of irreducibility, aperiodicity and positive recurrence and has stationary distribution $\phi(\cdot)$. In other words, the Metropolis Hasting algorithm converges to our target distribution $\phi(\cdot)$.

### 2.2.2 Gibbs sampling

An alternative route to posterior inference used in instances when the target distribution $\phi(\cdot)$ is multivariate and has closed-form conditional posterior distributions is Gibbs sampling [?]. Gibbs sampling generates $\theta_t$ samples by sampling each component-wise element of the vector $\theta_t$ from its conditional distribution, subject to keeping the other components of the vector $\theta_t$ fixed to their current values. The algorithm is as follows:

**Gibbs sampling algorithm:**

Take $\theta^i_t$ to be the $i^{th}$ coordinate of vector $\theta_t$ (of dimension $D$) and $\theta^{-i}_t$ be the vector of all coordinates of $\theta_t$ excluding the $i^{th}$ component, i.e. $\theta^{-i}_t = (\theta^1_t, \ldots, \hat{\theta}^i_t, \ldots, \theta^D_t)$. At the $t^{th}$ iteration, the following sequence of samples are taken:

$$\theta^1_{t+1} \sim \phi(\theta^1_t | \theta^2_t, \theta^3_t, \ldots, \theta^D_t)$$

$$\theta^2_{t+1} \sim \phi(\theta^2_t | \theta^1_t, \theta^3_t, \ldots, \theta^D_t)$$

$$\vdots$$

$$\theta^{D}_t \sim \phi(\theta^{D}_t | \theta^1_t, \theta^2_t, \ldots, \theta^{D-1}_t).$$

This produces a sequence of $\{\theta_0, \theta_1, \theta_2, \ldots\}$ samples that converges to the stationary distribution $\phi(\cdot)$. Gibbs sampling is a popular approach to posterior inference in instances when sampling from the conditional $\phi(\theta^1_t | \theta^2_t, \theta^3_t, \ldots, \theta^D_t)$ is manageable. The Gibbs sampler is a special case of the MH algorithm with a proposal distribution $q(\theta_{t+1} | \theta_t) = \phi(\theta^i_t | \theta^{-i}_t)$ [?], and thus guarantees algorithm convergence by satisfying the necessary Markov chain conditions of irreducibility, aperiodicity and positive recurrence.
2.2.3 Hamiltonian Monte Carlo

Another MCMC methodology aiming to efficiently sample \( \phi(\cdot) \) is the Hamiltonian Monte Carlo algorithm (HMC) \([?]\). HMC uses the principles of Hamiltonian dynamics to describe the evolution of a physical system as a function of its state pair \((q, p)\), where \( q \) is the position and \( p \) is the momentum of the system. The system is then defined by:

\[
\frac{dp}{dt} = -\frac{\partial H}{\partial q}, \quad \frac{dq}{dt} = -\frac{\partial H}{\partial p}
\]  

(2.4)

where \( H(q, p) \) is the Hamiltonian function. Frequently, \( H(q, p) \) coincides with the total energy of the system and assumed to take the form \( H(q, p) = U(q) + K(p) \), where \( U(q) = \text{potential energy} \) and \( K(q) = \text{kinetic energy} \). Crucially, (2.4) fully describes the trajectory of motion such that, for time \( t' \), the \( H(q, p) \) defines a mapping from any state \((q, p)\) at time \( t \) to state \((q', p')\) at time \( t' + t \).

The key to connecting Hamiltonian dynamics with MCMC is to construct a Hamiltonian function \( H(q, p) \) in terms of the target distribution \( \phi(\cdot) \). In particular, HMC assumes \( H(q, p) \) takes the form \( U(\theta_t) = -\log(\phi(\theta_t)) \), and typically assumes \( K(p) = -p^T M^{-1} p/2 \) where \( M \) is a symmetric, positive-definite matrix. There are many functional forms that \( K(p) \) can take, for further discussion refer to \([?]\). This produces the following dynamics:

\[
\frac{dp}{dt} = -\frac{\partial U}{\partial \theta}, \quad \frac{d\theta}{dt} = M^{-1} p.
\]

To simulate the evolution of this system it is necessary to discretise time. The dynamics can be approximated with arbitrary precision by solving these differential equations using Euler’s or Leapfrog methods. The HMC algorithm is as follows:

**Hamiltonian Monte Carlo algorithm:**

At the \( t^{th} \) iteration, sample a momentum variable \( p' \sim N(0, M) \). Then with \( p' \) and \( q_t \), simulate Hamiltonian dynamics from:

\[
\frac{dp}{dt} = -\frac{\partial U}{\partial \theta}, \quad \frac{d\theta}{dt} = M^{-1} p
\]

with the leapfrog method or some equivalent method to solve a set of differential equations to produce a final state \((\theta', p')\), and take \((\theta^*, p^*) = (\theta', -p')\). With the proposed state \((\theta^*, p^*)\), a MH step is performed with acceptance probability:

\[
\alpha(\theta, \theta^*) = \min \left( 1, \exp \left( -H(\theta^*, p^*) + H(\theta_t, p_t) \right) \right) = \min \left( 1, -U(\theta^*) + U(\theta_t) - K(p^*) + K(p_t) \right).
\]
If the candidate point is accepted, set $\theta_{t+1} = \theta^*$ otherwise $\theta_{t+1} = \theta_t$. This produces a Markov chain that converges to $\phi(\cdot)$. For further HMC details and review, refer to [?].

### 2.3 Implementation details

The algorithms covered in subsections 2.2.1, 2.2.2 and 2.2.3 all produce sequences of random samples \( \{\theta_0, \theta_1, \theta_2, \ldots\} \) eventually converging to the target distribution $\phi(\cdot)$. We now discuss approaches used to assess convergence and factors affecting the rate of convergence. There are many diagnostic measures used to indicate MCMC convergence, such as the Geweke diagnostic, Heidelberger and Welch statistic and Augmented Dicker-Fuller to mention a few. It is important to note however, that MCMC diagnostic tools are only indicators of MCMC convergence, rather than ‘proving’ convergence. For a deeper review of these MCMC convergence assessments, refer to [?]. During this work, we use the following diagnostic approaches to assess MCMC convergence:

**MCMC diagnostics:**

The core interest is in establishing whether the samples \( \{\theta_0, \theta_1, \theta_2, \ldots\} \) have converged to the stationary distribution $\phi(\cdot)$, and thus, originate from a single distribution. Throughout this work, we use the following criteria to assess MCMC convergence:

- **Trace plots:** These plot the sequence of sampled $\theta_t$ values against $t$ (MCMC iteration index). Once an MCMC algorithm has converged to the stationary distribution, samples should look like they originate from a single distribution. Furthermore, multiple MCMC chains from differing starting values of $\theta_0$ should eventually converge to the same stationary distribution $\phi(\cdot)$.

- **Heidelberger and Welch statistic:** This test is used to determine whether MCMC samples come from a stationary distribution (the null hypothesis). There are two phases to this test:

1) Phase one: the MCMC samples are iteratively tested for stationarity, i.e. we initially test the whole chain for stationarity, if it passes, then we infer the chain has converged (and move on to the second phase). Otherwise, we successively omit the first 10\%, 20\%, \ldots of the samples and test for stationarity until there are less than 50\% of the original samples remaining or the chain has at some point passed one of the stationarity tests. If the chain does not pass any of these iterative stationarity tests, we infer non-convergence and do not progress on to the second phase.
2.3. Implementation details

2) Phase two: Using the proportion of the chain which passed the previous phase, we calculate a 95% credible interval around the mean using the half-width test.

MCMC tuning:
Producing chains that efficiently converge often requires tuning of the parameters associated with the chosen MCMC methodology. In terms of the previously outlined algorithms, this relates to the chosen proposal distribution used in the MH algorithm, burn-in and chain length. In particular, we focus on the following:

- **MH \(q(\cdot | \theta_t)\) proposal:** The choice of \(q(\cdot | \theta_t)\) in the MH algorithm significantly impacts convergence efficiency. Much work has been done in investigating the link between \(q(\cdot | \theta_t)\) and convergence performance [???], but generally speaking, optimal performance is achieved when the transition kernel is similar to the target distribution \(\phi(\cdot)\) [??]. Work has further looked at how incorporating information about the distribution of \(\phi(\cdot)\) into \(q(\cdot | \theta_t)\) can be used to help produce accelerated convergence and reduce correlation between MCMC samples [???].

- **Burn-in:** As discussed earlier, once the first instance \(\theta_t\) is sampled from the stationary distribution \(\pi(\cdot)\), all subsequent samples are also from the stationary distribution. It is however, not always straight-forward to know when this first sample from the stationary distribution is obtained. In theory, depending on the desired level of similarity between \(T_{t,t+1}^n\) and \(\phi(\cdot)\), the first sample can be analytically calculated [??], but this in practice is computationally infeasible. To circumvent this, a burn-in period is taken, i.e. the initial \(t = 1, \ldots, m\) samples are discarded and the remaining samples are assumed to originate from the stationary distribution.

- **Chain length:** Once a chain has converged to its stationary distribution, a decision on how many samples are needed for adequate precision is required. Often, Monte Carlo variance or calculations of the effective sample size based on MCMC chains are performed to assess whether the appropriate number of samples have been obtained.

Throughout this work, Bayesian inference is performed by implementations of 2.2.1, 2.2.2 and 2.2.3. Although the focus of this research is not the inferential procedures used during this work’s retail analytics modelling, particular attention is paid to tuning and MCMC chain convergence, which is assessed in line with the previously discussed points.
Chapter 3

Bayesian Nonparametrics

This Chapter gives an overview of the Bayesian nonparametric mixture modelling paradigm. We provide definitions of parametric and nonparametric models, with particular reference to the Dirichlet process, and discuss the motivations of using nonparametric models when handling complex and non-linear datasets. We further provide various relevant expositions of the Dirichlet process, and briefly mention approaches used for posterior inference for the Dirichlet process.

3.1 Bayesian Nonparametric mixture models

During section 2.1, we introduced the notion of probability models and the broad framework that statisticians interpret data through. To refresh the reader, the model (2.2) specified the Bayesian paradigm of statistical modelling which supposed a sequence $y_1, y_2, ..., y_n$ of instances drawn independently and identically from the probability models $P_\theta$ (with distribution $F(y \mid \theta)$), indexed by parameters $\theta \in \Theta$ (with parameter space $\Theta$), can be expressed as the following:

\begin{align*}
  y_i &\overset{iid}{\sim} F(y \mid \theta), \text{ for } i = 1, ..., n \\
  \theta &\sim \pi
\end{align*}

where $\pi$ is the distribution over parameter space $\Theta$.

We now introduce the notions of parametric and nonparametric models in relation to the model (3.1). We say the model has a parametric prior if $\Theta \subset \mathcal{L}$, where $\mathcal{L}$ is a finite linear spanning set. A set $\mathcal{L}$ is a finite linear spanning set if there exists a $k \in \mathbb{N}$ and elements $l_1, \ldots, l_k$ of $\mathcal{L}$, such that, all elements of $\mathcal{L}$ can be expressed as a linear combination of elements $l_1, \ldots, l_k$ [?]. Consequently, parametric models bound the dimension of the solution space, irrespective of the number of samples being modelled. This can lead to model misfit when data is characterised by heterogeneity beyond probability models with a finite parameter space.
Motivated by the shortcomings of parametric models at handling complex data, nonparametric models can be devised to circumvent these aforementioned issues. We define a nonparametric model as one with an infinite dimensional parameter space \( \Theta \), and for such a parameter space \( \Theta \), there exists no \( k \) and corresponding elements of \( \Theta \) such that these elements span \( \Theta \). Thus, in order for model \((2.2)\) to be Bayesian nonparametric, it is necessary to specify an infinite dimensional prior over the space \( \Theta \). One of the main strengths of nonparametric models over parametric models is their capability of capturing any distribution of the data. The caveat of this flexibility is the computational cost of requiring more data to adequately infer the correct model structure. This can be intuited from the following example; suppose one fits a simple linear regression to data. In this situation, inference for such a model requires less data as one borrows strength from the assumption the data lies on a straight line. In the case of a nonparametric model, no such linearity assumption is made and consequently, linearity is inferred from the data. This results in nonparametric methods requiring more data.

A popular branch of Bayesian nonparametrics (BNP) models are Bayesian nonparametric priors. Bayesian nonparametric priors can be thought of as the prior over probability measures, i.e. the distribution over distributions. Such nonparametric priors have useful applications to mixture modelling. Extending \((2.2)\) to a mixture model can be expressed hierarchically as:

\[
y_i \mid \theta_i \overset{\text{i.i.d.}}{\sim} F(y \mid \theta_i), \quad \text{for } i = 1, \ldots, n
\]

\[
\theta_i \mid G \overset{\text{i.i.d.}}{\sim} G
\]

\[
G \sim G_0
\]

(3.2)

where \( G_0 \) is some nonparametric prior over countable measures, and \( G \) is an instance of such a measure. The Dirichlet process is an example of such a nonparametric prior over countable measures, and is widely used in fields such as finance, medicine and survival analysis [???].

The forthcoming sections of this chapter introduce the Dirichlet distribution along with its basic properties. The Dirichlet distribution can be considered a semi-parametric prior, and will provide the reader with the prerequisite material to understand subsequent content. Latter sections will move on to defining the Dirichlet distribution’s nonparametric extension, the Dirichlet process, along with its properties and applications relevant to this work.

### 3.2 Notation & mathematical background

We introduce some prerequisite terminology and mathematical definitions needed to formally define the Dirichlet Process. In particular, we define the key concepts of a \( \sigma \)-algebra, a measurable
space, a probability measure and the measurable function known as the Dirac measure.

**Definition 4. σ-algebra**

We say $\mathcal{B}$ is a $\sigma$-algebra on the set $\mathcal{X}$ if it satisfies the following:

1. $\mathcal{X} \in \mathcal{B}$.

2. If $S$ is in $\mathcal{B}$, then $S^c \in \mathcal{B}$.

3. For all countable collections $\{E_i\}_{i \in \mathbb{N}} \in \mathcal{B}$, then $\cup_{i \in \mathbb{N}} E_i \in \mathcal{B}$.

**Definition 5. Measurable space**

We say the pair $(\mathcal{X}, \mathcal{B})$ is a measurable space if $\mathcal{B}$ defines a $\sigma$-algebra on the set $\mathcal{X}$.

**Definition 6. Probability measure**

Let $(\mathcal{X}, \mathcal{B})$ be a measurable space. We say a function $\mu : \mathcal{B} \to [0, 1]$ is a probability measure if the following are satisfied:

1. $\forall E \in \mathcal{B}$, $\mu(E) \geq 0$.

2. $\mu(\emptyset) = 0$.

3. $\mu(\mathcal{B}) = 1$.

4. For all countable collections $\{E_i\}_{i \in \mathbb{N}}$ of pairwise disjoint sets in $\mathcal{B}$, we have $\mu(\bigcup_{i=1}^{\infty} E_i) = \sum_{k=1}^{\infty} \mu(E_i)$.

**Definition 7. Dirac measure $\delta_x$**

The Dirac measure $\delta_x$ on a measurable space $(\mathcal{X}, \mathcal{B})$ is defined such that, for any measurable set $A \subset \mathcal{B}$ and $x \in \mathcal{X}$:

$$\delta_x(A) = \begin{cases} 0, & x \notin A \\ 1, & x \in A \end{cases}$$

Crucially, a probability distribution function can be thought of as a probability measure, and a $\sigma$-algebra can be informally thought of as the sensible, non-paradoxical sets upon which traditional probability distributions are defined.

### 3.3 Dirichlet distribution

The Dirichlet distribution is a popular distribution under the Bayesian formulation, and is frequently used as a semi-parametric prior $\pi$ over a parameter space $\Theta$. Informally, the Dirichlet distribution can be thought of as a prior over finite probability mass functions, and is defined as follows:
Definition 8. Dirichlet distribution

Let \( P = (P_1, ..., P_k) \) be the random components of a probability mass function, i.e. \( P_i \geq 0 \) for \( i = 1, ..., k \) and \( \sum_i^k P_i = 1 \). We say \( P \) is distributed according to a Dirichlet distribution with parameters \( \alpha = (\alpha_1, ..., \alpha_k) \) where \( \alpha_i \geq 0 \) for \( i = 1, ..., k \), denoted as \( P \sim \text{Dir}(\alpha) \), if its probability mass function \( f(P_1, \ldots, P_k | \alpha) \) satisfies:

\[
f(P_1, \ldots, P_k | \alpha) = \frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \prod_i P_i^{\alpha_i-1}
\]

where \( \Gamma(t) = \int_0^\infty x^{t-1}e^{-x}dx \). For \( k = 2 \), the Dirichlet distribution reduces to the Beta(\( \alpha_1, \alpha_2 \)) distribution.

An important observation of the Dirichlet distribution is its domain space; since \( P = (P_1, \ldots, P_k) \sim \text{Dir}(\alpha) \) where \( \sum_i^k P_i = 1 \) such that \( P_i \geq 0 \) for each \( i \), it follows that \( P \) is a probability mass function. Thus, the Dirichlet distribution can be thought of as a prior distribution over the space of finite probability mass functions. The Dirichlet distribution has the following properties:

\textbf{Dir(\( \alpha \)) moments:}

The mean and covariance of \( \text{Dir}(\alpha) \) are given by:

\[
E[P_i] = \frac{\alpha_i}{\sum_{i=1}^k \alpha_i} 
\]

\[
\text{Cov}[P_i, P_j] = \frac{\alpha_i \alpha_j}{\alpha_0^2 (\alpha_0 + 1)} \quad (i \neq j)
\]

respectively, where \( \alpha_0 = \sum_{i=1}^k \alpha_i \). These expressions provide insight into how the parameters \( \alpha \) contribute to samples generated from \( \text{Dir}(\alpha) \) in that, the \( \alpha_i \) control the relative likelihood of the \( i^{th} \) component and further contribute to the correlation between components.
Multinomial conjugacy:
A key property of the Dirichlet distribution is its conjugacy with the multinomial distribution.
The multinomial distribution is defined as follows; suppose we take \( n \) independent samples
of \( k \) mutually exclusive categories with the \( i^{th} \) category having success probability \( P_i \) \((i = 1, \ldots, k)\), then the multinomial distribution is the distribution over the number of occurrences
\( x_i \) of category \( i \) for \( i = 1, \ldots, k \), and has probability mass function given by:

\[
f(x_1, \ldots, x_k | n, P_1, \ldots, P_k) = \frac{n!}{x_1!x_2! \ldots x_k!} \prod_{i=1}^{k} P_i^{x_i}. \tag{3.6}
\]

We denote this distribution as Multinomial\((k, n, P)\). Importantly, suppose \( P \sim Dir(\alpha) \) and
\( X | P \sim Multinomial(k, n, P) \) where \( X = (x_1, \ldots, x_k) \), then the posterior distribution is given
by \( P|X \sim Dir(\alpha + X) \). This is a significant result, as it means the Dirichlet distribution is the
conjugate distribution over the space of finite probability mass functions.

Aggregation property of the Dirichlet distribution:
Our final Dirichlet distribution property is the aggregation property. This property can be
informally thought of as the effect of clumping together different parts of probability space,
i.e. if one combines non-intersecting sectors of the space, then one has a Dirichlet distribution
over the new augmented space. More concretely, if one has a partition \( \{A_1, \ldots, A_r\} \) of the set
\( \{1, \ldots, k\} \), then:

\[
\left( \sum_{i \in A_1} Q_i, \sum_{i \in A_2} Q_i, \ldots, \sum_{i \in A_r} Q_i \right) \sim Dir\left( \sum_{i \in A_1} \alpha_i, \sum_{i \in A_2} \alpha_i, \ldots, \sum_{i \in A_r} \alpha_i \right).
\]

This property offers an intuition into the formulation of the Dirichlet distribution, and also
provides utility with respect to inferential methods which we will allude to in latter sections.
3.4 Dirichlet process

?, motivated by the difficulty of specifying nonparametric priors in the Bayesian framework, introduced the Dirichlet process as a class of nonparametric prior distributions. The Dirichlet process can be thought of as the prior distribution over countable measures, or the distribution over distributions. Ferguson argued that any Bayesian nonparametric prior should exhibit the following two appealing characteristics:

1. The support for the prior is large.
2. The posterior distributions given a sample of observations from the true probability distribution should be analytically manageable.

**Definition 9. Dirichlet process**

We say \( G \) is distributed according to a Dirichlet process with parameters \( G_0 \) (base distribution) and \( \nu \) (scale), denoted as \( G \sim \text{DP}(\nu G_0) \), if for any \( \sigma \)-algebra \( \mathcal{B} \) of the space \( \Theta \), and given any finite partition \( A_1, \ldots, A_k \subset \mathcal{B} \), we have the following property:

\[
(G(A_1), \ldots, G(A_k)) \sim \text{Dir}(\nu G_0(A_1), \ldots, \nu G_0(A_k))
\]

for any \( k \in \mathbb{N} \).

? proved the existence of such a stochastic process sharing these DP criteria by verifying the Kolmogorov consistency theorem [?]. The Dirichlet process has the following key properties:

**Discreteness:**

Samples of \( \text{DP}(\nu G_0) \) are discrete random measures. ? proved their discreteness by using an involved analysis of Gamma processes. Crucially, this discreteness allows for mixture modelling, as the probability of two samples coinciding from a \( G \) measure is non-zero, thus allowing a cluster interpretation.

**Realisations of** \( G \sim \text{DP}(\nu G_0) \) **are random probability measures:**

This is verified by an alternative definition provided by ?, which expresses a Dirichlet process \( G \) as:

\[
G = \sum_{i=1}^{\infty} p_i \delta_{\theta_i}
\]

with \( p_i = j_i / \sum_{i=1}^{\infty} j_i \), where \( j_i \) are random variables constructed from the distribution functions \( P(j_1 \leq x_1) = e^{N(x_1)} \), \( N(x) = -\alpha \int_{x}^{\infty} e^{-y} y^{-1} dy \) where \( \alpha > 0 \), and \( P(j_k \leq x_j | j_{k-1} = x_{k-1}) = \ldots\)
Chapter 3. Bayesian Nonparametrics

\( x_{k-1}, \ldots, j_1 = x_1 = e^{N(x_k) - N(x_{k-1})} \) for \( 0 < x_k < x_{k-1} \) and \( j \geq 2 \). They showed that this construction is a random probability measure and is in fact, a Dirichlet process. Ferguson further showed that for \( \text{DP}(\nu G_0) \) defined over measurable space \((\Theta, \mathcal{B})\), and \( Q \) a fixed probability measure on \((\Theta, \mathcal{B})\) with \( Q \ll \nu G_0 \), then for any \( m \) and measurable sets \( A_1, \ldots, A_m \), and \( \epsilon > 0 \):

\[
P(\|G(A_j) - Q(A_j)\| < \epsilon \text{ for } j = 1, \ldots, k) > 0.
\]

These two results have the following important implications; firstly, it verifies that \( G \) is a probability measure on the measurable space \((\Theta, \mathcal{B})\) and secondly, it demonstrates the fulfilment of the objective that the nonparametric prior support is sufficiently large. However, it is important to note that this alternative definition is not constructive, i.e. we are unable to generate samples from this definition due to the normalisation constant \( \sum_{i=1}^{\infty} j_i \).

**Moments of \( \text{DP}(\nu G_0) \):**

The mean and variance of \( \text{DP}(\nu G_0) \) are given by:

\[
\mathbb{E}(G(A)) = G_0(A)
\]

\[
\text{Var}(G(A)) = \frac{G_0(A)(1 - G_0(A))}{1 + \nu}
\]

respectively. These moments give us a sense of how the parameters of \( \text{DP}(\nu G_0) \) contribute to \( G \sim \text{DP}(\nu G_0) \) realisations; i.e. \( G_0 \) controls where realisations are centred, and \( \nu \) controls the degree to which realisations are close to \( G_0 \).

**DP conjugacy:**

A key property of the DP is its conjugacy with the multinomial. Since \( G \) is a random measure, we can sample \( \theta_i \) iid \( G \) for \( i = 1, \ldots, n \). The posterior of \( G \) given observed values \( \theta_1, \ldots, \theta_n \) is then given by:

\[
G|\theta_1, \ldots, \theta_n \sim \text{DP} \left( \nu + n, \frac{\nu}{\nu + n} G_0 + \frac{n}{\nu + n} \sum_{i=1}^{n} \delta_{\theta_i} \right).
\]

This means the DP is the conjugate prior over distributions that are closed under posterior updates given observations \([\theta]\). This has a theoretical significance, since it meets Ferguson’s second criterion, that posterior distributions given a sample from the true random measure from the DP should be tractable. This has practical significance, as it gives one routes into developing straight-forward procedures for posterior inference.
3.4 Dirichlet process

3.4.1 Blackwell-MacQueen urn scheme

? used a Polya urn scheme to provide an alternative definition of the Dirichlet process. This provided a generative model for producing random measure instances as specified by Ferguson’s Dirichlet process. More concretely, the DP is re-expressed as the following; given samples \( \theta_1, \ldots, \theta_n \) of \( G \) where \( G \sim \text{DP}(\nu G_0) \), \( G \) is integrated out of the joint distribution of \( \theta_1, \ldots, \theta_{n+1} \). The posterior predictive distribution then becomes:

\[
\theta_1 \sim G_0 \\
\theta_{n+1}|\theta_1, \ldots, \theta_n \sim \frac{\nu}{\nu + n} G_0(\theta_{n+1}) + \frac{1}{\nu + n} \sum_{i=1}^{n} \delta_{\theta_i}(\theta_{n+1}).
\]  

(3.7)

Crucially, ? establish two key properties of the limiting distribution of \( G \). Firstly, that as \( n \rightarrow \infty \), the urn scheme converges with probability 1 to a discrete random measure \( G \), and secondly, that this \( G \) is identical to a sample from \( \text{DP}(\nu G_0) \). From the joint distribution induced from \( p(\theta_1, \ldots, \theta_{n+1}) \), and along with the definition of exchangeability and De Finetti’s theorem, ? establish critically that \( \theta_1, \ldots, \theta_n | G \sim G \) is equivalent to the DP and that these \( \theta_1, \ldots, \theta_{n+1} \) are i.i.d draws from \( G \).

There is a clustering property implied from (3.7). By denoting \( \theta_1^*, \ldots, \theta_n^* \) as the set of unique values of \( \theta_1, \ldots, \theta_n \), we then can rewrite (3.7) as:

\[
\theta_1 \sim G \\
\theta_{n+1}|\theta_1, \ldots, \theta_n \sim \frac{\nu}{\nu + n} G(\theta_{n+1}) + \frac{1}{\nu + n} \sum_{i=1}^{n^*} n_i \delta_{\theta_i^*}(\theta_{n+1}).
\]  

(3.8)

where \( n_i \) is the number occurrences of \( \theta_i^* \), and \( n^* \) is the number of unique atoms of \( \theta_1, \ldots, \theta_n \).

(3.8) demonstrates the clustering structure and preferential attachment process at play, as the probability of the sample \( \theta_{n+1} \) being assigned to the atom \( \theta_i^* \) is \( \frac{n_i}{\nu + n n} \). This illustrates that there is a non-zero probability of being assigned to an existing cluster, as well as an increased probability for atoms to be added to larger clusters.

? investigated the expected number of unique mixture components that a sample \( \theta_1, \ldots, \theta_n \sim G \) induces. They derived:

\[
\mathbb{E}(n^*|\nu) = \sum_{i=1}^{n} \frac{\nu}{\nu + i - 1} \\
\approx \nu \log(\frac{\nu + n}{\nu}).
\]  

(3.9)
This shows the number of unique clusters $n^*$ grows logarithmically in $n$, which indicates the clustering is well defined, and allows straightforward inference to be developed around the $\nu$ parameter. The DP shares connections with the Chinese Restaurant Process (CRP) [3]. The CRP is a stochastic process which defines a distribution over partitions of the integers $\{1, \ldots, n\}$. The CRP can be intuited by the process of customers attending a restaurant and selecting tables to sit at. This is as follows; the first customer sits at the first table, and $n^{th}$ customer chooses table $c_n$, where probability of selecting the $k^{th}$ table is probability $\frac{n_k}{\nu+n-1}$, with $n_k$ being the number of customers at the $k^{th}$ table, and selects a new table with probability $\frac{\nu}{\nu+n-1}$, i.e.:

$$p(c_n \text{ table choice } | \ K \text{ partitions } ) = \begin{cases} 
\text{k^{th} table with probability } \frac{n_k}{\nu+n-1}, \\
\text{new table with probability } \frac{\nu}{\nu+n-1}.
\end{cases}$$

This is exchangeable with respect to the customer labels and the tables they sit at. This possesses a close resemblance with (3.8), which becomes clearer from the relabelled expression (3.7):

$$p(c_{n+1}|c_1, \ldots, c_n) = \frac{\nu}{\nu+n} \delta(c_{n+1} = K + 1) + \sum_{k=1}^{K} \frac{n_k}{\nu+n} \delta(c_{n+1} = k). \quad (3.10)$$

As with the DP, the CRP exhibits a preferential attachment process, as well as the distribution over the table partitions being the same as the distribution over the cluster sizes [3].

### 3.4.2 Sethuraman stick-breaking construction

Provided an alternative constructive definition of the DP, known as the stick-breaking construction. This is as follows; given $G$ satisfying:

$$G = \sum_{i=1}^{\infty} \pi_i \delta_{\theta_i},$$

$$\beta_i \overset{i.i.d.}{\sim} Beta(1, \nu),$$

$$\pi_i = \beta_i \prod_{l=1}^{i-1} (1 - \beta_l), \quad (3.11)$$

where $\theta_i \overset{i.i.d.}{\sim} G$, then $G \sim DP(\nu G_0)$. This construction has connections with (3.7), as it provides an intuitive distribution over the cluster partitions. More precisely, 3 verified that the probability of the draw $\theta_1$ of (3.7)’s urn scheme as the same probability of the partition denoted by $\pi_1$ in (3.11). This construction has useful applications to density estimation and mixture modelling.

### 3.5 Dirichlet process applications

Having formally introduced the DP, its various expositions and theoretical properties, we now describe the applications of the DP. In-particular, we discuss DP’s applications to mixture
modelling and density estimation.

3.5.1 Dirichlet process mixture model

The Dirichlet process mixture model (DPMM) was proposed by ? as a mixture model with a DP prior over the random mixing distribution. The DPMM can be hierarchically expressed as:

\[
\begin{align*}
y_i|\theta_i & \sim F(\theta_i) \\
\theta_i|G & \sim G \\
G & \sim \text{DP}(\nu G_0).
\end{align*}
\]  

(3.12)

As discussed in section 3.4.1, the samples \( G \sim \text{DP}(\nu G_0) \) induce a clustering structure over the \( \theta_1, ..., \theta_n \sim G \), which in turn induce a partition over the responses \( y_1, ..., y_n \). An example of (3.12) is given by (3.13), with \( N(\cdot, \sigma_1) \) as the kernel, and base distribution \( G_0 = N(0, \sigma_2) \):

\[
\begin{align*}
y_i|\theta_i & \sim N(\theta_i, \sigma_1^2) \\
\theta_i|G & \sim G \\
G & \sim \text{DP}(\nu G_0)
\end{align*}
\]  

(3.13)

where \( \sigma_1, \sigma_2 \) are fixed constants. The DPMM of (3.12) can be further re-expressed using allocation variables as:

\[
\begin{align*}
y_i|\theta_i & \sim F(y_i|\theta_{Z_i}) \\
Z_i|G_0 & \sim \sum_{j=1}^{\infty} \pi_j \delta_{\theta_j}(.) \\
G_0 & = \sum_{i=1}^{\infty} \pi_i \delta_{\theta_i} \\
\beta_i & \sim \text{Beta}(1, \nu) \\
\pi_i & = \beta_i \prod_{l=1}^{i-1}(1 - \beta_l) \\
\theta_i & \sim G_0
\end{align*}
\]

where \( Z_i \) denotes the index that the \( i^{th} \) data point has been allocated to. Though the observations \( y_1, ..., y_n \) are continuous, by discreteness of \( G \), we interpret two observations \( y_i, y_j \) being clustered together if they share the same allocation variable, i.e. \( Z_i = Z_j \), and therefore the same atom \( \theta_i = \theta_j \). Furthermore, if \( Z_i \neq Z_j \) for \( i \neq j \), it follows almost surely that \( \theta_i \neq \theta_j \), since the \( \theta_i \) realisations are unique for continuous \( G_0 \). This guarantees different allocations have different atoms. We derive the probability of the responses \( y_i, y_j \) for \( i \neq j \), belonging to the
same cluster as follows; from the exchangeability of labels of $i, j$, we can therefore relabel them as $i = 1, j = 2$. Thus, from (3.7) we obtain:

$$\theta_2 | \theta_1 \sim \frac{\nu}{\nu + 1} G(\theta_1) + \frac{1}{\nu + 1} \delta_{\theta_1}.$$ 

Hence, the probability of responses $y_1, y_2$ belonging to the same cluster as $\frac{1}{\nu + 1}$.

The parameters $G_0, \nu$ of DP play a significant role in mixture modelling. For fixed $G_0$, increasing $\nu$ decreases the variance of $G \sim DP(\nu G_0)$ samples. Consequently, $\nu$ ought to reflect the strength of an experimenter’s belief around which samples $G$ are centred around the measure $G_0$.

### 3.5.2 Density estimation

The DP has further applications to density estimation [???]. Density estimation essentially estimates some unknown distribution $F$, such that:

$$x_i \sim F(\cdot) \text{ for } i = 1, \ldots, n$$

given the samples $x_1, \ldots, x_n$. The Bayesian framework requires placing a prior distribution over the functional space of $F$. Parametric methods traditionally restrict the functional space by assuming the space can be expressed as a finite combination of mixing densities (of a known family). Bayesian nonparametric approaches however, assume an infinite dimensional prior over this functional space, examples of which include Polya trees, Berstein polynomials and other variants of the Dirichlet process. For the purposes of this work, we will focus on the DP’s utility to density estimation.

The DP’s approach to estimating a density $p(x)$ involves convolving a family of kernels $f(x|\theta)$ (parametrised by $\theta$) with a DP prior over the mixing proportions to produce a countably infinite mixture of smooth kernel functions. More concretely, suppose we aim to estimate a density $p(x)$ as a DP mixture of densities $f(x|\theta)$, we then write $p(x)$ as:

$$p(x) = \int f(x|\theta) dG_0(\theta)$$  \hspace{1cm} (3.14)
where $G \sim \text{DP}(\nu G_0)$. An alternative way of expressing (3.14) is by using the stick-breaking construction of (3.11) where given $G \sim \text{DP}(\nu G_0)$, (3.14) can be rewritten as:

$$p(x) = \int f(x|\theta) dG_0(\theta) = \int f(x|\theta) d\left(\sum_{i=1}^{\infty} \pi_i \delta_{\theta_i}\right) = \sum_{i=1}^{\infty} \pi_i f(x|\theta_i).$$

(3.15)

Thus, we can smoothly estimate the density of $p(x)$ as countably infinite mixture of $f(x|\theta_i)$ densities, where $\nu$ contributes to the degree of smoothness; with small $\nu$ producing smoother estimates and lumpier estimates otherwise. Figure 3.1 provides examples of measures generated from urn process of (3.7), along with the corresponding densities produced from a mixtures of normal kernels $f(\cdot | \theta, 0.5)$ when these measures are priors over $\theta$. Here we use the base distribution $G_0 = \text{N}(0, 3.0)$. We notice increasing $\nu$ in turn increases the number of unique atoms produced from the DP, as well as increasing the multi-modal behaviour of the mixtures.

The R code producing these simulations are included in appendix A.1.

Figure 3.1: Simulated DP measures produced from the urn process of (3.7) for 50 iterations, along with the corresponding densities produced from a mixtures of normal kernels $f(\cdot | \theta, 0.5)$ when these measures are priors over $\theta$. These measures and mixture of densities are produced over various $\nu$ values. The R code producing these plots is included in A.1.
3.6 Posterior inference

Broadly speaking, there are two routes for posterior inference used in DPMMs, marginal and conditional approaches, each relying on the differing representations of a DP. Marginal approaches involve integrating out the infinite dimensional measure of $G$, and then by utilising the property of (3.7) and the exchangeability of atoms $\theta_i$, then takes Gibbs samples from the distributions of $\theta_i | \theta_{-i}$ (where $\theta_{-i}$ is the vector of locations excluding the $i^{th}$ atom) for each $i$. This inferential procedure was made possible by the initial work of ?, and thus, represents the first step made towards enabling DPs to be an applicable methodology. ? provides an excellent summary of marginal approaches. Conditional samplers, first proposed by ?, do not marginalise out the infinite random measure of $G$, but instead involve imputing $G$, then sampling the cluster assignments for each of the location atoms $\theta_i$ from their posteriors. One of the initial challenges conditional approaches had was handling the infinite dimensional nature of $G$. Initial approaches relied on making finite approximations of a DP [??], but extensions of conditional samplers have included innovations around slice and retrospective samplers, which have offered marked improvements in many more challenging DP inferential contexts [????]. Although marginal and conditional methodologies are amongst the more popular approaches to DP inference, other methodological procedures exist, including sequential greedy search algorithms and variational approaches to name a few [??].

For the purposes of this work, we will focus on marginal approaches for posterior inference of DPMMs. Our motivations for orienting our inferential procedure around a marginal method, over other conditional approaches, are the following two closely related reasons. Firstly, marginal approaches circumvent the issues of the infinite dimensional nature the DP by marginalising over the random measure $G$. By exploiting this property of (3.7), and thus maintaining the infinite dimensional nature of the DP, we maintain a particularly favourable feature of the DP that is worth keeping. This is especially valuable in cases when the number of unique clusters characterising the data is uncertain, such as is with our retail analytics context where we devising a DPMM around unfamiliar and novel data. Our second, closely related reason, is the computational straightforwardness of the implementation of marginal inferential procedures, whilst still maintaining the trait of possessing the infinite dimensional nature of the DP. An important caveat of this is that, although more advanced conditional samplers exist that make it possible to maintain the infinite dimensional property of the DP, for example, ?’s retrospective sampler approach or ?’s slice sampler (and other related methods), they often involve nuanced and complicated updating strategies that make these procedures non-trivial
Our particular marginal method our inferential procedure will be largely based on ?’s algorithm 8, where $G_0$ is a non-conjugate prior with respect to the likelihood function of $f(\cdot | \theta)$. Their approach uses the DP exposition of (3.8), which then iteratively takes Gibbs samples of the $\theta_i$ conditionals, and then updates each of the unique $\theta_i$ atoms using a Metropolis-Hasting step. We then use ? methodology of updating the scale $\nu$ of the DP. More concretely, the algorithm is as follows:

1. ?’s approach iteratively samples the locations for each data-point by sampling the multinomial distribution of order $n^* + c$ (where $c$ is the chosen number of auxiliary components). More concretely, the resultant sample of $\theta_i$ equates to sampling a multinomial with probabilities:

$$P(\theta_i = \theta^*_k | \theta_{-i}, y_i, \theta^*_1, \ldots, \theta^*_{n^*+c}) \propto \begin{cases} \frac{n^*_k}{n^*+\nu} f(y_i | \theta^*_k) & \text{for } 1 \leq k \leq n^* \\ \frac{\nu/k}{n^*+\nu} f(y_i | \theta^*_k) & n^* < k \leq n^* + c \end{cases}$$

where:

$$\theta^*_k \sim G_0 \text{ for } k = n^* + 1, \ldots, n^* + c$$

2. The $\theta_k$ atoms are then updated for each of the unique clusters $k = 1, \ldots, n^*$. This avoids inefficiencies associated with having to pass through extremely low probability states to get to a higher probability states. This can be done by Metropolis Hastings updates.

3. Finally, by specifying $\nu \sim G(\tau_1, \tau_2)$ and introducing an auxiliary variable $\gamma$, enables $\nu$ to be Gibbs sampled. Specifically, we take the following samples:

$$(\gamma \mid \nu, n^*) \sim \text{Beta}(\nu + 1, n)$$

$$(\nu \mid \gamma, n^*) \sim \pi_\gamma \text{Gamma}(\tau_1 + n^*, \tau_2 - \log(\gamma)) + (1 - \pi_\gamma) \text{Gamma}(\tau_1 + n^* - 1, \tau_2 - \log(\gamma))$$

where the weights $\pi_\gamma$ are defined by $\pi_\gamma/(1 - \pi_\gamma) = (\theta + n^* - 1) / (n (\tau_2 - \log(\gamma)))$. The detailed steps of this facet of the DP inferential procedure will be elaborated on during section 5.3.3.
This DPMM inference methodology is simple and intuitive, and will be the primary methodology for posterior inference of DPMMs during this work. The implementation details of this approach are explained in more depth in subsequent Chapters.
Chapter 4

Elasticity clustering & related methodologies

This Chapter introduces and defines the interest around analysing cross-elasticity coefficients. In particular we discuss the relationship between these cross-elasticity coefficients and the sensitivity of sales of a product with respect to changes in its own price and the prices of competing products. As referenced in section 1.1, retailers are becoming increasingly interested in classifying and segmenting many of their processes, as it allows them to mitigate storage and computational costs as well as providing valuable insights with which can create a competitive advantage. One way this interest manifests itself is in retailers’ analysis of the sensitivity of their products’ sales to the price changes across competing products.

The subsequent sections are structured as follows: section 4.1 introduces the concept of a product’s sales sensitivity in the context of its cross-elasticity coefficients and outlines retailers’ motivation in clustering products in terms of their sales sensitivities. In particular we articulate how a product’s sales sensitivity is exhibited through these cross-coefficients and specify a class of regression models that such cross-elasticity coefficients can be generated from. The section continues on to outline the data by which our sales sensitivities analysis is motivated. Section 4.2 describes and reviews the traditional approaches used to investigate and analyse these cross-elasticity coefficients in the fields of retail analytics and econometrics, and outlines the methods used to interpret the differences between cross-elasticity coefficients. This section concludes by highlighting the shortcomings of these current approaches.
4.1 Elasticity clustering background

Characterising products by how sensitive their sales are to their competitor prices is of particular interest to supermarkets. The price at which a product is offered to the consumer is arguably one of the most important controls a retailer has. Unsurprisingly, retailers’ analytics teams are constantly striving to develop models and inferential methods that provide insights into how price fluctuations that propagate throughout stores will impact the sales of products whose prices have not changed. Consequently, retailers are invested in understanding how these price sensitivities are manifested, as they view it as core to their business operations. More concretely, such a price sensitivity analysis and segmentation provides the following benefits to retailers:

1. Understanding how a product’s sales are sensitive to its competitor’s price changes allows store planners to decide the value of a given display combination, as it provides information on how a product’s sales are likely to react to the deviations of prices of other products. Retailers are increasingly interested in fully understanding the triggers around consumers purchasing decisions when faced with various display combinations and multiple product choices, and retailers are aware that prices are a key factor that drives these decisions. For instance, a poor display combination could be one that consists entirely of products characterised by their sales being primarily driven by the prices of its competition. This would lead to margin cannibalisation - where profit made on one product is offset by the loss of profit of another product. A characterisation of products in terms of their sales sensitivities would allow store planners to circumvent such pitfalls, and generally empower them to make better pricing and display decisions.

2. The exercise of segmenting products in terms of their sales sensitivity can reveal hidden structure and an informative narrative of data that could provide valuable insights. For example, it is common practice of retailers to cluster consumers according to their product preferences; this allows retailers to efficiently summarise consumers’ tastes, which can be used to create personalised recommender systems, improve sales forecasts and consumer loyalty. Such a price sensitivity segmentation could allow retailers to improve their personalised services by understanding consumers’ purchasing patterns in terms of their preferences to a particular price sensitivity segmentation. Such improvements could provide a significant advantage in the competitive landscape of consumer retail.

3. In addition to the product display and possible consumer personalisation benefits that a sales sensitivity segmentation would provide, a sales sensitivity segmentation would
reduce storage costs and would improve the efficiency of this analysis compared to any manual method that compares these sale sensitivities product by product. Theoretically, a supermarket with $N$ products could want to understand all of the sale sensitivities between all products, i.e. how do the sales of a given product change with respect to price changes of every other product. Such an analysis would lead to needing to store $N \times N$ quantities, which can be impractical for large $N$. Segmentation or clustering methodologies capable of reducing the dimension of such an analysis to a more simple generating process that neatly characterise products into groups or clustering is appealing to retailers. Consequently, where possible, retailers are interested in developing efficient summaries of their data, to mitigate storage and computational costs.

Although these aforementioned benefits are by no means exhaustive, it provides the reader an overview of the motivations behind why retailers are interested in product segmentation.

### 4.1.0.1 Cross-elasticity demand models

Having discussed the desire to categorise products in terms of their sales sensitivities, we now introduce one approach of quantitatively deriving a product’s sensitivity in sales with respect to changes to its competitors prices. We do this by introducing the notion of a product’s *price elasticity of demand*, which we define as the rate of change of the quantity demanded of a product with respect to changes in its own price. More formally, by defining $P$ as the price of the product and $S(P, x)$ as the sales of the product as some function of its price $P$ and other variables $x = (x_1, \ldots, x_k)$, we then define $\psi$, the *price elasticity of demand*, as:

$$\psi = \frac{\partial S(P, x)}{\partial P}.$$  \hfill (4.1)

This describes the nature in which a product’s sales changes with respect to changes in its own price. The majority of products that retailers offer to their markets have elasticity of demands such that $\psi \leq 0$, i.e. increases in prices lead to decreases in sales. Importantly, more negative $\psi$ implies that small increases in the price leads to large reductions in sales, hence indicating a product that is highly price sensitive.

We can extend this definition to the concept of a product’s sensitivity in sales with respect to changes in price of another product. We define a *product i’s cross elasticity of demand with respect to product j*, as the rate of change between the quantity demanded of product $i$ with respect to a change in the price of product $j$. More concretely, by defining $P_i$ as the price of product $i$, $P_j$ as the price of product $j$, $S_i(P_j, x)$ as the sales of product $i$ as some function of
and other variables $x$, we then define the $\chi_{ij}$, the product $i$’s cross elasticity of demand with respect to product $j$, as:

$$\chi_{ij} = \frac{\partial S_i(P_j, x)}{\partial P_j}.$$  \hfill (4.2)

Similarly as before, this encapsulates the effect that changes in a product’s prices affects another product’s sales. The majority of products that retailers offer to their markets have cross-elasticity of demands such that increases/decreases in another product’s prices lead to increase/decreases in sales of another product. As before, the larger $\chi_{ij}$ is, the more sensitive product $i$’s sales are to changes in product $j$’s price.

Having defined (4.1) and (4.2) as a concept of a sales sensitivity measures, we now introduce a method of calculating these quantities. Importantly, we need to assume a functional form for $S_i(P_j, x)$. There are many approaches that link the relationship between price elasticity of demand with sales, such as market share models, attraction models, structural equation and consumer utility modelling \[????\]. Market share and attraction models generally interpret consumers’ demand for particular products as having a multiplicative structure with normally distributed errors. The proportion of demand exhibited for each of these products relative to other competitor products is then described as a function of the original attraction of the products \[??\]. Structural equation modelling approaches involves using factor analysis and multivariate regression techniques to analyse the graphical structure that describes the relationships between the variables of interest \[??\]. Random utility modelling assumes that consumer preferences between two or more options are discrete decisions that are made with respect to random utility functions that describes an individual’s underlying objective in which they strive to maximise \[????\]. All of these models have been successful at describing how the direct and cross-elasticity quantities impact demand at numerous levels of aggregation. However, for the purposes of our analysis, we focus on the functional form known as the Working-Leser equations. The Working-Leser regression models are parametric models that predict the demand of a product given covariate data. More precisely, for a category of $N$ different products (also referred to as items), we estimate the set of coefficients $\{c_i, \chi_{ij}, \psi_i | 1 \leq j \leq m_j, 1 \leq i \leq N\}$ derived from the system of $N$ regression models:

$$y_{it} = \log(S_{it}) = c_i + \psi_i \log(P_{it}) - \sum_{j=1}^{m_i} \chi_{ij} \log(P_{ijt}) + \epsilon_{it},$$ \hfill (4.3)
where:

\[ S_{it} = \text{sales of item } i \text{ at time } t, \]
\[ \psi_i = \text{item } i\text{'s direct elasticity}, \]
\[ P_{it} = \text{item } i\text{'s price at time } t, \]
\[ \chi_{ij} = \text{item } j\text{'s cross elasticity with item } i, \]
\[ P_{ijt} = \text{price of item } i\text{'s } j^{th} \text{ cross item at time } t, \]
\[ c_i = \text{item } i\text{'s additive constant}, \]
\[ m_i = \text{number of cross competitors of item } i, \]
\[ \epsilon_{it} \sim N(0, \sigma_i^2). \]

These and similar models are widely used in a range of econometrics and retail analytics settings \[?\], but in practice companies often use much more sophisticated versions of this model, taking into account a larger range of covariates and that include autoregressive terms, time-dependencies via smoothing and seasonality modelling. Crucially, this model assumes that the log of sales of each product are conditionally independent, conditioned on the aforementioned covariates. The popularity of these models is that the output of fitted models is highly transparent as the coefficients are straight-forward to interpret and are computationally efficient to implement. This is of key importance, as companies can use models not only to predict demand but also gain a greater insight into the underlying phenomena at play.

Importantly, the cross-elasticity output of models such as \((4.3)\) allows us to quantify the sales sensitivities of products with respect to the price changes of its relevant competition. In particular, the vector of a product’s direct- and cross-elasticity coefficients, \( (\psi_i, \chi_{i1}, \ldots, \chi_{im_i}) \), are the quantities conveying a product’s sales sensitivities with respect to changes in a product’s own and competitor prices. Our aim is to cluster products as a function of their direct- and cross-elasticity coefficients vectors \( (\psi_i, \chi_{i1}, \ldots, \chi_{im_i}) \).
4.1.1 PriceStrat and cross-elasticity output

We now describe the dataset our sales sensitivities analysis and elasticity clustering is motivated by. Access to this dataset was permitted by dunnhumby ltd, and comprises the relative cross-elasticity vectors for a set of products from a large UK supermarket retailer. These relative cross-elasticity vectors have been generated from a cross-elasticity regression model known as PriceStrat, which is closely related to (4.3). Although the precise mechanics of how these estimates are obtained are highly engineered, the general form of the model is given by the following regression:

\[
\log(S_{i,t}) = c_i - \varphi_i \log(Q_{i,t}) + \sum_{j=1}^{n_i} \varphi_i \eta_{ij} \log(P_{i,j,t}) + f(Q_{i,1:T}, P_{i,1:n_i,1:T}) + \epsilon_{i,t},
\]  

(4.4)

where, for each product \( i \) and time \( t \), \( S_{i,t} \) denotes its sales, \( Q_{i,t} \) its price, \( P_{i,j,t} \) the price of its \( j^{th} \) competitor product, \( \varphi_i \) its direct elasticity and \( \eta_{ij} \) product \( j \)'s relative cross-elasticity with product \( i \) (as a multiple of the direct elasticity) and \( c_i \) is some additive constant. We use the notation \( 1:n \) to denote the set \( 1,\ldots,n \). The map \( f(\cdot) \) involves nuanced data aggregation and smoothing, seasonality patterns relevant to retail sales, as well as additional information on display combinations and promotions specifically engineered to induce \( \epsilon_{i,t} \sim N(0,\sigma_i^2) \). Here \( n_i \) is the number of competitor products of product \( i \), which are pre-selected using expert knowledge.

The regression coefficients are estimated using shrinkage methods, so that only \( l_i \) of the \( \eta_{ij} \)'s are non-zero, with the remaining exactly equal to 0. To ease notation and terminology in latter sections, we assume that competitor products are labelled such that product \( i \)'s relative cross-elasticity coefficients \( \eta_{ij} \) are decreasing in magnitude with increasing \( j \) and that all products have the same potential number of competitors, i.e. \( n_i := n \ i = 1,\ldots,N \), and from now onwards refer to these relative cross-elasticity coefficients as the cross-elasticity coefficients. Table 4.1 provides some toy examples of the cross-elasticity vectors typically observed from dunnhumby’s implementation of (4.4). These examples are provided to allow the reader to intuit and visualise the data in question. Although a clustering approach of the regression coefficients can be performed alongside the regression, this is often computationally prohibitive in any context where the original predictive sales model is highly tailored and engineered, such as with model (4.4). Consequently, any clustering methodology that clusters the cross-elasticity vectors separately from the regression analysis is often the preferred route to any sales sensitivity analysis.
4.1. Elasticity clustering background

The cross-elasticity data is such that, for each product $i$, we observe a decreasing set of entries of a larger vector, censored to only the top few entries with the remaining values set to 0. Thus, the data are in the form:

$$X = \{ \eta_{i,1:n} : \eta_{i,n-l_i+1} \leq \eta_{i,n-l_i+2} \leq \ldots \leq \eta_{i,n}, \text{with } \eta_{i,j} \text{ censored to } 0 \text{ for } 1 \leq j \leq n-l_i \}$$

(4.5)

where $\eta_i$ is the cross-elasticity vector of dimension $n$ for product $i$, which has $l_i$ uncensored ordered entries, with the remaining being censored.

4.1.1.1 Challenges

We now describe some of the pertinent artefacts typical of the data (4.5) and the associated challenges.

1. **Preselection of significant cross competitors:** Although companies could store the entire data of (4.5) as a matrix containing all the cross-elasticities for each pair of products, this in practice would be computationally prohibitive. Consequently, companies induce sparsity through expert preselection of competitor products and shrinkage. They often do this by the use of highly tailored black-box sparse regression sales models [??] and only measure the cross-elasticities for a small number of competitors for each product, with the remaining entries being treated as missing or negligible. This resultant cross-elasticity coefficient data implicitly reflects only the top competitors within the market and thus induces an inherent informative missingness that means a global interpretation of the behaviour of the entire market may not be directly available. Thus, any proposed clustering strategy ought to reflect that the data of (4.5) is indeed pre-selected using expert knowledge to represent the top competitor products across a market (and therefore subject to error), with omitted entries being treated as zero or missing minor competitors and furthermore, should accommodate instances where relevant competitors have been omitted from the original regression that should have been included.

2. **Varying dimensions:** Due to the shrinkage previously, many of the remaining $\eta_{ij}$’s are exactly equal to 0. This effectively leads to the $\eta_{i,1:n}$ having different dimensions for differing $i$ as some products may have more or less zero shrinkage than others due the amount of competition with the market that they encounter. Consequently, any clustering methodology has to support this variation in dimensions exhibited in the cross-elasticities coefficients. Figure 4.2(a) shows the histograms of the varying dimensions ($l_i$) of cross-elasticity vectors $\eta_{i,1:n}$.

3. **Strictly decreasing sequences:** As a consequence of the $\eta_{ij}$ reordering, the indi-
Table 4.1: Ordered elasticity output $\varphi$ and $\eta$ for two fictional products, *Bobby’s puffs* and *Lucan’s Salted crisps*. For each product we have columns of order elasticity coefficients $\varphi_i, \varphi_i \eta_{ij}$ along with the respective sequences of $\eta_{ij}$, which demonstrates the decreasing nature of data from model (4.4). The number of potential cross competitors is set to $n_i = 6$, although the number of terms censored to 0 differs. Importantly, the set of competitors can differ for each of the products and in instances where there is a shared competitor (as with *Supermarket puffs* in this case), the value of $\varphi_i \eta_{ij}$, as well as its position in the ordering, need not be consistent across products.

<table>
<thead>
<tr>
<th>Bobby’s Cheesy puffs</th>
<th>Lucan’s Salted crisps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relevant competitors</td>
<td>$\varphi_1, \varphi_1 \eta_{1j}$</td>
</tr>
<tr>
<td>$\varphi_i \eta_{16}$</td>
<td>Bobby’s puffs</td>
</tr>
<tr>
<td>$\varphi_i \eta_{15}$</td>
<td>Supermarket puffs</td>
</tr>
<tr>
<td>$\varphi_i \eta_{14}$</td>
<td>Harry’s puffs</td>
</tr>
<tr>
<td>$\varphi_i \eta_{13}$</td>
<td>Supermarket Nuts</td>
</tr>
<tr>
<td>$\varphi_i \eta_{12}$</td>
<td>Bobby’s Tortillas</td>
</tr>
<tr>
<td>$\varphi_i \eta_{11}$</td>
<td>Tommy’s chips</td>
</tr>
<tr>
<td>$\varphi_i \eta_{11}$</td>
<td>Tommy’s puffs</td>
</tr>
</tbody>
</table>

Individual entries of cross-elasticity vectors $\eta_{i,1:n}$ are strictly decreasing in magnitude, i.e. $\eta_{i,j} \leq \eta_{i,j+1}$. One of the key aspects in the analysis of cross-elasticity coefficients is the relative decay between successive values of the cross-elasticity coefficients, as this conveys the degree of competition a product encounters with the market. Figure 4.1 plots the histograms of the marginal elasticity entries ($\eta_{ij}$) of $x_{i,1:n}$ which demonstrates the varying decay rates across the cross-elasticity vectors. Figure 4.2(b) plots some real data examples of decreasing cross-elasticity coefficients. We observe the entries of the cross-elasticity vectors $x_{i,1:n}$ are decaying at differing rates and further notice $\eta_{i,1:n}$ have differing number of entries (and thus differing dimensions).

Our goal is to summarise products’ sensitivity in sales by clustering them by their cross-elasticities coefficient vectors. We however want to do this in a way that handles the three aforementioned challenges. Namely, we want to cluster these cross-elasticity vectors according to the distribution of their competition in the market that accommodates not only clustering products with similar decay rates and cross-elasticity dimension, but also reflects that these entries are assumed to represent a product’s most significant competitors (and further deals with possible omitted competitors).

## 4.2 Analysis of cross-elasticity coefficients

We now introduce the existing work done in the analysis of direct- and cross-elasticity coefficients generated from models closely related to (4.3). This section aims to give the reader an understanding of the current methodologies employed to investigate the differences in magnitudes between the direct- and cross-elasticity coefficients of differing products. We then explore the narrative of what these differences convey. The section then moves on to highlight some of
4.2. Analysis of cross-elasticity coefficients

4.2.1 Current approaches

We now outline the main contributions and describe the broad pattern employed among research into the analysis of direct and cross-elasticity coefficients generated from cross-elasticity regression models. Generally speaking, the body of research into direct- and cross-elasticity coefficients can be largely split into two fields. The first body of work is from the public health perspective which strives to understand how the public’s consumption of specific food categories are sensitive to competing prices of other relevant categories. These papers are ultimately interested in finding insights aimed to aid public policy. In such studies, researchers broadly investigate how factors such as ethnicity and income are related to the sales sensitivities of food groups, with findings being used help reduce obesity and other diet related disorders. The second body of work into the analyses of direct- and cross-elasticity coefficients is from a retailers’ perspective, in which they try to better understand and improve their business operations. These motivations are varied, but examples range from investigating the differing store-wise pricing policies and quantifying the effect of promotional strategies to profit maximisation across product categories. Retailers ultimately aim to understand the substitutional relationship between products that these cross-elasticity coefficients convey, whether in terms of measuring

![Histograms](https://via.placeholder.com/150)

(a) Histogram of $\eta_{i,(10)}$ marginals

(b) Histogram of $\eta_{i,(9)}$ marginals

(c) Histogram of $\eta_{i,(8)}$ marginals

(d) Histogram of $\eta_{i,(7)}$ marginals

Figure 4.1: Various plots of $\eta_i$ data
brand loyalty or supermarket preference, and use this substitutional knowledge to help improve business practices.

Both bodies of work generally employ very similar post-processing methodologies when interpreting the direct- and cross-elasticity output exhibited from cross-elasticity demand models. As stated earlier, cross-elasticity regressions models for product $i$ are generally of the form:

\[
    f(S_i) = c + \psi_i g(P_i) + \sum_{j=1}^{n} \psi_i \eta_{ij} g(P_{ij}) + h(x) + \epsilon_i
\]

where $c$ is some additive constant, $x$ are some relevant covariate information and $f(\cdot)$, $g(\cdot)$ and $h(\cdot)$ are functional forms depending on the context and $\epsilon_i$ is additive noise. Very often the functional forms of $f(\cdot)$ and $g(\cdot)$ are such that $f(x) = x$ or $f(x) = \log(x)$ and similarly for $g(\cdot)$. Variations of this generalised model are frequently used to generate direct- and cross-elasticity coefficients, examples of which include Almost-Ideal-Demand models \cite{1} or for the Working-Leser demand system \cite{2}. However, independent of the original regression details, the direct- and cross-elasticity output is essentially the same as it ultimately gives rise to data of the form (4.5), i.e.:

\[
    X = \{\eta_{i,1:n} : \eta_{i,n-l_i+1} \leq \eta_{i,n-l_i+2} \leq \ldots \leq \eta_{i,n}, \text{with } \eta_{i,j} \text{ censored to 0 for } 1 \leq j \leq n - l_i\}
\]

where $\eta_i$ is the cross-elasticity vector of dimension $n$ for product $i$, which has $l_i$ uncensored ordered entries, with the remaining being censored. The regression output from various implements of cross-elasticity models is of this form as long as there is a single coefficient that summarises the relationship between the change of demand with respect to the change of a product’s price, which is the case in our cross-elasticity analysis. There are then systematic
trends in the post-processing analysis of data (4.5), which are as follows.

Firstly, the majority of work focuses on direct-elasticities, and in instances where cross-elasticity effects are considered, the possible competitors are usually constrained to a preselected group of products (either through expert pre-selection or a set of products determined by some variable selection technique) that are chosen to assess the substitutability of some product category of interest [??]. In such studies, the relative substitutability of products and general sensitivity to sales are assessed by raw comparisons of the direct- and cross-elasticities across the products. However, the relative sizes between the direct- and cross-elasticity coefficients are not assessed, rather the cross elasticity effects are simply compared to one another when they exist.

Secondly, the decision about how many competitors are selected in the cross-elasticity regression is generally considered secondary in such analyses. The relative importance and relevance of the inclusion (or omission) of cross competitors is determined from consideration of statistical significance, by either looking at p-values or some other equivalent variable selection method. The potential values of the omitted competitors is then ignored in the corresponding direct- and cross-elasticity analysis.

Often, much research is interested in understanding how an entire category or set of brands are affected by the price deviations of their competition or other relevant group of products [?]. In such analyses, the method of segmenting and categorising the direct- and cross-elasticity coefficients is done a priori before the analysis begins. In particular, the models are implemented across the relevant products and the direct- and cross-elasticity coefficients are aggregated across the chosen categories. The corresponding category-wise analysis of this output is generally done by studying summary statistics across these predefined segments. More concretely, given category-wise coefficient data \( X = \{(\psi_{i}^{(k)}, \eta_{i1}^{(k)}, \ldots, \eta_{im}^{(k)}) \mid (\psi_{i}^{(k)}, \eta_{i1}^{(k)}, \ldots, \eta_{im}^{(k)}) \text{ coefficients from category } k\} \), summaries for the direct- and cross-elasticities are often taken to be\( \frac{1}{N_k} \sum_{i=1}^{N_k} \psi_{i}^{(k)} \) and \( \frac{1}{N_k} \sum_{i=1}^{N_k} \eta_{ij}^{(k)} \) for some cross effect product \( j \) and where \( N_k \) is the number products in the category \( k \). These summaries are often calculated as the mechanism of conveying the aggregate price elasticity across some predefined set of products. These category-wise summary statistics are then compared relative to one another and heterogeneity, where it exists, is discussed along with the relevant implications within the given modelling context [??]. Existing approaches generally demonstrate that differences between these elasticity summary statistics across different categories of \( X_k \) do exist, which supports the hypothesis that products exhibit fundamentally different sales sensitivities.
4.2.2 Shortcomings of existing analyses

As the previous subsection illustrates, there has been much research in investigating the differences in sales sensitivities between differing products and the implications such heterogeneity has at strategic pricing and policy decision making levels. However, there are arguably some systematic weaknesses of how these direct- and cross-elasticities are analysed. Here are some of the pertinent shortcomings that we see in the current approaches of analysing these direct- and cross-elasticity coefficients that could be improved upon:

1. **Direct elasticities**: Much of the research in the price sensitivity analysis of products looks only as far as a product’s direct elasticities, i.e. a product’s sales sensitivity is summarised in terms of its own price changes. Although the primary driver of a product’s sales is its own price, there may be additional information in how a product’s sales changes with respect to changes in its relevant competitor’s price changes. Typically, these effects are frequently overlooked and it is our hypothesis that products may not only be characterised by their direct-elasticities, but also by the nature of their cross-elasticities.

2. **Relative cross elasticity magnitudes**: In the cases when cross-elasticities are considered, information about the magnitudes of leading cross-elasticity coefficients relative to the direct elasticity is not considered. Information on this decay between the direct-elasticity and leading cross-elasticity coefficients could be another way of segmenting and interpreting the differences between these sales sensitivities.

3. **Top competitor assumption**: In the majority of direct- and cross-elasticity coefficient analysis, the decision of which products qualify as significant in implementations of models such as (4.3) is often fixed a priori. Thus, any interpretation of cross-elasticity coefficients with respect to one another ought to reflect that the number and particular competitors are selected to represent the most significant competitors a product has across the entire market. This is crucially important, as the relative decay of the cross-elasticity coefficients may convey information about the values of possibly omitted regression coefficients had they been included in the original regression. This is an additional aspect that we believe products could differentiate themselves from one another with respect to their sales sensitivities.

4. **Summary statistics**: The current approaches only consider summary statistics of the elasticity coefficients. Although high-level summary statistics capture headline information on the cross-elasticity coefficients, characterising the entire distribution of these elasticity coefficients would allow experimenters to compare the similarity or dissimilarity between
4.2. Analysis of cross-elasticity coefficients

elasticity coefficients in a mixture modelling framework. Currently however, elasticity coefficients are simply eye-balled and little thought is given to a possible generating process describing the structure of the observed elasticity coefficients. The lack of such a distributional characterisation of elasticity coefficients makes it difficult to cluster and segment the sales sensitivities of products in systematic way.
This Chapter is largely based on a paper due to be published in JRSSC titled “Dirichlet Process Mixtures of Order Statistics with Applications to Retail Analytics”. arXiv:1805.05671

This Chapter presents a novel methodology that clusters products in terms of their cross-elasticity coefficients, and thus allows us to segment the universe of supermarket products in terms of their relative sales sensitivities. We achieve this by developing a Bayesian non-parametric modelling framework and interpreting our observed data of (4.5) as realisations of variable length order statistics sequences. Crucially, by reframing the data of (4.5) as variable length order statistics sequences, it allows us to specify a distributional form that characterises the cross-elasticity coefficient data. This in turn, allows us to define a mixing kernel that quantifies the degree of similarity between different cross-elasticity coefficients and therefore accommodates a mixture modelling setting. We will show this succinctly handles the partial censoring and allows for computationally straight-forward inference on the unobserved entries of the cross-elasticity matrix. Our approach uses tools from survival analysis to address inherent censoring mechanisms, together with a Dirichlet Process mixture model that allows products to be clustered into distinct groups. By using the Exponentiated Weibull distribution as a mixture kernel [?], we are able to account for both light and heavy tail behaviour apparent in the data. As we will discuss later, the Exponentiated Weibull distribution has several unique properties which makes it ideal for modelling order statistics. We develop efficient sampling mechanisms by using ?’s algorithm 8 and provide interpretations and visualisations of the fitted output. Our approach fully characterises the entire cross-elasticity vector, offering two distinct benefits. Firstly, by interpreting these elasticity vectors as order statistic sequences, we can directly cluster products by all of their cross-elasticity coefficients and conveniently handle
5.1 Relevant distributions

As discussed in the previous Chapter, the cross-elasticity data at hand is such that, for each product $i$, we observe a decreasing set of entries (i.e. observed order statistics) of a larger vector, that have been censored for sparsity purposes to only the top few entries. Mathematically speaking, the data are in the form:

$$X = \{x_{i,1:n} : x_{i,n-l_i+1} \leq x_{i,n-l_i+2} \leq \ldots \leq x_{i,n}, \text{with } x_{i,j} \text{ censored to 0 for } 1 \leq j \leq n-l_i\},$$

where $x_i$ is the cross-elasticity vector of dimension $n$ for product $i$, which has $l_i$ uncensored ordered entries, with the remaining being censored. We recast these decreasing sequences of varying length as variable length order statistics sequences, which we will go onto define along with a distribution known as Exponentiated Weibull which we propose as a suitable mixing kernel for such sequences.

5.1.1 Order statistics of continuous distributions

The order statistics of a random sample are the reordered observations in terms of increasing size. More concretely, given a continuous distribution variable $X$ and observations $x_{1:n} \overset{i.i.d.}{\sim} X$, their varying dimensions. Secondly, it provides a framework for predicting censored entries which can shed light on potentially important competitors which have been omitted from the original regression. Hence, our approach neatly handles the challenges outlined in 4.1.1.1. We implement our proposed methodology on three datasets, two simulated examples and one from real cross-elasticity data generated from a large UK supermarket retailer's regression model output which we were given access to through dunnhumby's secured servers. We show that our proposal successfully partitions the space of products in terms of their sales sensitivities.
the order statistics \( x_{(1)}, \ldots, x_{(n)} \) are given by:

\[
x_{(1)} < x_{(2)} < \ldots < x_{(n)}.
\]  

(5.1)

The \( j \)th order statistic of (5.1) is denoted as \( x_{(j)} \) and thus, \( x_{(1)} \) and \( x_{(n)} \) are the smallest and largest observations respectively. Given a density function \( f(x) \) of a continuous random variable \( X \), the density of the \( j \)th order statistic \( x_{(j)} \), denoted by \( f_{(j)}(x) \) is given by [7]:

\[
f_{(j)}(x) = n f(x) \binom{n-1}{j-1} F(x)^{j-1} (1 - F(x))^{n-j}.
\]

(5.2)

An implicit assumption of the cross-elasticity coefficient data is that the elasticities represent the top competitors a product encounters throughout the entire market, i.e. these cross-elasticity coefficients are the largest in magnitude a product will encounter across all of its competitors.

Hence, we then make the following assumption that the partially observed cross-elasticity vector \( x_{i,1:n} \), of length \( n \) with \( l_i \) non-zero entries in fact corresponds to the top \( l_i \) order statistics of a random sample of size \( n \). We term each of these vectors of the top \( l_i \) order statistics as **variable length order statistics sequences**, and denote them as \( x_{i,1:n} = (x_{i,(n)}, \ldots, x_{i,(n-(l_i-1))}) \). We also denote the \( j \)th order statistic of sequence \( x_{i,1:n} \) by \( x_{i,(j)} \). For notational ease, we drop the \( i \) index for the remaining of this section. The density of \( x \mid l \) denoted as \( f_{(n):(n-l+1)} \) is given by:

\[
f_{(n):(n-l+1)}(x \mid l) = f_{(n):(n-l+1)}(x_{(n)}, \ldots, x_{(n-l+1)} \mid l)
= \frac{n!}{(n-l)!} F(x_{(n-(l-1))})^{n-l} \prod_{j=1}^{l} f(x_{(n+j-l)}).
\]

(5.3)

By the independence of \( x_{(n-j)} \mid x_{(n-j+1)}, \ldots, x_{(n-j+l)} \) and by (5.3), the density of the conditional distribution of \( x_{(n-j)} \mid x_{(n-j+1)}, l \) for \( j < l \) (denoted as \( f_{(n-j)|(n-j+1)} \)) is given by:

\[
f_{(n-j)|(n-j+1)}(x_{(n-j)} \mid x_{(n-j+1)}, l) = (n-j) f(x_{(n-j)}) \frac{F(x_{(n-j)})^{n-j}}{F(x_{(n-j+1)})^{n-j}}
\]

(5.4)

and thus the density of the joint sample \( x \mid l \) can also be expressed in hierarchical format:

\[
f_{(n):(n-l+1)}(x \mid l) = f(x_{(n)}) \prod_{j=1}^{l-1} f_{(n-j)|(n-j+1)}(x_{(n-j)} \mid x_{(n-j+1)}, l)
\]

(5.5)
5.1. Relevant distributions

Finally, the joint density of a variable length order statistics sequence \( \mathbf{x} \), denoted as \( f_{\text{vloss}} \), can therefore be expressed as:

\[
f_{\text{vloss}}(\mathbf{x}, l) = f_{\text{vloss}}(l) \times f_{\text{vloss}}(\mathbf{x} | l) = p(l) \times f_{(n):n-l+1}(x | l)
\]

(5.6)

since \( f_{\text{vloss}}(\mathbf{x} | l) \) (the density of an observed vector of order statistics \( \mathbf{x} | l \)) is precisely \( f_{(n):n-l+1}(x | l) \), and where \( f_{\text{vloss}}(l) \) is simply the probability mass function over the length of the sequence, which we denote as \( p(l) \). Here we assume that \( l \) and the non-zero entries of \( \mathbf{x} \) are independent.

Much work has been done in the study of the theoretical properties of order statistics [?], from which they have been applied to areas such as modelling the reliability of software and to the modelling of recommender systems [??]. A relevant field of order statistics which bears resemblance to our problem set-up lies in the field of reliability analysis, known as \( k \)-out-of-\( n \) systems. A \( k \)-out-of-\( n \) system models the failure of \( k \) out of \( n \) components within a finite time horizon. The set of \( k \) ordered values of the time until failure (censored or not) can then be modelled as the observed order statistics of a base distribution. Much of the relevant non-parametric work has focused on flexibly learning the underlying base distributions [??] and building hierarchical versions of these models [?].

In the current context, we observe the top few order statistics of the cross-elasticity vector, with the remaining entries treated as missing. This type of data is akin to the format of models in survival analysis, where the probability of survival decreases over time and may be right-censored. One aspect important to the success of Bayesian non-parametric models in survival analysis is the choice of kernel, as it impacts whether the relevant statistics and survival functions are recoverable. As a consequence, much attention is paid to the choice of kernel. Notably, a hierarchical structure in the base measure was introduced by (?, whereas ? and ? used Gamma and Weibull kernels within a Dirichlet process mixture model framework respectively. The Exponentiated Weibull distribution was shown to be a distribution that could model non-monotone hazards [?], which in our context correspond to order statistics terms whose modes exist but are not necessarily light-tailed.
5.1.2 Exponentiated Weibull distribution

Following the formulation of our observations as order statistics of random samples, the choice of the underlying distribution of $X$ will determine the behaviour of the corresponding order statistics. Here we are interested in a distribution which can allow for a range of light and heavy tail behaviour and provide interpretable analytical expressions for the distribution of its order statistics. We thus assume that these random samples are distributed according to the Exponentiated Weibull distribution. A random variable $X$ is distributed according to the Exponentiated Weibull (EW) distribution, denoted as $X \sim EW(\alpha, \beta, \lambda)$, if its probability density and distribution function are given by

$$f(x) = \alpha \beta \lambda^\beta x^{\beta - 1} \left(1 - e^{-(\lambda x)^\beta}\right)^{\alpha - 1} e^{-(\lambda x)^\beta} \quad \text{(5.7)}$$

and

$$F(x) = \left(1 - e^{-(\lambda x)^\beta}\right)^\alpha \quad \text{(5.8)}$$

respectively, where $x > 0$, $\lambda > 0$, $\beta > 0$, $\alpha > 0$. The Exponentiated Weibull is an extension to the standard Weibull distribution through the inclusion of the additional parameter $\alpha$, which allows the distribution to have a wide range of tail behaviours. Similarly to the Weibull distribution, $\lambda$ is a scale parameter whereas $\beta$ controls the tail behaviour of the distribution; distributions are heavy tailed for $\beta < 1$ and light-tailed otherwise. Furthermore, decreasing $\beta$ monotonically increases the mean and variance, kurtosis and skew of the EW distribution. The impact of $\alpha$ depends on both the value $\alpha \beta$ and whether $\alpha < 1$; increasing $\alpha$ increases symmetry around the mean and mode. These different modal, asymptotic and tail behaviours are summarised in Table 5.1. Figure 5.1 demonstrates various density plots for differing combinations of $(\alpha, \beta, \lambda)$, various asymptotic, modal and tail behaviours are observed.

<table>
<thead>
<tr>
<th>Ranges of $\alpha, \beta$</th>
<th>$x \to 0$</th>
<th>Mode</th>
<th>Order statistic marginal tails</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha &gt; 1, \beta &gt; 1, \alpha \beta &gt; 1$</td>
<td>$f(x) \to 0$</td>
<td>$\approx \frac{1}{\lambda} \left[\frac{2(\alpha \beta - 1)}{\beta (\alpha + 1)}\right]^{1/\beta}$</td>
<td>Light</td>
</tr>
<tr>
<td>$\alpha &gt; 1, \beta &lt; 1, \alpha \beta &gt; 1$</td>
<td>$f(x) \to 0$</td>
<td>$\approx \frac{1}{\lambda} \left[\frac{2(\alpha \beta - 1)}{\beta (\alpha + 1)}\right]^{1/\beta}$</td>
<td>Heavy</td>
</tr>
<tr>
<td>$\alpha &gt; 1, \beta &lt; 1, \alpha \beta &lt; 1$</td>
<td>$f(x) \to \infty$</td>
<td>none</td>
<td>Heavy</td>
</tr>
<tr>
<td>$\alpha &lt; 1, \beta &gt; 1, \alpha \beta &lt; 1$</td>
<td>$f(x) \to \infty$</td>
<td>none</td>
<td>Light</td>
</tr>
<tr>
<td>$\alpha &lt; 1, \beta &gt; 1, \alpha \beta = 1$</td>
<td>$f(x) \to \lambda$</td>
<td>$0$</td>
<td>Light</td>
</tr>
</tbody>
</table>

5.1.3 EW distribution application to order statistics

There are some key properties of the EW distribution that lead to useful applications to order statistics and variable length order statistics sequences. The joint density of (5.3) under the EW
5.1. Relevant distributions

The distribution for fixed order sequences of lengths $l$ is given by:

$$f_{(\alpha, \beta, \lambda)}(x | l) = \frac{n!}{(n-l)!} \left(1 - e^{-(\lambda x(n-l))^{\beta}}\right)^{n-l} \prod_{j=1}^{l} f(x_{(n+j-l)})$$  \quad (5.9)$$

where $f$ is the EW density function of (5.7). The EW distribution handles censoring naturally, since the censored, joint and conditional densities under the EW distribution belong to the same family, i.e. $x_{(n-j)} | x_{(n-j+1)} \sim EW_{x(n-j) < x(n-j+1)} ((n-j) \alpha, \beta, \lambda), 1 \leq j \leq n-1$ are also readily available. This means that the properties and interpretability of the EW distribution transparently carry over to its order statistics. Finally, the EW can account for both light and heavy tails, allowing us to capture different types of decay behaviours of the elasticity vectors. Figure 5.2 provides some examples of order statistics sequences, which demonstrate various decay behaviours and tail behaviours that can be produced under the EW kernel.

Figure 5.1: EW density for $(\alpha, \beta, \lambda) = (1.2, 0.8, 1.0)$ [black solid], (1.55, 0.8, 1.0) [blue dashed], (0.24, 5.0, 1.0) [red dotted] and (1.8, 1.4, 0.5) [green dashed-dotted lines] respectively.

Figure 5.2: Left panel: realisations of order statistics sequences with $EW(\alpha, \beta, \lambda)$ kernel for combinations $(\alpha, \beta, \lambda) = (0.2, 0.6, 0.7)$ [black solid], (0.5, 1.5, 1.5) [blue dashed], (4, 5, 1.5) [red dotted] respectively. Right panel: Density plots of $f(k)(x)$ with EW(0.5, 1.5, 1.5) kernel for orders $k=10$ [dotted], 9 [dashed] and 8 [solid].
5.2 Model

Our ultimate goal is to characterise the behaviour of different products in terms of their cross-elasticity coefficients. To this end, we use the EW distribution as a representation of cross-elasticity decay behaviour. However, in order to account for different behaviour across products, we additionally cluster products that potentially correspond to the same EW distribution. We thus model the entire set of cross-elasticity vectors non-parametrically as a Dirichlet Process Mixture Model \[?] as outlined in the section 3.5.1.

5.2.1 Nonparametric mixture model of variable length order statistic sequences

We now propose a DPMM of variable length order statistics sequences on mixtures of distributions satisfying (5.9). Placing a \(\text{DP}(\nu G_0)\) on the distributions of (5.9) is an attractive approach to handling the complex multi-modalities, decay rates and variable lengths that order statistics sequences can exhibit as discussed in Section 5.1.2. Thus, the DPMM of variable length order statistics sequences expressed in hierarchical format of (5.5) by:

\[
\begin{align*}
\nu & \sim \text{Gamma}(\tau_1, \tau_2), \\
G & \mid \nu \sim \text{DP}(\nu G_0), \\
(\alpha_i, \beta_i, \lambda_i, w_i) & \mid G \sim G, \\
l_i & \sim 1 + \text{Binomial}(n - 1, w_i), \\
x_{i,j} & \sim \text{EW}(\alpha_i, \beta_i, \lambda_i), \ j = 1, \ldots, n,
\end{align*}
\]

(5.10)

where \(i = 1, 2, \ldots, N\) are the number of observations and for each observation vector \(i\), with all but the top \(l_i\) entries being censored. The final line of (5.10) can also be expressed through the iterative formulation:

\[
\begin{align*}
x_{i,(n-j)} \mid x_{i,(n-j+1)} & \sim \text{EW}_{x_{i,(n-j)}<x_{i,(n-j+1)}} \left( (n-j)\alpha_i, \beta_i, \lambda_i \right), \ 1 \leq j \leq l_i - 1 \\
x_{i,(n)} & \sim \text{EW}(n\alpha_i, \beta_i, \lambda_i).
\end{align*}
\]

(5.11)

which follows from equation (5.5). We treat the lengths \(l\) and observations \(x_{i,(j)}\) of \(x\) as independent to allow detection of competitor omissions and to ease computation. Since cross-elasticity coefficients are identically distributed a priori, each individual coefficient has the same probability of being censored, leading to a Binomial prior on \(l_i\); to avoid the degenerate case of empty cross-elasticity vectors, we force one of the Bernoulli trials to be 1. It important to note, that \(w\) plays an important role in (5.10) that is particularly relevant to our modelling setup. The inclusion of \(w\) as a cluster level parameter, rather than being a global parameter, is especially important for the following two reasons. Firstly, a characterising feature of cross-
5.2. Model

Elasticity coefficient vectors are their lengths, not just the relative decay between entries (which is captured by \( \alpha, \beta, \lambda \) parameters). These lengths of the cross-elasticity coefficient vectors have an important retail analytics interpretation, in that they convey the number of significant competitors a product has throughout the market - which is typically seen as one of the defining features of a product (and group of products). Secondly, \( w \) conveys the level of truncation, and thus implicitly the number of censored observations. For clusters with few censored coefficients, we expect larger values of cluster-wise \( w \) values. Consequently, the cluster-wise variable \( w \) has important ramifications in section of 5.4.4.1 (and subsequent sections), where we define two statistics relevant to our retail analytics analysis; namely the omitted competitors (OC) and aggregate competition (AC) statistics. These statistics strive to respectively describe the relative magnitude of possible omitted cross-elasticity coefficients and the total effect of competition a product receives throughout the market (detailed definitions of these are provided in section 5.4.4.1). Both definitions crucially rely on an accurate summarisation of the number of non-censored coefficients a cluster typically observes. Consequently, a cluster-wise variable of \( w \) is key for this accurate summarisation, with possible misfit being likely if \( w \) were a global parameter.

The base distribution \( G_0 \) is a key aspect of the DP(\( \nu G_0 \)) as it specifies the prior over \((\alpha, \beta, \lambda, \omega)\) atoms which defines the cluster structure of the model; here we specify \( G_0 \) as:

\[
G_0 (\alpha, \beta, \lambda, w) = \text{Gamma} (\alpha | \alpha_1, \alpha_2) \times \text{Gamma} (\beta | \beta_1, \beta_2) \times \\
\times \text{Gamma} (\lambda | \lambda_1, \lambda_2) \times \text{Beta} (w | a, b). \quad (5.12)
\]

The hyperparameters \((a, b, \alpha_1, \alpha_2, \beta_1, \beta_2, \lambda_1, \lambda_2)\) are treated as fixed, chosen depending on the modelling context and reflecting prior expertise. The prior for \( \nu \) is assumed to be Gamma \((\tau_1, \tau_2)\), allowing the relation \( \mathbb{E} [N^* | \nu] = \nu \log \left( \frac{\nu + N}{\nu} \right) \) (where \( N^* \) is the number of occupied clusters) to inform our prior expectation of the number of clusters. As discussed during section 3.5.2, and illustrated by the simulation study of figure 3.1, the \( \nu \) parameter of DP(\( \nu G_0 \)) acts as smoothing parameter controlling the degree of ‘smoothness’ of density estimates, and equivalently, the number of unique mixture components induced by the DP(\( \nu G_0 \)) prior. Consequently, it is important to place uncertainty over the \( \nu \) parameter, especially in the context of variable length order statistic sequences where it is not clear how to fix \( \nu \), as it not obvious how many unique clusters will exist in the retail analytics dataset. Although it should be noted, in some modelling contexts, the number of unique clusters is often driven by the data.
5.3 Posterior inference

We now present an efficient Markov Chain Monte Carlo (MCMC) procedure for obtaining samples from the posterior of \( p(\alpha, \beta, \lambda, w, \nu \mid X) \) according to the model proposed by (5.10) with:

\[
X = \{ x_{i,1:n} : x_{i,n-l_i+1} \leq x_{i,n-l_i+2} \leq \ldots \leq x_{i,n}, \text{with } x_{i,j} \text{ censored to 0 for } 1 \leq j \leq n-l_i \},
\]

where \( x_i \) includes the variable length order statistics sequence of length \( l_i \) (uncensored ordered entries), with the remaining \((n-l_i)\) being censored. Our posterior inference methodology as outlined in section 3.6, consists of three steps to obtaining samples from \( p(\alpha, \beta, \lambda, w, \nu \mid X) \) for each MCMC iteration: sampling the atoms \((\alpha, \beta, \lambda, w)\) of the DP \((\nu G_0)\) for each order statistics sequence; sampling the cluster-wise atoms for each of the unique clusters (as induced by \( DP(\nu G_0) \)), and finally, sampling the \( \nu \) scale parameter. As discussed during section 3.6, this DPMM inference methodology is simple, intuitive and circumvents issues relating to the infinite dimensional nature of the DP by marginalising over the random measure \( G \), and then updating the cluster allocations of the data-points. These three steps are manifest as follows:

5.3.1 Sample from \( p(\theta_i \mid \theta_{-i}, \nu, x_i) \)

We initiate by using the Polya urn exposition of a DP [?] by taking a Gibbs sample of \( \theta_i = (\alpha_i, \beta_i, \lambda_i, w_i) \) atoms associated to observation \( x_i \) using:

\[
p(\theta_i \mid \theta_{-i}, \nu, X) = q_0^* H_i + \sum_{k=1}^{N^*} q_k^* \delta \theta_k^* \tag{5.13}
\]

where \( q_0^* \propto \nu \int f(x_i \mid \theta) G_0 (d\theta) \) and \( q_k^* \propto N_k^* f(x_i \mid \theta_k^*, \nu) \) subject to \( \sum_{k=0}^{N^*} q_k^* = 1 \). Here \( f(x_i \mid \theta) = f_{(n):(n-l_i+1)}(x_i \mid l_i, \alpha, \beta, \lambda) p(l_i \mid w) \), where \( f_{(n):(n-l_i+1)} \) is specified in (5.9) and the conditional distribution \( p(l_i \mid w) = \binom{n-1}{l_i-1} w^{l_i-1} (1-w)^{(n-l_i)} \). \( H_i \) is the posterior distribution for \( \theta \) based on the prior distribution \( G_0 \) of (5.12) with likelihood \( f(x_i \mid \theta, \nu) \). Here \( \theta_{-i} \) denotes the vectorised atoms of \( \theta \) excluding the \( i^{th} \) atom \( \theta_i \), \( \{ \theta_1^*, \ldots, \theta_{N^*}^* \} \) denotes the unique values of \( \theta_i \), \( N^* \) the number of unique clusters induced by the DP and \( N_k^* \) the number of points assigned to atom \( \theta_k^* \).

As calculating the integral \( q_0^* \) is intractable, we use algorithm 8 [?] to approximate \( q_0^* \) by a weighted mixture of likelihoods by taking \( c \) auxiliary components sampled from the prior distribution \( G_0 \). Concretely, we sample \( \theta_i = (\alpha_i, \beta_i, \lambda_i, w_i) \) by sampling from the multinomial
distribution of degrees of freedom of order $N^* + c$ with entries

$$\theta_k^* \overset{i.i.d.}{\sim} G_0 \text{ for } k = N^* + 1, \ldots, N^* + c$$

$$G_0 = Beta(w \mid a, b) \times Gamma(\alpha \mid \alpha_1, \alpha_2) \times Gamma(\beta \mid \beta_1, \beta_2) \times Gamma(\lambda \mid \lambda_1, \lambda_2)$$

with probabilities

$$P(\theta_i = \theta_{i,k}^* \mid \theta_{-i}, x_i, \theta_{1}^*, \ldots, \theta_{N^*+c}^*) \propto \begin{cases} \frac{N^*_i}{N-1+p} f(x_i \mid \theta_{i,k}^*) & \text{for } 1 \leq k \leq N^* \\ \frac{w_i}{N-1+p} f(x_i \mid \theta_{i,k}^*) & \text{for } N^* < k \leq N^* + c \end{cases}$$

where

$$f(x_i \mid \theta_{i,k}^*) = \binom{n-1}{l_i-1} (w_{k_i}^*)(l_i-1) (1-w_{k_i}^*)(n_l-l_i) \times F(x_{i,(n_l-l_i)}) \prod_{j=1}^{l_i} f(x_{i,(n_l-l_i)-j}) \alpha_{k_i}^* \beta_{k_i}^* \lambda_{k_i}^*$$

(5.14)

where $F(x \mid \alpha, \beta, \lambda) = \left(1 - e^{-(\lambda x)^\beta}\right)^\alpha$ and $f(x \mid \alpha, \beta, \lambda) = \alpha \beta x^{\beta-1} \left(1 - e^{-(\lambda x)^\beta}\right)^{-1} e^{-(\lambda x)^\beta}$.

The number auxiliary components $c$ chosen determines the level $q_0^*$ is approximated to.

### 5.3.2 Sample from $p(\alpha^*, \beta^*, \lambda^*, w^* \mid \nu, x_{i;C_i=k})$

The $\theta_k$ atoms are then updated for each of the unique clusters $k = 1, \ldots, N^*$ to avoid inefficiencies associated with having to pass through extremely low probability states to get to a higher probability states. This is achieved by updating $\theta_k$ to be a single sample generated from the posterior $p(\theta_k \mid \nu, x_{i;C_i=k})$, for each $k = 1, \ldots, N^*$. As taking exact samples from $p(\theta_k \mid \nu, x_{i;C_i=k})$ is intractable for our choice of kernel (5.9) and prior $G_0$ (5.12), the Metropolis Hastings algorithm is used to sample from $p(\theta_k \mid \nu, x_{i;C_i=k})$, for $k = 1, \ldots, N^*$.

This involves following $t_{MH}$ iterations of the Metropolis Hastings procedure, and saving the final $t_{MH}^{th}$ sample as $\theta_k$, for each $k = 1, \ldots, N^*$. The primary motivation of these Metropolis Hastings updates is to perturb the unique locations of $\{\theta_{1}^*, \ldots, \theta_{N^*}^*\}$, for $k = 1, \ldots, N^*$, and thus avoiding the Polya Urn sweeps getting stuck, rather than obtaining a complete representation of the entire posterior distribution specified by $p(\theta_k \mid \nu, x_{i;C_i=k})$.

The precise Metropolis Hastings procedure is as follows: to ease notation, we suppress the asterisks from the exponents in this subsection. For $t_{MH} = 1, \ldots, T_{MH}$ iterations, we draw new parameters using a Normal proposal for $(\alpha_k', \beta_k', \lambda_k', w_k')$ centred at the current points...
Chapter 5. Elasticity clustering using Dirichlet process mixtures

\((\alpha^{tMH}_k, \beta^{tMH}_k, \lambda^{tMH}_k, w^{tMH}_k)\) with standard deviations \(\sigma_{\alpha}, \sigma_{\beta}, \sigma_{\lambda}, \sigma_w\) respectively:

\[
(\alpha^{tMH}_k, \beta^{tMH}_k, \lambda^{tMH}_k, w^{tMH}_k) \sim N(\alpha^{tMH}_k, \sigma_{\alpha}^2) \times N(\beta^{tMH}_k, \sigma_{\beta}^2) \times N(\lambda^{tMH}_k, \sigma_{\lambda}^2) \times N(w^{tMH}_k, \sigma_w^2)
\]  
(5.15)

Then, with probability

\[
a = \min \left(1, \frac{\pi(\alpha_k', \beta_k', \lambda_k', w_k' | x_{\{i:C_i=k\}})}{\pi(\alpha_k, \beta_k, \lambda_k, w_k | x_{\{i:C_i=k\}})} \right),
\]

set \((\alpha^{tMH+1}_k, \beta^{tMH+1}_k, \lambda^{tMH+1}_k, w^{tMH+1}_k) = (\alpha_k', \beta_k', \lambda_k', w_k')\)

otherwise \((\alpha^{tMH+1}_k, \beta^{tMH+1}_k, \lambda^{tMH+1}_k, w^{tMH+1}_k) = (\alpha^{tMH}_k, \beta^{tMH}_k, \lambda^{tMH}_k, w^{tMH}_k)\).

Here

\[
\pi(\alpha_k, \beta_k, \lambda_k, w_k | x_{\{i:C_i=k\}}) \propto \alpha_k^{a-1}e^{-\alpha_k \lambda_k} \times \beta_k^{b-1}e^{-\beta_k \lambda_k} \times \lambda_k^{1-n}e^{-\lambda_k^2} \times w_k^{a-1}(1-w_k)^{k-1} \\
\times \prod_{x_{i:C_i=k}} \left[ w_k^{(i-1)}(1-w_k)^{(n-i)} F(x_{i,(n-(i-1))} | \alpha_k, \beta_k, \lambda_k) \right].
\]

This is performed for each of the unique clusters \(k = 1, \ldots, N^*\), and initiated after \(t_{init}\) iterations of step 5.3.1. This is to allow the \(\theta_i = (\alpha_i, \beta_i, \lambda_i, w_i)\) atoms produced from the Polya urn sampling to settle into the appropriate number of unique clusters. If the Metropolis Hastings iterations are initiated immediately after the first sweep of step 5.3.1 (i.e. \(t_{init} = 0\)), then the sampler can spend unnecessary time updating the unique locations of many \(\{\theta_i^1, \ldots, \theta_i^{N^*}\}\) induced by the DP, which is typically large in \(N^*\) during the initial phases of the Polya urn sweeps of 5.3.1. This is ultimately unnecessary, as typically these unique locations of \(\{\theta_i^1, \ldots, \theta_i^{N^*}\}\) eventually get merged into larger grouping of locations during the latter phases of completed cycles of step 5.3.1. The scales of the proposal normal distributions \((\sigma_w, \sigma_\alpha, \sigma_\beta, \sigma_\lambda)\) should be tuned depending on the dataset. The details of this tuning will likely involve paying attention to the acceptance rate during the iterations of the Metropolis Hasting algorithm across each of the unique clusters \(k = 1, \ldots, N^*\). The primary mechanism at optimising this acceptance rate during this research is via \((\sigma_w, \sigma_\alpha, \sigma_\beta, \sigma_\lambda)\) tuning parameters, as these parameters control the jump size proposed during the random walk outlined in (5.15). If these standard deviations \((\sigma_w, \sigma_\alpha, \sigma_\beta, \sigma_\lambda)\) are too small, then the acceptance rate will be too high and thus likely will result in highly autocorrelated samples. Alternatively, if these standard deviations \((\sigma_w, \sigma_\alpha, \sigma_\beta, \sigma_\lambda)\) are too large, then the acceptance will be too low as the sampler will be ‘stuck’ at its current position.

Much work has been done into the performance of the Metropolis Hasting algorithm around...
effective proposal distributions and parameter tuning, but a good guide for an optimal acceptance rate was proposed by ???, and this was the approach adopted during this research. Other more sophisticated Metropolis Hasting approaches can be adopted over the (5.15) proposals, for example, an adaptive Metropolis Hastings algorithm that adapts the standard deviations of \((\sigma_w,\sigma_\alpha,\sigma_\beta,\sigma_\lambda)\) during the iterations according to the acceptance rate as proposed by ??? could be beneficial depending on the modelling circumstances.

5.3.3 Sample from \(p(\nu | \theta_1, \ldots, \theta_N, N^*, X)\)

As discussed during section 3.5.2, and illustrated by the simulation study of figure 3.1, the \(\nu\) parameter is key in contributing to the extent of smoothness of density estimates as well as the number of unique clusters induced from \(G \sim \text{DP}(\nu G_0)\) samples. Consequently, we place uncertainty over the \(\nu\) parameter, and specify \(\nu \sim G(\tau_1, \tau_2)\). This prior specification, when augmented with an additional auxiliary variable \(\gamma\) (the definition of which will be detailed in the preceding discussion), induces a convenient conjugacy property that allows straight-forward Gibbs sampling for \(\nu\).

Assuming a continuous prior on \(\nu\), ??? showed:

\[
p(\nu | \theta_1, \ldots, \theta_N, N^*, X) = p(\nu | N^*) \\ \propto p(\nu)p(N^* | \nu)
\]  

(5.16)

since the data \(X\) is conditionally independent of \(\nu\), given \(N^*\) and locations \(\theta_1, \ldots, \theta_N\) (and thus partition proportions), and furthermore, since the locations \(\theta_1, \ldots, \theta_N\) are conditionally independent of \(\nu\), given \(N^*\) and data \(X\). From (5.16), we notice a simple Gibbs procedure can be derived, i.e. given \(\nu\), we resample the \(\theta_1, \ldots, \theta_N\) parameters (according the procedural steps outlined as in sections 5.3.1 and 5.3.2) and hence \(N^*\). Then at each iteration, we sample from the condition posterior of \(p(\nu | N^*)\). ??? further showed when \(\nu \sim G(\tau_1, \tau_2)\), (5.16) can be re-expressed as:

\[
p(\nu | N^*, X, \theta_1, \ldots, \theta_N) \propto p(\nu)p(N^* | \nu) \\ \propto p(\nu)\nu^{N^*}(\nu + N) \int_0^1 x^\nu(1-x)^{N-1}dx. 
\]  

(5.17)

The key observation about (5.17) is that the distribution of \(p(\nu | N^*)\) is in fact the marginal distribution of a joint distribution for \(\nu\) and another continuous variable \(\gamma\) (the auxiliary variable) such that \(p(\nu, \gamma | N^*) \propto p(\nu)\nu^{N^*-1}(\nu + N)\gamma^\nu(1-\gamma)^{N-1}\) for \(\nu > 0\) and \(\gamma \in (0,1)\). ??? finally
showed the conditional distributions of \( p(\nu | \gamma, N^*) \) and \( p(\gamma | \nu, N^*) \) were given by:

\[
p(\gamma | \nu, N^*) \propto \gamma^\nu (1 - \gamma)^{N^*-1} \\
p(\nu | \gamma, N^*) \propto \nu^{\tau_1 + N^*-1} e^{\nu(\tau_2 - \log(\gamma)) + N} + \nu^{\tau_1 + N^*-2} e^{\nu\tau_2 - \log(\gamma)) + N}
\]

(5.18)

which can easily be recognised as a beta density and a mixture of gamma densities respectively. Thus, to sample \( \nu \) at each Gibbs iteration with current values of \( N^* \) and \( \nu \), we initially sample the auxiliary variable \( \gamma \) from the beta distribution specified in (5.18), then conditioned on this \( \gamma \) and \( N^* \), we sample a new \( \nu \) from the mixture of Gamma distributions specified in (5.18).

More concretely, by specifying \( \nu \sim G(\tau_1, \tau_2) \), and introducing the auxiliary variable \( \gamma \), we then take the following samples:

\[
(\gamma | \nu, N^*) \sim \text{Beta}(\nu + 1, N) \\
(\nu | \gamma, N^*) \sim \pi_\gamma G(\tau_1 + N^*, \tau_2 - \log(\gamma)) + (1 - \pi_\gamma) G(\tau_1 + N^* - 1, \tau_2 - \log(\gamma))
\]

where the weights \( \pi_\gamma \) is defined by \( \pi_\gamma / (1 - \pi_\gamma) = (\theta + N^* - 1) / (N (\tau_2 - \log(\gamma))) \) after each of the steps outlined in sections 5.3.1 and 5.3.2. This concludes one complete iteration of our posterior inference procedure.

This three step procedure of 5.3.1, 5.3.2 and 5.3.3 is followed for both the real analytics and simulated studies presented in the subsequent sections. It should be noted, that the tuning of the MCMC parameters \( (t_{\text{init}}, c, \sigma_w, \sigma_\alpha, \sigma_\beta, \sigma_\lambda, T_{\text{MH}}) \) - the iterations waited till the Metropolis Hastings procedure is initiated, the number auxiliary components used in during Neal’s algorithm of step 5.3.1, the standard deviations of the proposal distributions and the number samples taken during the Metropolis Hasting algorithm of step 5.3.2 - should ultimately be tuned to produce good mixing amongst the final sampled atoms of \( (\alpha_k^t, \beta_k^t, \lambda_k^t, w_k^t) \), across the index of the MCMC sampler \( t = 1, \ldots, T \) (\( T \) being the total number of MCMC samples) as well as across the unique atoms \( k = 1, \ldots, N_t^* \), where \( N_t^* \) is the number of unique clusters at iteration \( t \). During the retail analytics and simulation studies described in this research, these MCMC tuning parameters are selected as \( (t_{\text{init}}, c, \sigma_w, \sigma_\alpha, \sigma_\beta, \sigma_\lambda, T) = (200, 150, 0.009, 0.05, 0.02, 0.05, 5) \) for the retail analytics dataset, and \( (t_{\text{init}}, c, \sigma_w, \sigma_\alpha, \sigma_\beta, \sigma_\lambda, T) = (100, 100, 0.018, 0.010, 0.015, 0.015, 100) \) for the simulation studies.
5.4 Results

We now illustrate how our methodology works in practice by performing two simulation studies, before proceeding to a real retail analytics dataset. The first example (subsection 5.4.1) generates data from our model. The second example (subsection 5.4.2) generates data using a Gamma distribution as a kernel (rather than EW). We fit our model to both datasets using vague priors.

5.4.1 Simulated data 1

We generate data using parameters for the mixtures of (5.19) which demonstrate the various behaviours that variable length order statistics sequences from an EW kernel can exhibit, namely a mixture of light and heavy tails with varying rates of order statistics terms $x_i, (20)$ convergence to $0$, lengths, different decay rates and varying modal behaviours. Specifically, we draw 1500 samples from the following DPMM of variable length order statistics sequences of (5.10):

$$G = 0.4\delta_{\theta_1} + 0.35\delta_{\theta_2} + 0.25\delta_{\theta_3}$$
$$\begin{align*}
&(\alpha_i, \beta_i, \lambda_i, w_i) \mid G \sim G, \ i = 1, \ldots, 1500, \\
&l_i \sim 1 + \text{Binomial}(19, w_i), \ i = 1, \ldots, 1500, \\
&x_{i,j} \sim \text{EW}(\alpha_i, \beta_i, \lambda_i), \ j = 1, \ldots, 20,
\end{align*}
$$

where $\theta = (\alpha, \beta, \lambda, w)$, with $\theta_1^* = (0.15, 0.8, 0.91, 0.65)$, $\theta_2^* = (2.5, 3.3, 0.35, 0.75)$, $\theta_3^* = (0.64, 1.7, 0.4, 0.9)$. Thus, for each variable length order statistics sequences observation vector $i$, the first $l_i$ entries (i.e. variability in truncation between observations) correspond to the top $l_i$ order statistics of the random sample $x_{i,j}, j = 1, \ldots, 20$, with the remaining entries being censored.

5.4.2 Simulated data 2

This simulated example differs from the former simulation study in that the data is simulated from a mixture of gamma distributions rather than a mixture EW distributions. The purpose of fitting our model to a mixture of Gamma distributions instead of a mixture of EW distributions is to test the inference in a less optimistic setting and establish whether the EW kernel is sufficiently flexible to capture the decay of order statistics sequences from a set of mixtures that are not a mixture of EW distributions. The mixture components $\theta_1^*, \theta_2^*, \theta_3^*$ are selected to produce simulated mixtures that imitate the mixtures of (5.19).

We generate 1500 samples from the following DPMM of variable length order statistics se-
sequences from the following mixture model

\[
G = 0.4\delta_{\theta_1} + 0.35\delta_{\theta_2} + 0.25\delta_{\theta_3},
\]

\[ (\alpha_i, \beta_i, w_i) \mid G \sim G, \ i = 1, \ldots, 1500, \]

\[ l_i \sim 1 + \text{Binomial}(19, w_i), \ i = 1, \ldots, 1500, \]

\[ x_{i,j} \sim \text{Gamma}(\alpha_i, \beta_i), \ j = 1, \ldots, 20, \]

where \( \theta = (\alpha, \beta, w) \), with \( \theta_1^* = (0.15, 0.5, 0.65) \), \( \theta_2^* = (1.7, 1.0, 0.75) \), \( \theta_3^* = (32, 10, 0.9) \), similarly as before.

### 5.4.3 Prior distributions and posterior sampling

We fit our EW mixture model using the following vague priors for \( (\alpha, \beta, \lambda, w) \) and \( \nu \) (DP scale parameter):

\[
(\alpha, \beta, \lambda, w) \sim \text{Gamma}(1, 0.1) \times \text{Gamma}(1, 0.1) \times \text{Gamma}(1, 0.1) \times \text{Beta}(1, 1)
\]

\[ \nu \sim \text{Gamma}(1, 1) \]

respectively.

The priors for \( \alpha \), \( \beta \) and \( \lambda \) imply a mean of 10 and variance 100, a rather vague choice centred away from the true values. The Beta prior for \( w \) corresponds to a uniform distribution, assuming no prior information about the number of non-censored entries. The motivation of these priors is to establish the effectiveness of inferential procedure, and we do this by specifying vague priors relative to the known true values of the mixture components. We use the steps outlined in Section 5.3 for parameter inference and perform 9000 MCMC iterations with 1000 burn-in, and thin every 9 samples. We present the MCMC output based on the inference methodology of Section 5.3 on the simulated data of (5.4.1) and (5.4.2).

Since individual clusters are not identifiable (up to permutations), an additional identifiability criterion is required in order to perform cluster-wise inference. \( \theta \) proposed an approach of estimating the optimal co-memberships partitions based on a Bayesian clustering regime that involved the posterior coincidence probability matrix \( \rho_{ij} = P(C_i = C_j) \) (computed as \( \rho_{ij} = \frac{1}{S} \sum_{s=1}^{S} \mathbb{1}[\theta^s_i = \theta^s_j] \) where \( S \) is the number of MCMC samples and \( \theta^s_i \) is the location of \( i^{th} \) data point at the \( s^{th} \) sample). They in-particular proposed minimising the linear loss function of the posterior expected loss of the posterior marginal coincidence probabilities, which
is equivalent to maximising:

\[ l(C^*, K) = \sum_{(i, j) \in M} \mathbb{I}[C^*_i = C^*_j] (\rho_{ij} - K) \]  

(5.21)

where \( M = \{(i, j) : i < j; i, j \in \{1, \ldots, N\}\} \), \( K = \frac{b}{a+b} \in [0, 1] \) where \( b \) is the penalty of misclassifying two points into different clusters (when they should be) and \( a \) the penalty of misclassifying two points being the same cluster (when they shouldn’t be), \( C \) is a given clustering of the observations (up to permutation). With such a partition \( C^* \), we are able to define cluster assignments \( C^*_i \) for each observation \( x_i \). Represented as an integer programming optimisation, solving (5.21) exactly is an NP hard problem, which makes it challenging to solve directly.

A variety of methodologies have been devised to deal with this computational intensiveness, and during this work, we focus on using two approaches, namely, the ?’s integer programming approach and ?’s agglomerative hierarchical clustering with average linkage approach, using \((1 - \rho_{ij})\) as the metric representing the distance between observations \(i, j\). We denote these respective methodologies as \( C^*_{LG} \) and \( C^*_{HC} \). The ? can be considered a more principled estimate of the optimal partitioning solution to (5.21) [?], as it devises a novel heuristic item-swapping algorithm guaranteed to approximate the true optimal partitioning, whereas the ?’s approach only considers the subset of partitions induced from the hierarchical clustering cutting procedure, which are not guaranteed to be optimal. It should be noted however, that this more principled solution of \( C^*_{LG} \) comes with the computational caveat of being more challenging to compute compared to the \( C^*_{HC} \) partition. Where feasible, we opt for computing \( C^*_{LG} \) partition.

Implementations calculating the partitions of \( C^*_{LG} \) and \( C^*_{HC} \) is performed using the \texttt{minbinder()} function from the \texttt{mcclust} R library [?].

Figures 5.3 and 5.4 present fitted MCMC output which includes the histogram of the number of occupied clusters \( N^* \), heatmap of the posterior marginal coincidence probabilities (cluster co-membership) and density estimates of the order statistics sequences \( x_i \) for the each of the simulated studies (5.4.1) and (5.4.2) respectively. We observe that in both of simulation studies (5.4.1) and (5.4.2) that the marginal density estimates closely match the corresponding histograms of the data. This importantly demonstrates that even in the case when mixtures of variable order statistic sequences are generated from mixtures other than \( EW(\alpha, \beta, \lambda) \), our model specified in (5.10) can successfully produce sound density estimates and correctly allocate the relative data-point partitioning induced by the mixture models in these examples.

During the simulated studies outlined in sections 5.4.1 and 5.4.2, we use the partition induced by \( C^*_{HC} \) to compute cluster-wise point estimates of various quantities of interest, and table
5.2 summarises our MCMC output for the simulated study (5.4.1) using the clusters defined by $C^*_{HC}$. The motivation for using $C^*_{HC}$ over $C^*_{LG}$ on the simulated datasets is based on the scaling issues related to computing $C^*_{LG}$ for increasing $N$. The runtime of calculating $C^*_{HC}$ over the simulated datasets of 5.4.1 and 5.4.2 was satisfactorily fast, taking less than 1 minute to execute. We opted for $C^*_{HC}$ over $C^*_{LG}$ in the simulation studies due to the runtime of calculating $C^*_{LG}$ being computationally infeasible for $N = 1500$ (taking at least in the order of days rather than minutes). Table 5.2 provides the estimates yielded from the inferential procedure outlined in Section 5.3 based on the simulated data of (5.19), grouped according to the partitions induced by $C^*_{HC}$. This simulation study was the case where the data was generated from a mixture of EW($\alpha, \beta, \lambda$) distributions. We observe the estimates from table 5.2 are very close to the true parameter values, which are also contained within the 95% credibility intervals indicating the inference is working effectively.
5.4. Results

Table 5.2: Posterior means and (2.5%, 97.5%) credible intervals for the parameters \((\alpha, \beta, \lambda, w)\) of each cluster partitions induced by \(C^*_{HC}\) from simulated data (5.19). The top and bottom rows show the true parameters and number of observations assigned to each cluster respectively.

<table>
<thead>
<tr>
<th>((\alpha, \beta, \lambda, w))</th>
<th>Cluster 1</th>
<th>Cluster 2</th>
<th>Cluster 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\alpha)</td>
<td>0.15 (0.14, 0.17)</td>
<td>2.5 (2.2, 2.9)</td>
<td>0.67 (0.59, 0.75)</td>
</tr>
<tr>
<td>(\beta)</td>
<td>0.78 (0.72, 0.84)</td>
<td>3.3 (3.1, 3.5)</td>
<td>1.7 (1.5, 1.8)</td>
</tr>
<tr>
<td>(\lambda)</td>
<td>0.90 (0.79, 1.0)</td>
<td>0.35 (0.34, 0.36)</td>
<td>0.41 (0.38, 0.44)</td>
</tr>
<tr>
<td>(w)</td>
<td>0.64 (0.64, 0.65)</td>
<td>0.75 (0.75, 0.76)</td>
<td>0.90 (0.89, 0.91)</td>
</tr>
<tr>
<td>(N)</td>
<td>609</td>
<td>546</td>
<td>345</td>
</tr>
</tbody>
</table>

Table 5.3 provides the estimates yielded from the inferential procedure outlined in Section 5.3 based on the simulated data of (5.20), grouped according to the partitions induced by \(C^*_{HC}\). This simulation study was the study where the data was generated from a mixture of \(\text{Gamma}(\alpha, \beta)\) distributions. The motivation of this simulation study was to challenge our inferential procedure with mixture distributions not generated under our initial \(\text{EW}(\alpha, \beta, \lambda)\) assumptions. We observe the estimates from table 5.3 are not close to the true \(\text{Gamma}(\alpha, \beta)\) parameter values, which is expected, as the parameters are unlikely to agree with the inferred \(\text{EW}\) parameters as assumed by our inferential procedure. Though our inferential procedure does, as indicated by the partitions \(C^*_{HC}\), establish that the data was produced from a three mixture distributions and the density estimates indicated by figure 5.4 are effective at describing the decay of the order statistic sequences despite the data being generated from a mixture of \(\text{Gamma}\) distributions.

Table 5.3: Posterior means and (2.5%, 97.5%) credible intervals for the parameters \((\alpha, \beta, w)\) of each cluster partitions induced by \(C^*_{HC}\) from simulated data (5.20). The top and bottom rows show the true parameters and the number of observations assigned to each cluster respectively.

<table>
<thead>
<tr>
<th>((\alpha, \beta, w)) =</th>
<th>Cluster 1</th>
<th>Cluster 2</th>
<th>Cluster 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\alpha)</td>
<td>0.18 (0.16, 0.20)</td>
<td>1.4 (1.2, 1.7)</td>
<td>6.1 (5.1, 7.4)</td>
</tr>
<tr>
<td>(\beta)</td>
<td>0.87 (0.81, 0.94)</td>
<td>1.1 (1.07, 1.2)</td>
<td>2.7 (2.5, 2.9)</td>
</tr>
<tr>
<td>(\lambda)</td>
<td>0.89 (0.79, 1.0)</td>
<td>0.69 (0.62, 0.77)</td>
<td>0.42 (0.40, 0.45)</td>
</tr>
<tr>
<td>(w)</td>
<td>0.66 (0.65, 0.67)</td>
<td>0.75 (0.74, 0.76)</td>
<td>0.90 (0.89, 0.91)</td>
</tr>
<tr>
<td>(N)</td>
<td>535</td>
<td>591</td>
<td>374</td>
</tr>
</tbody>
</table>

5.4.4 Retail analytics dataset

We apply our method of order statistics clustering to a retail analytics dataset based on the aggregated demand from a large UK supermarket retailer. The dataset consists of the cross-elasticiities for a category of supermarket products of the format described in Section 4.1.1.

\[ X = \{ \eta_i, 1:n : \eta_i, (n - l_i + 1) \leq \ldots \leq \eta_i, (n) \}, \text{with } \eta_i, (j) = \text{censored to 0 for } 1 \leq j \leq n - l_i \}, \]
Figure 5.4: Posterior probability cluster co-membership probability heatmap, histograms and density estimates for $N^*$, $l$ and a few order statisticsof simulated data (5.20). These density plots demonstrate the EW kernel successfully describing the mixture of decay sequences. Although subfigure 5.4(b) indicates that the sampling procedure is often producing four occupied clusters, seemingly counter to subfigure 5.4(a)'s three clusters, we note that DP inference can produce partitions consisting of a few singleton clusters consisting of one datapoint. This is the case here, on the simulated mixture of Gamma distributions (the challenging study), where the inferential procedure largely achieves its task of good density estimates and overall partitions, but spends some time producing singleton clusters.

where we have observed only the top $l_i$ order statistics of each cross-elasticity vector $\eta_i$. To allow for straightforward interpretation we focus on the snacks category which consists of $N = 275$ products, consequently our data consists of $N = 275$ vectors of cross-elasticity coefficients. For this study, a maximum of $n = 10$ competitors is considered a priori to reflect a product’s most significant competitors. The snack category consists of the following product line breakdown: 22.5% traditional flavoured crisps (salted, cheese and salt and vinegar), 33.1% exotic flavoured crisps (crisps excluding traditional flavours), 8.73% tortillas, 8.00% popcorn, 7.64% nuts, 4.73% dips, 2.18% pretzels and 13.1% other peripheral quick snack products. Figure 5.5 shows summary plots for the snacks category in this study. The plots provide histograms of the lengths $l_i$ of $\eta_i$ as well as the top two terms of the sequences and along with smooth density estimates produced from the DP model fit. The histogram of the top order statistics demonstrates spikes centred around 0.0 and 1.0, suggesting possible multi-modality.
Figure 5.5: Histograms of the number of observed entries in each cross-elasticity vector, as well as the top two entries of the cross-elasticity vectors, with corresponding density estimates from our model. The censored entries (corresponding to 0 elasticities) have been omitted from the histograms.

5.4.4.1 Omitted competitors & aggregate competition

We introduce two statistics relevant to the retail analytics setting; omitted competitors and mean aggregate competition. These notions have key interpretations in the retail analytics context and will allow us to assess model fit.

**Definition 1: Omitted competitors**

Since cross-elasticity vectors arise as the outcome of penalised regression, it is natural to assume that coefficients are shrunk to zero as the result of a penalisation threshold. However under this regime, it is possible for potentially important competitor products to have been inadvertently omitted from the regression equation, meaning that the cross-elasticity vector should have included additional uncensored entries. The objective of the omitted competitors (OC) statistic is to assess whether the truncation has occurred prematurely by predicting the subsequent term of the observed order statistics sequence (i.e. \( \eta_{i,(n-l)} \) of \( \eta_i \)), and assessing whether this predicted value is sufficiently large. Concretely, we say an elasticity vector contains omitted competitors if its variable length order statistics sequence satisfies:

\[
OC = \mathbb{E}_{\tilde{l}}(\tilde{\eta}_{(n-l)}) \left[ \tilde{\eta}_{(n-l)} \mid \alpha, \beta, \lambda, w \right] \geq \epsilon,
\]

for some truncation constant \( \epsilon > 0 \) and where \( \tilde{\eta}_{(n-l)} \) represents the random quantity of the \( (n-l) \)th order statistic of \( n \) i.i.d. \( \text{EW}(\alpha, \beta, \lambda) \) samples with \( \tilde{l} \sim 1 + \text{Binomial}(n - 1, w) \). In other words, \( \tilde{\eta} \) has the same distribution as \( \eta_i \), but without any censoring. Thus the OC statistic represents the expected value of the 1st censored term of a cross-elasticity vector \( \tilde{\eta}_i \) were we to have observed it. The value of \( \epsilon \) should be chosen to represent a ‘small value’ within the modelling context. We set \( \epsilon = 0.05 \) as a sensible value to deem truncation (and will be fixed for our subsequent analysis) as it implies that if log price deviations of the next competitor is
expected to account for more than 5% of equivalent log prices changes of the product’s own cross-elasticity coefficient $\varphi_1$, we conclude this as a significant omission in the sales model. One of the benefits of interpreting the cross-elasticities as *variable length order statistic sequences* is the utility it provides with respect to defining OC statistic by casting censored observations into a missing data framework. The OC statistic crucially relies on being able to make a prediction of the subsequent value of a cross-elasticity vector were it to be observed. The *variable length order statistic sequence* model, by capturing the sequential decay of these decreasing sequences, allows inferences on subsequent entries of these cross elasticity vectors that flexibly incorporates the rates of decay across the previous entries.

**Definition 2: Aggregate Competition**

One of the primary interests of the analysis is characterising products in terms of their sales sensitivities with respect to their competitors’ prices. We introduce the notion of aggregate competition (AC) to summarise the total effect of competition on a product’s sales through its competitors’ prices changes. We achieve this by defining the aggregate competition of product $i$ as the sum of the top $l$ cross-elasticity coefficients. Concretely, the AC of a cross elasticity vector distribution is given by:

$$AC = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=n-l+1}^{n} \eta_i(j).$$

The AC can be thought of as the total percentage effect that log price deviations of the top $l$ elasticity terms (where $l$ is the expected number of competitors terms) has with respect to the equivalent prices changes of the product’s own log price. For example, if a product’s AC is 0.25, it means that if the log price decrease across each of its competitors was 1 unit, then the product’s log price would need to decrease by 0.25 to offset the loss of sales its competitors’ prices changes would have had on the product’s sales. Thus a large AC indicates a product’s sales are significantly impacted by its competitors’ prices.

### 5.4.4.2 MCMC output

We present the MCMC output of the real retail cross-elasticity dataset and using the following priors for $(\alpha, \beta, \lambda, w)$ and $\nu$ (DP scale parameter):

$$(\alpha, \beta, \lambda, w) \sim Gamma(7, 7/10) \times Gamma(0.5, 1) \times Gamma(1, 1) \times Beta(2, 3)$$

$$\nu \sim Gamma(5, 1)$$
respectively. These priors are selected to reflect a prior expectation of the decay and typical length of the cross-elasticity vectors in the retail analytics context. The prior $w \sim \text{Beta}(2, 3)$ is selected to prefer cross elasticities of length 5, and $\lambda, \nu$ are uninformatively chosen. The priors over $(\alpha, \beta)$ are selected to reflect prior knowledge of the modal nature of the coefficients and the expected heavy tail nature of the cross-elasticity coefficients. Figure 5.5 shows density estimates of the number of observed entries of the cross-elasticity vector, as well as the top two observed values in each vector, showing that our model is capturing these observed quantities very well. In particular, in Figure 5.5 we observe a spike of very small values for the top order statistics $\eta_{(10)}$ and $\eta_{(9)}$, which the model is able to accommodate through a small value of $\alpha\beta$. Figure 5.6 provides heatmaps of pairwise posterior distributions of the parameters which demonstrate a neat separation between pairwise atoms. Interestingly, we observe that larger values of $\lambda$ (corresponding to a smaller mean) are associated with larger values of $w$; instead, $\beta$ values are inversely associated to values of $w$, suggesting that the censoring in this case is largely driven by $\beta$.

Figure 5.7 presents a histogram of the number of unique clusters $N^*$ and a heatmap of the posterior marginal coincidence probabilities (cluster co-membership). We see that the Maximum A Posteriori number of clusters is 3, with two large and one small cluster. Table 5.4 provides the category breakdown of each cluster, together with the number of observations in each as well as OC and AC values. It also includes the posterior mean and 2.5% and 97.5% posterior credible intervals of $(\alpha, \beta, \lambda, w)$ for each of optimal clusters.

For retail analytics dataset outlined in 5.4.4, we use the partition induced by $C^*_{LG}$ to compute cluster-wise point estimates of various quantities of interest. The runtime of calculating the $C^*_{LG}$ partitions over the retail analytics dataset was computationally feasible for $N = 275$ (executing in within one to two hours). To assess model fit, we calculate the posterior predictive p-values [?] of AC for each of the clusters defined by $C^*_{LG}$. Posterior predictive p-values involves generating repetitions $X^{rep}$ from the predictive distribution $p (X^{rep} | \alpha, \beta, \lambda, w)$ for each MCMC sample and calculating $p\text{-value} = 2(1 - p(T(X^{rep}) > T(X) | X))$ for some test statistic $T(X)$, in this case the aggregate competition. Figure 5.8 provides predictive posterior p-values plots on the observed aggregate competition AC over each cluster, compared against histograms of generated AC statistics over predictive replicates of $X$. These all comfortably fall within the 95% prediction intervals.
Figure 5.6: Heatmaps of pairwise posterior distributions of the parameters. The scales have been specifically selected here to produce interpretable heatmaps. It should be noted, that although three distinct clusters have been produced as a result of our inferential procedure, the smallest cluster (≈ 6% of snacks category) is only vaguely noticeable at this current scale. Even at scales where the smallest cluster could in theory be visible, it is difficult by-eye to pickup on a ≈ 6% cluster through a heatmap due to its low density nature.

Figure 5.7: Left panel: histogram of $N^\ast$. Right panel: heatmap of cluster co-membership probabilities (re-grouped with respect to $C_{LG}^\ast$).
5.4. Results

Predictive AC samples

Density

(a) Cluster 1

(b) Cluster 2

(c) Cluster 3

Figure 5.8: Histograms of AC samples with the 2.5%, 97.5% quantiles (red-dashed lines) and AC (solid black line) for each cluster induced by $C^*_{LG}$.

5.4.4.3 MCMC trace plots

Here we discuss convergence of our inferential procedure on the retail analytics dataset. One of issue in diagnosing MCMC convergence of a DPMM is the ‘label switching’ problem. The label switching problem relates to the non-identifiability of mixture components under symmetric priors, this makes it challenging to understand MCMC convergence of a DPMM since mixture components can merge, appear or disappear through MCMC sweeps which creates difficulties in diagnosing the convergence of clusterwise locations hard. We deal with this issue by providing trace plots of the atoms across all the unique clusters (demonstrating the convergence of the atom’s locations) and the trace plot of the unique clusters $N^*$ (demonstrating the convergence of $DP(\nu G_0)$ measure). Figure 5.9 provides traces of the atoms across all unique clusters $\left(\left(\alpha^*_k\right)^t,\left(\beta^*_k\right)^t,\left(\lambda^*_k\right)^t,\left(w^*_k\right)^t\right)$ of $DP(\nu G_0)$ samples for the iterations $t = 1, \ldots, T$ across the unique atoms $k = 1, \ldots, N^*_t$, where $N^*_t$ is the number of unique clusters at iteration $t$ and the trace of $N^*_t$. We plot the $\sqrt{\cdot}$ traces of $(\alpha, \beta, \lambda)$ to induce similar scales for graphical convenience. All plots indicate satisfactory convergence.

5.4.4.4 Retail analytics discussion

Considering the clusters given by $C^*_{LG}$ and linking them to the corresponding categories, we see interesting breakdowns. Firstly, the first cluster has a high concentration of traditional flavoured crisps and no nut products, whereas the second cluster has a lower representation of traditional crisps. Finally, the third cluster comprises nuts, pretzels and the other product categories.

The first and second clusters appear not to have competitor products omitted from their regression models since $OC_1 = 0.038, OC_2 = 0.031 < \epsilon$ and thus indicate that we do not expect any of the unobserved cross-elasticities to be of any significance. However, the third cluster
Figure 5.9: Trace plots of MCMC samples for unique atoms of $(\sqrt{\alpha}, \sqrt{\beta}, \sqrt{\lambda}, w)$ parameters and $N^*$ on dunnhumby's cross elasticity data of the snack category. As discussed in section 5.3.2, for good overall mixing across all unique clusters induced by the DP($\nu G_0$), it is important to have good mixing during the Metropolis Hastings phase for each of unique location updates of $(\alpha^*_k, \beta^*_k, \lambda^*_k, w^*_k)$ for $k = 1, \ldots, N^*$ (the second step of the inferential procedure outlined in 5.3). As a consequence, special attention should be paid to the Metropolis Hastings acceptance rate from sampling $p(\alpha^*, \beta^*, \lambda^*, w^* | \nu, x_{i : C_i = k})$ across each of the unique clusters $k = 1, \ldots, N^*$. Depending on the modelling context, more sophisticated MCMC methodologies, such as adaptive Metropolis Hastings may be appropriate to produce optimal mixing rates across each of the unique clusters $k = 1, \ldots, N^*$.

exhibits competitor omission since $OC_3 = 0.96 > \epsilon$. This implies that, according to the model, we expect to find at least one more competitor with a non-negligible cross-elasticity.

The posterior mean values of parameters of the first cluster are $\alpha_1 = 0.16$, $\beta_1 = 1.51$ with $w_1 = 0.34$ and an aggregate competition of $AC_1 = 0.55$, which indicates that the marginal distributions of the cross-elasticities are of a light-tailed nature. This is in line with the fact that this cluster largely consists of traditional crisps, which are a fiercely competitive product line, where products have multiple substitutes and thus a high degree of sales sensitivity is expected. The second cluster exhibits similar behaviour, with posterior mean parameters $\alpha_2 = 0.06$, $\beta_2 = 1.71$, $w_2 = 0.24$ and an aggregate competition of $AC_2 = 0.42$, also implying a light-tailed distribution. The third cluster is rather different; its posterior mean parameters $\alpha_3 = 5.73$, $\beta_3 = 10.88$ suggest a very light-tailed distribution. This cluster largely consists
5.4. Results

of vectors with only a single cross-elasticity entry (through \( w_3 = 0.016 \)), and the model sug-
gests that an additional competitor may have been missed (or does not exist) as indicated by
\( OC_3 = 0.96 > \epsilon \). It is important to note however, that this cluster’s attribute of having only
a single cross-elasticity entry is not its defining trait. This cluster is also characterised by the
parameters of \( \alpha_3 = 5.73, \beta_3 = 10.88 \), since this combination of parameters produces a very
slow decay rate between respective entries of the order statistic sequences (markedly slower
than the decay of the first and second clusters). This slow decay rate leads to the aggregate
competition of the third cluster being \( AC_3 = 1.11 \), i.e. price changes of the leading competitor
products account for 1.11 of equivalent prices changes of the product’s own price changes.
These parameters suggest that these products are pure substitutes, i.e. products only bought
as an alternative due to other equivalent products being unavailable or too expensive. This
third cluster neatly demonstrates the value of clustering on both the parameters of \((\alpha, \beta, \lambda)\) and
\( w \), since the \((\alpha, \beta, \lambda)\) parameters control the decay between successive order statistic sequence
entries, and \( w \) controls the number of relevant competitors a product receives throughout the
market. If \( w \) were a global parameter, it is likely that products comprising of the third cluster
would still need their own unique cluster, separate to that of the second and third clusters,
due to the distinctive decay rate between the entries of the order statistic sequences. This is
supported by figure 5.5, where we see a small spike centred around 1.0 in the histogram of
\( \eta_{i,(10)} \). This spike corresponds to the order statistic sequences of the third cluster, that are
distinctly different of the other entries of \( \eta_{i,(10)} \) that decay markedly quicker.

With respect to the expected values of the order statistic entries themselves, we observe
similar order statistic patterns between the first and second clusters; each of the first order
statistics entries accounts for a roughly similar amount of its leading direct elasticity (28% and
30% respectively), however the decay rate between the subsequent order statistics of the
first cluster is significantly slower than that of the second cluster (roughly 55% − 60% of their
previous value compared with 40% − 45%). This decay rate observation between subsequent
order statistics entries supports the discrepancy between each of the first and second cluster’s
AC statistics as well as the first cluster comprising of food items which traditionally have a
high number of competitors than in the second cluster. Similarly as before, the third cluster
differs significantly from the first and the second. Its first order statistic entry accounts for 98% of
its leading direct elasticity and has a slower decay rate between successive order statistic
sequences, each of these artefacts being significantly different from that of the previous clusters.

Retailers also wish to understand the behaviour of their product range at a less granular
level, e.g. at a category level. Clustering of cross-elasticity profiles provides a means to extract a new summary profile for a subset of products through a principled data-driven approach. Crucially, these can aid store planners and business specialists in the retail analytics domain to better understand the optimal pricing and display combinations. For example, products in the third cluster are highly sensitive to their leading cross-effect products, but otherwise are unaffected by the bulk of products around them. On the other hand, products in the first and second clusters are cannibalized by their competitor products, meaning that increasing the sale of one product decreases the sale of another, but with the second cluster being more robust to these prices changes than the first.

Table 5.4: Retail analytics cluster-wise inference. Posterior means and (2.5%,97.5%) credible intervals for each of the four parameters $(\alpha, \beta, \lambda, w)$ along with other breakdown statistics for each the clusters induced by $C^*_LG$.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Cluster 1</th>
<th>Cluster 2</th>
<th>Cluster 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>0.16 (0.036, 0.28)</td>
<td>0.06 (0.030, 0.21)</td>
<td>5.73 (2.49, 10.19)</td>
</tr>
<tr>
<td>$\beta$</td>
<td>1.51 (1.10, 2.13)</td>
<td>1.71 (1.20, 2.39)</td>
<td>10.88 (5.88, 17.06)</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>3.89 (1.75, 5.81)</td>
<td>2.29 (1.59, 4.47)</td>
<td>1.17 (1.09, 1.29)</td>
</tr>
<tr>
<td>$w$</td>
<td>0.34 (0.21, 0.41)</td>
<td>0.24 (0.20, 0.39)</td>
<td>0.016 (0.0017, 0.042)</td>
</tr>
<tr>
<td>$N$</td>
<td>110</td>
<td>149</td>
<td>16</td>
</tr>
<tr>
<td>OC</td>
<td>0.038</td>
<td>0.031</td>
<td>0.96</td>
</tr>
<tr>
<td>AC</td>
<td>0.55</td>
<td>0.42</td>
<td>1.11</td>
</tr>
<tr>
<td>trad crisps (22.5 %)</td>
<td>30.9 %</td>
<td>18.1 %</td>
<td>6.25 %</td>
</tr>
<tr>
<td>exotic crisps (33.1 %)</td>
<td>33.6 %</td>
<td>35.6 %</td>
<td>6.25 %</td>
</tr>
<tr>
<td>tortillas (8.73 %)</td>
<td>11.81 %</td>
<td>7.38 %</td>
<td>0 %</td>
</tr>
<tr>
<td>popcorn (8.00 %)</td>
<td>8.18 %</td>
<td>8.05%</td>
<td>6.25 %</td>
</tr>
<tr>
<td>nuts (7.64 %)</td>
<td>0%</td>
<td>8.72 %</td>
<td>50.0%</td>
</tr>
<tr>
<td>dip (4.73 %)</td>
<td>4.55%</td>
<td>5.37%</td>
<td>0 %</td>
</tr>
<tr>
<td>pretzels (2.18 %)</td>
<td>0.909 %</td>
<td>2.01 %</td>
<td>12.5%</td>
</tr>
<tr>
<td>other (13.1 %)</td>
<td>10.00%</td>
<td>14.8%</td>
<td>18.8%</td>
</tr>
<tr>
<td>$E(\eta_{10})$</td>
<td>0.287</td>
<td>0.304</td>
<td>0.984</td>
</tr>
<tr>
<td>$E(\eta_{9})$</td>
<td>0.156</td>
<td>0.124</td>
<td>0.963</td>
</tr>
<tr>
<td>$E(\eta_{8})$</td>
<td>0.092</td>
<td>0.053</td>
<td>0.937</td>
</tr>
<tr>
<td>$E(\eta_{7})$</td>
<td>0.055</td>
<td>0.022</td>
<td>0.926</td>
</tr>
</tbody>
</table>

5.5 Summary & future work

We have presented a Bayesian nonparametric mixture model for censored ordered data, using the Exponentiated Weibull distribution as a kernel. Our approach allows for flexible modelling of cross-elasticity coefficients and lends itself to meaningful interpretation. We implemented our methods on a dataset of cross-elasticities, focusing on quantities of interest in the retail analytics context, such as the aggregate competition and potential omitted competitors. Our model was able to capture several interesting features in the data through the corresponding clustering.

These methods can potentially be extended in several directions. Firstly, one could intro-
duce structure between the distribution of the length of the order statistics sequences and the kernel distribution. This may allow borrowing of information between these two sources of information, although it will become more computationally cumbersome. Secondly, one could relax the assumption of ordered observations to account for observations only ordered in expectation. Although in the cross-elasticity context this was not appropriate, in applications such as sports analytics it may be more reflective of the data. For example, the best athlete will not always have the best performance at a competition; instead, the ranking corresponds to average performance. Finally, we would like to explore combinations of different product categories to investigate similarities in market behaviour between otherwise disparate products.
Chapter 6

Slow-moving-inventory & related methodologies

This Chapter introduces and defines the issues of demand forecasting for a class of retail products known as slow-moving inventory (SMI). Demand forecasting of products is of particular interest to retailers as it impacts their businesses on various operational levels; this Chapter introduces SMI forecasting and the issues related to these forecasts.

The subsequent sections are structured as follows; Section 6.1 defines the notion of SMI and describes the intermittent demand of such products along with related difficulties with forecasting SMI. The section continues on to outline the motivation behind forecasting SMI and articulates the opportunities that forecasting for this type of products offers. Section 6.2 outlines the existing work into intermittent demand forecasting of SMI. Section 6.3 describes a class of regression models known as zero-inflated regression models and discusses their relevance to the issues related to forecasting SMI. Section 6.4 describes a class of point process know as Hawkes processes and describes its relevance to the temporal aspects of forecasting SMI.

6.1 Slow-moving inventory background

One of the main objectives of retail analytics is to build predictive models for the demand of products that companies offer to their markets. Generally speaking, demand forecasting for products with high volumes of sales have been extensively studied in the literature [??], and as a consequence, retailers have been successful at developing these forecasting models and have well understood the effect that traditional covariates such as price, cross-prices, seasonal indicators have on the demand of their products. However, there are a class of products known as slow-moving inventory, and for these products the sales volumes are significantly smaller than with most products, i.e. where a sale occurs only 5% of days. Crucially, at this level of sales
6.1. Slow-moving inventory background

volume, the data arising from the sales process of SMI is fundamentally different to products with a high sales volume, and this difference often makes demand forecasting harder. We will expand on why it is harder in subsequent sections. In spite of these difficulties, retailers remain interested in forecasting SMI for the same reasons that they are interested in forecasting demand for products with higher sales volumes, such as:

1. Understanding how variables such as a product’s price, the promotional activity a product has undergone and how the seasonal trends affects a product’s demand, and therefore revenue from the product, is crucially important to retailers as it allows them to understand the factors that drive demand. This allows retailers to make sensible decisions over how to set the variables they have control over. For example, a forecasting model allows experimenters to measure the effects variables have on revenue critically allows retailers to allocate marketing resources, optimise prices and understand the affect of promotional activity. Models capable of accurately assessing the link between demand and these variables create a competitive advantage by improving the retailers’ decision making.

2. Demand forecasting for products allows retailers to manage their inventory. Thus, being able to forecast demand can allow retailers to manage their supply chain optimally both in terms of distribution between regional stores and to know when to reorder further inventory from suppliers, both of which provide a measurable improvement to their business operations. For example, a demand forecaster at a store level allows retailers to reduce the opportunity costs associated with under/over stocking and avoid the potential loss of stock. showed that many companies have inventories that over stock products due to inaccurate demand forecasts.

Figure 6.1 provide plots for four SMI sales processes along with \( \log(\text{price}) \) in £ over 364 trading days. For each product, the daily count corresponds to the aggregated sales of a touchscreen tablet across five large supermarkets within south London. These plots illustrate that the sales volumes are ‘inflated’ with an excess of zero sales and demonstrates an unclear correlation between demand and changes in prices and seasonal affects. We further observe a clustering effect in the succession of sales in a product’s own demand series.
6.1.1 Challenges of slow-moving inventory forecasting

There are various aspects that make forecasting SMI difficult, but for the purposes of this work, we focus on the following three issues:

1. **Zero-inflation:** The sparsity of the sales signal that occurs for SMI products leads to an inflation of zeros (days with no sales), and these excess zero sales limit the degree to which forecasting methodologies can be deployed. In particular, this inflation of zeros has two impacts, firstly it means the implementation of traditional demand forecasting models such as (4.3) is untenable, due to the errors having non-standard distributions. Secondly, this zero-inflation often induces a low correlation between the covariates that retailers’ traditionally utilise in forecasting and the sale response. This makes establishing a compelling explanatory narrative for what drives demand difficult due to the high level of uncertainty and low correlation between covariates and response.

2. **Temporal dynamics:** Another not fully understood aspect of SMI intermittent demand data is the dependency between future demand and historical demand. This temporal component to SMI demand takes the form of ‘bursty’ sales across different products, i.e. that
sales of product A cause further sales of product A in the future, and contemporaneous structure, i.e. that sales of product B cause sales of product C in the future. Such features have been shown to be prevalent in previous forecasting work [??]. During this work, the ‘bursty’ SMI demand is often referred to as self- and cross-excitation respectively. These bursts could be the result from some common external factor that cannot be accounted for by available covariates. An example of a common external factor could be an unexpected twitter campaign promoting a product whereas a strict dependency on the previous sales history could be the positive word-of-mouth between consumers having brought the product within a social network. This dependency of the sales processes on its recent history as well as on the sales process history of other products possibly offers a route to improve the performance of predictive models for the SMI demand process by extending the models to consider and capture autocorrelation/contemporaneous nature of sales.

3. Short Sale cycles: One of the practical concerns related to forecasting SMI is that SMI products are frequently stocked and sold for a relatively limited amount of time (short sale cycles). This has implications on training predictive SMI demand models, as over fitting issues can arise by little covariate and demand history and the added issue that the collective time series may not exist over the same entire time period. This lack of sales signal obfuscates how the traditional variables used in forecasting models (prices, promotions, seasonality) are linked with the volume of demand. This is particular important to retailers, as they want to understand the effect that controls have on the underlying sales process.

6.2 Analysis of slowing-moving-inventory forecasting

Much work has been done in the field of SMI forecasting, with a wide range of different methodological approaches having been used to address the challenges that forecasting SMI demand presents. Broadly speaking however, the bulk of the methodological contributions have been from the fields of machine learning, exponential smoothing (and related methods) to more traditional statistical approaches.

Exponential smoothing and related methods have been a popular class of methodology for intermittent demand forecasting of SMI products. Exponential smoothing is a sequential forecasting methodology with attempts to forecast future observations as a weighted moving average
of past observations over time. More concretely, [?] expressed exponential smoothing as:

\[
\begin{align*}
l_0 &= y_0 \\
l_t &= \alpha y_{t-1} + (1 - \alpha)l_{t-1}
\end{align*}
\]  

(6.1)

where \(y_t\) is sales observation at time \(t\), \(l_t\) is the latent ‘smoothed forecast’ used to predict \(y_t\) and \(\alpha \in (0, 1)\) is the smoothing factor. Exponential smoothing and related methods have been extensively applied to a wide range of signal processing applications [??], but a relevant variation that has been heavily applied to forecast intermittent demand of SMI is known as Croston’s method [?]. Croston’s method decomposes the demand data into a count process \(y_t\) of instances of non-zero demand and \(l_t\) of inter demand intervals. Croston’s approach then makes forecasts of future observations as the ratio of these two non-zero demand and inter demand intervals, i.e. \(\frac{y_t}{l_t}\), assuming independence between the demand size and the inter-demand intervals. Extensions to Croston’s method [??] have included accommodating the possibility when there is no longer a demand for the marketed product and ? developed an unbiased estimator of Croston’s method, which was shown to outperform Croston’s original estimator on theoretically generated data. For a detailed review of Croston’s method, its extensions and related exponential smoothing methods refer to [??].

In spite of exponential smoothing approaches being widely applied to forecasting intermittent demand of SMI, there have been significant methodological innovations in the literature that offer improved forecasts accuracy and explanatory power, which make these original exponential smoothing approaches less compelling [??]. This relative underperformance in terms of forecast accuracy and explanatory power is arguably for a range of reasons, but it may be a result of the lack of statistical underpinning many of these methods have. ? showed the stochastic process of Croston’s method to be inconsistent with intermittent demand in that Croston’s method is non-stationary and defined on a continuous space. This lack of statistical underpinning is a significant drawback to these methods as it means hypothesis testing, forecasting distributions and a framework for regression analysis are not readily available. Consequently, these approaches are often thought of as ‘ad hoc’ testing methods, and ones where it is often not straight-forward to optimise or select parameters [??].

Machine learning approaches to intermittent demand forecasting of SMI have largely used neural networks and perceptrons methodologies [??], and less extensively support-vector-machines [??]. Machine learning algorithms have a tradition as being methodologies capable of
finding the complex non-linearities that underlie data generating processes with minimal model specification [??], and are appealing approaches to apply to intermittent demand forecasting. Consequently, neural networks and related methodologies have been widely applied to intermittent demand forecasting [??], and have demonstrated the flexibility to accommodate for dependencies between non-zero demand and the inter-demand intervals, temporal phenomena of bursty or lumpy sales patterns as well as for the nuanced interactions between various intermittent demand between different time series.

One of the appealing aspects of neural networks and other related methodologies is the ability to fit complex and non-linear datasets. However, neural networks and alike often require a significant amount of data to train on [??]. Further demonstrated neural nets are outperformed by conventional statistical methods on smaller samples. This can inhibit machine learning algorithms’ application to intermittent demand forecasting as the training signal is typically sparse and thus there can be a risk of overfitting. Regularised versions of machine learning methodologies have been successfully applied in the context on intermittent demand [??], but a significant amount of data is necessary for these approaches to be applicable. Finally, machine learning methods’ often do not have an interpretable stochastic process underpinning them. Consequently, answering hypothesis like questions such as the benefits of information sharing, measuring the effect variables have and quantifying certainty are not easily assessable. Machine learning approaches struggle with this interpretability, which is of particular interest in our SMI setting.

Finally, many statistical methodologies have been devised to handle intermittent demand forecasting of SMI. These approaches predominately make use of count models [??], but also include state-space models [??], modified Markov models [??], or more traditional time series models [??]. Count models are often employed as a route to handling the zero-inflation observed in intermittent demand data, and consequently models such as generalised hurdle negative binomial model, beta-binomial model and hurdle shifted Poisson models, including others, have been successfully applied to forecasting intermittent demand [??]. In addition to handling the excess of zeros in intermittent demand, extensions to static count models have included incorporating temporal dynamics to capture the lumpy and bursty nature that is often observed in such demand processes. Implemented various count models with damped and undamped recurrence relations on the mean of count distributions where they demonstrate a marked improvement over static traditional models that exclude temporal dynamics. Their approach however, does not make use of any explanatory variables or information borrowing between
the intermittent demand series. Further extensions have included hierarchical expositions of count models that aim to pool information across related demand series. ? brought together a hierarchical Bayesian approach to information pool across the intermittent demand of different products with an $AR(1)$ process on the mean of the count process and further incorporated explanatory variables within the regression framework, but do not accommodate any regression or temporal framework on the zero-process.

State-space, Markov chain and time series approaches for intermittent demand forecasting of SMI have generally focused on capturing the temporal and bursty aspect of intermittent demand. ? use an approximate Bayesian method with a latent state process to describe the burstiness of demand and ? similarly used a mixture of zero and Poisson distributions to demonstrate a significant improvement to Croston and related methods. ? used a modified Markov chain model to estimate intermittent demand and show that Markov chain based methods can capture the irregular nature of intermittent demand. Finally, with respect to more traditional times series modelling, ? and ? used conventional Bayesian times series and integer autoregressive moving average models for predicting intermittent demand.

However, there are open questions that the current statistical approaches do not sufficiently answer. Firstly, many existing approaches do not fully explore the effect covariates have on intermittent demand forecasting of SMI. Some have included a regression framework that could accommodate covariates, but the majority of applications have not. This leads us to ask whether the uncertainty exhibited in intermittent demand processes could be explained away if conditioned on the appropriate variables such as a prices or seasonality. Secondly, few approaches attempt to address the issues of hierarchical borrowing, temporal dynamics and zero-inflation in a unified way. For the approaches that do, they have not clearly separated out the benefits that covariates, hierarchical borrowing and temporal dynamics have in the context of intermittent demand forecasting. One of our key objectives is to understand the benefits that each of these modelling contributions bring to SMI forecasting. Thirdly, in the approaches incorporating temporal dynamics, though on the whole demonstrate significant benefits in each of the contexts they are applied to, are typically of linear forms that could arguably not be flexible enough in other modelling scenarios. These linear forms are typically moving averages or $AR(1)$ processes that only account for the most recent history. Temporal processes taking into account more of the history and allow for more complex linearities may be beneficial. Finally, none of approaches allow for any contemporaneous correlation across the intermittent demand series, i.e. the lumpiness in intermittent demand series occurs independently across products.
It would be interesting to investigate whether there is a contemporaneous dependency across the bursty demand across products and whether such a structure could provide a utility to forecasting intermittent demand.

For the purposes of this work, we favour statistical modelling approaches. This is because on balance, statistical approaches’ capability of neatly quantifying uncertainty, the ability to assess the effect of covariates and information borrowing in a hierarchical fashion are directly relevant to our objectives as outlined earlier in this Chapter.

6.3 Zero-modified distributions

The over dispersion of zero sales exhibited in intermittent demand is one of key difficulties when attempting to develop accurate demand forecasts. Much work into over dispersed count data has been done, and there are many contexts where common count distributions such as the Poisson and negative binomial distributions are not sufficiently flexible to capture the over dispersion of count data often exhibited in real-life settings [??].

Many statistical approaches have been developed to model zero-inflated data count data, such as hurdle models, zero-inflated models, Neyman type A distribution, threshold models and Birth process models [?]. For the purposes of this work, we focus on hurdle models as a route to modelling the inflation of zero sales, the reasons for which will be made clear in the subsequent section.

6.3.1 Hurdle models

introduced the hurdle regression model to handle the inflation of zeros in count data that traditional count models could not adequately account for. The hurdle model defines a distribution over \( \{0, 1, \ldots\} \), and assumes these counts can be split into two separate processes; a process accounting exclusively for the 0’s (the hurdle), and a process accounting for the non-zero counts.

More concretely, given an observation \( y \) assumed to be distributed according to a hurdle model, the probability mass function is given by:

\[
p(y) = \begin{cases} 
\pi, & \text{for } y = 0 \\
(1 - \pi) g(y), & \text{for } y \geq 1 
\end{cases}
\]

(6.2)

where \( \pi \in [0, 1] \) is the probability of zero count and \( g(\cdot) \) is some probability mass function over the positive integers. The mean and variance of the hurdle distribution is given by \( \mathbb{E}[Y] = (1 - \pi) \mathbb{E}_g[Y] \) and \( \mathbb{V}[Y] = (1 - \pi) \mathbb{V}_g[Y] + \pi \mathbb{E}_g[Y] \) respectively, where \( \mathbb{E}_g[Y] \) and \( \mathbb{V}_g[Y] \) are the expectation and variance of the positive count distribution induced from \( g(\cdot) \).
originally assumed a truncated Poisson parametrised by \( \lambda \) over the positive integers, and included explanatory variables \( w_i, x_i \) with respective regression coefficients \( \theta, \beta \) to allow \( \pi, \lambda \) to vary. Then, given a collection of \( y_i \)'s with associated explanatory variables \( w_i, x_i \), the hurdle model was given by:

\[
p(y_i) = \begin{cases} 
\pi_i, & \text{for } y_i = 0 \\
(1 - \pi_i) \exp(-\lambda_i)^{\pi_i} (1 - \exp(-\lambda_i))^{y_i} & \text{if } y_i \geq 0 
\end{cases}
\]  

(6.3)

with link functions:

\[
\logit(\pi_i) = w_i \theta \text{ and } \log(\lambda_i) = x_i \beta.
\]

Here \( x_i \) and \( w_i \) are \( p \times 1 \) and \( q \times 1 \) vectors of covariate data and \( \theta \) and \( \beta \) are \( p \times 1 \) and \( q \times 1 \) vectors of regression coefficients.

The hurdle model has been applied to a range of contexts which include ecology modelling \([?]\), resource planning within medicine \([?]\) to modelling insurance claims \([?]\). The hurdle model is closely related to the zero-inflated Poisson model \([?]\), which is essentially a two state mixture model that either mixes to a degenerate zero-distribution with probability \( \pi \in [0, 1] \), or alternatively with probability \( (1 - \pi) \) takes a sample from a untruncated Poisson distribution.

Though similar, these models differ in the following ways: firstly, the hurdle model assume the zero and non-zero processes are separable, as 0 observations arise exclusively from the degenerate 0 distribution and not from the count distribution over \( \{1, \ldots\} \). Consequently, the likelihood of hurdle can be separable. Hurdle models further differ from their zero-inflated counterparts in their ability at accommodating deflated models that zero-inflated models cannot.

Extensions over the original hurdle model have been vast, but generally have included alternative specifications of link functions in both the count and zero-inflation components \([?]\), as well as to allow for mixed effects \([?]\) to mention a few. One of the relevant developments has been the application of hierarchical Bayesian equivalents of hurdle and zero-inflated models. As discussed in Chapter 2, a Bayesian hierarchical model where samples are i.i.d can be expressed as in (2.3), and has appealing benefits in terms of information borrowing. Many of these hierarchical extensions have been implemented in the context of multivariate longitudinal data analysis. Multivariate longitudinal data comprise multiple time series data whose observations are recorded at the same time across each of the multiple time series. Within the context of the hurdle model of (6.3), by denoting \( y_{it} \) as the observation at time \( t \) for \( i^{th} \) time series, \( i = 1, \ldots, d \), and Hurdle \((\pi_i, \lambda_i)\) as the hurdle distribution with parameters \( \pi_i \) and \( \lambda_i \), we can
then express a $d$-dimensional longitudinal hurdle model across multiple longitudinal series as:

$$
y_{it} \sim \text{Hurdle}(\pi_i, \lambda_i), \text{ for } t = 1, \ldots, n$$

$$\pi_i, \lambda_i \sim \Upsilon(\omega)$$

$$\omega \sim \Pi$$

(6.4)

for $i = 1, \ldots, d$ where $\Pi$ is the prior of the hyper-parameters $\omega$ that parametrise $\Upsilon$. Such models are effective at information pooling across different time series whilst still capturing between-subject heterogeneity as well as demonstrating optimal small-sample properties [?????]. This is particularly relevant to our intermittent demand setup, as the ability to pool parameter information between the intermittent demand series could allow a route to handling the issues related to the sparsity of the demand signal. Though it should be noted, to model the busyness of demand, we disregard the i.i.d assumption of (6.4) to accommodate the more interest temporal dynamics typically exhibited in intermittent demand.

During this work, we opt for hurdle models over zero-inflated equivalents for the following reasons. Firstly, one of the appealing traits of hurdle models is the scope for the likelihood being separable. This decoupling of the count and zero processes can allow inference to be simplified. Furthermore, there are known issues associated with zero-inflated models in terms of estimation and identifiability as well as having poorer fit when compared to their hurdle model equivalents [??].
6.4 Temporal point processes

One of key aspects of intermittent demand is the bursty and lumpy nature of sales across time. As mentioned in section 6.2, the inclusion of temporal dynamics in SMI forecasting indicates significant benefits when compared to static models without such dynamics. There have been a range of approaches that incorporate temporal dynamics into demand forecasting. In the intermittent demand context, these have included state-space modelling, ARIMA process, modified Markov chain models as well as other approaches. For a detailed review of various forecasting approaches, refer to [?].

For the purposes of this work however, we consider a relevant class of stochastic processes known as temporal point processes, that could have utility at describing the bursty and intermittent nature of sales of SMI products. Informally speaking, a temporal point process consists of an ordered sequence of arrival times of some particular events. Temporal point processes have been widely applied to a range of contexts from forecasting earthquake activity [?], modelling market events data [?] to predicting Twitter tweet popularity [?]. Temporal point processes have proved to be a good class of model for predicting the arrival times of future events.

6.4.1 Point processes background

We now introduce some prerequisite terminology and mathematical definitions before covering the relevant temporal point processes that are relevant to this work. Namely, we define the concepts of a counting process, a point process and a conditional intensity function.

**Definition 10. Counting process**

A counting process is a stochastic process \{N(t), for \( t \geq 0 \)\} that satisfies the following:

1. \( N(t) \) is defined over the positive integers \( \mathbb{N}^+ \).
2. \( N(j) \leq N(t) \) for \( j \leq t \).
3. \( N(0) = 0 \).

**Definition 11. Point process**

A point process is a collection of random variables \( \{t_1, t_2, \ldots \} \) that satisfies the following:

1. \( t_1 \in [0, \infty] \) for \( t \geq 0 \).
2. \( P(t_1 \leq t_2 \leq \ldots) = 1 \) almost surely.
3. The number of points in some bounded region of \( \mathbb{R} \) is almost surely finite.
We introduce $H(t)$ to denote the history of a point process instances prior to time $t$, i.e. $H(t) = (t_j : \text{for all } t_j < t)$. Having now defined a point process, it is now necessary to specify a distributional form over a finite collection of $\{t_1, t_2, \ldots, t_J\}$, i.e.:

$$f(t_1, t_2, \ldots, t_J) = \prod_{j=1}^{J} f(t_j \mid H(t_j)).$$

Instead of specifying the conditional arrival distribution $f(t_j \mid H(t_j))$ directly, we can instead characterise a temporal process by a conditional intensity function.

**Definition 12. Conditional intensity function**

A conditional intensity function $\lambda(t)$ of an associated counting process $N(t)$ (or equivalently point process $\{t_1, t_2, \ldots, t_J\}$) is given as:

$$\lambda(t) = \lim_{h \to 0} \frac{\mathbb{E}(N(t + h) - N(t) \mid H(t))}{h},$$

given the righthand limit exists. Crucially, the conditional intensity function, if it exists, uniquely defines distributions over a point process given in (6.5) (in the finite dimensional case).

The conditional intensity function $\lambda(t)$ is a useful route to intuiting a temporal point process. More concretely, given a conditional intensity function $\lambda(t)$ and a sufficiently small interval of time $[t, t + \delta t)$, the probability of a new occurrence happening within this interval given $H(t)$ is:

$$P(\text{event} \in [t, t + \delta t) \mid H(t)) = \lambda(t)dt$$

This allows an understanding of the count and point processes induced from the functional form of $\lambda(t)$. The greater $\lambda(t)$ is during the time interval $[t, t + \delta t)$, the more probable of observing an event occurring during this interval is. Consequently, $\lambda(t)$ should reflect a point processes dynamics observed in the data [??]. Two popular point processes are the **homogeneous the non-homogenous Poisson process** (HPP) and **non-homogenous Poisson process** (NHPP). A counting process $N(t)$ defines a HPP if its conditional intensity function is given by $\lambda(t) = \lambda_0$. Similarly, a counting process $N(t)$ defines a NHPP if its conditional intensity function $\lambda(t) = \varphi(t)$, where $\varphi(t)$ is a function independent of its past history and depends only variables defined at current time $t$.

Homogeneous and non-homogeneous Poisson processes have been widely applied to a range of temporal settings [??]. However, in spite of these simpler models, in situations where conta-
gion effects are observed, the assumption that the conditional intensity function is independent of its history can be violated. To accommodate this dependency on history, exciting processes have been introduced. Exciting processes are processes that are inclined to ‘cluster’ over the domain they occur over. One such process known to accommodate event arrival clustering is the Hawkes process.

### 6.4.2 Hawkes process

The Hawkes process was introduced as a self-exciting point process to capture the temporal clustering of events within a counting process \([N(t)]\). More concretely, given a count process \(N(t)\) and associated point process \(\{t_1, t_2, \ldots\}\), a Hawkes process can be defined by its conditional intensity function \(\lambda(t)\) given by:

\[
\lambda(t) = \varphi(t) + \int_0^t \kappa g(t - u) \, dN(u)
\]

\[
= \varphi(t) + \sum_{t_j < t} \kappa g(t - t_j)
\]  \hspace{1cm} (6.6)

where \(\varphi(t)\) is the background intensity rate, \(g(\cdot) \geq 0\) some continuous excitation kernel that controls the extent counts/events cluster together and \(\kappa > 0\) is a trigger constant. A Hawkes process induces a clustering among the count process \(N(t)\), where increases in \(\kappa\) in turn increases the probability of an event occurring in the future.

The Hawkes process can be thought of as a generalisation of a non-homogeneous Poisson process. The non-homogeneous Poisson processes assumes \(\lambda(t) = \varphi(t)\), i.e. its conditional intensity is purely a function of its current time and independent of the history of previous events. This corresponds to a Hawkes process with \(\kappa = 0\) and \(\varphi(t) \neq \text{constant}\).

This demonstrates the differences between the Hawkes process and the closely related non-homogeneous Poisson process, in that a Hawkes process accounts for the history of event occurrences that allows for excitation dynamics. Figures 6.2 and 6.3 give an example of a counting process \(N(t)\) induced from a homogeneous Poisson process and a Hawkes process respectively for 200 units of time. Interestingly, both have the same background rate, but the addition of \(\sum_{t_j < t} e^{-(t-t_j)}\) in the conditional intensity function of the Hawkes process induces a strong clustering amongst the \(N(t)\). This is indicated by the quick succession of blue dotted lines compared to the more evenly spaced counts generated under the HPP case.
6.4. Temporal point processes

Figure 6.2: Plots of the count process $N(t)$ (as indicated by the solid blacks dots) and temporal point process $\{t_1, t_2, \ldots\}$ (as indicated by the sequence of vertical blue dotted lines) induced from the homogeneous Poisson process with conditional intensity function $\lambda(t) = \lambda_0 = 0.05$.

Figure 6.3: Plots of the count process $N(t)$ (as indicated by the solid blacks dots) and temporal point process $\{t_1, t_2, \ldots\}$ (as indicated by the sequence of vertical blue dotted lines) induced from the Hawkes process with conditional intensity function $\lambda(t) = 0.05 + 0.3 \sum_{t_j < t} e^{-(t-t_j)}$.

The incorporation of a point process’s history in the conditional intensity function of the Hawkes process allows for excitation and contagion dynamics to be captured that a HPP and NHPP can not. Consequently, in situations where self-excitation phenomena exist, the Hawkes process has demonstrated to be a good temporal point process that accommodates for such dynamics [??].
6.4.3 Multivariate Hawkes Process

A Hawkes process can be generalised to a multivariate setting. The corresponding multivariate Hawkes Process can be thought of as a multivariate point process whose individual point process entries are defined by a conditional intensity function given by a Hawkes process that incorporates excitation from the other point process entries. Before defining the multivariate Hawkes process more formally, we introduce the concept of a *multivariate counting process*.

**Definition 13. Multivariate counting process**

A stochastic process \(\{N(t), \text{for } t \geq 0\}\) is a multivariate counting process of \(d\) dimensions if it satisfies:

1. \(N^i(t)\) defines a counting process for every \(i \in \{1, \ldots, d\}\), where \(N^i(t)\) is the \(i^{th}\) entry of \(N(t)\).

2. The sum of the coordinates of \(N(t)\), \(\sum_{i=1}^{d} N^i(t)\), also defines a counting process.

We then denote \(\{t^i_j\}_{i=1, \ldots, d}\) as the associated \(d\)-dimensional multivariate point process induced from the multivariate count process \(N(t)\) \([?]\). The multivariate Hawkes process is then defined through its conditional intensities \(\lambda^i(t)\), \(i = 1, \ldots, d\), given by:

\[
\lambda^i(t) = \phi^i(t) + \sum_{k=1}^{d} \int_0^t \kappa_{ki} g_{ki}(t-u) dN^k(u)
\]

\[
= \phi^i(t) + \sum_{k=1}^{d} \sum_{i \neq j} \kappa_{ki} g_{ki}(t-t_j^k) \tag{6.7}
\]

where \(\phi^i(t)\) is the background intensity of the counting process \(N^i(t)\), \(g_{ki}()\) and \(\kappa_{ki} > 0\) are the excitation kernel and constant corresponding to the counting process \(N^k(t)\)’s effect on counting process \(N^i(t)\). By the inclusion of \(g_{ki}()\) with \(k \neq i\) in the intensity function of \(\lambda^i(t)\) allows for mutual excitation across the multivariate counting processes. Multivariate Hawkes processes have been successfully applied to a wide range of disciplines. ? proposed a scheme allowing for inter-excitation and inhibition across different social media events and themes and use a triggering kernel exponential in time and Gaussian in space to capture cross excitation and inhibition in tweets in different topics and geographies. ? use a multivariate Hawkes process to model information spread across sparse low-rank social networks. A multivariate Hawkes process is a possible methodology that could capture the suspected dependency between the ‘lumpy’ sales of intermittent demand series.
6.4.4 Discretised Hawkes process

As outlined earlier, the Hawkes process has useful applications in modelling the clustering phenomena often exhibited in point process data. However, it is important to note that the Hawkes process is defined over the continuous space. Consequently, it is necessary to create a discretised equivalent to the Hawkes process that is applicable in the intermittent demand forecasting setting where forecasts are made on a daily level.

One relevant discretisation of excitation processes in modelling zero-inflated count data was that of ?, who interpreted a Hawkes process within the discrete setting. In particular, they let $E_t$ be the indicator for an event day where $E_t = 1$ if on day $t$, there was at least one non-zero count observed, and $E_t = 0$ if on day $t$, there was only a 0 count observed. They then assumed $E_t \sim \text{Bernoulli}(\pi_t)$ where $\pi_t$ is the probability of observing non-zero count on day $t$ with link function:

$$\eta(\pi_t) = -\log(1 - \pi_t) = \varphi(t) + \kappa \sum_{j < t} E_j g(t - j)$$  \hspace{1cm} (6.8)

where $\varphi(t)$ is mean intensity function on day $t$ (which takes no account of the history), $g(\cdot)$ is a discrete excitation function and $\kappa$ is the real valued excitation or inhibition constant. This essentially achieves the self-excitation dynamics that a traditional Hawkes process captures in the temporal point process setting. We observe from (6.8), that $\eta(\pi_t)$ elicits a behaviour such that for $\kappa > 0$, increases in $\varphi(t) + \kappa \sum_{j < t} E_j g(t - j)$ lead to increases the probability ($\pi_t$) of further such events occurring in the near future.

? demonstrated that this adequately captures the self-excitation exhibited in their count dataset when compared to other benchmark models. Such an approach is closely related to intermittent demand forecasting. Figure 6.4 plots two series of samples from a Bernoulli distribution with a Hawkes process term. It illustrates the variation in Bernoulli samples according to the differing parameters of the excitation kernel and trigger constant. We observe from this plot, that the maroon curve, by having a higher excitation constant $\kappa$, experienced much more excitation as exhibited by the densely packed maroon events dots, as opposed to the blue which are mostly isolated events.
Figure 6.4: Simulated example. Two series of samples are generated from $E_t \sim \text{Bernoulli}(p_t)$, with logit$(p_t) = \theta + \kappa \sum_{j<t} E_j g(t - j \mid \mu, \tau)$ for $t = 1, \ldots, 364$ where $g(\cdot \mid \mu, \tau)$ is the negative binomial density on the positive integers with mean and scale $\mu, \tau$. The blue dots are $E_t$ samples generated from $(\theta, \kappa, \mu, \tau) = (-3.2, 3.1, 1.0, 5.0)$ and the solid blue line is the corresponding $p_t$. The maroon dots are $E_t$ samples generated from $(\theta, \kappa, \mu, \tau) = (-2.5, 5.5, 60)$ and the solid maroon line is the corresponding $p_t$. We observe how the differing $(\theta, \kappa, \mu, \tau)$ lead to different clustering patterns and the underlying shape of the probability of seeing events.
Chapter 7

Slow-moving inventory prediction model

This Chapter is largely from a paper due to be submitted to The Annals of Applied Statistics titled “Bayesian hierarchical modelling of sparse count processes in retail analytics”. arXiv:1805.05657

This Chapter presents a novel forecasting methodology for the intermittent demand of slow-moving inventory. Our approach accommodates the structural features exhibited in slow-moving inventory sales data, namely; zero-inflation of sales, the temporal clustering within and across intermittent demand series and the inherent information sparsity within each series. We achieve this by developing a modelling, inferential and predictive method able to learn the dynamics of sparse count processes for SMI products with few to no sales. In particular, we flexibly introduce covariates into the self-exciting model for sparse processes through the link function of the hurdle model of (6.3) similarly to that of ?, introduce pricing covariates into the discretised background intensity of (6.6), and further extend the model to include a cross-excitation contribution allowing for differing intermittent demand series to excite one another. Similarly to the work of ?, we integrate individual products into a Bayesian hierarchical model that accommodates shrinkage and information passing across differing sparse count process, but further allows for excitation, seasonality and information pooling across intermittent demand series to exist in the zero-process component of the hurdle model.

The rest of this Chapter is organised as follows; section 7.1 describes the SMI demand data used in this work. Section 7.2 outlines our hierarchical Bayesian hurdle model with self- and cross-excitation components to model the multiple sparse count processes simultaneously. Section 7.3 presents the results of our sparse count process on the demand data of touchscreen tablets across five South London supermarkets. We conduct a detailed investigation to compare our model to its non-hierarchical equivalent and models without the self- and cross-excitation terms to highlight the benefits the information borrowing and excitation components and discuss the
implications of these results within the context of retail analytics. Section 7.4 concludes with a summary of our contributions and a discussion of possible future developments.

7.1 Data

Our data consist of 17 longitudinal SMI sales processes over 464 days of trading between the dates 1st October 2013 to 7th January 2015. For each product, the daily count corresponds to the aggregated sales of a touchscreen tablet across five large south London supermarkets of a leading UK supermarket retailer. Daily prices as well as seasonality characteristics are available as covariates during the 464 trading days, during which all of the 17 tablets were stocked and in circulation. We split the data into training and test sets, the first 364 trading days between 1st October 2013 to 29th September 2014 (a full trading year excluding Christmas), and the remaining 100 trading days between 30th September 2014 to 7th January 2015 kept as hold out test set. These training and test split gives a balance between providing sufficient training periods where we observe one full year to allow the learning of seasonal trends, whilst having test sets of a reasonable size to allow meaningful forecasts to be made on. This dataset is challenging since we only have one year to learn seasonality from and thus makes a hierarchical model formulation particularly applicable. It should be noted that this data was fully anonymised for general research purposes such that no individual shoppers, or any other sensitive data could be identified.

Table 7.1 provides summary statistics over the training set of the sale counts across the 17 tablet products. The demand across the category is primarily driven by one product, as it accounts for 75% of sales. However, the remaining products are extremely slow moving as indicated by the majority of them only having 0.5-5% non-zero sales days.
### Table 7.1: Summary statistics of SMI demand within tablet category on the training set. The brands have been anonymised with fictitious names for privacy purposes.

<table>
<thead>
<tr>
<th>Product</th>
<th>Brand</th>
<th>total sales</th>
<th>% non-zero sale days</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SPARK</td>
<td>1</td>
<td>0.27</td>
</tr>
<tr>
<td>2</td>
<td>TECHY</td>
<td>409</td>
<td>53.57</td>
</tr>
<tr>
<td>3</td>
<td>TECHY</td>
<td>36</td>
<td>4.12</td>
</tr>
<tr>
<td>4</td>
<td>GADGET</td>
<td>9</td>
<td>1.92</td>
</tr>
<tr>
<td>5</td>
<td>TECHY</td>
<td>5</td>
<td>1.37</td>
</tr>
<tr>
<td>6</td>
<td>TECHY</td>
<td>13</td>
<td>3.57</td>
</tr>
<tr>
<td>7</td>
<td>TECHY</td>
<td>13</td>
<td>3.57</td>
</tr>
<tr>
<td>8</td>
<td>GADGET</td>
<td>13</td>
<td>3.30</td>
</tr>
<tr>
<td>9</td>
<td>GADGET</td>
<td>2</td>
<td>0.27</td>
</tr>
<tr>
<td>10</td>
<td>GADGET</td>
<td>5</td>
<td>1.37</td>
</tr>
<tr>
<td>11</td>
<td>TECHY</td>
<td>1</td>
<td>0.27</td>
</tr>
<tr>
<td>12</td>
<td>TECHY</td>
<td>12</td>
<td>1.92</td>
</tr>
<tr>
<td>13</td>
<td>TECHY</td>
<td>2</td>
<td>0.55</td>
</tr>
<tr>
<td>14</td>
<td>TECHY</td>
<td>3</td>
<td>0.82</td>
</tr>
<tr>
<td>15</td>
<td>TECHY</td>
<td>9</td>
<td>0.82</td>
</tr>
<tr>
<td>16</td>
<td>TECHY</td>
<td>6</td>
<td>1.10</td>
</tr>
<tr>
<td>17</td>
<td>TECHY</td>
<td>3</td>
<td>0.82</td>
</tr>
</tbody>
</table>

These data demonstrate many of the pertinent features of SMI sales processes. Figure 7.1 contrasts the sales and respective prices of a faster-selling tablet against a slower one. The plots illustrate the zero-inflation and that the sales do not show a straightforward dependence on either the prices or the seasonal effects, as indicated by the little movement in demand with respect to changes in prices and season. A clustering effect in the succession of sales within their own demand series is also evident. For example, sales of the right-hand plot in Figure 7.1 fall during the month prior to the festive period, typically thought of as driving demand, but a quick succession of sales follows shortly after this month. This suggests that an excitation process not accounted for by covariate information, as sales bursts occur outside the effects explained by covariate data. Figure 7.2 provides plots suggesting the existence of possible cross-excitation of
tablet sales within a particular brand. We see that successive sales of a tablet in a given brand is often followed by a subsequent sale of another tablet of the same brand.

Figure 7.2: Plots of tablet sales across two brands over proportions of the training set. The left plot corresponds to the GADGET brand and the right plot to the TECHY brand. For each plot, the differing colours correspond to the sales of a particular product within the given brand. The model proposed in subsequent sections (in-particular section 7.2.2), will incorporate the cross-excitation dynamic that the sales of products within the GADGET brand can trigger further sales of differing products within the same GADGET brand (and likewise for the TECHY brand).

The model proposed in the latter section will have various boolean variables that indicate whether a sale has been triggered within a product’s own recent sales history, but also whether a recent sale has been triggered of a different product within the same brand category.

7.2 Model

We model the daily sales of SMI by explicitly modelling the absence of a sale, termed the ‘zero-process’, and the number of sales by the ‘count-process’. Our model uses a Bayesian hierarchical version of the hurdle model of (6.3), with self- and cross-excitation terms in the zero components and self-excitation terms in the count components. More concretely, given \( y_{it} \) sales of some product \( i \) on day \( t \) (where \( y_{it} \in \{0, 1, \ldots\} \)), the probability density function of \( y_{it} \) given covariates \( x_{it}, w_{it} \) is specified as:

\[
p(y_{it} | x_{it}, w_{it}, H_{it}, \tilde{H}_{it}, \theta_i, \beta_i) =
\begin{cases}
  p(w_{it}, H_{it}, \tilde{H}_{it}, \theta_i), & \text{for } y_{it} = 0 \\
  (1 - p(w_{it}, H_{it}, \tilde{H}_{it}, \theta_i)) f(y_{it} | \lambda(x_{it}, H_{it}, \beta_i)), & y_{it} \in \mathbb{N}^+
\end{cases}
\]

with link functions:

\[
\text{logit} \left( p \left( w_{it}, H_{it}, \tilde{H}_{it}, \theta_i \right) \right) = \varphi_i^z(t, w_{it}, \theta_i) + S^z_i(H_{it}, \theta_i) + \tilde{S}^z_i(\tilde{H}_{it}, \theta_i),
\]

\[
\text{log} \left( \lambda(x_{it}, H_{it}, \beta_i) \right) = \varphi_i^c(t, x_{it}, \beta_i) + S^c_i(H_{it}, \beta_i).
\]
Here

\[ x_{it} \text{ and } w_{it} \text{ are the } p \times 1 \text{ and } q \times 1 \text{ vectors of covariate data at time } t, \]

\[ \theta_i, \beta_i \text{ are the collection of coefficients of the zero and count processes respectively,} \]

\[ H_{it} \text{ & } \bar{H}_{it} \text{ are the history self and cross events respectively until } t, \]

\[ \varphi_i^z(t, w_{it}, \theta_i) \text{ is the background intensity function of } t, w_{it}, \theta_i \text{ for the zero process,} \]

\[ \varphi_i^c(t, x_{it}, \beta_i) \text{ is the background intensity function of } t, x_{it}, \beta_i \text{ for the count process,} \]

\[ S_{it}^z(H_{it}, \theta_i) \text{ is the self-excitation function of the zero process (as a function of } H_{it}, \theta_i), \]

all indexed by product \( i \), and with \( \tilde{S}_{it}^z(H_{it}, \theta_i) \) and \( \tilde{S}_{it}^z(H_{it}, \beta_i) \) defined similarly. Here \( p(w_{it}, \theta_i) \)

is the probability of observing a zero sale at time \( t \), and \( f(\cdot \mid \lambda(x_{it}, H_{it}, \beta_i)) \) is a probability

mass function defined on the positive integers parametrised by \( \lambda(x_{it}, H_{it}, \beta_i) \) (as function of \( x_{it}, H_{it}, \beta_i \)).

It is important to note that \( \theta_i \) is the a vector coefficients that represent

the collection of coefficients parametrising the \( \varphi_i^z(t, w_{it}, \theta_i), S_{it}^z(H_{it}, \theta_i) \) and \( \tilde{S}_{it}^z(H_{it}, \theta_i) \) pro-

cesses. \( \beta_i \) is defined similarly. For notational purposes, we express \( \varphi_i^z(t, w_{it}, \theta_i) = \varphi_i^z(t), \]

\( \varphi_i^c(t, x_{it}, \beta_i) = \varphi_i^c(t), S_{it}^z(H_{it}, \theta_i) = S_{it}^z, \tilde{S}_{it}^z(H_{it}, \theta_i) = \tilde{S}_{it}^z \) and \( S_{it}^c(H_{it}, \beta_i) = S_{it}^c \) in subsequent sections. We let \( E_{it} \)

be the indicator for an event day such that \( E_{it} = 1 \) if \( y_{it} \geq 1 \) (a day \( t \)

where at least one sales instance is observed) and \( E_{it} = 0 \) if \( y_{it} = 0 \) (a day \( t \) with no sales for

product \( i \)). The functional forms of the distributions, covariates, parameters, intensity, link and

excitation functions for each of the zero and count processes will be specified in more detail in

subsequent sections.

Our proposed model makes the following three extensions to existing approaches. Firstly,

we use covariates beyond seasonal information, in particular, we use price to assist in forecast-

ing the demand of products along boolean seasonal variables. Secondly, we extend the zero

process of hurdle models to include covariates in the background intensity, along with self- and

cross-excitation terms that aims to capture the auto-correlative and contemporaneous nature of

demand bursts across the SMI category. Thirdly, we build a Bayesian hierarchical model across

the sales \( y_{it} \) (the sales of product \( i \) at time \( t \)) of a SMI category to allow information borrowing.

7.2.1 Covariate data

We introduce covariate data into the model through the background intensity functions \( \varphi_i^z(t) \)

and \( \varphi_i^c(t) \) of (7.1) and (7.2). In the supermarket sales context, this corresponds to a product’s

own price along with seasonal effects (which are common for all products). In particular, these

covariates for a product \( i \) at time \( t \) are logarithm of its price, along with the indicator functions
of week day, month and Christmas period (where we define the Christmas period being the 30 trading days prior to the 25\textsuperscript{th} of December). We summarise these covariates as:

$$\log(p_{it}) = \log(price_{it}) = \text{logarithm price of SMI product } i \text{ at time } t,$$

$$s_t = \left(1_{(t \in \text{Christmas})}, 1_{(t \in \text{Mon})}, \ldots, 1_{(t \in \text{Sat})}, 1_{(t \in \text{Jan})}, \ldots, 1_{(t \in \text{Nov})}\right).$$

We use December and Sunday as reference values when all indicators are equal to zero. Using boolean indicators allows for a natural interpretation in an information borrowing scheme, and further avoids any explicit aggregation across the SMI product data, allowing us to easily handle any issues relating to products coming in and out of circulation. As mentioned in section 7.1, all of the 17 tablets during this analysis were stocked and in circulation, but it is important to note, that the use of boolean seasonal indicator variables is the primary mechanism by which our model handles the issue of non-overlapping sale periods.

We specify the background intensities $\varphi^z_i(t)$, $\varphi^c_i(t)$ of the zero and count processes of (6.3) for product $i$, as:

$$\varphi^z_i(t) = \theta_{i1} + \theta_{i2} \log(p_{it}) + \sum_{k=1}^{18} \theta_{i(k+2)} s_{kt} \quad (7.3)$$

$$\varphi^c_i(t) = \beta_{i1} + \beta_{i2} \log(p_{it}) \quad (7.4)$$

where \{\theta_{i1}, \ldots, \theta_{i20}\} and \{\beta_{i1}, \beta_{i2}\} are the parameters associated with the zero and count processes respectively for product $i$. The $j$ index of $\theta_{ij}$ ranges from $1 - 20$ to include the 1 additive constant, 1 log price variable, 6 week day, 11 month and 1 Christmas indicators. These drift functions (7.3) and (7.4) describe the background intensities of the processes absent of excitation. Thus, in the zero process, we expect the background intensity to depend on a linear combination of log(price), seasonal effects and some additive constant through a given link function, whereas in the count process, we expect the background intensity to depend on a linear combination of log(price) and some additive constant through a given link function. We restrict the background intensity of the count process to exclude seasonal effects to reduce model complexity and the possibility of over-fitting. It is important to note, that for a given product $i$ the count process only exists for $t$ with $E_{it} = 1$. This reduces the count process data has to train on compared to the zero process. We now denote these covariates as $w_{it} = (p_{it}, s_t)$ and $x_{it} = (p_{it})$ for the zero and count processes respectively in line with notation of (7.7).
7.2. Self- and cross-excitation

SMI demand of different but comparable products may occur in auto-correlative and contemporaneous ‘bursts’, in that, sales of a particular product may be followed by sales of a comparable product in the immediate future. These bursts can be a result of external advertising campaigns or viral dynamics, but importantly the apparent excitation not only happens auto-correlatively, but also contemporaneously across products. In the SMI context, cross-excitation is suspected to occur within brand, i.e. an instance of demand for a product leads to a higher probability of demand of a product from the same brand over the subsequent days. Concretely, we define \( \tilde{E}_{it} \) as the indicator for a cross event day of product \( i \) of some brand such that \( \tilde{E}_{it} = 1 \) if \( \sum_{k \in B \setminus \{i\}} y_{kt} \geq 1 \), where \( B \) is the set of indices corresponding to products of the brand, and \( \tilde{E}_{it} = 0 \) if \( \sum_{k \in B \setminus \{i\}} y_{kt} = 0 \). Thus the indicator \( \tilde{E}_{it} \) is 1 if there is at least one sale within the brand at time \( t \) and 0 otherwise. We denote the history of cross-events up to but not including \( t \) as \( \tilde{H}_{i(t-1)} = (\tilde{E}_{i1}, \ldots, \tilde{E}_{i(t-1)}) \).

The corresponding shot noise process with the self and cross-excitation of product \( i \) then becomes:

\[
S_{it} = \sum_{j < t} \kappa_i E_{it} g(t - j \mid \zeta_i) \quad (7.5)
\]

\[
\tilde{S}_{it} = \sum_{j < t} \tilde{\kappa}_i \tilde{E}_{it} g(t - j \mid \tilde{\zeta}_i) \quad (7.6)
\]

where \( \kappa_i, \tilde{\kappa}_i \) are the trigger constants for the self- and cross-excitation respectively and \( g \) is some probability mass function parametrised by \( \zeta_i \) and \( \tilde{\zeta}_i \) controlling the shape of future self and cross-excitation respectively. Our cross-excitation formulation of (7.6) is closely related to the multivariate Hawkes process [?], where we fix all cross-excitation kernels of a given product to 0 that correspond to a different brand, and have shared cross-excitation kernels with shared parameters for products corresponding to the same brand. We denote these collections of self- and cross-excitation parameters as \( \gamma_i = (\kappa_i, \zeta_i) \) and \( \tilde{\gamma}_i = (\tilde{\kappa}_i, \tilde{\zeta}_i) \) respectively.
7.2.3 Self and cross exciting hurdle model

Our SMI model uses the hurdle model specification of (7.7). In particular, for the zero process, we use the background intensity \( \varphi_i^z(t) \) of (7.3) along with self- and cross-excitation components specified in (7.5) and (7.6). For the count process, we use background intensity \( \varphi_i^c(t) \) of (7.4) with self-excitation term of (7.5). Our model is indexed by 17 longitudinal demand series from the tablets category over 464 (training+test) days of trading between the dates 1st October 2013 to 7th January 2015. The full details of the hurdle model mass function of the hurdle model is specified as:

\[
p(y_{it} | x_{it}, w_{it}, H_{it}, H_{it}, \theta_i, \beta_i) = \begin{cases} 
  p(w_{it}, H_{it}, H_{it}, \theta_i), & \text{for } y_{it} = 0 \\
  (1 - p(w_{it}, H_{it}, H_{it}, \theta_i)) f(y_{it} | \lambda(x_{it}, H_{it}, \beta_i), \phi), & y_{it} \in \mathbb{N}^+ 
\end{cases}
\] 

(7.7)

with \( f(y_{it} | \lambda, \phi) = \binom{y_{it} - 2 + \phi}{y_{it} - 1} \left( \frac{\lambda - 1}{\lambda - 1 + \phi} \right)^{y_{it} - 1} \left( \frac{\phi}{\lambda - 1 + \phi} \right)^{\phi} \) and \( \phi = 1 \) which is the probability mass function of the shifted negative binomial distribution (NB) and \( H_{it}, H_{it}, w_{it} \) and \( x_{it} \) are as defined in sections 7.2.2 and 7.2.1 respectively. We opt for a shifted NB distribution over a shifted Poisson distribution on the positive counts due to the known shortcomings of the Poisson distribution at not accommodating over-dispersion adequately [7]. We specify the link functions as:

\[
\logit \left( p \left( w_{it}, H_{it}, H_{it}, \theta_i \right) \right) = \varphi_i^z(t) + S^z_{it} + \tilde{S}^z_{it} \tag{7.8}
\]

\[
\log(\lambda(x_{it}, H_{it}, \beta_i)) = \varphi_i^c(t) + S^c_{it} \tag{7.9}
\]

\( \varphi_i^z(t) \) and \( \varphi_i^c(t) \) are as defined from (7.3) and (7.4) respectively. We define \( S^z_{it} = \sum_{s < t} \kappa^s_{it} E_{it}(t - s \mid \mu^s_i, \tau^s_i) \) and \( \tilde{S}^z_{it} = \sum_{s < t} \tilde{\kappa}^s_{it} E_{it}(t - s \mid \tilde{\mu}^s_i, \tilde{\tau}^s_i) \) similarly to (7.5) and (7.6) respectively with \( g(t \mid \mu, \tau) = \left( \frac{t - 2 + \tau}{t - 1 + \tau} \right) \left( \frac{\mu - 1}{\mu - 1 + \tau} \right)^{t-1} \left( \frac{\tau}{\mu - 1 + \tau} \right)^{\tau} \) as the shifted NB distribution. We similarly define \( S^c_{it} = \sum_{s < t} \kappa^c_{it} E_{it}(t - s \mid \mu^c_i, \tau^c_i) \). We denote the collection of shot parameters as \( \gamma_i = (\kappa^c_i, \mu^c_i, \tau^c_i) \), \( \gamma_i = (\kappa^c_i, \mu^c_i, \tau^c_i) \) and \( \gamma_i = (\kappa^c_i, \mu^c_i, \tau^c_i) \) and collectively denote \( \theta_i = (\theta_{i1}, \ldots, \theta_{i20}, \gamma^c_i, \gamma^z_i) \) and \( \beta_i = (\beta_{i1}, \beta_{i2}, \gamma^z_i) \).

Special attention is paid to the specification of hierarchical priors over the collection \( \theta_i \) and \( \beta_i \), as they are the mechanism through which we penalise complexity and pool information. In particular, we specify \( \theta_{ij} \sim N(\rho^z_j, (\sigma^z_j)^2) \) and \( \rho^z_j \sim N(\theta^z_j, (\zeta^z_j)^2) \) and fix \( (\sigma^z_j)^2 \) for \( j = 1, \ldots, 20 \) and similarly specify \( \beta_{ij} \sim N(\rho^c_j, (\sigma^c_j)^2) \) and \( \rho^c_j \sim N(\theta^c_j, (\zeta^c_j)^2) \) and fix \( (\sigma^c_j)^2 \) for each \( j = 1, 2 \). For parameters of the shot function \( S^z_{it} \), we specify \( \gamma^z_{ij} \sim \text{Gamma}(\eta^z_j, \nu^z_j) \) with \( \eta^z_j \sim \text{Gamma}(\alpha^z_j, \delta^z_j) \) and fix \( \nu^z_j \) for each \( j = 1, 2, 3 \). We specify priors on \( \gamma^z_{ij} \) and \( \gamma^z_{ij} \) similarly. The full details of hierarchical prior specification are contained in appendix B.1.3 and B.1.5. Thus, by denoting:
where over some interval of time, we can write the full posterior \( p_{\text{post}}(\theta_1, \ldots, \theta_{17}, \beta_1, \ldots, \beta_{17} | Y, X, W, H, \tilde{H}) \) as:

\[
p_{\text{post}}(\theta_1, \ldots, \theta_{17}, \beta_1, \ldots, \beta_{17} | Y, X, W, H, \tilde{H}) = p_{\text{post}}(\theta_1, \ldots, \theta_{17} | Y, W, H, \tilde{H}) \times p_{\text{post}}(\beta_1, \ldots, \beta_{17} | Y, X, H)
\]

since the posteriors over the zero and count processes are separable, each denoted as \( p_{\text{post}}(\theta_1, \ldots, \theta_{17} | Y, W, H, \tilde{H}) \) and \( p_{\text{post}}(\beta_1, \ldots, \beta_{17} | Y, X, H) \) respectively, where:

\[
p_{\text{post}}(\theta_1, \ldots, \theta_{17} | Y, W, H, \tilde{H}) = \prod_{i=1}^{17} \prod_{t=1}^{364} p(w_{it}, H_{it}, \tilde{H}_{it}, \theta_i)^{E_i} (1 - p(w_{it}, H_{it}, \tilde{H}_{it}, \theta_i))^{(1-E_i)} \times \prod_{j=1}^{20} f_{\text{norm}}(\theta_i | \rho_j^\gamma, (\sigma_j^\gamma)^2) f_{\text{norm}}(\rho_j^\beta | \delta_j^\gamma, (\zeta_j^\beta)^2) \times \prod_{j=1}^{3} f_{\text{gamma}}(\gamma_{ij} | \eta_j^\gamma, \nu_j^\gamma) f_{\text{gamma}}(\eta_j^\gamma | \alpha_j^\gamma, \delta_j^\gamma) \times \prod_{j=1}^{3} f_{\text{gamma}}(\gamma_{ij} | \eta_j^\gamma, \nu_j^\gamma) f_{\text{gamma}}(\eta_j^\gamma | \alpha_j^\gamma, \delta_j^\gamma)
\]

\[
p_{\text{post}}(\beta_1, \ldots, \beta_{17} | Y, X, H) = \prod_{i=1}^{17} \prod_{t \in T_i} f(y_{it} | \lambda(x_{it}, H_{it}, \beta_i), \phi) \times \prod_{j=1}^{20} f_{\text{norm}}(\beta_i | \rho_j^\gamma, (\sigma_j^\gamma)^2) f_{\text{norm}}(\rho_j^\beta | \delta_j^\gamma, (\zeta_j^\beta)^2) \times \prod_{j=1}^{3} f_{\text{gamma}}(\gamma_{ij} | \eta_j^\gamma, \nu_j^\gamma) f_{\text{gamma}}(\eta_j^\gamma | \alpha_j^\gamma, \delta_j^\gamma)
\]

where \( T_i = \{ t | y_{it} > 0 \} \), i.e. \( T_i \) are the set time indices corresponding to sale days for product \( i \) over some interval of time, \( E_i \) defined as in section 7.2.2 and \( f_{\text{gamma}}(x | \alpha, \beta) = \frac{\beta^n}{\Gamma(n)} x^{n-1} e^{-x\beta} \) and \( f_{\text{norm}}(x | \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma^2} e^{- \frac{(x-\mu)^2}{2\sigma^2}} \).
Chapter 7. Slow-moving inventory prediction model

7.3 Results

We fit variations of the model (7.7) to the 17 longitudinal SMI sales processes over 364 days of trading between the dates 1st October 2013 to 29th September 2014. We denote time interval over which we train our models as $T_{\text{train}}$. A hold out test set over 100 trading days between 30th September 2014 to 7th January 2015 is used to evaluate the predictive performance of the model variations for both the zero and count processes. We denote this test interval as $T_{\text{test}}$. As the zero and count processes are completely separable, we perform model inference and analysis separately.

7.3.1 Zero process variations

To assess the predictive benefits of the additions of self-excitation, cross-excitation and hierarchical components to the zero process of the hurdle model of (7.7), we implement cumulative variations of both the link functions as well as the hierarchical layering used in the modelling. These model variations are the following:

Z.1 Baseline model (Base\(_z^1\)): We learn the zero process of the hurdle model (7.7) with link function:

$$\text{logit} \left( p \left( w_{it}, H_{it}, \tilde{H}_{it}, \theta_i \right) \right) = \varphi_{i}^z$$

for each $i = 1, \ldots, 17$, i.e. a constant probability per product. This is the Bayesian baseline model as it estimates the zero-process independent of covariate information. The $\varphi_{i}^z$ is estimated using non-informative priors. The performance of this model is used to verify the relative benefits that covariate information brings to SMI zero-process modelling.

Z.2 Hierarchical Bayesian (HB\(_z^2\)): We learn the zero process of the hurdle model (7.7) with link function:

$$\text{logit} \left( p \left( w_{it}, H_{it}, \tilde{H}_{it}, \theta_i \right) \right) = \varphi_{i}^z(t)$$

for each $i = 1, \ldots, 17$ with the hierarchical prior formulation discussed in section 7.2.3. This model is implemented to establish a benchmark of the simplest regression model, i.e. a model that excludes information of previous events and is used to verify the relative benefits of self excitation and cross-excitation.

Z.3 Bayesian with self-excitation (BE\(_z^3\)): We learn the zero process of the hurdle model
(7.7) with link function:

\[
\text{logit} \left( p \left( w_{it}, H_{it}, \tilde{H}_{it}, \theta_i \right) \right) = \varphi_z^i(t) + S_{it}^z
\]

for each \( i = 1, \ldots, 17 \) but exclude the hierarchical layer of the priors articulated in section 7.2.3. More concretely, we fix the parameters \( (\rho_j^z, (\sigma_j^z)^2) \) and \( (\eta_j^z, \nu_j^z) \) as constants rather for each \( j \). This model is implemented to establish a benchmark of a model with excitation but without information borrowing between products and is used to verify the relative benefits of information borrowing between products.

**Z.4 Hierarchical Bayesian with self-excitation (HBE^z):** We learn the zero process of the hurdle model (7.7) with link function:

\[
\text{logit} \left( p \left( w_{it}, H_{it}, \tilde{H}_{it}, \theta_i \right) \right) = \varphi_z^i(t) + S_{it}^z
\]

for each \( i = 1, \ldots, 17 \) with the hierarchical prior formulation discussed in section 7.2.3. This model is implemented to demonstrate the possible benefits of self-excitation in the standard zero inflated regression model. We use HB^z as reference as to what self-excitation provides over the model that exclusively uses the regression covariates.

**Z.5 Fixed Bayesian with self-excitation (FBE^z):** We learn the zero process of the hurdle model (7.7) with link function:

\[
\text{logit} \left( p \left( w_{it}, H_{it}, \tilde{H}_{it}, \theta_i \right) \right) = \varphi_z^i(t) + S_{it}^z
\]

for each \( i = 1, \ldots, 17 \). In this implementation the parameters across the products are shared, i.e. the parameters of \( \varphi_z^i(t) \) and \( S_{it}^z \) are identical across all 17 products, and hence \( \theta_i = (\theta_{i1}, \ldots, \theta_{i20}, \gamma_i^z) = (\theta_1, \ldots, \theta_{20}, \gamma^z) \). This is the maximal information sharing regime where the borrowing is to the extent that the parameters are identical across all products. We fix the parameters \( (\rho_j^z, (\sigma_j^z)^2) \) and \( (\eta_j^z, \nu_j^z) \) as constants. This is compared with models BE^z and HBE^z to assess the benefits of information borrowing.

**Z.6 Bayesian with self and cross-excitation (BEC^z):** We learn the zero process of the
hurdle model (7.7) with link function:

$$\text{logit} \left( p \left( w_{it}, H_{it}, \tilde{H}_{it}, \theta_i \right) \right) = \varphi_i^2(t) + S_{it}^2 + \tilde{S}_{it}^2$$

for each $i = 1, \ldots, 17$ but exclude the hierarchical layer of the priors articulated in section 7.2.3. Prior specification is similar to that of BE\textsuperscript{z} but extended to include $\tilde{\gamma}_i^z$. This is a benchmark of a model with self and cross-excitation but without an information borrowing scheme.

Z.7 Hierarchical Bayesian with self and cross-excitation (HBEC\textsuperscript{z}): This model is the full model discussed in the section 7.7. We learn the zero process of the hurdle model (7.7) with link function:

$$\text{logit} \left( p \left( w_{it}, H_{it}, \tilde{H}_{it}, \theta_i \right) \right) = \varphi_i^z(t) + S_{it}^z + \tilde{S}_{it}^z$$

for each $i = 1, \ldots, 17$ with the hierarchical prior formulation discussed in section 7.2.3. The hyper-priors are selected to balance borrowing across products and penalising complexity. This hierarchical model will be cross referenced with model BEC\textsuperscript{z}.

Z.8 Fixed Bayesian with self and cross-excitation (FBEC\textsuperscript{z}): We learn the zero process of the hurdle model (7.7) with link function:

$$\text{logit} \left( p \left( w_{it}, H_{it}, \tilde{H}_{it}, \theta_i \right) \right) = \varphi_i^z(t) + S_{it}^z + \tilde{S}_{it}^z$$

for each $i = 1, \ldots, 17$. In this implementation the parameters across the products are shared, i.e. the parameters of $\varphi_i^z(t), S_{it}^z, \tilde{S}_{it}^z$ are identical across all of the 17 products, and hence $\theta_i = (\theta_{i1}, \ldots, \theta_{i20}, \gamma_i^z, \tilde{\gamma}_i^z) = (\theta_1, \ldots, \theta_{20}, \gamma^z, \tilde{\gamma}^z)$. This is the maximal information sharing regime to the extent that the parameters are identical across all products. We fix the parameters $(\rho_j^z, (\sigma_j^z)^2), (\eta_j^z, \tilde{\eta}_j^z)$ and $(\nu_j^z, \nu_j^z)$ as constants for each $j$. This model is compared with models BEC\textsuperscript{z} and HBEC\textsuperscript{z} to assess the benefits of information borrowing.

Parameter inference of models Base\textsuperscript{z}, HB\textsuperscript{z}, BE\textsuperscript{z}, HBE\textsuperscript{z}, FBE\textsuperscript{z}, BEC\textsuperscript{z}, HBEC\textsuperscript{z} and FBEC\textsuperscript{z} is performed by Hamiltonian Monte Carlo algorithm as outlined during section 2.2.3, and is implemented by the rstan library [?]. Inference is performed by the Hamiltonian Monte Carlo algorithm (via the rstan library) due to its ease at implementing Bayesian hierarchical models and because of its success at efficiently producing uncorrelated MCMC samples. The RStan code
for the model HBEC\textsuperscript{z} is included in appendix B.1.2. All other RStan models are simplifications of the HBEC\textsuperscript{z} implementation. Convergence was confirmed by Heidelberger Welch statistic across all models and parameters \cite{heidel}. The specification of hyper-priors is included in appendix B.1.3. Further MCMC implementation details are included in B.1.6.

### 7.3.2 Zero process fits

The predictive performance of models Base\textsuperscript{1-7}, HB\textsuperscript{z}, BE\textsuperscript{z}, HBE\textsuperscript{z}, FBE\textsuperscript{z}, BEC\textsuperscript{z}, HBEC\textsuperscript{z} and FBEC\textsuperscript{z} is assessed by calculating how capable each model is at predicting the probability of a sale occurring on a given day over the test interval $T^{test}$ (30\textsuperscript{th} September 2014 to 7\textsuperscript{th} January 2015) for each $i = 1, \ldots, 17$ given the history of self and cross events $H_{it}, \tilde{H}_{it}$, covariate information $w_{it}$ and posterior samples. We denote the $s^{th}$ posterior sample of $\theta_i$ of the $i^{th}$ product as $\theta_{s}^{i}$. The sales occurrence probabilities are based on the posterior samples $\Theta_{s}^{i}$ inferred from the training interval $T^{train}$ (between 1\textsuperscript{st} October 2013 to 29\textsuperscript{th} September 2014). More precisely, we apply the following methodology over the test interval:

1. On given day $t$ on the test interval and $s^{th}$ posterior sample, we compute the full predictive posterior distribution of the probability of a sale occurring based conditioned on $w_{it}, H_{it}, \tilde{H}_{it}, \theta_{s}^{i}$ for each product $i = 1, \ldots, 17$.

2. We observe $y_{i(t+1)}$ (the number of sales of product $i$ on day $t + 1$) for each $i = 1, \ldots, 17$ and update the self and cross event histories $H_{i(t+1)}, \tilde{H}_{i(t+1)}$ for $i = 1, \ldots, 17$.

3. Repeat steps for each $t$, for each sample $s$ and $i$ over the test period of 30\textsuperscript{th} September 2014 to 7\textsuperscript{th} January 2015.

This builds up a set of daily predictive posterior probabilities $p_{its}$ for each $s = 1, \ldots, S$ for the probability of a sale on a given day over $T^{test}$ for each $i = 1, \ldots, 17$ based on posterior samples inferred from $T^{train}$ conditioned on $w_{it}, H_{it}, \tilde{H}_{it}, \theta_{i}$. Parameter inference is performed only once (over the training interval), with the predictive posterior probabilities being computed from the inferred posterior values from the fixed training interval (i.e. parameter inference is not rerun over the additional days observed during the test interval).

To evaluate the predictive performance of the models we use the log pointwise predictive density \cite{ref} for each of the products $i = 1, \ldots, 17$. The log pointwise predictive density is a score that indicates the predictive accuracy of a model over a dataset - the larger the log pointwise predictive density score, the better predictive accuracy of a model. The log pointwise predictive
density \( \text{lppd}^z \) for the zero process is given by:

\[
\text{lppd}^z_i = \sum_{t \in T} \log \left( \frac{1}{S} \sum_{s=1}^{S} p_{its}^E (1 - p_{its}) (1 - E_{its}) \right)
\]

where \( p_{its} \) is the prediction probability of a sale occurring for product \( i \) from posterior sample \( s \) for some model of interest. Table 7.2 provides the \( \text{lppd}^z \) scores across products and models \( \text{Base}^z, \text{HB}^z, \text{BE}^z, \text{HBE}^z, \text{FBE}^z, \text{BEC}^z, \text{HBEC}^z \) and \( \text{FBEC}^z \). The subscript of \( \text{lppd} \) denotes the log pointwise predictive density for a given fitted model and product (e.g. \( \text{lppd}^z_{\text{HBEC},i} \) is the log pointwise predictive density for model \( \text{HBEC}^z \) and product \( i \)). We compute the \( \text{lppd}^z_i \) over both the test and training intervals \( T^{\text{test}} \) and \( T^{\text{train}} \) which we denote as \( \text{lppd}^z_{i,\text{test}} \) and \( \text{lppd}^z_{i,\text{train}} \) respectively.
Table 7.2: $\text{lppd}^{z,\text{test}}_i$ and $\text{lppd}^{z,\text{train}}_i$ scores of the zero process fits for the models $\text{Base}_1^z$, $\text{HB}_1^z$, $\text{BE}_1^z$, $\text{HBE}_1^z$, $\text{FBE}_1^z$, $\text{BEC}_1^z$, $\text{HBEC}_1^z$ and $\text{FBEC}_1^z$ and each product.

<table>
<thead>
<tr>
<th>Product $i$</th>
<th>$\text{lppd}^{z,\text{test}}_{\text{Base}_1^z, i}$</th>
<th>$\text{lppd}^{z,\text{test}}_{\text{HB}_1^z, i}$</th>
<th>$\text{lppd}^{z,\text{test}}_{\text{BE}_1^z, i}$</th>
<th>$\text{lppd}^{z,\text{test}}_{\text{HBE}_1^z, i}$</th>
<th>$\text{lppd}^{z,\text{test}}_{\text{FBE}_1^z, i}$</th>
<th>$\text{lppd}^{z,\text{test}}_{\text{BEC}_1^z, i}$</th>
<th>$\text{lppd}^{z,\text{test}}_{\text{HBEC}_1^z, i}$</th>
<th>$\text{lppd}^{z,\text{test}}_{\text{FBEC}_1^z, i}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.37</td>
<td>-3.16</td>
<td>-0.32</td>
<td>-2.04</td>
<td>-2.38</td>
<td>-0.32</td>
<td>-1.97</td>
<td>-2.30</td>
</tr>
<tr>
<td>2</td>
<td>-73.47</td>
<td>-65.66</td>
<td>-60.85</td>
<td>-55.87</td>
<td>-57.70</td>
<td>-60.42</td>
<td>$\mathbf{-55.18}$</td>
<td>-57.20</td>
</tr>
<tr>
<td>3</td>
<td>-7.33</td>
<td>-6.81</td>
<td>-6.18</td>
<td>-5.56</td>
<td>$\mathbf{-5.24}$</td>
<td>-6.23</td>
<td>-5.59</td>
<td>-5.27</td>
</tr>
<tr>
<td>4</td>
<td>-29.44</td>
<td>-28.27</td>
<td>-29.30</td>
<td>-28.54</td>
<td>-26.47</td>
<td>-29.00</td>
<td>-28.35</td>
<td>$\mathbf{-26.36}$</td>
</tr>
<tr>
<td>6</td>
<td>-3.67</td>
<td>-5.80</td>
<td>-2.55</td>
<td>-3.63</td>
<td>-2.77</td>
<td>$\mathbf{-2.54}$</td>
<td>-3.63</td>
<td>-2.79</td>
</tr>
<tr>
<td>7</td>
<td>-6.92</td>
<td>-7.42</td>
<td>-5.91</td>
<td>-5.98</td>
<td>$\mathbf{-5.70}$</td>
<td>-6.00</td>
<td>-6.07</td>
<td>-5.76</td>
</tr>
<tr>
<td>8</td>
<td>-6.74</td>
<td>-8.95</td>
<td>-6.47</td>
<td>-6.91</td>
<td>-6.38</td>
<td>-6.42</td>
<td>-6.77</td>
<td>$\mathbf{-6.29}$</td>
</tr>
<tr>
<td>9</td>
<td>-5.97</td>
<td>-7.27</td>
<td>$\mathbf{-5.68}$</td>
<td>-5.98</td>
<td>-6.03</td>
<td>-5.69</td>
<td>-5.93</td>
<td>-5.99</td>
</tr>
<tr>
<td>10</td>
<td>-9.91</td>
<td>-11.30</td>
<td>-10.76</td>
<td>-10.45</td>
<td>-9.74</td>
<td>-10.60</td>
<td>-10.22</td>
<td>$\mathbf{-9.57}$</td>
</tr>
<tr>
<td>12</td>
<td>-9.80</td>
<td>-11.86</td>
<td>-10.48</td>
<td>-10.53</td>
<td>-9.66</td>
<td>-10.30</td>
<td>-10.27</td>
<td>$\mathbf{-9.49}$</td>
</tr>
<tr>
<td>15</td>
<td>$\mathbf{-10.36}$</td>
<td>-11.15</td>
<td>-10.78</td>
<td>-10.49</td>
<td>-10.51</td>
<td>-10.83</td>
<td>-10.52</td>
<td>-10.55</td>
</tr>
<tr>
<td>16</td>
<td>$\mathbf{-5.61}$</td>
<td>-7.47</td>
<td>-6.12</td>
<td>-6.60</td>
<td>-6.27</td>
<td>-6.19</td>
<td>-6.61</td>
<td>-6.27</td>
</tr>
</tbody>
</table>

$\sum_{i=1}^{17} \text{lppd}^{z,\text{test}}_{\text{model}, i} = -242.10 - 238.82 - 214.37 - 209.50 - 213.98 - 213.35 - 207.65 - 212.58$

$\sum_{i=1}^{17} \text{lppd}^{z,\text{train}}_{\text{model}, i} = -708.26 - 699.89 - 609.45 - 662.65 - 675.64 - 608.48 - 662.84 - 675.89$
Interpreting Table 7.2’s lppd$^{z,\text{test}}$ and lppd$^{z,\text{train}}$ scores reveal some interesting findings. Firstly, we observe the model HB$^z$, the zero process model with covariate information, provides a significant improvement in predictive performance compared to baseline models Base$^z_1$ without covariate information. We further see that inclusion of a self-excitation component in 7.3.1 provides a marked improvement over the model HB$^z$ without self-excitation. Figure 7.3 demonstrates an example of the benefit of self-excitation inclusion by comparing the event day prediction performance between models HBE$^z$ and HB$^z$ over a portion of the test set. We observe inclusion of self-excitation produces a 95% credibility interval of model HBE$^z$ that captures a subsequent sale that model HB$^z$ does not immediately after the first sale at $t = 382$.

Table 7.2 further indicates the predictive benefits that hierarchical extensions provide over its non-hierarchical equivalents. Figure 7.4 illustrates an example of the benefit of these hierarchical extensions by comparing event day prediction performance between models HBE$^z$ and BE$^z$ over a portion of the test set. We observe that by information pooling across the intermittent demand series produces a 95% credibility interval of model HBE$^z$ that captures a sale at $t = 446$ (during the Christmas period). This is in spite of there being no sales over the Christmas period of the previous year for this product. In this way, the hierarchical model benefits from inferring parameter values from other intermittent demand series which have observed sales over the previous Christmas period.

Table 7.2 indicates that the cross-excitation expositions of models BEC$^z$, HBEC$^z$ and FBEC$^z$ offer an improvement in event day prediction over the test set compared to their non cross-excitation counterparts (i.e. HBE$^z$, HBE$^z$ and BE$^z$). Interesting, cross-excitation does not offer benefits in terms of the training set; but shows significant predictive gains in the test set.

Figure 7.3: Plots of the predictive models HB$^z$ (left plot) and HBE$^z$ (right plot) for product $i = 13$ over a portion of the test set. The blue and magenta dots represent self and cross event days respectively (i.e. $E_{it}$ and $\tilde{E}_{it}$). The black line is the estimated posterior mean of an event day observation (i.e. $p_{it}$) and the shaded region is the 95% credible interval of these estimates.
7.3. Results

Figure 7.4: Plots of the predictive models $BE^z$ (left plot) and $HBE^z$ (right plot) for product $i = 11$ over a portion of the test set. The blue and magenta dots represent self and cross event days respectively (i.e. $E_{it}$ and $\tilde{E}_{it}$). The black line is the estimated posterior mean of an event day observation (i.e. $p_{it}$) and the shaded region is the 95% credible interval of these estimates.

7.3.3 Count process variations

Similarly to section 7.3.2, the benefits of the excitation and hierarchical component to the count process are verified by implementing cumulative variations in the link functions and hierarchical layerings of the model. These model variations follow the same rationale as with the zero process. In particular:

C.1 Baseline model (Base$^c_i$): We learn the count process of the hurdle model (7.7) with link function:

$$\log(\lambda(x_{it}, H_{it}, \beta_i)) = \varphi^c_i$$

for each $i = 1, \ldots, 17$, i.e. a constant rate per product. This is the Bayesian baseline model as it estimates the zero-process independent of covariate information. The $\varphi^c_i$ is estimated using non-informative priors.

C.2 Hierarchical Bayesian (HB$^c$): We learn the count process of the hurdle model (7.7) with link function:

$$\log(\lambda(x_{it}, H_{it}, \beta_i)) = \varphi^c_i(t)$$

for each $i = 1, \ldots, 17$ with the hierarchical prior formulation discussed in section 7.2.3.

C.3 Bayesian with self-excitation (BE$^c$): We learn the count process of the hurdle model (7.7) with link function:

$$\log(\lambda(x_{it}, H_{it}, \beta_i)) = \varphi^c_i(t) + S^c_{it}$$
for each $i = 1, \ldots, 17$ but exclude the hierarchical layer of the priors articulated in section 7.2.3.

C.4 Hierarchical Bayesian with self-excitation (HBE\textsuperscript{c}): This is the full model discussed in the section 7.2.3. We learn the count process of the hurdle model (7.7) with link function:

$$\log(\lambda(x_{it}, H_{it}, \beta_i)) = \varphi^c(t) + S^c_{it}$$

for each $i = 1, \ldots, 17$ with the hierarchical prior formulation discussed in section 7.2.3.

C.5 Fixed Bayesian with self-excitation (FBE\textsuperscript{c}): We learn the count process of the hurdle model (7.7) with link function:

$$\log(\lambda(x_{it}, H_{it}, \beta_i)) = \varphi^c(t) + S^c_t$$

for each $i = 1, \ldots, 17$.

Parameter inference of models HB\textsuperscript{c}, BE\textsuperscript{c}, HBE\textsuperscript{c} and FBE\textsuperscript{c} is performed by Hamiltonian Monte Carlo sampling algorithm and is implemented by the rstan library [?]. The RStan code for the model HBE\textsuperscript{c} is included in appendix B.1.4. All other RStan models are simplifications of the HBE\textsuperscript{c} implementation. Convergence was confirmed by Heidelberger Welch statistic across all models and parameters [?]. The specification of these hyper-priors and constant of models HB\textsuperscript{c}, BE\textsuperscript{c}, HBE\textsuperscript{c}, and FBE\textsuperscript{c} is included in appendix B.1.5. For further MCMC implementation details refer to appendix B.1.7.

7.3.4 Count process fits

Similarly as with the zero processes outlined in section 7.3.2, we test the performance of the count variation models Base\textsuperscript{c}, HB\textsuperscript{c}, BE\textsuperscript{c}, HBE\textsuperscript{c} and FBE\textsuperscript{c} by calculating how capable each model is of predicting the volume of sales on event days (i.e. days when sale has been observed) over the test interval $T^{test}$ (between 30th September 2014 to 7th January 2015) for each $i = 1, \ldots, 17$ given the history of self events $H_{it}$, covariate information $x_{it}$ and posterior samples. We denote the $s^{th}$ posterior sample of $\beta_i$ of the $i^{th}$ product as $\beta^c_{is}$. The predictive distribution is based on the posterior samples fits inferred from the training interval $T^{train}$ (between 1st October 2013 to 29th September 2014). We apply the following methodology over the test interval:

1. On event day $t$ (i.e. $E_t = 1$) on the test interval and $s^{th}$ posterior sample, we compute the full predictive posterior distribution of the volume of sales occurring conditioned on
7.3. Results

\[ H_{it}, x_{it}, \beta^i \] for each \( i = 1, \ldots, 17 \).

2. We observe \( y_{i(t+1)} \) (the volume of sales of product \( i \) on day \( t+1 \)) for each \( i = 1, \ldots, 17 \) and update the self event histories \( H_{i(t+1)} \) for \( i = 1, \ldots, 17 \).

3. Repeat steps for each \( t \), for each sample \( s \) and \( i \) over the test period of 30th September 2014 to 7th January 2015.

This builds up a set of posterior rates \( \lambda_{its} \) for samples \( s = 1, \ldots, S \) for the probability of the number of sales on a given event day over \( T^{test} \) for each \( i = 1, \ldots, 17 \) based on our posterior sample fits inferred from \( T^{train} \) conditioned on \( x_{it}, H_{it}, \beta \). As with the zero process, parameter inference is performed only once over the training interval, and not rerun over the additional days observed during the test interval.

Similarly as with the zero process, we evaluate the predictive performance by calculating the log pointwise predictive density for each of the products \( i = 1, \ldots, 17 \). The log pointwise predictive density \( \text{lppd}^c \) for the count process is given by:

\[
\text{lppd}^c_i = \sum_{t \in T_i} \log \left( \frac{1}{S} \sum_{s=1}^{S} \left( y_{ik} - 2 + \phi \right) \left( \frac{\lambda_{its} - 1}{\lambda_{its} - 1 + \phi} \right)^{y_{ik} - 1} \left( \frac{\phi}{\lambda_{its} - 1 + \phi} \right)^{\phi} \right)
\]

where \( \phi = 1 \) and \( \lambda_{its} \) is the prediction mean of count sales occurring for product \( i \) from the \( s^{th} \) posterior sample for some model of interest and \( T_i = \{t|y_{it} > 0\} \), i.e. \( T_i \) are the set time indices corresponding to sale days for product \( i \) over some interval of time. Table 7.3 provides the \( \text{lppd}^c \) scores for across products and models Base\( _1 \), HB\( ^c \), HBE\( ^c \), BE\( ^c \) and FBE\( ^c \). Interpreting Table 7.3’s \( \text{lppd}^c_{model,i} \) scores reveals some interesting findings. Firstly, we observe that the model variations of HB\( ^c \), BE\( ^c \), HBE\( ^c \) and FBE\( ^c \) perform significantly better than the baseline model Base\( _1 \) with no covariates. This provides evidence in favour of the hypothesis that the model of HB\( ^c \), BE\( ^c \), HBE\( ^c \) and FBE\( ^c \) are capturing the SMI count process in a meaningful way. Similarly as with the zero process, Table 7.3 indicates the count process uniformly benefits from the inclusion of self-excitation. We further see that the count process benefits more from the hierarchical borrowing across the intermittent demand series. This is understandable given the level of sparsity in the count process. As Table 7.1 indicates, the order of sales that the each intermittent demand series has is very small (typically in the order 3-20 sales), and thus it may be expected that information borrowing would particularly benefit the individual models. An example of this additive strength of the hierarchical exposition of the count model variations is illustrated by Figure 7.5. This plot shows a histogram of \( y_{it} \) against the sum of \( \sum_{t:y_{it}=k} y_{it} \) (for product 12) with corresponding 95% credibility intervals of posterior predictive distributions.
Table 7.3: lppd\textsuperscript{c} scores of the count process fits for the models Base\textsuperscript{c}, HB\textsuperscript{c}, BE\textsuperscript{c}, HBE\textsuperscript{c} and FBE\textsuperscript{c} for each product and fitted model.

<table>
<thead>
<tr>
<th>Product (i)</th>
<th>(\text{lppd}^{c,\text{test}}_{\text{Base},i})</th>
<th>(\text{lppd}^{c,\text{test}}_{\text{HB},i})</th>
<th>(\text{lppd}^{c,\text{test}}_{\text{BE},i})</th>
<th>(\text{lppd}^{c,\text{test}}_{\text{HBE},i})</th>
<th>(\text{lppd}^{c,\text{test}}_{\text{FBE},i})</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>3</td>
<td>-0.08</td>
<td>-0.07</td>
<td>-0.08</td>
<td>-0.66</td>
<td>-0.82</td>
</tr>
<tr>
<td>4</td>
<td>-2.54</td>
<td>-1.40</td>
<td>-1.50</td>
<td>-1.60</td>
<td>-1.63</td>
</tr>
<tr>
<td>5</td>
<td>-4.99</td>
<td>-4.16</td>
<td>-7.92</td>
<td>-3.17</td>
<td>-2.89</td>
</tr>
<tr>
<td>6</td>
<td>-0.01</td>
<td>-0.00</td>
<td>-0.04</td>
<td>-0.22</td>
<td>-0.33</td>
</tr>
<tr>
<td>7</td>
<td>-3.98</td>
<td>-3.98</td>
<td>-3.80</td>
<td>-2.04</td>
<td>-2.05</td>
</tr>
<tr>
<td>8</td>
<td>-1.02</td>
<td>-1.09</td>
<td>-1.03</td>
<td>-0.68</td>
<td>-0.62</td>
</tr>
<tr>
<td>9</td>
<td>-3.46</td>
<td>-3.47</td>
<td>-3.47</td>
<td>-2.33</td>
<td>-2.55</td>
</tr>
<tr>
<td>10</td>
<td>-6.19</td>
<td>6.46</td>
<td>-6.48</td>
<td>-5.23</td>
<td>-5.51</td>
</tr>
<tr>
<td>11</td>
<td>-2.04</td>
<td>-2.05</td>
<td>-1.95</td>
<td>-0.66</td>
<td>-0.51</td>
</tr>
<tr>
<td>12</td>
<td>-1.57</td>
<td>-2.64</td>
<td>-1.63</td>
<td>-1.80</td>
<td>-1.73</td>
</tr>
<tr>
<td>13</td>
<td>-0.10</td>
<td>-0.08</td>
<td>-0.09</td>
<td>-0.55</td>
<td>-0.54</td>
</tr>
<tr>
<td>(\sum_{i=1}^{17} lppd^{c,\text{test}}_{model,i})</td>
<td>-53.64</td>
<td>-53.60</td>
<td>-51.32</td>
<td>-46.33</td>
<td>-47.70</td>
</tr>
<tr>
<td>(\sum_{i=1}^{17} lppd^{c,\text{train}}_{model,i})</td>
<td>-336.81</td>
<td>-335.21</td>
<td>-308.58</td>
<td>-325.15</td>
<td>-329.90</td>
</tr>
</tbody>
</table>

for the models BE\textsuperscript{c} and HBE\textsuperscript{c}. We observe that the hierarchical model variation (even without the excitation) produces much tighter credibility intervals around the observed data than the model without information borrowing. However, the best performing models are ones with both information borrowing and self-excitation. Figure 7.6 illustrates the optimal performance of HBE\textsuperscript{c} over HB\textsuperscript{c}. In this plot, we see the 95% credibility intervals produced from model HBE\textsuperscript{c} for the higher count instances (7+) capture the observed aggregated count instances, whereas the HB\textsuperscript{c} credibility intervals fail to do so. We further see the aggregate log pointwise predictive density of \(\sum_{i=1}^{17} lppd^{c,\text{train}}_{model,i}\) of table 7.3 provides more evidence that model HBE\textsuperscript{c} is the best fitting model, as this is maximised relative to the other hierarchical model variations.

Figure 7.5: Histograms of \(\sum_{t:y_{it}=k} y_{it}\) with corresponding 95\% credible intervals of the posterior predictive distributions for models BE\textsuperscript{c} (left plot) and HBE\textsuperscript{c} (right plot) for product \(i = 12\). The lower of 2.5\% credible interval (the lower bound of the whisker bars) for \(\sum_{t:y_{it}=1} y_{it}\) will at best be \(\sum_{t:y_{it}=1} y_{it}\) \textsuperscript{1}. This is since the count distribution is lower bounded by 1.
7.3. Results

7.3.5 Retail analytics discussion

The output of models outlined in sections 7.3.1 and 7.3.3 provides interesting interpretations from a retail analytics perspective. Firstly, we observe that covariate data $w_{it}$, $x_{it}$ as specified in 7.2.1 improves forecasting performance for the intermittent demand series of SMI products. This is indicated in both HB$^c$ and HB$^z$ - models with regression parameters and no form of excitation - outperforming their baseline counterparts on both the training and test sets. This importantly sheds light into the intermittent demand of SMI, in that it demonstrates covariate data such as prices and seasonality ought to be incorporated into training forecasting models as it seems predictions are improved from their inclusion.

Our findings further support the hypothesis that intermittent demand forecasting is improved when excitation dynamics are incorporated into models. This supports the findings of ? and ? in which they establish that models incorporating the recent demand history outperform temporally static models. This is important because it ultimately allows retailers to circumvent over-stocking that typically results from inaccurate forecasting [?]. However, our findings reveal some aspects of intermittent demand forecasting that go beyond the work of ? and ?. Namely, we establish that the temporal excitation exists even if you condition on the seasonal trends and pricing information of $w_{it}$, $x_{it}$. This suggests that temporal excitation is systematic and occurs beyond the variables traditionally utilised in forecasting models. We furthermore find that temporal excitation is manifested at lags greater than 1. Figure B.2 demonstrates that $\mu^*_i$ (the mean of excitation function of $g(\cdot \mid \mu, \tau)$) is approximately 2 across the majority of products, which implies that 2/3 of the probability mass of $g(\cdot \mid \mu, \tau)$ is placed on lags greater than or equal to 2. This is crucially important, as it indicates that a simple AR(1), or equivalent model only taking the most recent observation into account is possibly not enough compared to the Hawkes process that incorporates the entire history of events.

Figure 7.6: Histograms of $\sum_{t:y_{it}=k} y_{it}$ with corresponding 95% credible intervals of the log of the posterior predictive distributions of $\sum_{t:y_{it}=k} y_{it}$ (sale counts) for models HB$^c$ (left plot) and HBE$^c$ (right plot) for product $i = 2$. 

\[ \log(\sum_{t:y_{it}=k} y_{it}) \]
Thirdly, we also see strong support for the hypothesis that intermittent demand forecasting of SMI products benefits from hierarchical modelling. This is evident when we cross reference the parameter estimates with the posterior predictive distributions produced from each of the models. For example, Figure B.1 shows how a non-hierarchical model can suffer from not observing a range of sale counts on the training set which then translates to poor predictive performance on the test set. Figure B.3 further shows how information pooling to the extent that parameters are fixed across all the intermittent demand series can lead to misfit when heterogeneity appears to exist in the parameter estimates when compared to their hierarchical model counterparts. This has significant implications in the retail analytics context, as it suggests retailers should take into account the hierarchical structure exhibited in intermittent demand forecasting, as prediction is significantly improved when such structure is taken into account. Figure 7.7 are the forecasts of the intermittent demand of two slow-moving-inventory products using the combined zero and count models.

![Figure 7.7: Plots of the combined models HBEz and HBEc for product \( i = 4 \) (left plot) and product \( i = 12 \) (right plot) over the entire training and test sets. The solid blue lines represent the sales of the respective touchscreen tablets and the black dashed lines are the 95\% credible interval of the predictive posterior distribution of the sales counts. The dashed-dotted vertical black line at \( t = 365 \) represents the end and start of the training and test sets respectively.](image)

### 7.4 Summary & future work

During this work we introduce a hierarchical model for the sales of the slow-moving-inventory category of touchscreen tablets across five large supermarkets in south London. We modelled the sales process as a Bayesian hierarchical zero-inflated hurdle regression model with self and cross-excitation components. The model specification is interpretable and allows a deeper understanding of the predictive role that covariates, self-excitation and cross-excitation play in the sales process of slow-moving-inventory and further provides a fully specified predictive distribution over this process. We demonstrated that the hierarchical structure as well as the
self and cross-excitation additions offer a significant improvement in the predictive accuracy of this SMI sales process.

This model has important implications to the challenging issues that retail analytics face when developing SMI models. Firstly, it offers utility in terms of demand and profit forecasting that will allow retailers more accurate predictions of the sales distributions to aid with the issue of inventory management as well as price optimisation over short term horizons. It helps to explain the sources of variation and uncertainty that is exhibited in intermittent demand processes that previously was not well understood. The model also reveals a strong excitation component to these sales which could warrant further investigation as to what the potential underlying factors that could explain the observed excitation (e.g. marketing campaigns). We further note, that though there are many other approaches of specifying the cross-excitation relationship between pairwise products, our adopted approach of cross-excitation within brand provides an intuitive and computationally simple method of expressing the suspected temporal cross-correlation.

This work could be extended in many different directions. For example, a variable selection methodology could be introduced into the covariate predictors for each of the regression models. Our approach specified a priori the cross-excitation structure by defining an excitation event as a sale occurring within the same brand; it could be an interesting to assess whether the excitation structure could instead be inferred from the data.
Chapter 8

Conclusion

The field of retail analytics is a research area characterised by having a vast range of interesting challenges to choose from. These problems vary from the design of recommendation strategies awarding customer loyalty, to optimal inventory management, to understanding the full effect that marketing campaigns have on consumer demand, to name but a few. All of these problems, as a result of the growth of available data, offer a range of interesting routes that research could develop from.

During our work, we focused on two specific subclasses of problems within product clustering and demand forecasting. In particular, we tackled the issues of clustering products in terms of their cross-elasticity coefficients and forecasting the intermittent demand of SMI products. In both of these problem subclasses, we broadly achieved our objective. With respect to clustering products in terms of their sensitivities in sales, we developed a Bayesian nonparametric methodology that flexibly clusters products in terms of their cross-elasticities coefficients in a way that reflects the underlying structure and assumptions of the data. Similarly with forecasting the intermittent demand of SMI products, we developed a Bayesian hierarchical forecasting methodology that incorporates excitation dynamics and offers significant improvement over other forecasting benchmarks.

The contributions of this work subtly answer deeper questions about each of the problem areas of cross-elasticity coefficient analysis and intermittent demand forecasting. Namely, with interpreting products via their cross-elasticities, we were able address issues around designing a clustering strategy that accommodates the structure exhibited in cross-elasticity data and establishes a fundamental pattern in heterogeneity among these coefficients across different products. In terms of intermittent demand forecasting, we were able to develop a forecasting methodology able to incorporate excitation dynamics together with price and seasonal affects.
8.1 Future opportunities

This work helped to illustrate the possible applications that Bayesian nonparametric mixture modelling, excitation processes and hierarchical modelling has to the field of retail analytics. In particular, we further identify the following areas as interesting routes yet to be fully investigated:

1. **New product prediction:** As mentioned in section 6.1.1, one of the issues that retailers’ face is the problem of forecasting the demand of a product that has limited demand history, or has no demand history as it has not yet been released for general consumption. Consequently, making accurate demand forecasts with such little information can be challenging, and this is particularly felt with products yet to be launched, which makes optimal inventory management hard. Such a scenario could be a fruitful application of data dependent mixture of regressions, in which the sales across various products is described as a mixture model whose mixtures are dependent on data related to various product features (for example a product’s category, its initial selling price, brand information etc). Such a model would be capable at making forecasts of a product’s demand prelaunch by allocating a newly launched product to an appropriate mixture of regression models depending on the product’s relevant features.

2. **Social media data & temporal excitation:** One of the interesting aspects of intermittent demand our work established was the existence of a temporal excitation over and beyond what traditional covariates such as season and prices are able to explain. Such a finding opens up the question of whether this excitation can be described by the use of additional covariates that are typically not used in demand modelling. A possible avenue of investigation would be to better understand whether incorporating information of social media activity, for example the number @mentions a particular product or brand receives on twitter, or the amount of shares a photograph of a product around various social network sites has had, could help to explain movements of a product’s sales that was previously thought of as excitation. Investigating a regression framework that studies the link between publicly available data and demand models to answer questions of whether temporal excitation still exists or whether social media data can be used to predict demand could be worthy of future research.

3. **Store-level hierarchical demand modelling:** A further avenue of research would be to investigate whether there are any temporal or spatial dependencies in demand across different stores and products. In the work developed during Chapter 7, the demand of a
particular touchscreen tablet was aggregated across five different south London stores. An alternative approach could be instead to model the demand of each product at a store level and incorporate a multivariate Hawkes process across the temporal domain that allows demand excitation in a product to occur as a result of the same products being purchased from a neighbouring store. Such an approach may have important implications in terms of a retailers’ inventory management strategy, and one that lends itself to a hierarchical exposition in which information can be shared across the different regional stores as well as the multivariate Hawkes processes themselves.

We further believe our work has possible applications to fields other than retail analytics. Our Bayesian nonparametric clustering methodology of order statistics sequences could be applied to fields where data is inherently ordered and censored, such as is frequently the case in hazard rate modelling and software reliability analysis [??]. Such fields could benefit from our clustering approach by identifying possible heterogeneity that could offer powerful interpretations within their relevant contexts. With respect to the proposed Bayesian hierarchical forecasting methodology, our approach could be applied to modelling scenarios where there exists hierarchical sparse count processes where future counts are dependent on historical counts. An example of such a setting is the case with healthcare resource planning, which is often characterised by an excess of zero counts, longitudinal data and temporal dynamics [?]. Such problem areas could greatly benefit from our forecasting approach by more accurately forecasting the demand for resources and services.
Appendix A

Appendix to Chapter 3

A.1 Appendix

A.1.1 Code for sampled DP measures and density mixtures of figure 3.1

The code below produces the generating process as described during section 3.5.2 along with the figure 3.1.

\[
N = 50; \text{ alpha}_vec = c(1, 5, 15, 40); \text{ base}_mean = 0; \\
\text{ base}_sd = 3; \text{ reponse}_sd = 0.5;
\]

\begin{verbatim}
for (j in 1:length(alpha_vec)){
    ### Samples from DP(alpha, N(base_mean, base_sd))
    theta_atoms = array(0,1)
    for (i in 1:N){
        if (i==1){
            theta_atoms[i] = rnorm(1,base_mean,base_sd)
        } else{
            probability_vector = array(0, i)
            probability_vector[1] = alpha_vec[j]/(i-1+alpha_vec[j])
            probability_vector[2:i] = 1/(i-1+alpha_vec[j])
            temp_allocation_vec = rmultinom(1,1,probability_vector)
            if (temp_allocation_vec[1]){  
                theta_atoms = c(theta_atoms, rnorm(1,base_mean,base_sd))
            } else{
                temp_allocation_vec = temp_allocation_vec[c(-1)]
                theta_atoms = c(theta_atoms, theta_atoms[temp_allocation_vec==1])
            }
        }
    }
}
\end{verbatim}
unique_atoms = unique(theta_atoms)
no_unique_atoms = length(unique(theta_atoms))
weights = array(0, no_unique_atoms)

for (i in 1:no_unique_atoms){
    weights[i] = sum(theta_atoms == unique_atoms[i])/N
}
x = seq(-10, 10, 0.01)
for (i in 1:no_unique_atoms){
    if (i==1){
        truth = weights[i]*dnorm(x, unique_atoms[i], reponse_sd)
    } else {
        truth = truth+weights[i]*dnorm(x, unique_atoms[i], reponse_sd)
    }
}
par(mar = c(5, 5, 4, 5)+0.1)
plot(x, truth, type="l", xlab="", ylim=c(0, max(truth, table(theta_atoms)/N)), main="", col="blue", cex.axis=1.5, cex=1.5, ylab="")
points(unique_atoms, weights)
axis(side = 4, cex.axis = 1.5)
mtext(side = 4, line = 3, expression(pi[i]), cex=1.5)
mtext(side = 1, line = 3, 'x', cex=1.5)
mtext(side = 2, line = 3, 'density', cex=1.5)
Appendix B

Appendix to Chapter 7

B.1 Appendix

B.1.1 Parameter analysis

We now provide an analysis of the parameters generated from model of 7.3.3 and 7.3.1.

Figure B.1: Box plots of count process parameters $\beta_{81}$, $\beta_{82}$ from models BE$^c$, HBE$^c$ and FBE$^c$ with histograms of $\sum_{t:y_{8t} = k} y_{8t}$ with 95% credible intervals of the posterior predictive distributions for models HBE$^c$ and BE$^c$. These demonstrate model BE$^c$ being penalised compared to models FBE$^c$ and HBE$^c$ that allow information pooling. The box plot of $\beta_{81}$, $\beta_{82}$ estimates for model BE$^c$ are different to those of models HBE$^c$ and FBE$^c$, which leads to poor predictions for product $i = 8$ in model BE$^c$. These discrepancies arise from sale counts of product $i = 8$ being $y_{8t} < 3$ over the training set. However, over the test interval a sale count 3 is observed. This shows models HBE$^c$ and FBE$^c$ benefit from having ‘seen’ sale counts $> 2$ from other intermittent demand series that non-hierarchical models cannot.
(a) Box plots of $\mu_c^i$ for $i = 1, \ldots, 17$ for model HBE.

Figure B.2: Box plots of $\mu_c^i$ across all products for model HBE. The $\mu_c^i$ estimates being greater than 2 indicates the temporal excitation exhibited in that data typically occurs at lags greater than 1.
Figure B.3: Box plots of various count process parameters. This figure shows that parameters across the models HBE$^c$ and FBE$^c$ to be almost identical except for parameter $\kappa_{11}$. The root of this discrepancy derives from the significant excitation exhibited in product $i = 2$, as demonstrated from the HBE$^c$ estimate of $\kappa_{2}$ in plot B.3(e). This creates a skew in the shared parameter of $\kappa$ in model FBE$^c$, a skew that does not exist in the $\kappa_{11}$ of HBE$^c$. This lack of heterogeneity in model FBE$^c$ reduces the predictive accuracy on the test set compared to hierarchical equivalent model of HBE$^c$.

B.1.2 Stan code for HBEC$^z$

Below is the hierarchical exposition of the HBEC$^z$ as implemented by STAN.

```stan
data{
  int <lower=0> no_models;
```
int <lower=0> N;
int <lower=0> zero_reg_dim;
int <lower=0> HP_zero_dim;

matrix [N, zero_reg_dim] X_zero[no_models];
int <lower=0> events_times[no_models, N];
int <lower=0> no_events_lesst[no_models, N];
int <lower=0> y[no_models, N];

int <lower=0> N_test;

matrix [N_test, zero_reg_dim] X_zero_test[no_models];
int <lower=0> events_times_test[no_models, N_test];
int <lower=0> no_events_lesst_test[no_models, N_test];
int <lower=0> y_test[no_models, N_test];

real <lower=0> excitation_hyperparameters[HP_zero_dim, 4];
real regression_hyperparameters[zero_reg_dim, 4];

// CROSS data
real <lower=0> excitation_hyperparameters_CROSS[HP_zero_dim, 4];

int <lower=0> events_times_CROSS[no_models, N];
int <lower=0> no_events_lesst_CROSS[no_models, N];

int <lower=0> events_times_test_CROSS[no_models, N_test];
int <lower=0> no_events_lesst_test_CROSS[no_models, N_test];

parameters {
  vector [zero_reg_dim] beta_zero[no_models];
  real <lower=0> kappa_zero[no_models];
  real <lower=1> mu_zero[no_models];
  real <lower=0> tau_zero[no_models];
  // CROSS
real <lower=0> kappa_zero_CROSS[no_models];
real <lower=1> mu_zero_CROSS[no_models];
real <lower=0> tau_zero_CROSS[no_models];
real beta_normal_mu_priors[zero_reg_dim];
real <lower=0> excitation_alphapriors[HP_zero_dim];

// CROSS
real <lower=0> excitation_alphapriors_CROSS[HP_zero_dim];
}
model{
real ps [no_models , N];
real HP_zero[no_models, N];
real HP_zero_CROSS[no_models, N];

for (k in 1:no_models) {
beta_normal_mu_priors[1] ~ normal(regression_hyperparameters[k,1], regression_hyperparameters[k,2]);
}
excitation_alphapriors[1] ~ gamma(excitation_hyperparameters[1,1], excitation_hyperparameters[1,2]);
excitation_alphapriors[2] ~ gamma(excitation_hyperparameters[2,1], excitation_hyperparameters[2,2]);
excitation_alphapriors[3] ~ gamma(excitation_hyperparameters[3,1], excitation_hyperparameters[3,2]);

// CROSS
excitation_alphapriors_CROSS[1] ~ gamma(excitation_hyperparameters_CROSS[1,1], excitation_hyperparameters_CROSS[1,2]);
excitation_alphapriors_CROSS[2] ~ gamma(excitation_hyperparameters_CROSS[2,1], excitation_hyperparameters_CROSS[2,2]);
excitation_alphapriors_CROSS[3] ~ gamma(excitation_hyperparameters_CROSS[3,1], excitation_hyperparameters_CROSS[3,2]);

for (m in 1:no_models) {
for (k in 1:20) {
}
beta_zero[m,k] ~ normal(beta_normal_mu_priors[k], 0.05);
}
kappa_zero[m] ~ gamma(excitation_alphapriors[1], 1);
(mu_zero[m]−1) ~ gamma(excitation_alphapriors[2], 2);
tau_zero[m] ~ gamma(excitation_alphapriors[3], 2.5);
kappa_zero_CROSS[m] ~ gamma(excitation_alphapriors_CROSS[1], 8);
(mu_zero_CROSS[m]−1) ~ gamma(excitation_alphapriors_CROSS[2], 2);
tau_zero_CROSS[m] ~ gamma(excitation_alphapriors_CROSS[3], 2.5);

for (i in 1:N) {
HP_zero[m, i] <- 0;
HP_zero_CROSS[m, i] <- 0;

if (no_events_less[m, i]>0) {
for (j in 1:no_events_less[m, i]) {
HP_zero[m, i] <- HP_zero[m, i]
+kappa_zero[m]*exp(neg_binomial_2_lpmf(i−events_times[m, j]−1 |
(mu_zero[m]−1), tau_zero[m]));}
}
if (no_events_less_CROSS[m, i]>0) {
for (j in 1:no_events_less_CROSS[m, i]) {
HP_zero_CROSS[m, i] <- HP_zero_CROSS[m, i]
+kappa_zero_CROSS[m]*exp(neg_binomial_2_lpmf(i−events_times_CROSS[m, j]−1 |
(mu_zero_CROSS[m]−1), tau_zero_CROSS[m]));}
}
ps[m, i] <- 1/(1+exp(−X_zero[m, i]∗beta_zero[m]
−HP_zero[m, i]−HP_zero_CROSS[m, i]));
y[m, i] ~ bernoulli(ps[m, i]);
}}

B.1.3 Prior formulation of zero processes

Table B.1 specifies the prior structure of models Base$^z$, HB$^z$, BE$^z$, HBE$^z$, FBE$^z$, BEC$^z$, HBEC$^z$ and FBEC$^z$. 
<table>
<thead>
<tr>
<th>Parameter</th>
<th>$\varphi_1 \sim$</th>
<th>$\theta_{11} \sim$</th>
<th>$\theta_{12} \sim$</th>
<th>$\theta_{20} \sim$</th>
<th>$\theta_{11} = \theta_1 \sim$</th>
<th>$\theta_{12} = \theta_2 \sim$</th>
<th>$\theta_{20} = \theta_20 \sim$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varphi_1 \sim$</td>
<td>$N(-3, 3)$</td>
<td>$N(\mu_1^1, 0.05)$</td>
<td>$N(\mu_1^2, 0.05)$</td>
<td>$N(\mu_2^0, 0.05)$</td>
<td>$N(-3, 0.75)$</td>
<td>$N(0, 0.75)$</td>
<td>$N(0, 0.75)$</td>
</tr>
<tr>
<td>$\theta_{11} \sim$</td>
<td>$N(-3, 0.75)$</td>
<td>$N(\mu_1^1, 0.05)$</td>
<td>$N(\mu_1^2, 0.05)$</td>
<td>$N(\mu_2^0, 0.05)$</td>
<td>$N(-3, 0.75)$</td>
<td>$N(0, 0.75)$</td>
<td>$N(0, 0.75)$</td>
</tr>
<tr>
<td>$\theta_{12} \sim$</td>
<td>$N(\mu_1^2, 0.05)$</td>
<td>$N(0, 0.75)$</td>
<td>$N(\mu_1^2, 0.05)$</td>
<td>$N(\mu_2^0, 0.05)$</td>
<td>$N(-3, 0.75)$</td>
<td>$N(0, 0.75)$</td>
<td>$N(0, 0.75)$</td>
</tr>
<tr>
<td>$\theta_{20} \sim$</td>
<td>$N(\mu_2^0, 0.05)$</td>
<td>$N(0, 0.75)$</td>
<td>$N(\mu_2^0, 0.05)$</td>
<td>$N(0, 0.75)$</td>
<td>$N(-3, 0.75)$</td>
<td>$N(0, 0.75)$</td>
<td>$N(0, 0.75)$</td>
</tr>
<tr>
<td>$\theta_{11} = \theta_1 \sim$</td>
<td>$N(-3, 0.75)$</td>
<td>$N(0, 0.75)$</td>
<td>$N(0, 0.75)$</td>
<td>$N(0, 0.75)$</td>
<td>$N(-3, 0.75)$</td>
<td>$N(0, 0.75)$</td>
<td>$N(0, 0.75)$</td>
</tr>
<tr>
<td>$\theta_{12} = \theta_2 \sim$</td>
<td>$N(0, 0.75)$</td>
<td>$N(0, 0.75)$</td>
<td>$N(0, 0.75)$</td>
<td>$N(0, 0.75)$</td>
<td>$N(0, 0.75)$</td>
<td>$N(0, 0.75)$</td>
<td>$N(0, 0.75)$</td>
</tr>
<tr>
<td>$\theta_{20} = \theta_20 \sim$</td>
<td>$N(0, 0.75)$</td>
<td>$N(0, 0.75)$</td>
<td>$N(0, 0.75)$</td>
<td>$N(0, 0.75)$</td>
<td>$N(0, 0.75)$</td>
<td>$N(0, 0.75)$</td>
<td>$N(0, 0.75)$</td>
</tr>
</tbody>
</table>

Table B.1: Prior formulation of models Base, HB, BE, HBE, FBE, BEC, HBEC and FBEC.
### B.1.4 Stan code for HBE$	extsuperscript{c}$

Below is the hierarchical exposition of the HBE$	extsuperscript{c}$ as implemented by STAN.

```stan
functions {
    real neg_binomial_den(int t, real mu, real size) {
        real binomial_term;
        binomial_term <- exp(binomial_coefficient_log(t+size-1, t));
        return binomial_term*pow(mu/(mu+size), t)*pow(size/(mu+size), size);
    }

    real poisson_den(int t, real lambda_par) {
        return pow(lambda_par, t)*exp(-lambda_par)/tgamma(t+1);
    }
}

data{
    int <lower=0> no_models;
    int <lower=0> N;
    //int <lower=0> no_events [no_models];
    int <lower=0> count_reg_dim;
    int <lower=0> HP_count_dim;

    matrix [N, count_reg_dim] X_count [no_models];
    int <lower=0> events_times [no_models, N]; // event occurrence index
    int <lower=0> no_events_lesst [no_models, N];
    int <lower=0> y_count [no_models, N];

    int <lower=0> N_test;

    matrix [N_test, count_reg_dim] X_count_test [no_models];
    int <lower=0> events_times_test [no_models, N_test];
    int <lower=0> no_events_lesst_test [no_models, N_test];
    int <lower=0> y_count_test [no_models, N_test];

    real <lower=0> excitation_hyperparameters [HP_count_dim, 4];
}
```
real regression_hyperparameters[count_reg_dim,4];

parameters {
  vector[count_reg_dim] beta_count[no_models];
  real <lower=0, upper=10> kappa_count[no_models];
  real <lower=0> mu_count[no_models];
  real <lower=0> tau_count[no_models];
  real beta_mupriors;
  real <lower=0> beta_sigmapriors;
  real beta_price_alpha_priors;
  real <lower=0> beta_price_beta_priors;
  real <lower=0> excitation_alphapriors[HP_count_dim];
  real <lower=0> excitation_betapriors[HP_count_dim];
}

model{
  real HP_count[no_models, N];
  real lambda[no_models, N];

  beta_mupriors ~ normal(regression_hyperparameters[1,1],
                          regression_hyperparameters[1,2]);
  beta_sigmapriors ~ gamma(regression_hyperparameters[1,3],
                           regression_hyperparameters[1,4]);
  beta_price_alpha_priors ~ normal(regression_hyperparameters[2,1],
                                   regression_hyperparameters[2,2]);
  beta_price_beta_priors ~ gamma(regression_hyperparameters[2,3],
                                  regression_hyperparameters[2,4]);
  excitation_alphapriors[1] ~ gamma(excitation_hyperparameters[1,1],
                                     excitation_hyperparameters[1,2]);
excitation_betapriors[1] ~ gamma(excitation_hyperparameters[1,3]
, excitation_hyperparameters[1,4]);

excitation_alphapriors[2] ~ gamma(excitation_hyperparameters[2,1]
, excitation_hyperparameters[2,2]);
excitation_betapriors[2] ~ gamma(excitation_hyperparameters[2,3]
, excitation_hyperparameters[2,4]);

excitation_alphapriors[3] ~ gamma(excitation_hyperparameters[3,1]
, excitation_hyperparameters[3,2]);
excitation_betapriors[3] ~ gamma(excitation_hyperparameters[3,3]
, excitation_hyperparameters[3,4]);

for (m in 1:no_models) {
  beta_count[m, 1] ~ normal(beta_mupriors
, beta_sigmapriors);

  beta_count[m, 2] ~ normal(beta_price_alpha_priors
, beta_price_beta_priors);

  kappa_count[m] ~ gamma(excitation_alphapriors[1]
, excitation_betapriors[1]);
  (mu_count[m]−1) ~ gamma(excitation_alphapriors[2]
, excitation_betapriors[2]);
  tau_count[m] ~ gamma(excitation_alphapriors[3]
, excitation_betapriors[3]);

for (i in 1:N) {
  HP_count[m, i] <- 0;

  if (y_count[m, i] > 0){
    if (no_events_lesst[m, i] > 0) {
      for (j in 1:no_events_lesst[m, i]) {
        HP_count[m, i] <- HP_count[m, i]
+kappa_count[m] * neg_binomial_den(i−1−events_times[m, j])
      }
    }
  }
}
\begin{verbatim}
B.1. Appendix

\[
, (\text{mu_count}[m] - 1), \text{tau_count}[m]);
\]

\}

\lambda[m, i] \leftarrow \exp(X_{\text{count}}[m, i] \ast \text{beta_count}[m] + \text{HP_count}[m, i]);
\]
\text{target} += \text{neg\_binomial\_2\_log}(y_{\text{count}}[m, i], \lambda[m, i], 1)
\]
\] - \text{neg\_binomial\_2\_ccdf\_log}(0, \lambda[m, i], 1);
\]
\}
\}
\}
\}
\}
\end{verbatim}
Table B.2: Prior formulation of models Base\(^c\), HB\(^c\), BE\(^c\), HBE\(^c\) and FBE\(^c\).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Base(^c)</th>
<th>HB(^c)</th>
<th>BE(^c)</th>
<th>HBE(^c)</th>
<th>FBE(^c)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\varphi_i)</td>
<td>(\sim) (N(-4, 4))</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\beta_1)</td>
<td>(\sim) (N(\mu_1, 0.05))</td>
<td>(N(1, 0.75))</td>
<td>(N(\mu_2, 0.05))</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\beta_2)</td>
<td>(\sim) (N(-1, 0.75))</td>
<td>(N(1, 0.75))</td>
<td>(N(-1, 0.75))</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\beta_1 = \beta_2)</td>
<td>(\sim) (N(0, 0.75))</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\gamma_1)</td>
<td>(\sim) (G(1, 5))</td>
<td>(G(\eta_1, 5))</td>
<td>(G(\eta_1, 5))</td>
<td>(G(\eta_1, 5))</td>
<td></td>
</tr>
<tr>
<td>(\gamma_2)</td>
<td>(\sim) (G(\eta_2, 5))</td>
<td>(1+G(3, 1))</td>
<td>(1+G(\eta_2, 1))</td>
<td>(G(\eta_2, 1))</td>
<td></td>
</tr>
<tr>
<td>(\gamma_3)</td>
<td>(\sim) (G(4, 1))</td>
<td>(G(4, 1))</td>
<td>(G(4, 1))</td>
<td>(G(4, 1))</td>
<td></td>
</tr>
<tr>
<td>(\gamma_1 = \gamma_2)</td>
<td>(\sim) (G(1, 5))</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\gamma_1 = \gamma_3)</td>
<td>(\sim) (G(1, 5))</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\gamma_1 = \gamma_2 = \gamma_3)</td>
<td>(\sim) (G(1, 5))</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

B.1.5 Prior formulation of count processes

Table B.1.5 specifies the prior structure of models Base\(^c\), HB\(^c\), BE\(^c\), HBE\(^c\) and FBE\(^c\).
B.1.6 Zero process MCMC output

Parameter inference of zero process models are performed by HMC algorithm. For models HB\(^z\) and HBE\(^z\) we take 3000 samples with 1000 burn-in respectively and 2000 samples with 1000 burn-in for models FBEC\(^z\), BE\(^z\) and FBE\(^z\). For model HBEC\(^z\) we take 2500 samples and 1500 burnin and we take 4000 samples and 6000 burn-in for BEC\(^z\). For model Base\(^z\) we take 2000 samples and 1000 burnin. Figure B.4 provides some typical trace plots of fitted models.

Figure B.4: Trace plots of selected zero process parameters.
B.1.7 Count process MCMC output

Parameter inference is performed by the HMC sampling algorithm. For models BE\textsuperscript{c} and FBE\textsuperscript{c} we take 2000 samples with 1000 burn-in respectively. For model HB\textsuperscript{c} we take 3000 samples and 2000 burnin and for model HBE\textsuperscript{c} we take 5000 samples and 2000 burn-in. For model Base\textsuperscript{c} we take 1000 samples and 1000 burnin. Figure B.5 provides typical trace plots of fitted models.

![Trace plots of selected count process parameters.](image)

Figure B.5: Trace plots of selected count process parameters.