Statistical modelling approaches with Bayesian tensor factorisations

Zhongzhen Wang

Principal Supervisor: Professor Petros Dellaportas
Subsidiary Supervisor: Dr Ioannis Kosmidis

A thesis presented for the degree of
Doctor of Philosophy

Department of Statistical Science
University College London

01 August 2021
Declaration

I, Zhongzhen Wang, 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

We propose a flexible nonparametric Bayesian modelling of univariate and multivariate time series of count data based on conditional tensor factorisations. Our models can be viewed as infinite state space Markov chains of known maximal order with non-linear serial dependence or, with an introduction of appropriate latent variables, as a Bayesian hierarchical model with conditionally independent Poisson distributed observations. Inference about the important lags and their complex interactions is achieved via Markov chain Monte Carlo. When the observed counts are large, we deal with the resulting computational complexity of the model by performing an initial analysis in a training set of the data that is not used further in the inference and prediction. Our methodology is illustrated using simulation experiments and real-world data. Our Bayesian tensor factorisations model can have a good performance in inference and prediction on time series of count data that tends to be non-linear, and in the meanwhile, can deal with Markov chains of linear or log-linear count data. Moreover, our Bayesian tensor factorisations model can capture higher-order interactions among the lags and then, maximal orders, in time series where the actual order of Markov chain of count data and serial dependence are unknown.
Impact Statement

Studies of disease occurrence lead to accurate predictions of counts and, hopefully, prevent certain diseases from spreading. With more concerns on public health, including the spreading of normal influenzas, H1N1 virus, Ebola virus and outbreak of the most recent Coronavirus, the number counts or cases of patients infected in a certain length of time is well recorded. Therefore, we expect better models that can fit those time series of count data. Besides epidemiological studies, there is a surge in the applications of times series of count data in many other areas, including economics, finance, environmental science, social science and etcetera.

Traditional generalised linear models, such as Poisson autoregressive and integer-valued generalised autoregressive conditionally heteroscedastic models, have great performance in fitting and predicting linear or log-linear time series of counts. However, in many cases, they fail to deal with time series of counts that trends to be non-linear. Even though these generalised linear models can consider covariate effects, the prediction on multivariate time series of count data can hardly be their strength. This thesis provides a novel framework for analysing time series of count data by using techniques including Bayesian statistics and tensor factorisations. This methodology captures higher-order interactions among the past lags and then, maximal orders of time series, which facilitates the computation and helps solve the two problems mentioned above.

The insights of our approach provide the research community with suitable alternatives for modelling univariate or multivariate time series of count data, especially the ones to be non-linear. Our methodology sheds light on dependent time lags and predictors, which makes the model parsimonious and indicates interactions among predictors and the response. Our approach selects significant predictors or lags that decreases the uncertainty of the model.
List of Symbols

\( \theta \) parameter

\( \Theta \) parameter space

\( E \) expectation

\( \text{Var} \) variance

\( \text{Cov} \) covariance

\( \cup \) union

\( \cap \) intersection

\( \mathbb{N} \) natural number

\( \mathbb{R} \) real number

\( \mathbb{Z} \) integer number

\( \mathbb{Z}_{[i,j]} \) a set containing all integers from \( i \) to \( j \)
List of Acronyms

GLM    generalised linear model
BaGLM  Bayesian generalised linear model
GAM    generalised additive model
INGARCH integer-valued generalised autoregressive conditionally heteroskedastic
INAR   integer-valued autoregression
INMA   integer-valued moving average
BTF    Bayesian tensor factorisation
MCMC   Markov chain Monte Carlo
DP     Dirichlet process
CRP    Chinese restaurant process
i.i.d. independently and identically distributed
OLS    ordinary least square
SRS    sum of residual squares
MLE    maximum likelihood estimator
PAR    Poisson autoregression
BaPAR  Bayesian Poisson autoregression
PARAFAC parallel factor analysis
HOSVD  higher order singular value decomposition
AIC    Akaike information criterion
BIC    Bayesian information criterion
RFD    relative frequency distribution
ALPS   average log predictive score
MSE    mean squared error
SRoMSE square root of mean squared error
List of Figures

4.1 Results of simulation experiments for the case with $C_0 = 4$ categories, true important lags $\{4, 7\}$, and sample size to be $2,000$ for the dataset with the classification error rate. (a) Relative frequency distribution of how many important lags in the model; (b) Inclusion proportions of different lags; The rest six panels are the predictive conditional probabilities for different cases labelled in titles. In each panel, the cyan line shows the trace plot of conditional probability. The blue solid line shows the running mean. The black dash lines show the 5% and 95% running quantiles, respectively. $y$-axis represents the conditional probability and $x$-axis represents the MCMC Samplers after Burn-in. ...................................... 69

5.1 Results of simulation experiments for the case with $\nu_+ = 5$, $\nu_- = 10$, true important lags $\{1, 3\}$. In each panel, $\lambda(i_1, i_3)$ represents the conditional mean of $y_t$ given $y_{t-1} = i_1, y_{t-3} = i_3$, and the true value is shown in the lower left corner. In each panel, the conditional mean of $y_t|y_{t-1}, y_{t-3}$ is drawn in cyan line. The blue solid line shows the running mean, and the black dashed lines are the 5% and 95% quantiles for each case. $y$-axis represents the mean of Poisson parameter and $x$-axis does MCMC samplers after burn-in. Thinning rate is set to be 10. .................................................. 88

5.2 Trace plot of dataset Death from Drug Dependence (Toxicomania) obtained from the Cause of Death Register maintained by Statistics Norway from January 1969 to June 2009 monthly with 486 points. ............................. 89
5.3 Result of important lag selection for the Toxicomania dataset. Left panel: the inclusion proportions of different lags for the scenario with 388 training data points and 98 testing data points. Right panel: the inclusion proportions of different lags for the scenario with 437 training data points and 49 testing data points. The orders of past lags are in the horizontal and their corresponding relative frequency distributions to be important are in the vertical.

5.4 Trace Plots of the counts of stocks going up consisting of NASDAQ100 Index with different range of years. (a) 3 years (01-May-2016 to 30-Apr-2019), with 100 stocks and 753 data points in total; (b) 7 years (01-May-2012 to 30-Apr-2019), with 93 stocks and 1,759 data points in total; (c) 10 years (01-May-2009 to 30-Apr-2019), with 89 stocks and 2,515 data points in total.

5.5 Result of important lag selection for the NASDAQ100 datasets. (a): the inclusion proportions of different lags for the scenario with 602 training data points and 151 data points (3 years); (b): the inclusion proportions of different lags for the scenario with 677 training data points and 76 data points (3 years); (c): the inclusion proportions of different lags for the scenario with 1407 training data points and 352 data points (7 years); (d): the inclusion proportions of different lags for the scenario with 1583 training data points and 176 data points (7 years); (e): the inclusion proportions of different lags for the scenario with 2012 training data points and 503 data points (10 years); (f): the inclusion proportions of different lags for the scenario with 2263 training data points and 252 data points (10 years). The orders of past lags are in the horizontal and their corresponding relative frequency distributions to be important are in the vertical.

6.1 Trace plot of 514 time-series data points counting flu cases in Norway and Castilla-La Mancha (CLM), Spain, counted by each week from 09-Oct-2005 to 09-Aug-2015.
6.2 Result of important lag selection for the Norway flu dataset. Left panel: the inclusion proportions of different lags for the scenario with 411 training data points and 103 testing data points. Right panel: the inclusion proportions of different lags for the scenario with 462 training data points and 52 testing data points. The orders of past lags are in the horizontal and their corresponding relative frequency distributions to be important are in the vertical. .......................................................... 105

6.3 Result of important lag selection for the Castilla-La Mancha flu dataset. Left panel: the inclusion proportions of different lags for the scenario with 411 training data points and 103 testing data points. Right panel: the inclusion proportions of different lags for the scenario with 462 training data points and 52 testing data points. The orders of past lags are in the horizontal and their corresponding relative frequency distributions to be important are in the vertical. .......................................................... 106

7.1 Five regions in South-eastern Spain including Andalusia, Castilla-La Mancha, Illes Balears, Region de Murcia and Comunitat Valenciana according to Study Abroad Map (studyabroadmap.com) ............................................. 114

7.2 Trace plot of 514 time-series data points counting flu cases in Andalusia, Castilla-La Mancha, Illes Balears, Region de Murcia and Comunitat Valenciana counted by each week from 09-Oct-2005 to 09-Aug-2015. .... 115

7.3 Result of important lag selection for the south-eastern Spain flu dataset: the inclusion proportions of different lags for the scenario with 411 training data points and 103 testing data points. Vertical region: target region; Horizontal region: corresponding covariant region. For each panel, the orders of past lags are in the horizontal and their corresponding relative frequency distributions to be important are in the vertical. A: Andalusia; CLM: Castilla-La Mancha; IB: Illes Balears; RM: Region de Murcia; VC: Comunitat Valenciana. .................................................. 117
7.4 Result of important lag selection for the south-eastern Spain flu dataset: the inclusion proportions of different lags for the scenario with 462 training data points and 52 testing data points. Vertical region: target region; Horizontal region: corresponding covariant region. For each panel, the orders of past lags are in the horizontal and their corresponding relative frequency distributions to be important are in the vertical. A: Andalusia; CLM: Castilla-La Mancha; IB: Illes Balears; RM: Region de Murcia; VC: Comunitat Valenciana.

8.1 USA: Information of the clusterings. Upper panel: total number of clusters and their corresponding Poisson rates; Lower panel: weight for each cluster in logarithm. Both panels share the same x-axis.

8.2 USA: (a) The bar chart of relative frequency distribution for variables indicated as important. All the variables for selection is labelled on x-axis, and y-axis represents their corresponding relative frequency. The threshold for a variable to be important is 0.5, which is depicted in red. (b) The RFD of $\prod_{j=1}^{q} k_j$, or the total number of possible combinations of $(h_1, \ldots, h_q)$. (c) The RFD of the number of clusters of the probability kernel $\lambda_{h_1 \ldots h_q}$. y-axis in both (b) and (c) represent the number, and x-axis in both ones are the relative frequency.

8.3 UK: Information of the clusterings. Upper panel: total number of clusters and their corresponding Poisson rates; Lower panel: weight for each cluster in logarithm. Both panels share the same x-axis.

8.4 UK: (a) The bar chart of relative frequency distribution for variables indicated as important. All the variables for selection is labelled on x-axis, and y-axis represents their corresponding relative frequency. The threshold for a variable to be important is 0.5, which is depicted in red. (b) The RFD of $\prod_{j=1}^{q} k_j$, or the total number of possible combinations of $(h_1, \ldots, h_q)$. (c) The RFD of the number of clusters of the probability kernel $\lambda_{h_1 \ldots h_q}$. y-axis in both (b) and (c) represent the number, and x-axis in both ones are the relative frequency.
8.5 Brazil: Information of the clusterings. Upper panel: total number of clusters and their corresponding Poisson rates; Lower panel: weight for each cluster in logarithm. Both panels share the same x-axis. ........................................ 129

8.6 Brazil: (a) The bar chart of relative frequency distribution for variables indicated as important. All the variables for selection is labelled on x-axis, and y-axis represents their corresponding relative frequency. The threshold for a variable to be important is 0.5, which is depicted in red. (b) The RFD of $\prod_{j=1}^{q} k_j$, or the total number of possible combinations of $(h_1, \ldots, h_q)$. (c) The RFD of the number of clusters of the probability kernel $\lambda_{h_1 \ldots h_q}$. y-axis in both (b) and (c) represent the number, and x-axis in both ones are the relative frequency. ........................................ 129

8.7 France: Information of the clusterings. Upper panel: total number of clusters and their corresponding Poisson rates; Lower panel: weight for each cluster in logarithm. Both panels share the same x-axis. ................................. 130

8.8 France: (a) The bar chart of relative frequency distribution for variables indicated as important. All the variables for selection is labelled on x-axis, and y-axis represents their corresponding relative frequency. The threshold for a variable to be important is 0.5, which is depicted in red. (b) The RFD of $\prod_{j=1}^{q} k_j$, or the total number of possible combinations of $(h_1, \ldots, h_q)$. (c) The RFD of the number of clusters of the probability kernel $\lambda_{h_1 \ldots h_q}$. y-axis in both (b) and (c) represent the number, and x-axis in both ones are the relative frequency. ................................. 130
List of Tables

4.1 Mean classification error rate among variable length Markov chain model (VLMC), random forest model and Bayesian tensor factorisations model (BTF) based on 10 generated data sets for each one of seven scenarios. Length of training data against testing data: 1,600 : 400. Models with best performance are highlighted bold. .............................................. 68

5.1 Mean of average log predictive score (ALPS) (with standard deviations in brackets) between Bayesian Poisson autoregressive model and our Bayesian tensor factorisations model (BTF) based on 10 Poisson autoregression generated datasets for each one of 6 Scenarios. AIC and BIC columns indicate that the best model has been chosen with the corresponding criterion. Models with the best performance are highlighted bold. .............................................. 85

5.2 Mean of mean squared error (MSE) (with standard deviations in brackets) among Poisson autoregression, integer-valued generalised autoregressive conditionally heteroscedastic (INGARCH) model and our Bayesian tensor factorisations model (BTF) based on 10 Poisson autoregression generated data sets for each one of 6 Scenarios. AIC and BIC columns indicate that the best model has been chosen with the corresponding criterion. Models with the best performance are highlighted bold. .............................................. 85

5.3 Mean of average log predictive score (ALPS) (with standard deviations in brackets) between Bayesian Poisson autoregressive model and our Bayesian tensor factorisations model (BTF) based on 10 Poisson autoregression generated datasets for each one of 6 Scenarios. AIC and BIC columns indicate that the best model has been chosen with the corresponding criterion. Models with the best performance are highlighted bold. .............................................. 87
5.4 Mean of mean squared error (MSE) (with standard deviations in brackets) among Poisson autoregression, integer-valued generalised autoregressive conditionally heteroscedastic (INGARCH) model and our Bayesian tensor factorisations model (BTF) based on 10 Poisson autoregression generated data sets for each one of 6 Scenarios. AIC and BIC columns indicate that the best model has been chosen with the corresponding criterion. Models with the best performance are highlighted bold. .......................... 87

5.5 Average log predictive score (ALPS) (with models selected in brackets) between Bayesian Poisson autoregressive model and our simple Bayesian tensor factorisations (BTF) model. AIC and BIC columns indicate that the best model has been chosen with the corresponding criterion. Training and testing data sizes appear in the first column. Models with best performance are highlighted bold. ........................................... 90

5.6 Mean squared error (MSE) (with models selected in brackets) among Poisson autoregression, integer-valued generalised autoregressive conditionally heteroscedastic (INGARCH) model and our Bayesian tensor factorisations (BTF) model. AIC and BIC columns indicate that the best model has been chosen with the corresponding criterion. Training and testing data sizes appear in the first column. Models with best performance are highlighted bold. ........................................... 90

5.7 Average log predictive score (ALPS) (with models selected in brackets) between Bayesian Poisson autoregressive model and our simple Bayesian tensor factorisations model (BTF). AIC and BIC columns indicate that the best model has been chosen with the corresponding criterion. Training and testing data sizes appear in the first column. Models with best performance are highlighted bold. ........................................... 91

5.8 Mean squared error (MSE) (with models selected in brackets) among Poisson autoregression, integer-valued generalised autoregressive conditionally heteroscedastic (INGARCH) model and our Bayesian tensor factorisations model (BTF). AIC and BIC columns indicate that the best model has been chosen with the corresponding criterion. Training and testing data sizes appear in the first column. Models with best performance are highlighted bold. ........................................... 92
6.1 Mean of average log predictive score (ALPS) (with standard deviations in brackets) between Bayesian Poisson autoregressive model and our Bayesian tensor factorisations model (BTF) based on 10 Poisson autoregression generated datasets for each one of 6 Scenarios. AIC and BIC columns indicate that the best model has been chosen with the corresponding criterion. Models with the best performance are highlighted bold. ................................ 103

6.2 Mean of mean squared error (MSE) (with standard deviations in brackets) among Poisson autoregression, integer-valued generalised autoregressive conditionally heteroscedastic (INGARCH) model and our Bayesian tensor factorisations model (BTF) based on 10 Poisson autoregression generated data sets for each one of 6 Scenarios. AIC and BIC columns indicate that the best model has been chosen with the corresponding criterion. Models with the best performance are highlighted bold. ................................ 104

6.3 Average log predictive score (ALPS) between Bayesian Poisson autoregression and our Bayesian tensor factorisations model (BTF) for flu counts datasets in Norway and Castilla-La Mancha, Spain. AIC and BIC columns indicate that the best model has been chosen (in brackets) with the corresponding criterion. Training and testing data sizes appear in the second column. Models with best performance are highlighted bold. .......................... 105

6.4 Square root of mean squared error (SRoMSE) (with models selected in brackets) among Poisson autoregression, integer-valued generalised autoregressive conditionally heteroscedastic (INGARCH) model and our Bayesian tensor factorisations model (BTF). AIC and BIC columns indicate that the best model has been chosen with the corresponding criterion. Training and testing data sizes appear in the first column. Models with best performance are highlighted bold. .............................................. 106

7.1 Mean average log predictive score (with standard deviations in brackets) between Bayesian Poisson autoregressive model and our Bayesian tensor factorisations model (BTF) based on 10 non-linear generated data sets for each one of 6 Scenarios. AIC and BIC columns indicate that the best model has been chosen with the corresponding criterion. Models with the best performance are highlighted bold. .............................................. 113
### 7.2 Mean of mean squared error (MSE) (with standard deviations in brackets) among Poisson autoregression, integer-valued generalised autoregressive conditionally heteroscedastic (INGARCH) model and our Bayesian tensor factorisations model (BTF) based on 10 Poisson autoregression generated data sets for each one of 6 Scenarios. AIC and BIC columns indicate that the best model has been chosen with the corresponding criterion. Models with the best performance are highlighted bold.

### 7.3 Distance between regions in kilometre. A = Andalusia, CLM = Castilla-La Mancha, IB = Illes Balears, RM = Region de Murcia and VC = Comunitat Valenciana. Information found on Open Street Map (openstreetmap.org).

### 7.4 Average log predictive score (ALPS) between Bayesian Poisson autoregressive model and Bayesian tensor factorisations model (BTF) for multivariate flu counts datasets (with models selected in brackets). Multiple datasets include flu counts in Andalusia, Castilla-La Mancha, Illes Balears, Region de Murcia and Comunitat Valenciana. AIC and BIC columns indicate that the best model has been chosen with the corresponding criterion. Models with best performance are highlighted bold.

### 7.5 Square root of mean squared error (SRoMSE) (with models selected in brackets) among Poisson autoregression, integer-valued generalised autoregressive conditionally heteroscedastic (INGARCH) model and our Bayesian tensor factorisations model (BTF) for multivariate flu counts datasets. Multiple datasets include flu counts in Andalusia, Castilla-La Mancha, Illes Balears, Region de Murcia and Comunitat Valenciana. AIC and BIC columns indicate that the best model has been chosen with the corresponding criterion. Models with best performance are highlighted bold.

### 7.6 Important predictors selected for each time series by our BTF model for the scenario with 411 training data points and 103 testing data points. A = Andalusia, CLM = Castilla-La Mancha, IB = Illes Balears, RM = Region de Murcia and VC = Comunitat Valenciana.

### 7.7 Important predictors selected for each time series by our BTF model for the scenario with 462 training data points and 52 testing data points. A = Andalusia, CLM = Castilla-La Mancha, IB = Illes Balears, RM = Region de Murcia and VC = Comunitat Valenciana.
8.1 Comparison of models including the Bayesian generalised linear model (BaGLM), generalised additive model (GAM) and our general Bayesian tensor factorisations (BTF) model for COVID-19 datasets. Left table: comparison between BaGLM and BTF in terms of average log predictive score (ALPS). Right table: comparison between GAM and BTF in terms of square root of mean squared error (SRoMSE). Models with best performance are highlighted bold. Countries for comparison include USA, UK, Brazil and France. Response: the daily death count; Predictors: the daily confirmed cases in the past 10 days. Training dataset: 1st March to 15th July 2020; Testing dataset: 16th July to 25th July 2020.
Contents

1 Introduction 23

2 Bayesian Inference 27
   1 Bayesian Inference .................................................. 27
   2 Markov Chain Monte Carlo ........................................... 28
   3 MCMC Convergence Diagnostics ................................. 33
   4 Bayesian Nonparametric Models .................................. 34
   5 Dirichlet Process ...................................................... 35

3 Models for Time series of Count Data 45
   1 Integer-Valued Autoregression ................................. 45
   2 Integer-Valued Moving Average ................................. 47
   3 Poisson Autoregression ............................................. 48
   4 Integer-Valued Generalised Autoregression Conditional Heteroscedastic ................................. 49
   5 Multivariate Time Series of Counts .......................... 51
   6 Software and Package ............................................... 52

4 Bayesian Tensor Factorisations for Categorical Data 55
   1 Tensor Factorisations ................................................ 55
   2 Tensor Factorisations on Mixture Transition Distribution ............ 57
   3 Model Specification for Categorical Data .......................... 58
   4 Estimation and Inference ............................................. 61
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 First Stage – Estimation of $K$</td>
<td>62</td>
</tr>
<tr>
<td>4.2 Second Stage – Posteriors of Random Variables</td>
<td>64</td>
</tr>
<tr>
<td>5 Comparisons</td>
<td>65</td>
</tr>
<tr>
<td>6 Appendix</td>
<td>70</td>
</tr>
<tr>
<td>5 Bayesian Tensor Factorisations for Univariate Time Series of Counts</td>
<td>73</td>
</tr>
<tr>
<td>1 Model Specification</td>
<td>73</td>
</tr>
<tr>
<td>2 Estimation and Inference</td>
<td>76</td>
</tr>
<tr>
<td>2.1 First Stage – Estimation of $K$</td>
<td>77</td>
</tr>
<tr>
<td>2.2 Second Stage – Posteriors of Random Variables</td>
<td>80</td>
</tr>
<tr>
<td>3 Simulation Experiments</td>
<td>82</td>
</tr>
<tr>
<td>3.1 Log-Linear Datasets</td>
<td>84</td>
</tr>
<tr>
<td>3.2 Non-Linear Datasets</td>
<td>86</td>
</tr>
<tr>
<td>3.3 MCMC Diagnostics</td>
<td>86</td>
</tr>
<tr>
<td>4 Applications</td>
<td>87</td>
</tr>
<tr>
<td>4.1 Application 1</td>
<td>88</td>
</tr>
<tr>
<td>4.2 Application 2</td>
<td>89</td>
</tr>
<tr>
<td>5 Appendix</td>
<td>94</td>
</tr>
<tr>
<td>6 Bayesian Tensor Factorisations for Univariate Time Series of Large Counts</td>
<td>97</td>
</tr>
<tr>
<td>1 Model Specification</td>
<td>98</td>
</tr>
<tr>
<td>2 Estimation and Inference</td>
<td>100</td>
</tr>
<tr>
<td>2.1 First Stage – Estimation of $D$</td>
<td>100</td>
</tr>
<tr>
<td>2.2 Second Stage – Estimation of $K$</td>
<td>101</td>
</tr>
<tr>
<td>2.3 Third Stage – Posteriors of Other Random Variables</td>
<td>101</td>
</tr>
<tr>
<td>3 Simulation Experiments</td>
<td>101</td>
</tr>
<tr>
<td>4 Applications</td>
<td>103</td>
</tr>
<tr>
<td>7 Bayesian Tensor Factorisations for Multivariate Time Series of Count</td>
<td></td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Statistically, count data is a type of data where the observations can only take the natural number $N$ including zero $\{0, 1, 2, \ldots\}$ and these integers arise from counting instead of ranking. A time series is a series of indexed data points in the order of time and normally, taken at successively equally spaced points in time. There have been many approaches for modelling time series of count data. Jacobs and Lewis [1978a,b] introduced discrete autoregressive-moving average models, obtained by a probabilistic mixture of sequence of independent identically distributed (i.i.d.) discrete variables. Al-Osh and Alzaid [1987] firstly introduced the very popular first-order integer-valued autoregressive process, or INAR(1) process, developed from the idea of binomial thinning [Steutel and Van Harn, 1979]. The first-order integer-valued moving average (INMA) models, as well as their $p$-th order extensions, were considered in McKenzie [1988]; Al-Osh and Alzaid [1988]. An INMA process can be taken as the survivals of i.i.d. random variables with integer values, whose structure reflects the mechanism generating real-world data for many counting processes. Consequently, it is useful for modelling such processes. Another popular class is the generalised linear model (GLM) based. Many papers introduce GLM, including Nelder and Wedderburn [1972]; Dobson and Barnett [2008]; Fahrmeir and Tutz [2013]; Fox [2015]; McCullagh [2018]. A Poisson autoregressive model is an application of GLM, which assumes the mean of counts follow conditional Poisson distributions. Cameron and Trivedi [2001] investigated properties of GLM-based time series models. As an extension to a Poisson autoregressive process, an integer-valued generalised autoregressive conditional heteroscedastic (INGARCH) process was firstly proposed by Ferland et al. [2006]. Fokianos and Fried [2010] studied the problem of intervention effects generating various types of outliers in INGARCH processes. Fokianos [2011] provided an account of many properties
of INGRACH models.

Many real-world problems can be investigated by time series due to the explicitness on prediction through operations on probability. Time series with specific orders facilitate the computation of conditional probability by limiting the past conditions within these orders. With the development of bioscience, especially genetics, more hidden information on gene patterns is detected and categorised. Raftery and Tavaré [1994] used Markov chains to fit and analyse DNA sequences by considering four bases of nucleotides as adenine, cytosine, guanine and thymine. Raftery [1985] assessed the amount of electricity that could be generated from wind power, in Belmullet, Ireland, by coding hourly wind speeds into a time series with four states. Berchtold and Raftery [2002] applied the mixture transition distribution approach to model a binary time series describing whether a patient experienced epileptic seizures on 204 consecutive days. Similar to their categorical counterparts, there are also numerous studies on time series of count data. The applications of diverse time series approaches on count data are showing their significance under the most recent outbreak of Coronavirus or COVID-19, and also Severe Acute Respiratory Syndrome in 2002 to 2003, H1N1 virus in the summer of 2009 and Middle East Respiratory Syndrome firstly reported in Saudi Arabia in 2012. Hay and Pettitt [2001] studied the control of an infectious disease by a Bayesian approach on a time series of counts with covariates. Yang et al. [2018] analysed the time series of measles counts through a threshold autoregressive model, developed by Tong [2012], that allows for higher degree of flexibility in model parameters. Besides epidemiological studies, there is a surge in the applications of times series of counts in economics and finance. The studies of the transaction frequency and the return times of extreme events of a certain stock can provide better understanding of the stock price [Davis and Liu, 2016]. Other application areas of time series of counts include environmental science and social science. Studies of annual hurricane counts can suggest whether global warming causes more numerous hurricanes [Livsey et al., 2018]. Hasan et al. [2012] studied air pollution related emergency room visit data through a regression analysis on time series of counts. The analysis of word frequencies of party manifestos can be used to estimate the time series party positions [Slapin and Proksch, 2008].

GLM-based models are popular approaches for modelling both times series of continuous and discrete data, including typical models like INGARCH and Poisson autoregressive models. Analysis for many applications on times series of count data benefits from INGARCH and Poisson autoregressive models with identity or logarithmic link functions due to the models’ straightforward representations. While in the cases where time series data trends
to be non-linear, these representations can hardly have satisfactory performance on either fitting or prediction. In this thesis, we render a new approach for modelling times series of count data, which has sound predictive performance for modelling linear or log-linear time series of counts and outperforms some popular GLM-based models in terms of prediction on non-linear time series of counts for the data that we use. Furthermore, the model we provide can fit data sequence with the more general cases, such as more dimensions, which means the model can deal with both univariate times series and their multivariate counterparts. We also propose the use of our model in standard Poisson regression in which observations and covariates are observed in time. Our model for analysing time series of count data is inspired by Sarkar and Dunson [2016] that modelled categorical data through the Bayesian tensor factorisations (BTF).

The rest of this thesis is structured as follows.

In Chapter 2, we provide the framework of important techniques that are used in this thesis. These techniques include Bayesian inference and Bayesian nonparametrics, such as the Dirichlet process.

Chapter 3 introduces some popular approaches for modelling time series of count data, including INAR, INMA, Poisson autoregressive and INGARCH models. Chapter 3 provides some essential and basic pieces of knowledge for the models mentioned.

Chapter 4 studies the previous application of the BTF model on time series of categorical data in Sarkar and Dunson [2016], including the model specification, the estimation and inference, and model comparisons. The technique of tensor factorisation and the corresponding application is also introduced in this chapter.

Chapter 5 introduces our simple BTF approach for modelling univariate time series of count data with a small range. We show a fair predictive performance of our simple BTF approach in Chapter 5, while the approach fails to model the time series of relatively large counts. Then we generalise the simple BTF approach to a general BTF model that can tolerate a dataset of large or even infinite range in Chapter 6. Chapter 7 extends the previous two chapters into our general BTF model that deals with multivariate time series of count data with any range.

Chapter 8 extends the idea of our BTF model to the general Poisson regression. It provides the model specification, the estimation and inference, and the application of COVID-19 dataset to study the relationship between the daily death count and the daily confirmed count in the previous days.
Conclusions, including the summary of this thesis and the expected future work, are in Chapter 9.
Frequentists and Bayesians use different definitions of *probability*. A frequentist views probability as a long-term frequency of occurrence of an event and does not attach probabilities to hypotheses or to any fixed but unknown values in general. In contrast, her or his Bayesian counterpart regards probability as the degrees of belief. A Bayesian uses probability to represent the uncertainty in any event or hypothesis. This chapter introduces the framework of Bayesian statistics with emphasis on nonparametrics, tools with two Markov chain Monte Carlo (MCMC) algorithms including Gibbs sampling and Metropolis-Hastings, that are utilised in this thesis for posterior computations. We also mention MCMC convergence diagnostics.

1 Bayesian Inference

Suppose that we have some parameters \( \theta \) in a parameter space \( \Theta \), the theorem for Bayesian inference that computes the *posterior probability* can be expressed as

\[
p(\theta | E) = \frac{p(E | \theta)p(\theta)}{p(E)},
\]

(2.1)

where \( E \) is the *evidence* and \( p(E) \) can be regarded as the chance of the evidence’s happening. \( p(\theta) \) is a general belief of parameters or named as *prior probability*, that is the estimate of the probability of \( \theta \) before the current evidence \( E \) is observed. \( p(E | \theta) \) is the sampling distribution describing the variability in \( E \) given the assumed parameters \( \theta \). \( p(\theta | E) \) represents our updated belief on prior when the evidence \( E \) is observed, or the posterior probability.
The mathematical expression in (2.1) is straightforward. By given \( p(E), p(E \mid \theta) \) and prior \( p(\theta) \), we can derive our updated posterior probability \( p(\theta \mid E) \). Take our posterior as the new prior, and iteratively, the posterior can be updated as long as new data are observed. With good information on prior, one can improve the predictive ability of Bayesian inference. Another advantage of the structure in (2.1) is that it can allow hyperparameters to construct Bayesian hierarchical models, which provides improvements in both model fitting and predicting [Gelman et al., 2006], where a hyperparameter is a parameter of a prior distribution in Bayesian statistics.

Even though Bayesian inference has its significant advantages, inevitably, there still exist some drawbacks. Robert [2007]; Wasserman [2013] discussed the disadvantages of Bayesian inference. Firstly, there is no correct way to choose a prior. Therefore, Bayesian inferences require the capability to convert subjective prior beliefs into appropriate mathematics and without such capability, misleading results are likely to be generated. Secondly, (2.1) can produce a posterior distribution that are heavily influenced by the prior and it will be difficult to convince whose who disagree with the validity of the chosen prior. Moreover, the posterior calculation is often with a high computational cost, especially in models with numerous parameters. In order to facilitate the computation, one can use the MCMC approaches.

2 Markov Chain Monte Carlo

Monte Carlo

Before the description of MCMC algorithms, we introduce the Monte Carlo as the first step. Monte Carlo provides a direct method for performing simulation and integration. Due to its simplicity and directness, Monte Carlo is easy to use [Caflisch, 1998]. Suppose there is a complex integral

\[
\int_a^b h(x)dx = \int_a^b f(x)p(x)dx = \mathbb{E}_{p(x)}[f(x)].
\]

which can be decomposed into the production of a function \( f(x) \) and a probability density function \( p(x) \) defined over the interval \((a, b)\). One can rewrite the integral (2.2) as an expectation of \( f(x) \) over the density function \( p(x) \) as
Therefore, if we draw \( N \) random variables \( x_1, \ldots, x_N \) of from the probability density function \( p(x) \), the Monte Carlo integration can be expressed as

\[
\int_a^b h(x)dx = \mathbb{E}_{p(x)}[f(x)] \simeq \frac{1}{N} \sum_{i=1}^N f(x_i).
\]

One can use the above Monte Carlo integration (2.3) to approximate posterior distributions required for a Bayesian inference. Similarly, an integral \( I(y) = \int f(y \mid x)p(x)dx \) can be approximated as

\[
\hat{I}(y) = \frac{1}{N} \sum_{i=1}^N f(y \mid x_i)
\]

where \( x_i \)'s are draws from the probability density function \( p(x) \). The Monte Carlo standard error is given by

\[
\frac{1}{N} \left( \frac{1}{N-1} \sum_{i=1}^N \left( f(y \mid x_i) - \hat{I}(y) \right)^2 \right).
\]

Markov Chain

Norris and Norris [1998] provided an account of the elementary theory of Markov chains. Let \( X_t \) denote the value of a random variable at time \( t \) and the state space refer to the range of possible \( X \) values. The random variable is a Markov process if the transition probabilities between different values in the state space depend only on the random variable’s current state, or mathematically,

\[
p(X_{t+1} = s_j \mid X_t = s_i, \ldots, X_0 = \cdot) = p(X_{t+1} = s_j \mid X_t = s_i).
\]

The above equation (2.4) indicates that the current state is the only information needed to predict the future for a Markov random variable. A Markov chain is a sequence of random variables \( (X_0, \ldots, X_N) \) that are generated by a Markov process. A particular chain can be defined by its transition probabilities \( P(i, j) = P(i \rightarrow j) \), which is the probability that a process at state space \( s_i \) moves to \( s_j \) in a single step, or mathematically,

\[
P(i, j) = P(i \rightarrow j) = p(X_{t+1} = s_j \mid X_t = s_i).
\]

Let \( \pi_j(t) = p(X_t = s_j) \) denote the probability that a chain is in state \( j \) at time \( t \), and \( \pi(t) \) the row vector of the state space probabilities at time \( t \). Then at time \( t+1 \), the probability
that the chain has state value \( s_i \) can be expressed as

\[
\pi_i(t + 1) = p(X_{t+1} = s_i) \\
= \sum_k p(X_{t+1} = s_i \mid X_t = s_k)p(x_t = s_k) \\
= \sum_k P(k \to i) \pi_k(t) = \sum_k P(k, i) \pi_k(t).
\]

(2.5)

Define the probability transition matrix \( P \) as the matrix whose \( i, j \)-th element is \( P(i, j) \), the probability of moving from state \( i \) to \( j \), \( P(i \to j) \), satisfying \( \sum_j P(i, j) = \sum_j P(i \to j) = 1 \).

Then the above (2.5) can be rewritten as

\[
\pi(t + 1) = \pi(t)P
\]

and we can immediately show that

\[
\pi(t) = \pi(t - 1)P = (\pi(t - 2)P)P = \cdots = \pi(0)P^t.
\]

Let \( p_{ij}^{(n)} \) denote the probability that the process is in state \( j \) given that it started in state \( i \) \( n \) steps ago, or mathematically,

\[
p_{ij}^{(n)} = p(X_{t+n} = s_j \mid X_t = s_i),
\]

it shows that \( p_{ij}^{(n)} \) is the \( ij \)-th element of \( P^n \).

A Markov chain is irreducible if there exists a positive integer such that \( p_{ij}^{(n)} > 0 \) for all \( i, j \). Moreover, a chain is aperiodic when the number of steps required to move between two states is not required to be multiple of some integer, or in other words, the chain is not forced into some cycle of fixed length between certain states.

A stationary Markov chain has a stationary distribution \( \pi^* \) that satisfies

\[
\pi^* = \pi^*P.
\]

If a Markov chain has a unique stationary distribution, a sufficient condition is the detailed balance that can be expressed as, for all \( i, j \),

\[
P(i \to j)\pi^*_i = P(j \to i)\pi^*_j.
\]

(2.6)
As (2.6) holds, the Markov chain is regarded to be reversible and hence (2.6) is referred to as the reversibility condition.

Metropolis-Hastings Algorithm

We have introduced the Monte Carlo integration briefly at the beginning of this section. One difficulty with its application is in obtaining samples from some complex probability distribution \( p(x) \) in (2.3). In order to address this problem, Bayesian statisticians searched attempts from physicists by random sampling [Metropolis et al., 1953; Hastings, 1970], which resulted in the Metropolis-Hastings algorithm. A very detailed review of the Metropolis-Hastings method can be found in Chib and Greenberg [1995].

Suppose that our objective is to generate samples from some continuous target density function \( p(\theta) \) satisfying \( p(\theta) = f(\theta)/K \), where \( K \) is the normalising constant very difficult to be computed. By setting the number of iterations to be \( N \), for \( i \in \mathbb{Z}_{[1,N-1]} \), the Metropolis-Hastings algorithm steps are as follows.

1. Initiate a \( \theta_0 \) satisfying \( f(\theta_0) > 0 \).

2. Sample a candidate \( \theta^* \) from some proposal distribution \( q(\theta_i, \theta^*) \), which is the probability function of reaching \( \theta^* \) given a previous value \( \theta_i \).

3. Define \( a(\theta_i, \theta^*) \) as an acceptance rate function which means the probability of a move to be accepted towards \( \theta^* \) from \( \theta_i \), and

\[
a(\theta_i, \theta^*) = \min \left( \frac{p(\theta^*) q(\theta_i, \theta^*)}{p(\theta_i) q(\theta^*, \theta^*)}, 1 \right).
\] (2.7)

4. Sample a rate \( u \) from a uniform distribution \( \text{Uniform}(0,1) \),

- if \( u \leq a(\theta_i, \theta^*) \), we accept the candidate, namely, \( \theta_{i+1} = \theta^* \), and then we return to Step 2.
- Else if \( u > a(\theta_i, \theta^*) \), we reject the candidate, namely, \( \theta_i = \theta^* \), and then we return to Step 2.

A Metropolis-Hastings algorithm can be used to construct a stationary Markov chain, since it fulfils irreducibility, aperiodicity and reversibility.
Gibbs Sampling Algorithm

A Gibbs sampler, which was firstly introduced in Geman and Geman [1984], is a special case of a Metropolis-Hastings algorithm. It is with the acceptance rate function in (2.7), \(a(\cdot,\cdot)\), to be constant one. Casella and George [1992] provided a comprehensive introduction to the Gibbs sampler and Smith and Roberts [1993] showed the natural marriage of the sampler with Bayesian statistics in terms of obtaining posterior distributions. Each random value in a Gibbs sampling algorithm is always accepted due to that the rate function is one for certain. The objective is still to construct a Markov chain with values convergent to the target function. An important feature that the Gibbs sampler owns is that it only considers univariate conditional distributions, where all random variables but one are fixed. One significant advantage of this approach is that univariate conditional distributions, usually with simple forms, are far easier to simulate than their complex joint counterparts. Hence, we simulate \(m\) random variables sequentially from their \(m\) corresponding univariate conditionals instead of generating a \(m\)-dimensional vector using the fill joint distribution.

Suppose that one shows great interest to obtain the marginal distributions, \(p(x)\) and \(p(y)\), of a bivariate random variable \((x,y)\), the Gibbs sampler considers a sequence of conditional distribution, \(p(x \mid y)\) and \(p(y \mid x)\) rather than the joint distribution \(p(x, y)\). By setting some initial values \(x_0\) and \(y_0\), the Gibbs sampler follows as

\[
\begin{align*}
x_i &\sim p(x \mid y = y_{i-1}) \\
y_i &\sim p(y \mid x = x_i)
\end{align*}
\]

where \((x_i, y_i)\) represent the value of the bivariate random variable \((x,y)\) at the \(i\)-th iteration. It will generate a Gibbs sequence of length \(N\) after \(N\) iterations. The Gibbs sequence converges to a stationary distribution that is independent of the starting values, and the stationary distribution is the target distribution [Tierney, 1994].

Generally, suppose there are \(m\) random variables that need to be sampled, and we define \(\theta^{(\cdot-k)}\) as the vector of random variables excluding the \(k\)-th random variable, therefore, for the \(i\)-th iteration, we can sample the \(k\)-th random variable \(\theta_i^{(k)}\) from the distribution

\[
\theta_i^{(k)} \sim p\left(\theta^{(k)} \mid \theta^{(1)} = \theta_i^{(1)}, \ldots, \theta^{(k-1)} = \theta_i^{(k-1)}, \theta^{(k+1)} = \theta_i^{(k+1)}, \ldots, \theta^{(m)} = \theta_i^{(m)}\right),
\]

then the approximated marginal density for each random variable \(\theta^{(k)}, k \in \mathbb{Z}_{[1,m]}\), can be
computed as
\[ \hat{p}_N(\theta^{(k)}) = \frac{1}{N} \sum_{i=1}^{N} p\left( \theta^{(k)} | \theta^{(1)} = \theta^{(1)}_1, \ldots, \theta^{(k-1)} = \theta^{(k-1)}_i, \theta^{(k+1)} = \theta^{(k+1)}_{i-1}, \ldots, \theta^{(m)} = \theta^{(m)}_i \right). \]

3 MCMC Convergence Diagnostics

MCMC methods enable its user to successively sample values from a convergent Markov chain when properly defined and implemented; however, it is often difficult to decide at what point is it reasonable to believe that the samples are truly representative of the underlying stationary distribution of the Markov chain [Cowles and Carlin, 1996]. MCMC diagnostics can be functional to check whether the quality of a sample generated through an MCMC algorithm is sufficient to provide an accurate approximation of the target distribution.

Cowles and Carlin [1996] comprehensively reviewed MCMC diagnostic tools for checking the convergence of stationary Markov chains. They are many approaches for testing the MCMC convergence, including the fixing-width stopping rules [Jones et al., 2006], effective sample size [Robert and Casella, 2013] and Raftery-Lewis diagnostic [Raftery and Lewis, 1991]. Recently, Vehtari et al. [2021] have showed that the convergence diagnostics of Gelman-Rubin diagnostic [Gelman et al., 1992] has serious flaws when the chain has a heavy tail or when the variance varies across the chains, and proposes an alternative rank-based diagnostic that fixes the problems.

In this thesis, we use trace plots for the MCMC diagnostics. These plots clearly show the sampled values of \( \theta_i \) or some \( f(\theta_i) \) against the \( i \)-th iteration of MCMC. Much information can be displayed with a trace plot, such as the running mean of samplers, 5% and 95% quantiles of samplers. It is advisable to run multiple parallel chains so as to check convergence through trace plots, by judging whether all chains look similar and oscillate around similar point estimates.

Some tunings can facilitate the effective convergence of a Markov chain. In this thesis, we use burn-in. Burn-in is the idea to discard some initial iterations during the simulation, which is to avoid values sampled by these initial iterations significantly influenced by starting values and bad information about the target distribution. Beside the burn-in approach, the adjustment of the chain length is another way. Once the convergence of a stationary distribution is reached, the number of samplers’ decision influences efficiency.
CHAPTER 2. BAYESIAN INFERENCE

More details can be found in Brooks et al. [2011].

4 Bayesian Nonparametric Models

A Bayesian nonparametric model is a Bayesian model on an infinite-dimensional parameter space. The parameter space is typically chosen as the set of all possible solutions for a given learning problem [Orbanz and Teh, 2010]. In other words, a Bayesian nonparametric model has a potentially infinite number of parameters, which can outperform its parametric counterpart in terms of data modelling.

Suppose that we have a sequence \( \{y_1, \ldots, y_n\} \) drawn independently and identically from some probability distribution \( F(\cdot \mid \theta) \), where \( \theta \) is in some parameter space \( \Theta \) with a distribution \( \pi \). The model can be expressed as

\[
\begin{align*}
y_i \sim F(y \mid \theta), \\
\theta \sim \pi,
\end{align*}
\]

where \( i = 1, \ldots, n \). Hierarchically, (2.8) can be expressed as, for \( i = 1, \ldots, n \),

\[
\begin{align*}
y_i \mid \theta_i & \sim F(y \mid \theta_i), \\
\theta_i \mid G & \sim G, \\
G & \sim G_0,
\end{align*}
\]

where \( G \) is a discrete probability measure, defined as follows, and \( G_0 \) is a base nonparametric prior over countable measures.

**Definition 4.1 (σ-algebra).** \( S \) is a σ-algebra on the set \( X \) if these following properties are satisfied,

- \( S \) is not empty;
- if \( A \in S \) then \( A^c \in S \);
- if \( A_1, A_2, \cdots \in S \), then \( \cup_{i=1,2,\cdots} A_i \in S \).

**Definition 4.2 (Measure Space).** \((X, S)\) is a measure space if \( S \) defines a σ-algebra on \( X \).

**Definition 4.3 (Probability Measure).** Suppose we have a measure \( \mu_m \) over \((X, S)\) which is a function \( \mu_m : S \to [0, \infty) \), then
• \( \mu_m(\emptyset) = 0; \)

• if \( A_1, A_2, \ldots \in S \) are disjoint, \( \mu_m(\bigcup_{i=1,2,\ldots} A_i) = \sum_i \mu_m(A_i); \)

• a probability measure is a measure where \( \mu_m(\mathcal{X}) = 1. \)

A Bayesian model can be defined to be parametric if the parameter space \( \Theta \) belongs to a finite set, and similarly, a Bayesian nonparametric model is with an infinite parameter space \( \Theta \). A Bayesian nonparametric prior can be regarded as a prior distribution over a distribution and is applied to mixture models.

**Definition 4.4 (Mixture distribution).** Suppose we have a probability distribution \( p(\cdot) \), it can be called as a mixture distribution if \( p(\cdot) \) can be written in the form

\[
p(\cdot) = \int f(\cdot \mid \theta) dG(\theta), \tag{2.9}
\]

where \( \theta \in \Theta \) and \( G \) is a discrete probability measure for some parameter space \( \Theta \). By introducing a weight vector \( (\pi_1, \ldots, \pi_n) \) with \( \sum_{i=1}^n \pi_i = 1 \), \( G(\theta) \) can be expressed as

\[
G(\theta) = \sum_{i=1}^n \pi_i \delta_{\theta_i}, \tag{2.10}
\]

where \( \delta_{\theta_i} \) is a Dirac measure. The expression in (2.9) can be developed as

\[
p(\cdot) = \int f(\cdot \mid \theta) dG(\theta)
= \int f(\cdot \mid \theta) d\left( \sum_{i=1}^n \pi_i \delta_{\theta_i} \right)
= \sum_{i=1}^n \pi_i f(\cdot \mid \theta_i).
\]

5 Dirichlet Process

**Dirichlet Distribution**

Before mentioning the Dirichlet Process, we provide a brief introduction to the Dirichlet distribution. A Dirichlet distribution is a generalisation of a Beta distribution, therefore, a Dirichlet distribution is one of the classes of continuous multivariate distributions. For the Bayesian nonparametrics, a Dirichlet distribution is always preferred since it can
be regarded as a prior over finite probability mass functions, and its conjugacy with a multinomial distribution.

Suppose we have $K$ random components of a probability mass function $\pi = (\pi_1, \ldots, \pi_K)$ with the criteria $\pi_k \geq 0$ for any $k \in [1, K]$ and $\sum_{k=1}^{K} \pi_k = 1$. Define $\alpha = (\alpha_1, \ldots, \alpha_K)$ as a parameter vector with $\alpha_k \geq 0$ for any $k = 1, \ldots, K$, the Dirichlet distribution on the $K$-dimensional probability simplex $\pi$ can be expressed as

$$p(\pi \mid \alpha) = \frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)} \prod_k \pi_k^{\alpha_k-1},$$

(2.11)

with $\Gamma(\tau) = \int_{0}^{\infty} x^{\tau-1} \exp(x)dx$, or we can rewrite (2.11) as

$$\pi \sim \text{Dirichlet}(\alpha).$$

A Beta distribution is a special case of a Dirichlet distribution by setting $K = 2$, which can be written as

$$p(\pi_1, \pi_2 \mid \alpha_1, \alpha_2) = \frac{\Gamma(\alpha_1 + \alpha_2)}{\Gamma(\alpha_1)\Gamma(\alpha_2)} \pi_1^{\alpha_1-1}\pi_2^{\alpha_2-1}. \tag{2.12}$$

With $\pi_1 + \pi_2 = 1$, (2.12) is clearly a Beta($\alpha_1, \alpha_2$) distribution.

By setting $\alpha_0 = \sum_{k=1}^{K} \alpha_i$, the expectation and variance of $\pi_k$ for $k \in [1, K]$ can be expressed as

$$\mathbb{E}[\pi_k] = \frac{\alpha_k}{\alpha_0} \quad \text{and} \quad \text{Var}[\pi_k] = \frac{\alpha_k(\alpha_0 - \alpha_k)}{\alpha_0^2(\alpha_0 + 1)}. \tag{2.13}$$

For any $1 \leq k, j \leq K$ and $k \neq j$, the covariance of $(\pi_k, \pi_j)$ is

$$\text{Cov}[\pi_k, \pi_j] = -\frac{\alpha_k \alpha_j}{\alpha_0^2(\alpha_0 + 1)}. \tag{2.14}$$

A Dirichlet distribution possesses an important property that it is **conjugate** with a multinomial distribution, or a Dirichlet distribution is a **conjugate prior** for a multinomial distribution as a likelihood function. In Bayesian statistics, if the posterior distribution is in the same probability distribution family as the prior probability distribution, the prior and posterior are then referred to as **conjugate distributions**, and the prior is called a conjugate prior for the likelihood function.

Suppose we want to sample some label of cluster $z_i$, from $z = \{z_i\}_{i \in [1,n]}$, for a data
sequence with $n$ data points, then we can have a model for the $i$-th data as

$$z_i \mid \pi \sim \text{Multinomial}(\pi),$$

$$\pi \mid \alpha \sim \text{Dirichlet}(\alpha). \quad (2.13)$$

According to (2.13), the joint distribution over $z_i$ and $\pi$ can be expressed as

$$p(\pi \mid \alpha) \times \prod_{i=1}^{n} p(z_i \mid \pi) = \frac{\Gamma(\alpha_0)}{\prod_k \Gamma(\alpha_k)} \prod_{k=1}^{K} \pi_k^{\alpha_k-1} \times \prod_{k=1}^{K} n_k^{\alpha_k},$$

where $n_k = \sum_{i=1}^{n} \{z_i = k\}$.

Then the posterior distribution is

$$p(\pi \mid z, \alpha) = \frac{\Gamma(n + \alpha_0)}{\prod_k \Gamma(n_k + \alpha_k)} \prod_{k=1}^{K} \pi_k^{n_k + \alpha_k - 1},$$

or we can write it in the form of another Dirichlet distribution, which is

$$\pi \mid z, \alpha \sim \text{Dirichlet}(n_1 + \alpha_1, \ldots, n_K + \alpha_K). \quad (2.14)$$

(2.14) proves that a Dirichlet distribution is conjugate with a multinomial distribution. More properties of the Dirichlet distribution can be found in Ghosh and Ramamoorthi [2003].

**Dirichlet Process**

Ferguson [1973] introduced the Dirichlet process (DP) that can be thought of as an infinite-dimensional Dirichlet distribution and the prior distribution over distributions. DPs can fulfil two desirable properties of a prior distribution for Bayesian nonparametric problems, which are

- the support of the prior distribution is expected to be large, theoretically, $(-\infty, \infty)$;
- posterior distributions given a sample of observations from the true probability distribution should be manageable analytically.

A Dirichlet process is defined as a random probability measure $G$ over a measure space $(\mathcal{X}, \mathcal{S})$ such that for any finite set of measurable sets $A_1, \ldots, A_K \in \mathcal{S}$ partitioning $\mathcal{X}$, or in
other words,

\[ A_i \cap A_j = \emptyset \] for \( 1 \leq i, j \leq K \) and \( i \neq j \),

\[ A_1 \cup \cdots \cup A_K = \mathcal{X}, \]

one can have the following representation

\[ (G(A_1), \ldots, G(A_K)) \sim \text{Dirichlet}(\alpha G_0(A_1), \ldots, \alpha G_0(A_K)), \tag{2.15} \]

where \( \alpha \) and \( G_0 \) are parameters for the DP. \( \alpha \) is a scale or referred as the concentration, and \( G_0 \) is the base distribution. \( (2.15) \) can be rewritten in a more straightforward way for the DP as

\[ G \sim \text{DP}(\alpha, G_0). \tag{2.16} \]

For any \( A \) that is a measure subset of \( \mathcal{X} \), the mean and variance of \( G(A) \) can be expressed as

\[ \mathbb{E}[G(A)] = G_0(A), \]
\[ \text{Var}[G(A)] = \frac{G_0(A)(1 - G_0(A))}{\alpha + 1} \]

respectively. Namely, \( G_0 \) is the mean of \( G \) and the larger the \( \alpha \) is, the smaller the variance.

Like a Dirichlet distribution, a DP is conjugate with a multinomial distribution. If we define a random variable \( \theta_i \) for each observation \( y_i, i = 1, \ldots, n \), following

\[ \theta_i \mid G \sim G, \]

by given the prior distribution in \( (2.16) \), one can easily obtain the conjugacy between a Dirichlet process and a multinomial distribution as the posterior distribution is

\[ (G(A_1), \ldots, G(A_K)) \mid \theta_1, \ldots, \theta_n \sim \text{Dirichlet}(\alpha G_0(A_1) + n_1, \ldots, \alpha G_0(A_K) + n_K) \tag{2.17} \]

where \( n_k = \sum_{i=1}^{n} \{ \theta_i \in A_k \} \) for \( k = 1, \ldots, K \). According to \( (2.17) \), the posterior over \( G \) is a DP, since \( (2.17) \) is valid for all finite measurable partitions \( A_1, \ldots, A_K \). By utilising the notation of Dirac measure \( \delta_{\theta_i} \) which is a point mass located at \( \theta_i \) and \( n_k = \sum_{i=1}^{n} \delta_{\theta_i}(A_k), \)
we can have an alternative expression for the posterior as

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

which not only indicates that a DP is conjugate with a multinomial distribution but also does that the base distribution posterior is a weighted average between the prior base distribution \( G_0 \) and the empirical distribution \( \sum_{i=1}^{n} \delta_{\theta_i} / n \). The first part of the posterior of the base distribution is proportional to the concentration scale \( \alpha \), and the second part is proportional to the number of observations \( n \). Therefore, \( \alpha \) has a close connection with the prior, since if \( \alpha \to 0 \) or \( \alpha \ll n \), the posterior of the base distribution is dominated by the empirical distribution which is a close approximation of the true underlying distribution, which shows the property of being consistent for the DP [Teh, 2010].

**Blackwell-MacQueen Urn Scheme**

Teh [2010] also provided a very detailed introduction to the Blackwell-MacQueen Urn Scheme. Again one draws \( G \sim \text{DP}(\alpha, G_0) \) and an i.i.d. sequence \( \theta_1, \ldots, \theta_n \sim G \). Consider the predictive distribution for \( \theta_{n+1} \) conditional on \( \theta_1, \ldots, \theta_n \) and with \( G \) being marginalised out. For a measurable set \( A \), one has

\[
p(\theta_{n+1} \in A \mid \theta_1, \ldots, \theta_n) = \mathbb{E}[G(A) \mid \theta_1, \ldots, \theta_n] = \frac{1}{\alpha + n} \left( \alpha G_0(A) + \sum_{i=1}^{n} \delta_{\theta_i}(A) \right)
\]

where the last step follows from the posterior base distribution of \( G \) given the first \( n \) observations. With \( G \) to be marginalised out,

\[
\theta_{n+1} \mid \theta_1, \ldots, \theta_n \sim \frac{1}{\alpha + n} \left( \alpha G_0 + \sum_{i=1}^{n} \delta_{\theta_i} \right). \tag{2.18}
\]

(2.18) shows that by given \( \theta_1, \ldots, \theta_n \), the posterior base distribution is also the predictive distribution of \( \theta_{n+1} \).

Blackwell et al. [1973] introduced an urn scheme which takes the sequence of predictive distributions in (2.18), which is also called as the Polya or Hoppe urn scheme [Hoppe, 1984]. Suppose a random variable \( \theta \) drawn from \( \theta \sim G \) represents the colour painted on a ball, and there is an urn that contains previously drawn balls. Then the scheme, which can be utilised to indicate the existence of a DP, is as follows.
1 Draw $\theta_1 \sim G_0$ since there is no ball in the urn. We paint the ball with this colour and throw it to the urn.

2 For the $(n + 1)$-th draw, one

- either draws a ball with a new colour $\theta_{n+1}$ from $\theta_{n+1} \sim G_0$, with probability $\frac{\alpha}{\alpha+n}$, and throw this ball into the urn,
- or takes one ball out of the urn, which is to draw $\theta_{n+1}$ from $\theta_{n+1} \sim \sum_{i=1}^{n} \delta_{\theta_i}$, with probability $\frac{n}{\alpha+n}$ paints a new ball with the same colour and throws both of the balls into the urn.

Instead of the indication of a DP, the above Blackwell-MacQueen urn scheme shows two other properties. The first one is that the sequence in (2.18) is infinitely exchangeable. It means that for every $n$, the probability of drawing $\theta_1, \ldots, \theta_n$ by (2.18), in that order, is equal to the probability of generating them in any order. Moreover, the predictive distribution (2.18) has point masses on the past draws $\theta_1, \ldots, \theta_n$, which implies that $G$ has point masses as well, and it is a discrete distribution.

**Chinese Restaurant Process**

Instead of the properties just mentioned in the previous part, (2.18) also possesses an extra property of clustering. It is inevitable that the values of draws are repeated, so if we let $\theta_1^*, \ldots, \theta_K^*$ be the unique values in $\theta_1, \ldots, \theta_n$, and $n_k$, for $k = 1, \ldots, K$, be the repeated number of $\theta_k^*$, then the predictive distribution in (2.18) can be rewritten as

$$
\theta_{n+1} \mid \theta_1, \ldots, \theta_n \sim \frac{1}{\alpha+n} \left( \alpha G_0 + \sum_{k=1}^{K} n_k \delta_{\theta_k^*} \right).
$$

(2.19) also implies a property of clustering, that is the larger $n_k$ is, the higher the probability it will have.

A Chinese restaurant process was discussed in Aldous [1985]; Pitman [2006], and it is a stochastic process that defines a probability distribution over partitions of integers. Suppose there is a Chinese restaurant with infinite tables, and each table has an infinite capacity of customers. The first customer sits at the first table, and the second one can choose to either sit with the first customer at the first table or choose a new table. With the expression of (2.19) and in general, the $(n + 1)$-th customer choose to sit at table $k$ with a probability proportional to the customers already sitting around the table, $n_k$, or choose an empty table with a probability proportional to $\alpha$. By given $n$ observations, for
$0 \ll \alpha < n$, the number of cluster $K$ has the mean

$$
\mathbb{E}[K \mid n] = \frac{\alpha}{\alpha + i - 1} = \alpha (\psi(\alpha + n) - \psi(\alpha)) \\
\simeq \alpha \log \left(1 + \frac{n}{\alpha}\right)
$$

and the variance

$$
\text{Var}[K \mid n] = \alpha (\psi(\alpha + n) - \psi(\alpha)) + \alpha^2 (\psi'(\alpha + n) - \psi'(\alpha)) \\
\simeq \alpha \log \left(1 + \frac{n}{\alpha}\right)
$$

where $\psi(\cdot)$ is the digamma function and $\psi'(\cdot)$ represents the first order derivative of the digamma function.

**Stick-Breaking Construction**

Besides the Blackwell-MacQueen urn scheme, the stick-breaking process is another popular and classic construction of the DP, which was firstly purposed by Sethuraman [1994]. Suppose we have a stick with length one, for the first step, we break it at a point with the portion $V_1$ and set $\pi_1$ to be the length of the stick we broke off. Following this scheme, we can obtain $\pi_2, \pi_3, \ldots, \pi_\infty$, which makes the infinite number of clusters theoretically reachable.

(2.10) represents a finite limit form for $G$, and with $K \to \infty$, one can obtain $G$ as an atomic distribution, or

$$
G = \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k^*}.
$$

(2.20)

With (2.20), we can represent the stick-breaking construction in a statistical way as

$$
G = \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k^*},
$$

$$
\pi_k = V_k \prod_{l=1}^{k-1} (1 - V_l),
$$

(2.21)

$$
V_k \sim \text{Beta}(1, \alpha),
$$

$$
\theta_k^* \sim G_0.
$$

then $G \sim \text{DP}(\alpha, G_0)$, the proof of which can also be found in Sethuraman [1994]. Let $\pi = (\pi_1, \pi_2, \ldots, \pi_\infty)$, the stick-breaking distribution over $\pi$ in (2.21) is also referred to
as the Griffiths-Engen-McCloskey (GEM) distribution and written as \( \pi \sim \text{GEM}(\alpha) \). The expression in (2.21) is straightforward, and due to this, the stick-breaking construction leads to many extensions for the DP [Ishwaran and James, 2001].

**Application of Dirichlet Process**

One of the most popular and important DP applications is to cluster data by mixture models [Escobar and West, 1995; Neal, 1992; Rasmussen, 2000]. We can recognise the nonparametric property of a DP in a mixture model with a countably infinite number of clusters or components.

Suppose we have observations \( y_1, \ldots, y_n \), with their corresponding latent parameters \( \theta_1, \ldots, \theta_n \), and each \( y_i \), where \( i = 1, \ldots, n \), has a distribution \( F(\theta_i) \). Each \( \theta_i \) is i.i.d. from \( G \), then we can have a hierarchical structure for \( y_i \) as

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

(2.22)

With the stick-breaking process in (2.21), we can denote \( z_i \) as a latent cluster variable taking a value from 1, 2, \ldots, \infty with the probability vector \( \pi = (\pi_1, \pi_2, \ldots, \pi_\infty) \). Then we can develop (2.22) as

\[
\begin{align*}
y_i \mid z_i, \{\theta_k^*\} &\sim F(\theta_{z_i}^*), \\
z_i \mid \pi &\sim \text{Multinomial}(\pi), \\
\pi \mid \alpha &\sim \text{GEM}(\alpha), \\
\theta_k^* \mid G_0 &\sim G_0.
\end{align*}
\]

(2.23)

According to terminologies of mixture models, \( \pi \) is the mixing proportion vector, \( \theta_k^* \) is a cluster parameter, \( F(\theta_k^*) \) is the distribution over data with the \( k \)-th cluster, and \( G_0 \) is defined to be the prior over cluster parameters.

The representation of the DP mixture model in (2.23) is clearly an infinite mixture model containing a countably infinite number of clusters. Due to the construction scheme of \( \pi_k \), \( \pi_k \) decreases exponentially with the increase of \( k \); therefore, a limited amount of clusters are sampled to model the observations. Theoretically, the DP mixture model in (2.23) is different from any finite mixture model where the number of clusters utilised to model the data is fixed. Because of this property, the actual number of clusters in the DP mixture
model can be automatically inferred from data by the usual Bayesian posterior inference framework, investigated in Neal [2000]. However, for mixture models with a finite number of clusters, the inference is always with difficulties due to the number of clusters is fixed. Throughout this thesis, we use the stick-breaking process wherever a DP is needed.
Chapter 3

Models for Time series of Count Data

In this chapter, we introduce some popular approaches for modelling times series of count data. Popular approaches include the integer-values autoregressive (INAR) model, integer-valued moving average (INMA) model, Poisson autoregression (PAR) model and integer-valued generalised autoregression conditional heteroscedastic (INGARCH) model.

1 Integer-Valued Autoregression

We introduce the very popular integer-valued autoregression model with the first order, or INAR(1), originally developed by McKenzie [1985] in this section. The approach binomial thinning, introduced in Steutel and Van Harn [1979], is used to develop INAR(1). \{y_t\} is a process, or a collection of random variables that are ordered in time, and suppose there is another i.i.d. process \{\epsilon_t\} with \alpha \in [0, 1], we have the recursion

\[ y_t = \alpha \circ y_{t-1} + \epsilon_t, \quad t \geq 1, \]

which is also called as the thinning operation. When \( y_t \) takes count value (non-negative integer), the random variable \( \alpha \circ y_{t-1} := \sum_{i=1}^{y_{t-1}} b_i \) is regarded as a binomial thinning operator if counting series \( b_i \) are independent Bernoulli trials according to B(1, \alpha) which is binomial distribution [Weiß, 2008]. Since the representation \( y_{t-1} \) is equivalent to \( y_t \) when \( t \)
can take any value, then we take $\alpha \circ y_t := \sum_{i=1}^{b_i} y_i$, we have

$$P(b_i = 1) = 1 - P(b_i = 0) = \alpha$$

and with all random variables being independent.

The expectation of $\alpha \circ y_t$ is

$$\mathbb{E}(\alpha \circ y_t) = \alpha \mathbb{E}(y_t),$$

with $1 \circ y_t = y_t$ and $0 \circ y_t = 0$. The variance of $\alpha \circ y_t$ is

$$\text{Var}(\alpha \circ y_t) = \alpha^2 \text{Var}(y_t) + \alpha(1 - \alpha)\mathbb{E}(y_t).$$

Al-Osh and Alzaid [1987] summarised some properties of INAR(1) processes, one of which is about the expectation and variance of process $\{y_t\}$. For a stationary INAR(1) process $\{y_t\}$ for non-negative integer $t$, we have

$$\mu_y := \mathbb{E}(y_t) = \frac{\mu_{\epsilon}}{1 - \alpha},$$

$$\sigma_y^2 := \text{Var}(y_t) = \frac{\alpha \mu_{\epsilon} + \sigma_{\epsilon}^2}{1 - \alpha^2},$$

where $\mu_{\epsilon}$ and $\sigma_{\epsilon}^2$ are the mean and variance of process $\{\epsilon_t\}$.

The probability generating function of taking value $z$ for INAR(1) process $\{y_t\}$, denoted by $p_y(z)$, satisfies

$$p_y(z) = p_y(1 - \alpha + \alpha z)p_{\epsilon}(z),$$

where $p_{\epsilon}(z)$ is the probability generating function of process $\{\epsilon_t\}$ to take value $z$.

Al-Osh and Alzaid [1988] provided the regression properties of stationary INAR(1) processes, where states the expressions for the conditional mean and variance as

$$\mathbb{E}(y_t \mid y_{t-1}) = \alpha y_{t-1} + \mu_{\epsilon},$$

$$\text{Var}(y_t \mid y_{t-1}) = \alpha(1 - \alpha)y_{t-1} + \sigma_{\epsilon}^2.$$
with the marginal probability generating function when taking value $z$ as

$$p_y(z) = \exp\left(\frac{\lambda}{1-\alpha}(z-1)\right),$$

i.e. the process is generated by a Poisson distribution with parameter $\frac{\lambda}{1-\alpha}$.

### 2 Integer-Valued Moving Average

McKenzie [1988]; Al-Osh and Alzaid [1988] introduced the integer-valued moving average (INMA) models with two further approaches proposed by Brännäs and Hall [2001]. Similar to the above section, we assume an i.i.d. process $\{\epsilon_t\}$ with non-negative integer range, and some coefficients $\beta_0, \beta_1, \ldots, \beta_p \in [0, 1]$, with $\beta_p \neq 0$, we have an INMA($p$) process $\{y_t\}$ with the expression

$$y_t = \beta_0 \circ \epsilon_t + \beta_1 \circ \epsilon_{t-1} + \cdots + \beta_p \circ \epsilon_{t-p}.$$

The expressions of expectation and variance of a stationary INMA($p$) process are

$$\mathbb{E}(y_t) = \mu \sum_{j=0}^{p} \beta_j,$$

$$\text{Var}(y_t) = \mu \sum_{j=0}^{p} \beta_j + (\sigma^2_{\epsilon} - \mu) \sum_{j=0}^{p} \beta^2_j.$$

Moreover, the probability generating function of $y_t$ taking the value $z$ is

$$p_y(z) = \prod_{j=0}^{p} p_{\epsilon}(1 - \beta_j + \beta_j z).$$

Again, suppose there is a process $\epsilon_t$ follows a Poisson distribution with parameter $\lambda$, or in other words, $\mu_{\epsilon} = \sigma^2_{\epsilon} = \lambda$ and $p_{\epsilon}(z) = \exp(\lambda(z-1))$, then the INMA($p$) model can be called as a Poisson INMA($p$) model. The probability generating function of $y_t$ taking the value $z$ is

$$p_y(z) = \prod_{j=0}^{p} \exp(\lambda \beta_j (z-1)) = \exp\left(\lambda \sum_{j=0}^{p} \beta_j (z-1)\right),$$

therefore, $\{y_t\}$ is generated by a Poisson distribution with parameter $(\lambda \sum_{j=0}^{p} \beta_j)$. More details of the regression of Poisson INMA($p$) model can be found in Weiss [2008].
3 Poisson Autoregression

A Poisson autoregressive (PAR) model is an application of GLM. The details of GLM can be found in many papers including Dobson and Barnett [2008]; Fahrmeir and Tutz [2013]; Fox [2015]; McCullagh [2018]; Nelder and Wedderburn [1972]. A possible Poisson autoregression model of \( y_t \) given \( y_{t-1}, \ldots, y_{t-q} \), for \( t = 1, \ldots, T \), is

\[
y_t \sim \text{Poisson}(\lambda_t), \quad \lambda_t = \beta_0 + \sum_{i=1}^{q} \beta_i y_{t-i},
\]

(3.1)

where \( \beta_0 > 0 \), \( \beta_i \geq 0 \) for \( i = 1, \ldots, q \) with \( \sum_{i=1}^{q} \beta_i < 1 \).

In this thesis, we compare the predictive performance of our proposed Bayesian tensor factorisations model against its competitors by criteria from both the views of frequentist and Bayesian, respectively. Therefore, it is beneficial and respectful to introduce the mechanism of some time series models from the perspective of frequentists.

Cameron and Trivedi [2001] investigated the maximum likelihood estimator (MLE) of the PAR model. Let \( \beta = (\beta_0, \beta_1, \ldots, \beta_q) \), and \( x_t = (1, y_{t-1}, \ldots, y_{t-q}) \), then the log-likelihood function can be expressed as

\[
\log L(\beta) = \sum_{t=1}^{T} \left\{ y_t x_t' \beta - \exp(x_t' \beta) - \ln y_t! \right\}.
\]

(3.2)

According to (3.2), the first derivative of log-likelihood with respect to \( \beta \) is

\[
\frac{\partial \log L(\beta)}{\partial \beta} = \sum_{t=1}^{T} (y_t - \exp(x_t' \beta)) x_t,
\]

(3.3)

and the objective is to find the Poisson MLE \( \hat{\beta}_{MLE} \) such that (3.3) to be zero, i.e.

\[
\sum_{t=1}^{T} \left( y_t - \exp(x_t' \hat{\beta}_{MLE}) \right) x_t = 0.
\]

Moreover, the covariance matrix of \( \hat{\beta}_{MLE} \) is

\[
\text{Cov}(\hat{\beta}_{MLE}) = \left( \sum_{t=1}^{T} \lambda_t x_t x_t' \right)^{-1}.
\]

In practice, models in (3.1) pose several difficulties for fitting, since the parameter \( \lambda_t \) has to be non-negative [Fokianos, 2012]. Then an alternative expression of the autoregression
model, by using a logarithmic link function, can be written as

\[ \log(\lambda_t) = \beta_0 + \sum_{i=1}^{q} \beta_i \log(y_{t-i} + 1), \]

(3.4)

where \(|\beta_1|, \ldots, |\beta_q| < 1\) with \(\sum_{i=1}^{q} |\beta_i| < 1\).

4 Integer-Valued Generalised Autoregression Conditional Heteroscedastic

An integer-valued generalised autoregression conditional heteroscedastic (INGARCH) model can be regarded as an extension of the previous PAR model. A process \(y_t\) of full order \(q\) can be expressed as

\[ y_t \sim \text{Poisson}(\lambda_t), \]

\[ \lambda_t = \beta_0 + \sum_{i=1}^{q} \beta_i y_{t-i} + \sum_{j=1}^{p} \alpha_j \lambda_{t-j}, \]

(3.5)

where \(\beta_0 > 0, \beta_i \geq 0\) for \(i = 1, \ldots, q\), \(\alpha_j \geq 0\) and \(j = 1, \ldots, p\). Such a process is referred as INGARCH\((p, q)\) process. Ferland et al. [2006] discussed many properties of INGARCH\((p, q)\) process, one of which is that in order to fulfil the stationary of the process, the coefficients must satisfy necessarily the condition

\[ \sum_{i=1}^{q} \beta_i + \sum_{j=1}^{p} \alpha_j < 1. \]

(3.6)

Frequentists estimate (3.5) by quasi conditional maximum likelihood estimation. In statistics, a maximum likelihood estimate maximises the actual log likelihood function for the data and statistical model. In contrast, a quasi-maximum likelihood estimate is an estimate of a parameter \(\theta\) in a model that is formed by maximising a function that is related to the logarithm of the likelihood function.

According to (3.6), one can easily have the parameter space

\[ \Theta = \{ \theta \in \mathbb{R}^{p+q+1} : \beta_0, \beta_1, \ldots, \beta_q, \alpha_1, \ldots, \alpha_p \geq 0, \sum_{i=1}^{q} \beta_i + \sum_{j=1}^{p} \alpha_j < 1 \}. \]
Then the conditional quasi log-likelihood function can be written as

\[
L(\theta) = \sum_{t=1}^{T} \log p_t(y_t; \theta) \\
= \sum_{t=1}^{T} (y_t \log(\lambda_t(\theta)) - \lambda_t(\theta)),
\]

where \( p_t(y_t; \theta) \) is the probability density function of a Poisson distribution when \( y_t \) taking the value \( y \) given all the historical information. The conditional score function is

\[
S_T(\theta) = \frac{\partial L(\theta)}{\partial \theta} \\
= \sum_{t=1}^{T} \left( \frac{y_t}{\lambda_t(\theta)} - 1 \right) \frac{\partial \lambda_t(\theta)}{\theta}.
\]

The conditional information matrix is

\[
G_T(\theta) = \sum_{t=1}^{T} \text{Cov}(\frac{\partial L(\theta; y_t)}{\partial \theta}) \\
= \sum_{t=1}^{T} \left( \frac{1}{\lambda_t(\theta)} \right) \left( \frac{\partial \lambda_t(\theta)}{\partial \theta} \right) \left( \frac{\partial \lambda_t(\theta)}{\partial \theta} \right)^\top.
\]

The quasi maximum likelihood estimator \( \hat{\theta}_T \) of \( \theta \) is derived by

\[
\hat{\theta} := \hat{\theta}_T = \arg \max_{\theta \in \Theta} L(\theta).
\]

A more general form of GLM can be expressed in the form

\[
g(\lambda_t) = \beta_0 + \sum_{i=1}^{q} \beta_i \tilde{g}(y_{t-i}) + \sum_{j=1}^{p} \alpha_j g(\lambda_{t-j}), \quad (3.7)
\]

where \( g : \mathbb{R}^+ \to \mathbb{R} \) is called as the link function and \( \tilde{g} : \mathbb{N}_0 \to \mathbb{R} \) is a transformation function. It is clear that (3.5) is a special case of (3.7), with identical link function and transformation function.

(3.7) can be regarded as a log-linear model if one takes \( \nu_t = g(\lambda_t) = \log(\lambda_t) \), \( \tilde{g}(y_{t-i}) = \log(y_{t-i} + 1) \), with a new form

\[
\nu_t = \beta_0 + \sum_{i=1}^{q} \beta_i \log(y_{t-i} + 1) + \sum_{j=1}^{p} \alpha_j \nu_{t-j}. \quad (3.8)
\]
The above form of log-linear model was studied by Fokianos and Tjøstheim [2011]; Woodard et al. [2011]; Douc et al. [2013]. According to Liboschik et al. [2017], there is a clear difference between model (3.8) and (3.5). Model (3.8) allows the existence of negative serial correlation, comparing with only positive serial correlation in (3.5). There are also some differences between two models in terms of covariates, which was discussed in Fokianos and Tjøstheim [2011].

The parameter space of log-linear model (3.8) is changed comparing with that of (3.5). The new parameter space is

\[
\Theta = \{ \theta \in \mathbb{R}^{p+q+1} : |\beta_1|, \ldots, |\beta_q|, |\alpha_1|, \ldots, |\alpha_q| < 1, \sum_{i=1}^q |\beta_i| + \sum_{j=1}^p |\alpha_j| < 1 \}.
\] (3.9)

Further discussion can be found in the appendix of Liboschik et al. [2017].

5 Multivariate Time Series of Counts

We have introduced some approaches for modelling univariate time series of counts in the previous sections, while the literature on their multivariate counterparts is rather sparse, and most of the models are enlightened by traditional autoregressive time series models. Franke and Rao Subba [1993]; Latour [1997] firstly introduced the first-order and its \( p \)-th order integer-valued autoregression model. Veraart [2019] presented a framework for multivariate time series of counts that is based on a mixed moving average process driven by Levy noise. Karlis [2016] is an account of some recent developments in multivariate time series of count data.

GLM-based time series models such as Poisson autoregressive and INGARCH models can be applied on modelling time series with covariate effects. Liboschik et al. [2017] suggested a naive improvement on (3.5) to accommodate the effect of covariates, as

\[
y_t \sim \text{Poisson}(\lambda_t), \\
\lambda_t = \beta_0 + \sum_{i=1}^q \beta_i y_{t-i} + \sum_{j=1}^p \alpha_j \lambda_{t-j} + \sum_{m=1}^{M-1} \eta_m x_{m,t},
\] (3.10)

for a \( M \)-dimensional multivariate time series where \( y_t \) is the target time series and the other \( M-1 \) time series are assumed to be the covariates of \( y_t \). \((x_{1,t}, \ldots, x_{M-1,t})\) is the time-varying covariate vector, and \((\eta_1, \ldots, \eta_{M-1})\) is the corresponding parameter vector.
that contains the effects of covariates. In order to avoid negativity for the Poisson rate in (3.10), we introduce the logarithmic transformation as

\[ y_t \sim \text{Poisson}(\lambda_t), \]
\[ \nu_t = \beta_0 + \sum_{i=1}^{q} \beta_i \log(y_{t-i} + 1) + \sum_{j=1}^{p} \alpha_j \nu_{t-j} + \sum_{m=1}^{M-1} \eta_m x_{m,t}, \]  

(3.11)

where \( \nu_t = \log(\lambda_t) \). The parameter space of (3.11) is exactly the one in (3.9).

The details of the constraints can be found in Liboschik et al. [2017], and the model in (3.11) was discussed in past papers including Fokianos [2011]; Fokianos and Tjøstheim [2011]. (3.11) has the advantage over (3.10) when fitting times series with covariates since there is no constraint for covariate parameters, moreover, the logarithmic representation in (3.11) allows negative serial correlation while (3.10) only accepts positive serial correlation.

Throughout the thesis, we consider (3.11) as the representation to fit an \( M \)-dimensional multivariate Poisson autoregressive model of order \( q \) that predicts \( y_{\ell,t} \) with the covariate vector \((y_{1,t-1}, \ldots, y_{-1,t-1}, y_{l+1,t-1}, y_{M,t-1})\) as

\[ y_{\ell,t} \sim \text{Poisson}(\lambda_{\ell,t}), \]
\[ \log(\lambda_{\ell,t}) = \beta_{\ell,0} + \sum_{i=1}^{q} \beta_{\ell,i} \log(y_{\ell,t-i} + 1) + \sum_{m \neq \ell} \eta_m y_{m,t-1}, \]  

(3.12)

where \( \beta_{\ell,0}, \beta_{\ell,i} \) and \( \eta_m \) are unknown parameters. Similarly, we fit an \( M \)-dimensional multivariate INGARCH model as

\[ y_{\ell,t} \sim \text{Poisson}(\lambda_{\ell,t}), \]
\[ \log(\lambda_{\ell,t}) = \beta_{\ell,0} + \sum_{i=1}^{q} \beta_{\ell,i} \log(y_{\ell,t-i} + 1) + \sum_{j=1}^{p} \alpha_{\ell,j} \log(\lambda_{\ell,t}) + \sum_{m \neq \ell} \eta_m y_{m,t-1}, \]  

(3.13)

with more unknown parameters \( \alpha_{\ell,j} \).

6 Software and Package

We use the package tscount [Liboschik et al., 2017] in R for modelling time series of count data through the Poisson autoregressive and INGARCH models. The package analyses time series of count data by utilising likelihood-based estimation methods. The package allows the identical link function and log linear link function. tscount prefers GLM-based
models than the models based on thinning operators because of two reasons, which are that (i) GLM-based models can describe negative correlations in a straightforward way, and (ii) there is a rich toolkit available for this class of models. Functions of \texttt{tscout} are very similar to \texttt{R} functions \texttt{arima} and \texttt{glm}, and inspired by \texttt{MASS} [Venables and Ripley, 2013].

Due to the significant advantages GLM-based models have over the models based on thinning operators mentioned above, we choose PAR and INGARCH as the models to compare with the BTF model on time series of count throughout this thesis.
Chapter 4

Bayesian Tensor Factorisations for Categorical Data

The main purpose of this thesis, which is to model times series of count data by the Bayesian tensor factorisations (BTF) model, is inspired by Sarkar and Dunson [2016], where an approach of Bayesian nonparametrics is used to model time series of categorical data. In this chapter, we introduce tensor factorisations and replicate a previous application of BTF in Sarkar and Dunson [2016] for modelling time series of categorical data.

1 Tensor Factorisations

In a succinct sentence, tensors are multidimensional arrays of numerical values that generalise matrices to multiple dimensions. A tensor factorisation, or a tensor decomposition, is any approach that can express a tensor as a sequence of elementary operations acting on other, often simpler tensors. When mentioning tensor factorisations, we can be reminded of none but the two most popular approaches: the parallel factor analysis (PARAFAC) and higher-order singular value decomposition (HOSVD).

Harshman [1970]; Harshman and Lundy [1994] introduced PARAFAC as an approach to decompose a $D_1 \times \cdots \times D_p$ dimensional tensor $\mathcal{M} = \{m_{x_1,\ldots,x_p}\}$ as the sum of tensors with rank one, that is

$$m_{x_1,\ldots,x_p} = \sum_{h=1}^{k} g_h \prod_{j=1}^{p} u_h^{(j)}(x_j),$$

where $\mathcal{G} = \{g_h\}$ is a core tensor, and $\mathcal{U} = \{u_h^{(j)}(x_j)\}$ are the weights.
De Lathauwer et al. [2000] proposed HOSVD inspired by Tucker [1966] as another approach to decompose $\mathcal{M}$ as

$$m_{x_1,\ldots,x_p} = \sum_{h_1=1}^{k_1} \cdots \sum_{h_p=1}^{k_p} g_{h_1,\ldots,h_p} \prod_{j=1}^{p} u_{h_j}^{(j)}(x_j), \quad (4.2)$$

similarly, where $G = \{g_{h_1,\ldots,h_p}\}$ is a core tensor, and $U = \{u_{h_j}^{(j)}(x_j)\}$ are the weights.

According to the expressions in both (4.1) and (4.2), it is easy to see that an HOSVD is a general form of a PARAFAC where the core tensor $G$ is diagonal. Therefore, an HOSVD can achieve better data compression than a PARAFAC [Sarkar and Dunson, 2016].

In recent years, the applications of BTF are very popular in various interdisciplinary fields. Tang et al. [2018] studied the multidimensional electroencephalography in neuroinformatics through an approach of BTF. Based on the global correlation along a spectrum and non-local self-similarity across space, Wei and Fu [2019] employed the low-rank property in this representation to design a hierarchical probabilistic model based on BTF to capture the inherent spatial-spectral correlation of hyperspectral image. Xiong et al. [2017] proposed a BTF for detecting thermoacoustic instabilities in gas turbine engines in real-time. In the field of machine learning, the usage and applications of tensor factorisations are common these decades. Porteous et al. [2008] introduced a novel generative Bayesian probabilistic model for unsupervised matrix and tensor factorisation, leading to finding low dimensional data representations. For the tasks like multidimensional data analysis in structural toxicogenomics and functional neuroimaging, Khan et al. [2016] introduced Bayesian multi-tensor factorisation. Lower dimensional cases can be found in Hardoon et al. [2004]; Kolda and Bader [2009]. When considering the problem of learning probabilistic models for complex relational structures between various types of objects, Sutskever et al. [2009] proposed the clustered BTF model, that embeds a factorised representation of relations in a nonparametric Bayesian clustering framework. Schein et al. [2015] developed a BTF model, by using dyadic events relating to international political relations, to infer a low-dimensional, interpretable representation of salient patterns from the data. Zhang et al. [2014] proposed a probabilistic BTF model for synthesising a single word vector representation and per-perspective linear transformations from any number of word similarity matrices.
2 Tensor Factorisations on Mixture Transition Distribution

For convenience, we let \( \mathbb{N} \) denote the set of natural numbers and \( \mathbb{Z}_{[i,j]} \) denote the set contains all integers from \( i \) to \( j \).

Sarkar and Dunson [2016] considered a discrete time series \( \{y_t\} \), for \( t \in \mathbb{Z}_{[1,T]} \) with \( C_0 \) categories, where \( T \) is the last observed time, with a transition probability \( p(y_t \mid \{y_{t-j} \}_{j \in \mathbb{Z}_{[1,q]}}) \). Assume that the distribution of \( y_t \) depends only on the previous \( q \) time points, the transition probability can then be rewritten as

\[
p(y_t \mid y_{t-1}, \ldots, y_1) = p(y_t \mid \{y_{t-j} \}_{j \in \mathbb{Z}_{[1,q]}}).
\]

In the above equation, \( q \) is defined as the maximal order, i.e. \( y_{t-q} \) is the most distant lag that \( y_t \) has serial dependence on. If the distribution of \( y_t \) varies with the values of all previous \( q \) consecutive time points, i.e. \( y_t \) has serial dependence on \( \{y_{t-j} \}_{j \in \mathbb{Z}_{[1,q]}} \), then the sequence of Markov chain is of full order \( q \), in other word, \( y_t \) depends on all consecutive \( q \) point in the past. In contrast, \( q = 0 \) corresponds to serial independence, i.e. \( p(y_t \mid \{y_{t-j} \}_{j \in \mathbb{Z}_{[1,q]}}) = p(y_t) \).

We can use a Markov chain to explain high-order dependencies among consecutive observations of random variables. However, in practice, with the increase of possible values of random variables and the order of chain, the number of independent parameters will soar exponentially, and the process will be inefficient and time-costing for modelling. For a chain with \( C_0 \) states, there are \( C_0 - 1 \) parameters in the conditional distribution of \( y_t \), which can potentially vary arbitrarily with every possible combination of the previous variables’ levels. For a Markov chain of maximal order \( q \), there are a total of \( C_0^q \) such combinations, and hence the number of independent parameters in the full model is \( (C_0 - 1)C_0^q \). This number increases exponentially in the order of the chain, making estimation infeasible. Therefore, it is very important to define flexible, parsimonious and interpretable representations. Raftery [1985] introduced a mixture transition distribution (MTD) model for modelling high-order Markov chains, with the conditional probability

\[
p(y_t = i_0 \mid y_{t-q} = i_q, \ldots, y_{t-1} = i_1) = \sum_{k=1}^{q} \lambda_k \pi_{i_k i_0}, \tag{4.3}
\]

where \( \pi_{i_k i_0} \) represents the probabilities of a transition matrix with \( C_0 \times C_0 \) dimensions, and \( \lambda_k \) is the weight parameter associated with lag \( k \). It is obvious that in (4.3) there are
only $C_0(C_0 - 1) + (q - 1)$ independent parameters, and it is a huge reduction of a Markov chain full model, which has $(C_0 - 1)C_0^q$ independent parameters.

Lim [2005] defined a $q$-array of real numbers representing an order-$q$ tensor as $A = \{a_{c_1 \ldots c_q}\} \in \mathbb{R}^{d_1 \times \ldots \times d_q}$, and the covariant multi-linear matrix multiplication of $A$ by matrices $M_1 = \{m_{c_1t_1}^{(1)}\} \in \mathbb{R}^{d_1 \times s_1}, \ldots, M_q = \{m_{c_qt_q}^{(q)}\} \in \mathbb{R}^{d_q \times s_q}$ as

$$A(M_1, \ldots, M_q) := \left\{ \sum_{c_1=1}^{d_1} \ldots \sum_{c_q=1}^{d_q} a_{c_1 \ldots c_q} m_{c_1t_1}^{(1)} \ldots m_{c_qt_q}^{(q)} \right\} \in \mathbb{N}^{s_1 \times \ldots \times s_q}.$$

Let $S_r^{-1} = \{x \in \mathbb{R}^r : x_j \geq 0, \sum_{j=1}^r x_j = 1\}$ denote the $(r - 1)$-dimensional probability simplex, the joint probability distribution of $y_1, \ldots, y_q$ can be expressed as a probabilistic PARAFAC factorisation as

$$p(y_1 = c_1, \ldots, y_q = c_q) = k_1 \sum_{h_1=1}^{k_1} \cdots k_q \sum_{h_q=1}^{k_q} g_{h_1} \prod_{j=1}^{q} u^{(j)}_{h_j}(y_j = c_j),$$

where $(g_1, \ldots, g_k)'$ is a vector of mixture probabilities and

$$\{u^{(j)}_{h_1}(y_j = 1), \ldots, u^{(j)}_{h_q}(y_j = d_j)\} \in S_{d_j-1}$$

is a vector of mixture probabilities of in the $h$-th component.

A probabilistic HOSVD factorisation for the above joint probability distribution can be expressed as

$$p(y_1 = c_1, \ldots, y_q = c_q) = k_1 \sum_{h_1=1}^{k_1} \cdots k_q \sum_{h_q=1}^{k_q} \lambda_{h_1, \ldots, h_q}(y_1) \prod_{j=1}^{q} \pi^{(j)}_{h_j}(y_{j-}) \pi^{(j)}_{h_j}(y_j),$$

3 Model Specification for Categorical Data

When considering a problem with regressing a categorical response variable $y_t \in \mathbb{Z}_{[1,C_0]}$, on categorical predictors $y_{t-j} \in \mathbb{Z}_{[1,C_0]}$, where $j \in \mathbb{Z}_{[1,q]}$, we can build the conditional probabilities $p(y_t \mid \{y_{t-j}\}_{j \in \mathbb{Z}_{[1,q]}})$ as the elements of a $C_0 \times C_0 \times \cdots \times C_0$ dimensional tensor, which is shown in Yang and Dunson [2016]; Sarkar and Dunson [2016], as

$$p(y_t \mid \{y_{t-j}\}_{j \in \mathbb{Z}_{[1,q]}}) = k_1 \sum_{h_1=1}^{k_1} \cdots k_q \sum_{h_q=1}^{k_q} \lambda_{h_1, \ldots, h_q}(y_t) \prod_{j=1}^{q} \pi^{(j)}_{h_j}(y_{t-j}).$$
for $t \in \mathbb{Z}_{[q+1,T]}$, where $k_j \in \mathbb{Z}_{[1,C_0]}$ for all $j \in \mathbb{Z}_{[1,q]}$ and the parameters $\lambda_{h_1,\ldots,h_q}(y_t)$ and $\pi_{h_j}^{(j)}(y_{t-j})$ are all non-negative and satisfy the constraints

$$\sum_{y_t=1}^{C_0} \lambda_{h_1,\ldots,h_q}(y_t) = 1, \text{ for each combination } (h_1,\ldots,h_q),$$

$$\sum_{h_j=1}^{k_j} \pi_{h_j}^{(j)}(y_{t-j}) = 1, \text{ for each pair } (j, y_{t-j}).$$

Example 3.1. We illustrate the above HOSVD model in an example. Assume there are only two important lags, i.e. $q = 2$, for the response $y_t$ taking categorical values in the set $\{1,2,3\}$, i.e. $C_0 = 3$. Moreover, we take $k_1 = k_2 = C_0 = 3$; then, (4.4) can be rewritten as

$$p(y_t \mid y_{t-1}, y_{t-2}) = \sum_{h_1=1}^{k_1=3} \sum_{h_2=1}^{k_2=3} \lambda_{h_1,h_2}(y_t) \prod_{j=1}^{q=2} \pi_{h_j}^{(j)}(y_{t-j})$$

$$= \lambda_{11}(y_t)\pi_1^{(1)}(y_{t-1})\pi_1^{(2)}(y_{t-2}) + \lambda_{12}(y_t)\pi_1^{(1)}(y_{t-1})\pi_2^{(2)}(y_{t-2})$$

$$+ \lambda_{13}(y_t)\pi_2^{(1)}(y_{t-1})\pi_1^{(2)}(y_{t-2}) + \lambda_{21}(y_t)\pi_2^{(1)}(y_{t-1})\pi_1^{(2)}(y_{t-2})$$

$$+ \lambda_{22}(y_t)\pi_2^{(1)}(y_{t-1})\pi_2^{(2)}(y_{t-2}) + \lambda_{23}(y_t)\pi_2^{(1)}(y_{t-1})\pi_3^{(2)}(y_{t-2})$$

$$+ \lambda_{31}(y_t)\pi_3^{(1)}(y_{t-1})\pi_1^{(2)}(y_{t-2}) + \lambda_{32}(y_t)\pi_3^{(1)}(y_{t-1})\pi_2^{(2)}(y_{t-2})$$

$$+ \lambda_{33}(y_t)\pi_3^{(1)}(y_{t-1})\pi_3^{(2)}(y_{t-2}).$$

and with the constraints

$$\sum_{y_t=1}^{C_0=3} \lambda_{h_1,h_2}(y_t) = 1, \text{ for each combination } (h_1, h_2), \text{ with } h_1 \in \mathbb{Z}_{[1,3]}, h_2 \in \mathbb{Z}_{[1,3]};$$

$$\sum_{h_1=1}^{k_1=3} \pi_{h_1}^{(1)}(y_{t-1}) = 1, \sum_{h_2=1}^{k_2=3} \pi_{h_2}^{(2)}(y_{t-2}) = 1, \text{ for } y_{t-1} \in \mathbb{Z}_{[1,3]}, y_{t-2} \in \mathbb{Z}_{[1,3]}.$$

The factorisation in (4.4) is straightforward, but still complex, therefore, latent allocation variables are introduced to simplify the factorisation. With latent allocation variables $z_{j,t}$, the response values are conditionally independent, and then, the factorisation can be equivalently represented through the following hierarchical formulation:

$$y_t \mid z_{j,t} = h_j \sim \text{Multinomial}\left(\{1,\ldots,C_0\}, \{\lambda_{h_1,\ldots,h_q}(1),\ldots,\lambda_{h_1,\ldots,h_q}(C_0)\}\right), \quad (4.5)$$

$$z_{j,t} \mid y_{t-j} \sim \text{Multinomial}\left(\{1,\ldots,k_j\}, \{\pi_1^{(j)}(y_{t-j}),\ldots,\pi_{k_j}^{(j)}(y_{t-j})\}\right), \quad (4.6)$$

for $t \in \mathbb{Z}_{[q+1,T]}$, where $j \in \mathbb{Z}_{[1,q]}$. Note that $\text{Multinomial}(\{\mathcal{C}\}, \{\mathcal{p}\})$ represents a multinomial
distribution with \( \mathcal{C} \) to be the set of values it takes, and \( \mathbf{p} \) to be the corresponding probability density function.

(4.5) and (4.6) are equivalent to (4.4). To see this, note that

\[
p(y_t \mid \{y_{t-j} \}_{j \in \mathbb{Z}_{1,q}}) = \sum_{\tilde{z}} p(y_t \mid \{z_{j,t} \}_{j \in \mathbb{Z}_{1,q}}, \{y_{t-j} \}_{j \in \mathbb{Z}_{1,q}}) p(\{z_{j,t} \}_{j \in \mathbb{Z}_{1,q}} \mid \{y_{t-j} \}_{j \in \mathbb{Z}_{1,q}}),
\]

where \( \tilde{z} \) represents all values \( \{z_{j,t} \}_{j \in \mathbb{Z}_{1,q}} \) can take. Since \( y_t \) is conditionally independent of \( \{y_{t-j} \}_{j \in \mathbb{Z}_{1,q}} \) given \( \{z_{j,t} \}_{j \in \mathbb{Z}_{1,q}} \), we have

\[
p(y_t \mid \{y_{t-j} \}_{j \in \mathbb{Z}_{1,q}}) = \sum_{\tilde{z}} p(y_t \mid \{z_{j,t} \}_{j \in \mathbb{Z}_{1,q}}) p(\{z_{j,t} \}_{j \in \mathbb{Z}_{1,q}} \mid \{y_{t-j} \}_{j \in \mathbb{Z}_{1,q}})
= \sum_{\tilde{z}} p(y_t \mid \{z_{j,t} \}_{j \in \mathbb{Z}_{1,q}}) \prod_{j=1}^q p(z_{j,t} \mid y_{t-j}),
\]

since \( z_{j,t} \) is only dependent on \( y_{t-j} \) for \( j \in \mathbb{Z}_{1,q} \).

Throughout this chapter, we set \( \lambda_{h_1, \ldots, h_q} = \{\lambda_{h_1, \ldots, h_q}(1), \ldots, \lambda_{h_1, \ldots, h_q}(C_0)\} \) to be the vectors that constitute the core tensor. In a more parsimonious representation, we let the kernels \( \lambda_{h_1, \ldots, h_q} \) be shared amongst the label combinations \( (h_1, \ldots, h_q) \) through probabilistic clustering. The DP prior, derived by stick-breaking scheme, of our model can be represented as

\[
\lambda_{h_1, \ldots, h_q} \sim \sum_{l=1}^{\infty} \pi_l^* \delta_{\lambda_l^*}, \text{ independent for each } (h_1, \ldots, h_q),
\]

\[
\lambda_l^* = \{\lambda_l^*(1), \ldots, \lambda_l^*(C_0)\} \sim \text{Dirichlet}(\alpha, \ldots, \alpha), \text{ independent for } l = 1, \ldots, \infty, \\
\pi_l^* = V_l \prod_{m=1}^{l-1} (1 - V_m), \quad V_l \sim \text{Beta}(1, \alpha_0), \text{ independent for } l = 1, \ldots, \infty,
\]

given that \( \delta_{\lambda_l^*} \) is a Dirac function concentrating on \( \lambda_l^* \), and \( \pi_l^* \) is a function constructed by stick-breaking process for each \( l \). The DP prior theoretically allows infinite components, and the number of components occupied by \( \prod_{j=1}^q k_j \) mixture kernels is finite and likely much smaller than \( \prod_{j=1}^q k_j \), leading to a significant reduction in the effective number of parameters of the model.

Then we introduce latent variables \( z_t^* \) and \( z_{h_1, \ldots, h_q}^* \), for \( t \in \mathbb{Z}_{[q+1,T]}, h_j \in \mathbb{Z}_{[1,k_j]} \) and \( j \in \mathbb{Z}_{[1,q]} \),
so that we can convert (4.5) into the following representations

\[ p(z^*_1, \ldots, z^*_q = l) = \pi^*_l, \quad \text{independently for each } (h_1, \ldots, h_q), \]
\[ \left( \lambda_{h_1, \ldots, h_q} \mid z^*_1, \ldots, z^*_q = l \right) = \lambda^*_l, \]
\[ \left( z^*_t \mid \{ z_{j,t} \}_{j \in \mathbb{Z}_{[1,q]}} = \{ h_j \}_{j \in \mathbb{Z}_{[1,q]}}, \right) = z^*_h, \]
\[ (y_t \mid z^*_t = l) \sim \text{Multinomial} \left( \{ 1, \ldots, C_0 \}, \{ \lambda^*_l(1), \ldots, \lambda^*_l(C_0) \} \right). \]

And for (4.6), we assign independent priors on \( \pi^j(y_{t-j}) \) as

\[ \pi^j(y_{t-j}) = \{ \pi^j_1(y_{t-j}), \ldots, \pi^j_k(y_{t-j}) \} \sim \text{Dir}(\gamma_j, \ldots, \gamma_j), \quad (4.8) \]

where \( \gamma_j \) is the rate for Dirichlet distribution. We set \( \gamma_j = 1/C_0 \).

Finally, the model is completed by assigning priors on \( k_j \), as

\[ p(k_j = \kappa) = p_{0,j}(\kappa) \propto \exp(-\varphi j \kappa), \]

where \( j \in \mathbb{Z}_{[1,q]}, \kappa \in \mathbb{Z}_{[1,c_0]} \). Notice that \( \varphi \) controls \( p(k_j = \kappa) \) varying from a uniform distribution when \( \varphi = 0 \), to some exponential distribution as \( \varphi \neq 0 \), and thus, the number of important lags for the proposed conditional tensor factorisation; for all our experiments throughout this chapter, we set \( \varphi = 0.5 \). We set \( \alpha_0 = 1 \) for the Beta prior of \( V_l, \alpha = 1/C_0 \) for the Dirichlet prior in (4.7). Finally, we truncate the series (4.7), by assuming

\[ \lambda_{h_1, \ldots, h_q} \sim \sum_{l=1}^{L} \pi^*_l \delta(\lambda^*_l), \]

and we set \( L = 100 \).

4 Estimation and Inference

Even though the expression of the probability distribution after the Dirichlet process seems to be more complicated than what it was, it becomes straightforward for Bayesian approaches and inference.

We define variables in the BTF model as follows. \( z^* := \{ z^*_1, \ldots, z^*_q \} \) for all the combinations of \( (h_1, \ldots, h_q) \), \( K := \{ k_j \}_{j \in \mathbb{Z}_{[1,q]}}, y := \{ y_t \}_{t \in \mathbb{Z}_{[q+1,T]}}, w := \{ \{ y_{t-j} \}_{j \in \mathbb{Z}_{[1,q]}} \}_{t \in \mathbb{Z}_{[q+1,T]}}, z := \{ \{ z_{j,t} \}_{j \in \mathbb{Z}_{[1,q]}} \}_{t \in \mathbb{Z}_{[q+1,T]}}, \lambda^* := \{ \lambda^*_1 \}_{t \in \mathbb{Z}_{[1,L]}}, \pi^* := \{ \pi^*_l \}_{l \in \mathbb{Z}_{[1,L]}}, \) and \( \pi_K := \{ \{ \pi^j(\omega) \}_{\omega=1}^{C_0} \}_{j \in \mathbb{Z}_{[1,q]}}, \)

\[ \mathcal{L}_{h_1, \ldots, h_q} \sim \sum_{l=1}^{L} \pi^*_l \delta(\lambda^*_l), \]

and we set \( L = 100 \).
The joint density can be expressed as $p(y, z, z^*, w, \lambda^*, \pi_0, \pi_K)$, and in detail,

$$p(y, z, z^*, w, \lambda^*, \pi_0, \pi_K) = p(y, z, z^* \mid w, \lambda^*, \pi_0, \pi_K)p(w, \lambda^*, \pi_0, \pi_K)$$

$$= p(y \mid z, z^*, w, \lambda^*, \pi_0, \pi_K)p(z \mid z^*, w, \lambda^*, \pi_0, \pi_K)p(z^* \mid w, \lambda^*, \pi_0, \pi_K)p(w, \lambda^*, \pi_0, \pi_K).$$

According to the joint density, we have the conditional independence structure, and the full conditionals are as follows.

$$p(y \mid z, z^*, w, \lambda^*, \pi_0, \pi_K) \propto p(y \mid z, z^*, \lambda^*),$$

$$p(z \mid y, z^*, w, \lambda^*, \pi_0, \pi_K) \propto p(z \mid y, z^*, \lambda^*)p(z \mid \pi_0),$$

$$p(z^* \mid y, z, w, \lambda^*, \pi_0, \pi_K) \propto p(z^* \mid y, z, \lambda^*)p(z^* \mid \pi_0),$$

$$p(\lambda^* \mid y, z, z^*, w, \pi_0, \pi_K) \propto p(\lambda^* \mid z, z^*, \lambda^*)p(\lambda^*),$$

$$p(\pi_0 \mid y, z, z^*, w, \lambda^*, \pi_K) \propto p(z^* \mid \pi_0)p(\pi_0),$$

$$p(\pi_K \mid y, z, z^*, w, \lambda^*, \pi_0) \propto p(z \mid w, \pi_K)p(\pi_K).$$

### 4.1 First Stage – Estimation of $K$

**Posterior of $K$**

Note that there is still a conditional of variable missing in the above representations, which is that of $K := \{k_j\}_{j \in \mathbb{Z}_{1, q}}$. Concentration on $K$ is an innovative key developed by Sarkar and Dunson [2016], where states that in the situations $\gamma_j$ in (4.8) independent on $k_j$. By defining $z_j := \{z_{t, j}\}_{t \in \mathbb{Z}_{0+1, T}}$ and $w_j := \{w_{t-j}\}_{t \in \mathbb{Z}_{0+1, T}}$, one can integrate out $\pi^{(l)}(y_{t-j})$ to represent $p(k_j \mid z_j, w_j)$ in a closed-form expression, which is, for $k_j = \max(z_j), \ldots, C_0$,

$$p(k_j \mid z_j, w_j) \propto p_0(k_j) \prod_{j=1}^{C_0} \frac{\Gamma(k_j \gamma_j)}{\Gamma(k_j \gamma_j + n_{j, \omega})} \quad \text{(4.9)}$$

with $n_{j, \omega} = \sum_{t \in \mathbb{Z}_{0+1, T}} \mathbb{I}\{y_{t-j} = \omega\}$, for $j \in \mathbb{Z}_{1, q}$

When $n_{j, \omega}$ is large, we can use Stirling’s approximation $\Gamma(n + a)/\Gamma(n) \approx n^a$ [Robbins, 1955; Feller, 2008] for (4.9) as

$$p(k_j \mid z_j, w_j) \approx \left( \sum_{l=\max(z_j)}^{C_0} \frac{p_{0, j}(l)}{p_0(k_j)} \prod_{j=1}^{C_0} \frac{n_{j, \omega}^{(k_j-l)\gamma_j}}{n_{j, \omega}^{(k_j-l)\gamma_j}} \right)^{-1}.$$
Then the conditional of $K$ is

$$p(K | \ldots) = \prod_{j=1}^{q} p(k_j | z_j, w_j).$$

We use the value of $k_j$ to infer whether the $j$-th lag is important or not, for any $j \in Z_{1,q}$, if $k_j = 1$, the $j$-th lag is not selected as important; else the $j$-th lag is selected as important.

Note that the dimensions of random variables $z^*$ and $\pi_K$ vary with $K$, which makes the design of efficient MCMC algorithms infeasible. In order to facilitate the efficiency, we adopt a two-stage sampler in Yang and Dunson [2016]. The first stage is to estimate the $K$, referred to as the approximate sampler in this thesis, and then sample the other random variables while keeping the $K$ fixed in the second stage.

### Approximate Sampler

The levels of $y_{t-j}$ are partitioned into $k_j$ clusters $\{C_{j,r} : r \in Z_{1,k_j}\}$ with each cluster $C_{j,r}$ assumed to correspond to its own latent class $h_j = r$. With independent Dirichlet priors on the mixture kernels $\lambda_{h_1,\ldots,h_q} \sim \text{Dirichlet}(\alpha, \ldots, \alpha)$ marginalised out, the likelihood conditional on the cluster configuration $C = \{C_{j,r} : j \in Z_{1,q}, r \in Z_{1,k_j}\}$ is given by

$$p(\{y_t\}_{t \in Z_{q+1,T}} | C) = \prod_{(h_1, \ldots, h_q)} \frac{\text{Beta}\{\alpha + n_{h_1,\ldots,h_q}(1), \ldots, \alpha + n_{h_1,\ldots,h_q}(C_0)\}}{\text{Beta}(\alpha, \ldots, \alpha)},$$

where $n_{h_1,\ldots,h_q}(\omega) = \sum_{t \in Z_{q+1,T}} \mathbb{1}\{y_t = \omega, y_{t-1} \in C_1, \ldots, y_{t-q} \in C_q, h_q\}$. Then we do the following for $j \in Z_{1,q}$.

1. If $1 \leq k_j \leq C_0$, we propose to either increase $k_j$ to $(k_j + 1)$ or decrease $k_j$ to $(k_j - 1)$.

2. If an increase move is proposed, we randomly split a cluster of $y_{t-j}$ into two clusters. We accept this move with acceptance rate based on the approximated marginal likelihood.

3. If a decrease move is proposed, we randomly merge two clusters of $y_{t-j}$ into a single cluster. We accept this move with acceptance rate based on the approximated marginal likelihood.

Take $K^*$ and $C^*$ are the updated model index and cluster, and denote $\alpha(\cdot; \cdot)$ as the Metropolis-Hastings acceptance rate, $L(\cdot)$ as the likelihood function and $q(\cdot \to \cdot)$ as the
proposal function, we can have
\[
\alpha(K,C;K^*,C^*) = \frac{L(\{y_{t}\}_{t\in[1,T]};K^*,C^*)q(K^*,C^* \to K,C)}{L(\{y_{t}\}_{t\in[1,T]};K,C)q(K,C \to K^*,C^*)}.
\]

4.2 Second Stage – Posteriors of Random Variables

This section contains the information of all the other random variables given the estimated \(K\). The computations of posteriors can be found in the appendix. Note that \(p(y \mid z,z^*,\lambda^*)\) follows the factorisation
\[
p(y \mid z,z^*,\lambda^*) = \prod_{t \in Z_{[q+1,T]}} p(y_t \mid \lambda^*_{z_{1,t},...,z_{q,t}}(C_0)).
\]

1 Sample \(z_{h_1,...,h_q}^*\) for each \((h_1,...,h_q)\) from
\[
p(z_{h_1,...,h_q}^* = l \mid \ldots) \propto \pi_i^* \prod_{\omega=1}^{C_0} \{\lambda_i^*(\omega)\}^{n_{h_1,...,h_q}(\omega)}.
\]
where \(n_{h_1,...,h_q}(\omega) = \sum_{t \in Z_{[q+1,T]}} I\{z_{1,t} = h_1, \ldots, z_{q,t} = h_q, y_t = \omega\}\).

2 Sample \(V_l\) for \(l \in Z_{[1,L]}\) from
\[
V_l \mid \ldots \sim \text{Beta}(1 + n_{l}^*, \alpha_0 + \sum_{k \geq l} n_{k}^*),
\]
where \(n_{l}^* = \sum_{(h_1,...,h_q)} I\{z_{h_1,...,h_q}^* = l\}\), and update \(\pi_i^*\) accordingly.

3 Sample \(\lambda_i^*(1), \ldots, \lambda_i^*(C_0)\) with \(l \in Z_{[1,L]}\) from
\[
\{\lambda_i^*(1), \ldots, \lambda_i^*(C_0)\} \mid \ldots \sim \text{Dirichlet}\{\alpha + n_{l}(1), \ldots, \alpha + n_{l}(C_0)\},
\]
where \(n_{l}(\omega) = \sum_{(h_1,...,h_q)} I\{z_{h_1,...,h_q}^* = l\}n_{h_1,...,h_q}(\omega)\).

4 For \(j \in Z_{[1,q]}\) and \(\omega \in Z_{[1,C_0]}\), sample
\[
\{\pi_{1,j}(\omega), \ldots, \pi_{k_j,j}(\omega)\} \mid \ldots \sim \text{Dirichlet}\{\gamma_j + n_{j,\omega}(1), \ldots, \gamma_j + n_{j,\omega}(k_j)\},
\]
where \( n_{j,\omega}(h_j) = \sum_{t \in \mathbb{Z}_{[q+1,T]}} 1\{z_{j,t} = h_j, y_{t-j} = \omega\}\).

5 Sample \( z_{j,t} \) for \( j \in \mathbb{Z}_{[1,q]} \) and \( t \in \mathbb{Z}_{[q+1,T]} \) from

\[
p(z_{j,t} = h | z_{j',t} = h_{j'}, j' \neq j, \ldots) \propto \pi_h^{(j)}(y_{t-j}) \lambda_{h_1,\ldots,h_{j-1},h,h_{j+1},\ldots,h_q}(y_t).
\]

5 Comparisons

Competitors and Criterion

Following Sarkar and Dunson [2016], we select the Variable Length Markov Chain and Random Forest models to compare the BTF model on predicting time series of categorical data.

A Variable Length Markov Chain (VLMC) model has attracted interest in information theory and machine learning, and it is a popular strategy to model higher-order Markov chains which are based on trees with conditioning sequences of different lengths as nodes and leaves. Stationary VLMC model for categorical data on a finite space was studied in Bühlmann et al. [1999].

Suppose a stationary process \((Y_t)_{t \in \mathbb{Z}}\) with finite categorical space \(Y\) and with finite order \(K\). Denote by \(y_i^t = (y_j, y_{j-1}, \ldots, y_i)\), with \(i \leq j\), and \(i, j\) are integers with infinity range. Denote by \(c\): a function by \(Y_{\infty}^0 \rightarrow Y_{\infty}^0\), which maps \(c: y_{0,\infty}^0 \mapsto y_{0,\infty}^0\), where

\[
S = S(y_{0,\infty}^0) = \min\{K; p(Y_1 = y_1 | Y_{-\infty}^0 = y_{-\infty}^0) = p(Y_1 = y_1 | Y_{-K+1}^0 = y_{-K+1}^0)\}
\]

for all \(y_1 \in Y\). Note that \(S \equiv 0\) indicates there is no dependence. Then let \(0 \leq K \leq \infty\) be the smallest integer such that

\[
|c(y_{0,\infty}^0)| = S(y_{0,\infty}^0) \leq K
\]

for all \(y_{-\infty}^0 \in Y^\infty\). \(K\) is called the order of the context function \(c(\cdot)\), and if \(K < \infty\), \((Y_t)_{t \in \mathbb{Z}}\) is called a stationary VLMC of order finite \(K\). Rissanen [1983] provided a tree structured context algorithm to fit a VLMC. More details about VLMC including the methodology, computing and software can be found in Mächler and Bühlmann [2004].

A random forest based model that, like VLMC, is also tree-based but, unlike VLMC, does not enforce a strict top-down search. According to Breiman [2001], suppose there is an ensemble of \(k\) classifiers \(h_1(\cdot), h_2(\cdot), \ldots, h_k(\cdot)\), with the training set with random from the
distribution of random vector $X$ and $Y$, and for any $y$ in $Y$, then the marginal function for a random forest can be defined as

$$mr(X, y) = \text{av}_k(\text{h}_k(X) = y) - \max_{j \neq y} \text{av}_k(\text{h}_k(X) = j),$$

where $mr(,)$, $I(,)$ represent the marginal function and indicator function, respectively, and $av_k$ is the average number of votes at $X, y$ for each classifier.

Now, we set $h_k(X) = h(X, \Theta_k)$. If there is a set of probability zero on the sequence space $\Theta_1, \Theta_2, \ldots$, so for outside of the set of probability zero, for all $X$,

$$\frac{1}{N} \sum_{n=1}^{N} I(h(\Theta_n, X) = j) \rightarrow P(\Theta)(h(\Theta, X) = j),$$

then, we can rewrite the marginal function for a random forest as

$$mr(X, y) = P(\Theta)(h(\Theta, X) = y) - \max_{j \neq y} P(\Theta)(h(\Theta, X) = j),$$

and the strength of the set of classifiers $\{h(X, \Theta)\}$ is $E_{X,y}mr(X, y)$.

Denote by

$$\hat{j}(X, y) = \arg \max_{j \neq y} P(\Theta)(h(\Theta, X) = j),$$

the marginal function for a random forest is

$$mr(X, y) = P(\Theta)(h(\Theta, X) = y) - P(\Theta)(h(\Theta, X) = \hat{j}(X, y))$$

$$= E_{\Theta} [I(h(\Theta, X) = y) - I(h(\Theta, X) = \hat{j}(X, y))].$$

Breiman [2001] further introduced the structure and estimation of a random forest model, and Liaw et al. [2002] have developed a software in R.

**Definition 5.1 (Classification error rate).** The mathematical definition of classification error rate (CER) can be expressed as

$$1 - \left( \sum_{t \in T'} I(\hat{y}_t = y_t) \right) / T',$$

where $T'$ and $T'$ denote the testing dataset and the size of it respectively. $\hat{y}_t$ is the estimator by each model at time $t$ in the testing dataset.

**Settings for Models**
The assumed most distant important lag $q$ is set to be the actual most distant important lag plus two throughout simulation experiments, as in Sarkar and Dunson [2016]. For example, if the most distant lag that $y_t$ depends on is $y_{t-8}$, then $q$ is set to be 10.

All MCMC runs for the BTF model are based on the following burn-in and posterior samples respectively: For the first stage of the BTF model, 1,000 and 2,000 for selecting the important lags and their corresponding number of inclusions; and, for the second stage, 2,000 and 5,000 for sampling the rest of the parameters. The thinning rate for the last stage is set to be 10.


**Details of Datasets**

We design seven scenarios with different numbers of categories and important lags for model comparison. For each scenario, we generate 10 datasets with 2,000 data points each, by separating the first 1,600 data points as the training set and the last 400 data as the testing set. $C_0, \{i_1, \ldots\}$ denotes the number of categories with important lags. We generate all the datasets as follows. Suppose there are $v$ important lags $\{y_{i_1}, \ldots, y_{i_v}\}$, if $\sum_{j=1}^{v} y_{i_j}$ is even, then $y_t \sim \text{Multi}(\pi_e)$; else $y_t \sim \text{Multi}(\pi_o)$. We fix

$$
\pi_o = (0.02, 0.2, 0.08, 0.7) \quad \text{and} \quad \pi_e = (0.7, 0.02, 0.2, 0.08).
$$

The outcome of comparisons is shown in the following Table 4.1. The variable length Markov chain model has unsatisfactory predictive performance when important lags are distant, whilst the random forest and the BTF model have relatively sound performance. The table indicates that in all scenarios, the BTF model outperforms its competitors in terms of prediction because of the least classification error rate.

**MCMC Diagnostics**

We use a dataset from the last scenario in the previous part of the comparison. The detail of the time series of 2,000 categorical data points is as follows.

- There are four categories, or $C_0 = 4$, which means that $y_t$ takes the value from $Z_{[1,4]}$;
- Assume the maximum lag is $7+2$, i.e. $q = 9$;
Categories, \{ Important Lags \} & VLMC & Random Forest & BTF \\
4, \{ 1 \} & 0.340 & 0.322 & 0.289 \\
4, \{ 3 \} & 0.353 & 0.330 & 0.303 \\
4, \{ 7 \} & 0.655 & 0.313 & 0.295 \\
4, \{ 10 \} & 0.661 & 0.317 & 0.309 \\
4, \{ 1, 3 \} & 0.347 & 0.364 & 0.343 \\
4, \{ 2, 5 \} & 0.485 & 0.350 & 0.337 \\
4, \{ 4, 7 \} & 0.653 & 0.346 & 0.314 \\

Table 4.1: Mean classification error rate among variable length Markov chain model (VLMC), random forest model and Bayesian tensor factorisations model (BTF) based on 10 generated data sets for each one of seven scenarios. Length of training data against testing data: 1,600 : 400. Models with best performance are highlighted bold.

- \( y_t \) only depends on the fourth and seventh lags ahead i.e. \( y_{t-4} \) and \( y_{t-7} \);
- If \( y_{t-4} + y_{t-7} \) is odd, we generate \( y_t \) by the probability \( p_o \), where \( p_o = (0.02, 0.2, 0.08, 0.7) \). Otherwise, generate \( y_t \) by the probability \( p_e \), where \( p_e = (0.7, 0.02, 0.2, 0.08) \).
- Generate the first ten data points randomly taking the integers from 1 to \( C_0 \).

The MCMC diagnostics for the dataset is illustrated in Figure 4.1. Panel (a) describes the relative frequency distribution (RFD) of how many important lags are there for the BTF according to the training dataset, and there are two important lags indicated. Panel (b) illustrates which lags are selected as important, also in terms of RFD, and it shows that the fourth and seventh lags, \( y_{t-4} \) and \( y_{t-7} \), are not independent to \( y_t \). Both (a) and (b) show perfect convergence of MCMC, with RDFs of both to be 1. The rest six panels in Figure 4.1 illustrate the MCMC diagnostics for conditional probability given by the BTF model. The titles of these six panels are in the format of \( 'p(y_t \mid y_{t-4}, y_{t-7})' \). For example, the panel with title \( 'p(1 \mid 1, 1)' \) shows the MCMC diagnostics for the conditional probability of \( y_t \) taking the category 1 by given \( y_{t-4} \) to be 1 and \( y_{t-7} \) to be 1. The true conditional probabilities for these panels are 0.7, 0.08, 0.02, 0.2, 0.7 and 0.2 respectively. In these six panels, the cyan line represents the trace plot of conditional probability in each case. The blue solid lines in the plots are the running mean and the running mean of the \( s \)-th sampler after burn-in. The black dash lines are the 5% and 95% running quantiles for each case. According to these trace plots for the MCMC, the running means and quantiles are very stable and showing convergence in all of the plots. Moreover, there is an agreement between true conditional values and running means.
Figure 4.1: Results of simulation experiments for the case with $C_0 = 4$ categories, true important lags \{4, 7\}, and sample size to be 2,000 for the dataset with the classification error rate. (a) Relative frequency distribution of how many important lags in the model; (b) Inclusion proportions of different lags; The rest six panels are the predictive conditional probabilities for different cases labelled in titles. In each panel, the cyan line shows the trace plot of conditional probability. The blue solid line shows the running mean. The black dash lines show the 5% and 95% running quantiles, respectively. $y$-axis represents the conditional probability and $x$-axis represents the MCMC Samplers after Burn-in.
6 Appendix

Detail of the Gibbs samplers in Section 4.2 are as follows.

- **Step 1**

\[
p(z_{h_1,\ldots,h_q}^* = l \mid \ldots) \propto p(z_{h_1,\ldots,h_q}^* = l)p(y \mid \lambda_{z_{1,t},\ldots,z_{q,t}}^*)
= \pi_l^* \prod_{t \in Z_{[q+1,T]}} \{\lambda_t^*(y_t)\}^1\{z_{1,t} = h_1, \ldots, z_{q,t} = h_q\}
= \pi_l^* \prod_{\omega \in Z_{[1,C_0]} \{\lambda_\omega^*(\omega)\n^{h_1,\ldots,h_q(\omega)}
\]

where

\[
n_{h_1,\ldots,h_q}(\omega) = \sum_{t \in Z_{[q+1,T]}} 1\{z_{1,t} = h_1, \ldots, z_{q,t} = h_q, y_t = \omega\}.
\]

- **Step 2**

\[
p(V_l \mid \ldots) \propto p(V_l)p(z_l^* \mid V_l)
= (V_l)^{1-1}(1 - V_l)^{a_0-1}(V_l)^{n_l^*} (1 - V_l)^{\sum_{k>l} n_k^*}
(V_l \mid \ldots) \sim \text{Beta}(1 + n_l^*, a_0 + \sum_{k>l} n_k^*),
\]

where

\[
n_l^* = \sum_{(h_1,\ldots,h_q)} 1\{z_{h_1,\ldots,h_q}^* = l\}.\]

- **Step 3**

\[
p(\lambda_l^* \mid \ldots) \propto p(\lambda_l^*)p(y \mid \lambda_{z_{1,t},\ldots,z_{q,t}}^*)
= \prod_{\omega \in Z_{[1,C_0]} \{\lambda_\omega^*(\omega)\n^{1-n_l^*(\omega)}
\}
(\lambda_l^* \mid \ldots) \sim \text{Dirichlet}\{\alpha + n_l^*(1), \ldots, \alpha + n_l^*(C_0)\},
\]

where

\[
n_l^*(\omega) = \sum_{(h_1,\ldots,h_q)} 1\{z_{h_1,\ldots,h_q}^* = l\}n_{h_1,\ldots,h_q}(\omega).
\]

- **Step 4**

\[
p(\pi_{1,\ldots,k}^*(\omega), \ldots, \pi_{k,\ldots,k}^*(\omega) \mid \ldots) \propto p(\pi_{1,\ldots,k}^*(\omega), \ldots, \pi_{k,\ldots,k}^*(\omega)) \cdot p(z \mid \pi_{1,\ldots,k}^*(\omega), \ldots, \pi_{k,\ldots,k}^*(\omega))
\]
CHAPTER 4. BAYESIAN TENSOR FACTORISATIONS FOR CATEGORICAL DATA

\[
= \prod_{i=1}^{k_j} \pi_i^{\gamma_j - 1} \prod_{t \in \mathbb{Z}_{[q+1,T]}} \prod_{i=1}^{k_j} \mathbb{1}\{z_{j,t} = h_j, w_{j,t} = \omega\} \\
= \prod_{i=1}^{k_j} \pi_i^{\gamma_j + \sum_{t \in \mathbb{Z}_{[q+1,T]}} \mathbb{1}\{z_{j,t} = h_j, w_{j,t} = \omega\} - 1}
\]

\{\pi_1^{(j)}(\omega), \ldots, \pi_{k_j}^{(j)}(\omega) | \ldots\} \sim \text{Dirichlet}\{\gamma_j + n_{j,\omega}(1), \ldots, \gamma_j + n_{j,\omega}(k_j)\},

where

\[n_{j,\omega}(h_j) = \sum_{t \in \mathbb{Z}_{[q+1,T]}} \mathbb{1}\{z_{j,t} = h_j, w_{j,t} = \omega\}\]

- Step 5

\[p(z_{j,t} = h | \ldots) \propto p(z_{j,t} = h | \pi_h^{(j)}(y_{t-j}))p(y_t | z_{j,t} = h)\]

\[= \pi_h^{(j)}(y_{t-j}) \lambda_{y_1,\ldots,y_{h-1},h,h+1,\ldots,y_q}(y_t).\]
Chapter 5

Bayesian Tensor Factorisations for Univariate Time Series of Counts

In this chapter, we introduce our simple BTF model for dealing with univariate time series of counts. When assuming the range of datasets to be small, our simple BTF can have fair predictive performance against other approaches for modelling time series of count data. The chapter consists of the model specification, the estimation as well as inference, and applications.

1 Model Specification

Consider a one-dimensional time-indexed sequence of random variables with size $T$, $\{y_t\} \in \mathbb{Z}_{[1,T]}$, where $y_t$ takes values in $\mathbb{N}$. Furthermore, we define the filtration of $y_t$ as $\mathcal{F}_t = \{y_{1:t}\}$. We build a probabilistic model by assuming that the transition probability law of the sequence $\{y_t\}$ conditional on $\mathcal{F}_{t-1}$ is that of a Markov chain of maximal order $q$ and $c_0$ states:

$$p(y_t \mid \mathcal{F}_{t-1}) = p(y_t \mid \{y_{t-j}\}_{j \in \mathbb{Z}_{[1,q]}}),$$

for $t \in \mathbb{Z}_{[q+1,T]}$.

This formulation includes the possibility that only a subset of the previous $q$ values affects $y_t$. We follow Sarkar and Dunson [2016] and introduce a series of latent variables. First, a collection $K := \{k_j\}_{j \in \mathbb{Z}_{[1,q]}}$ that determines two things: if $k_j = 1$ then $y_{t-j}$ does not affect
\( y_t \); any other value of \( k_j \) determines how many values of \( y_{t-j} \) affect \( y_t \). Thus, conditional on \( k_j \), the parameter space is reduced to \( \prod_{j \in \mathbb{Z}_{[1,q]}} k_j \). We also define time-dependent random variables \( Z_t := \{ z_{j,t} \}_{j \in \mathbb{Z}_{[1,q]}} \) together with a collection of indices \( H := \{ h_{j} \}_{j \in \mathbb{Z}_{[1,q]}} \). Furthermore, we define \( \mathcal{H} := \{ h_{j} \in \mathbb{Z}_{[1,k_j]}, j \in \mathbb{Z}_{[1,q]} \} \). The connection among \( Z_t \), \( H \) and \( \mathcal{H} \) is that for at any \( t \in \mathbb{Z}_{[q+1,T]} \), \( Z_t \) is sampled with the value \( H \in \mathcal{H} \).

The conditional transition probability law (5.1) can then be written as a Bayesian hierarchical model, for \( j \in \mathbb{Z}_{[1,q]} \), \( H \in \mathcal{H} \) and \( t \in \mathbb{Z}_{[q+1,T]} \), as

\[
\begin{align*}
y_t \mid Z_t &= H \sim \text{Poisson}(\lambda_H), \quad (5.2) \\
z_{j,t} \mid y_{t-j} &\sim \text{Multinomial} \left( \mathbb{Z}_{[1,k_j]}, \left\{ \pi_{h_j}^{(j)}(y_{t-j}) \right\} \right), \quad (5.3)
\end{align*}
\]

(5.2) and (5.3) imply that

\[
p(y_t \mid \{ y_{t-j} \}_{j \in \mathbb{Z}_{[1,q]}}) = \sum_{H \in \mathcal{H}} \text{PD}(y_t; \lambda_H) \prod_{j \in \mathbb{Z}_{[1,q]}} \pi_{h_j}^{(j)}(y_{t-j}). \tag{5.4}
\]

where \( \text{PD}(y, \lambda) \) represents the Poisson density function for random variable \( y \) with Poisson rate \( \lambda \), with constraints

\[
\begin{align*}
\lambda_H &\geq 0, \text{ for any } H \in \mathcal{H}, \\
\sum_{h_{j=1}}^{k_j} \pi_{h_j}^{(j)}(y_{t-j}) &\leq 1, \text{ for each combination of } (j, y_{t-j}).
\end{align*}
\]

The formulation (5.4) is referred as a conditional tensor factorisation with the Poisson density \( \text{PD}(y_t; \lambda_H) \) being the core tensor. It can also be interpreted as a Poisson mixture model with \( \prod_{j \in \mathbb{Z}_{[1,q]}} \pi_{h_j}^{(j)}(y_{t-j}) \) being the mixture weights that depend on previous values of \( y_t \). The original Markov chain in (5.1) indicates that there are as many as \( c_0 \) parameters for deciding \( y_t \), and by using the Bayesian hierarchical model in (5.2) and (5.3), the parameter space is reduced to \( \prod_{j \in \mathbb{Z}_{[1,q]}} k_j \). \( k_j \) is maximal number of clusters that the values of \( y_{t-j} \) can be separated into for predicting \( y_t \). For instance, there is a time series with \( c_0 \) unique values. \( y_t \) depends on \( y_{t-1} \) only, and if \( y_t \sim \text{Poisson}(0) \) given that \( y_{t-1} > 1 \), and \( y_t \sim \text{Poisson}(1) \) given that \( y_{t-1} \leq 1 \), then \( k_1 = 2 \) since the values of \( y_{t-1} \) are separated into two clusters that determine the distribution of \( y_t \). The latent allocation variable \( z_{j,t} \) in (5.3), sampled from \( \mathbb{Z}_{[1,k_j]} \), decides which cluster \( y_{t-j} \) belongs for predicting \( y_t \). In the meanwhile, the structure in (5.2) and (5.3) provides a judgement whether \( y_t \) depends on \( y_{t-j} \). If \( k_j = 1 \),
$z_{j,t}$ can only take the value one, and the distribution of $y_t$ does not vary with $y_{t-j}$.

(5.4) is equivalent to (5.2) and (5.3). To see this, note that

$$p(y_t \mid \{y_{t-j}\}_{j \in \mathbb{Z}_{[1,q]}}) = \sum_{H \in \mathcal{H}} p(y_t \mid Z_t = H, \{y_{t-j}\}_{j \in \mathbb{Z}_{[1,q]}}) p(Z_t = H \mid \{y_{t-j}\}_{j \in \mathbb{Z}_{[1,q]}}).$$

Since $y_t$ is conditionally independent of $\{y_{t-j}\}_{j \in \mathbb{Z}_{[1,q]}}$ given $Z_t = H$, we have

$$p(y_t \mid \{y_{t-j}\}_{j \in \mathbb{Z}_{[1,q]}}) = \sum_{H \in \mathcal{H}} p(y_t \mid Z_t = H) p(Z_t = H \mid \{y_{t-j}\}_{j \in \mathbb{Z}_{[1,q]}})$$

$$= \sum_{H \in \mathcal{H}} p(y_t \mid Z_t = H) \prod_{j \in \mathbb{Z}_{[1,q]}} \pi_{H_j}^{(j)}(y_{t-j})$$

$$= \sum_{H \in \mathcal{H}} \text{PD}(y_t; \lambda_H) \prod_{j \in \mathbb{Z}_{[1,q]}} \pi_{H_j}^{(j)}(y_{t-j}).$$

From (5.4), the expectation of $y_t$ conditional on $\{y_{t-j}\}_{j \in \mathbb{Z}_{[1,q]}}$ is

$$\mathbb{E}(y_t \mid \{y_{t-j}\}_{j \in \mathbb{Z}_{[1,q]}}) = \sum_{H \in \mathcal{H}} \lambda_H \prod_{j \in \mathbb{Z}_{[1,q]}} \pi_{H_j}^{(j)}(y_{t-j}). \quad (5.5)$$

(5.4) represents a mixture of Poisson models into a tensor factorisation model. However, $\prod_{j \in \mathbb{Z}_{[1,q]}} k_j$ can still be large, and the possible number of Poisson rates $\lambda_H$ can be huge, resulting in impractical inference. A more parsimonious representation for our tensor factorisation model is obtained by adopting a Dirichlet process for Poisson rates $\lambda_H$.

Independently, for each $H \in \mathcal{H}$, we use the stick-breaking construction

$$\lambda_H \sim \sum_{i=1}^{\infty} \pi_i \delta(\lambda_i^*), \quad (5.6)$$

where $\delta(.)$ is an indicator function, and independently, for $l \in \mathbb{Z}_{[1,\infty)}$, $\lambda_i^* \sim \text{Gamma}(a, b)$, $\pi_i^* = V_l \prod_{s=1}^{l-1} (1 - V_s)$, $V_l \sim \text{Beta}(1, \alpha_0)$ representing a label-clustered Poisson rate, which is generated by a Gamma distribution. By introducing an additional set of latent variables $\mathbb{Z}_H$ that labels $H$ from $\mathbb{Z}_{[1,L]}$, and $\mathbb{Z}_{Z_t}$ representing the label of $Z_t$ at time $t \in \mathbb{Z}_{[q+1,T]}$, we
complete by model formulation as

\[ p(Z^*_H = l) = \pi^*_l, \quad \text{independently for each } H \in \mathcal{H}, \]

\[ (\lambda_H \mid Z^*_H = l) = \lambda^*_l, \]

\[ (Z^*_{zl} \mid Z_t = H) = Z^*_H, \]

\[ (y_t \mid Z^*_z = l) \sim \text{Poisson}(\lambda^*_l). \]

For \( \omega \in \mathbb{Z}_{[0,c_0-1]} \), we assign independent priors on \( \pi^{(j)}(\omega) \) as

\[ \pi^{(j)}(\omega) = \{\pi_1^{(j)}(\omega), \ldots, \pi_{K_j}^{(j)}(\omega)\} \sim \text{Dirichlet}(\gamma_j, \ldots, \gamma_j). \] (5.7)

with \( \gamma_j = 0.1 \). Also, we follow Sarkar and Dunson [2016] and set priors

\[ p(k_j = \kappa) \propto \exp(-\varphi j \kappa), \] (5.8)

where \( j \in \mathbb{Z}_{[1,q]}, \kappa \in \mathbb{Z}_{[1,c_0]} \). Notice that \( \varphi \) controls \( p(k_j = \kappa) \) varying from a uniform distribution when \( \varphi = 0 \), to some exponential distribution as \( \varphi \neq 0 \), and thus, the number of important lags for the proposed conditional tensor factorisation; for all our experiments throughout this chapter, we set \( \varphi = 0.5 \). Following Viallefont et al. [2002], we place for the Gamma density of \( \lambda^*_l \) parameters \( a \) as the mid-range of \( y_t \) in the training dataset

\[ a = \frac{1}{2} \left[ \max(\{y_t\}_{t \in \mathbb{Z}_{[q+1,T]}}) - \min(\{y_t\}_{t \in \mathbb{Z}_{[q+1,T]}}) \right] \]

and \( b = 1 \). We set \( a_0 = 1 \) for the Beta prior of \( V_t \). Finally, we truncate the series (5.6), by assuming

\[ \lambda_H \sim \sum_{l=1}^{L} \pi^*_l \delta(\lambda^*_l), \]

and we set \( L = 100 \).

## 2 Estimation and Inference

There are quite a few variables in our BTF model. Let

\[ y := \{y_t\}_{t \in \mathbb{Z}_{[q+1,T]}}, \quad W := \{\{y_t-j\}_{j \in \mathbb{Z}_{[1,q]}\}_{t \in \mathbb{Z}_{[q+1,T]}}, \quad Z := \{Z_t\}_{t \in \mathbb{Z}_{[q+1,T]}}, \]

\[ Z^* := \{Z^*_H\}_{H \in \mathcal{H}}, \quad \lambda^* := \{\lambda^*_l\}_{l \in \mathbb{Z}_{[1,L]}}, \quad \pi^* := \{\pi^*_l\}_{l \in \mathbb{Z}_{[1,L]}}, \quad \pi_K := \{\{\pi^{(j)}(\omega)\}_{\omega \in \mathbb{Z}_{[0,c_0-1]}}\}_{j \in \mathbb{Z}_{[1,q]}}, \]
the joint density of the above variables can be expressed as
\[ p(y, Z, Z^*, W, \lambda^*, \pi^*, \pi_K) \].

We propose a two-stage procedure for sampling variables. The first stage is to estimate
\( K \) by using the training dataset. This stage can eliminate the uncertainty of
\( K \), and thus improve the efficiency of our model. The second stage is to sample all the other
random variables through Gibbs sampling, by data from the training dataset as well as the estimated \( K \).

### 2.1 First Stage – Estimation of \( K \)

Yang and Dunson [2016] developed a scheme that completes the estimation of
\( K \) as the first step. The estimation of \( K \) is completed through a stochastic variable search scheme
in George and McCulloch [1997] which is based on an approximated marginal likelihood.

There are two main advantages of estimating \( K \) before sampling other random variables.
Firstly, it can be beneficial to the inference by fixing the numbers of inclusions of the tensor.
Secondly, the sampling of \( K \) can indicate whether a predictor is important or not, which
facilitates the model’s parsimony. We use the value of \( k_j \) to infer whether the predictor
\( y_{t-j} \) is important or not. That is, for any \( j \in Z_{[1,q]} \), if \( k_j = 1 \), then \( y_{t-j} \) is not important
for \( y_t \) and if \( k_j > 1 \), then \( y_{t-j} \) is important for \( y_t \). The posterior and approximate sampler
of \( K \) is provided as follows.

**Posterior of \( K \)**

Concentration on \( K = \{k_j\}_{j \in Z_{[1,q]}} \) is an innovative key developed by Sarkar and Dunson
[2016], where states that the posterior of \( k_j \) can be expressed as

\[
p(k_j | \ldots) \propto \exp(-\varphi j k_j) \prod_{\omega=0}^{c_0-1} \frac{\Gamma(k_j \gamma_j)}{\Gamma(k_j \gamma_j + n_{j,\omega})}
\] (5.9)

with \( k_j = \max\left( \{z_{j,t}\}_{t \in Z_{[q+1,T]}} \right) \), \( c_0 \), \( n_{j,\omega} = \sum_{t \in Z_{[q+1,T]}} \mathbb{1}\{y_{t-j} = \omega\} \).

When \( n_{j,\omega} \) is large, we can use Stirling’s approximation \( \Gamma(n+a)/\Gamma(n) \approx n^a \) [Robbins, 1955; Feller, 2008] for (5.9) as

\[
p(k_j | \ldots) \approx \left( \sum_{l=\max(\{z_{j,t}\}_{t \in Z_{[q+1,T]}})}^{c_0} \frac{\exp(-\varphi j l) \prod_{\omega=0}^{c_0-1} n_{j,\omega}^{(k_j-l)\gamma_j}}{\exp(-\varphi j k_j) \prod_{\omega=0}^{c_0-1} n_{j,\omega}^{(k_j-l)\gamma_j}} \right)^{-1}.
\]
Proof for (5.9): We follow the proof in Appendix B of Sarkar and Dunson [2016]. Define $Z_j := \{z_{j,t}\}_{t \in \mathbb{Z}_{[q+1,T]}}$, $W_j := \{y_{t-j}\}_{t \in \mathbb{Z}_{[q+1,T]}}$ and $\pi(j) := \{\pi(j) (\omega)\}_{\omega \in \mathbb{Z}_{[0,c_0-1]}}$ from (5.7). According to the prior of $k_j$ in (5.8), it is dependent on $j$. Moreover, $Z_j$ and $W_j$ depend on $j$. Then the probability function of the posterior of $k_j$ can be expressed as

$$p(k_j \mid \ldots) = p(k_j \mid Z_j, W_j).$$

(5.10)

By the Bayesian rule, (5.10) can be rewritten as

$$p(k_j \mid Z_j, W_j) = \frac{p(k_j \mid W_j)p(Z_j \mid W_j, k_j)}{p(Z_j \mid W_j)}$$

$p(k_j \mid W_j)$ is equivalent to $p(k_j)$ as shown in (5.8), and $p(Z_j \mid W_j, k_j)$ is derived in the follows.

$$p(Z_j \mid W_j, k_j) = \int p(Z_j \mid W_j, k_j, \pi(j))p(\pi(j))d\pi(j)$$

$$= \prod_{\omega=0}^{c_0-1} \left( \prod_{l=1}^{k_j} \frac{\Gamma(\gamma_j + n_{j,\omega}(l)) \Gamma(k_j \gamma_j)}{\Gamma(k_j \gamma_j + n_{j,\omega})} \right)$$

$$= \left( \prod_{\omega=0}^{c_0-1} \left( \prod_{l=1}^{\max(Z_j,\omega)} \frac{\Gamma(\gamma_j + n_{j,\omega}(l)) \Gamma(\gamma_j)}{\Gamma(\gamma_j + n_{j,\omega})} \right) \right) \left( \prod_{\omega=0}^{c_0-1} \frac{\Gamma(k_j \gamma_j)}{\Gamma(k_j \gamma_j + n_{j,\omega})} \right),$$

(5.11)

where $Z_{j,\omega} = \{z_{j,t} : y_{t-j} = \omega, t \in \mathbb{Z}_{[q+1,T]}\}$, $n_{j,\omega} = \sum_{t \in \mathbb{Z}_{[q+1,T]}} I\{y_{t-j} = \omega\}$ and $n_{j,\omega}(l) = \sum_{t \in \mathbb{Z}_{[q+1,T]}} I\{z_{j,t} = l, y_{t-j} = \omega\}$.

Moreover, we have

$$p(Z_j \mid W_j) = \sum_{k_j = \max(Z_j,\omega)}^{c_0} p(Z_j \mid W_j, k_j)p(k_j)$$

$$= \left( \prod_{\omega=0}^{c_0-1} \prod_{l=1}^{\max(Z_j,\omega)} \frac{\Gamma(\gamma_j + n_{j,\omega}(l)) \Gamma(\gamma_j)}{\Gamma(\gamma_j + n_{j,\omega})} \right) \left( \sum_{k_j = \max(Z_j)}^{c_0} p(k_j) \prod_{\omega=0}^{c_0-1} \frac{\Gamma(k_j \gamma_j)}{\Gamma(k_j \gamma_j + n_{j,\omega})} \right)$$

(5.12)

Combining (5.8), (5.10), (5.11) and (5.12), we have, for $k_j = \max(Z_j), \ldots, c_0$,

$$p(k_j \mid \ldots) \propto p(k_j) \prod_{\omega=0}^{c_0-1} \frac{\Gamma(k_j \gamma_j)}{\Gamma(k_j \gamma_j + n_{j,\omega})}.$$
Approximate Sampler

Recall that the dimensions of random variables $Z^*$ and $\pi_K$ vary with $K$, which makes the design of efficient MCMC algorithms infeasible. In order to facilitate the efficiency, we adopt a two-stage sampler to facilitate the efficiency of the MCMC algorithm. The first stage is to estimate the $K$, referred to as the approximate sampler in this thesis and then sample the other random variables while keeping the $K$ fixed in the second stage. The levels of $y_{t-j}$ are partitioned into $k_j$ clusters $\{C_{j,r} : r \in \mathbb{Z}_{[1,k_j]}\}$ with each cluster $C_{j,r}$ assumed to correspond to its own latent class $h_j = r$. With independent Dirichlet priors on the mixture kernels $\lambda_H \sim \text{Gamma}(a, b)$ marginalised out, the likelihood of our targeted response $\{y_t\}_{t \in \mathbb{Z}_{[q+1,T]}}$ conditional on the cluster configuration $C = \{C_{j,r} : j \in \mathbb{Z}_{[1,q]}, r \in \mathbb{Z}_{[1,k_j]}\}$ is given by

$$p(\{y_t\}_{t \in \mathbb{Z}_{[q+1,T]}} | C) = \prod_{H \in \mathcal{H}} \int_0^\infty \left( \prod_{t \in \mathbb{Z}_{[q+1,T]}} \frac{(y_t \xi)!}{\lambda_H} \right)^{-1} \exp \left( -(\sum_{t \in \mathbb{Z}_{[q+1,T]}} \xi \lambda_H) \right) d\lambda_H$$

$$= \prod_{H \in \mathcal{H}} \int_0^\infty \left( \prod_{t \in \mathbb{Z}_{[q+1,T]}} \frac{(y_t \xi)!}{\lambda_H} \right)^{-1} \exp \left( -(\sum_{t \in \mathbb{Z}_{[q+1,T]}} \xi \lambda_H) \right) d\lambda_H$$

$$= \prod_{H \in \mathcal{H}} \frac{1}{(1/b)^a \Gamma(a)} \left( \prod_{t \in \mathbb{Z}_{[q+1,T]}} \frac{(y_t \xi)!}{\lambda_H} \right)^{-1} \Gamma \left( a + \sum_{t \in \mathbb{Z}_{[q+1,T]}} y_t \xi \right)$$

$$= \prod_{H \in \mathcal{H}} \frac{1}{(1/b)^a \Gamma(a)} \left( \prod_{t \in \mathbb{Z}_{[q+1,T]}} \frac{(y_t \xi)!}{\lambda_H} \right)^{-1} \Gamma \left( a + \sum_{t \in \mathbb{Z}_{[q+1,T]}} y_t \xi \right),$$

where $\xi = 1 \{y_{t-1} \in C_{1,h_1}, \ldots, y_{t-q} \in C_{q,h_q}\}$. Then we do the following for $j \in \mathbb{Z}_{[1,q]}$.

1. If $1 \leq k_j \leq c_0$, we propose to either increase $k_j$ to $(k_j + 1)$ or decrease $k_j$ to $(k_j - 1)$.

2. If an increase move is proposed, we randomly split a cluster of $y_{t-j}$ into two clusters. We accept this move with acceptance rate based on the approximated marginal likelihood.

3. If a decrease move is proposed, we randomly merge two clusters of $y_{t-j}$ into a single cluster. We accept this move with acceptance rate based on the approximated marginal likelihood.

Take $K^*$ and $C^*$ are the updated model index and cluster, and denote $\alpha(\cdot; \cdot)$ as the
CHAPTER 5. BAYESIAN TENSOR FACTORISATIONS FOR UNIVARIATE TIME SERIES OF COUNTS

Metropolis-Hastings acceptance rate, \( L(\cdot) \) as the likelihood function and \( q(\cdot \rightarrow \cdot) \) as the proposal function, we can have

\[
\alpha(K, C; K^*, C^*) = \frac{L(\{y_t\}_{t \in \mathbb{Z}^{[q+1, T]}}, K^*, C^*)q(K^*, C^* \rightarrow K, C)}{L(\{y_t\}_{t \in \mathbb{Z}^{[q+1, T]}}, K, C)q(K, C \rightarrow K^*, C^*)}.
\]

2.2 Second Stage – Posteriors of Random Variables

In the second stage, we aim to sample the other random variables through posterior computations conditional on the estimated \( K \). Recall that

\[
y := \{y_t\}_{t \in \mathbb{Z}^{[q+1, T]}}, \quad W := \{\{y_{t-j}\}_{j \in \mathbb{Z}^{[1, q]}}\}_{t \in \mathbb{Z}^{[q+1, T]}}, \quad Z := \{Z_t\}_{t \in \mathbb{Z}^{[q+1, T]}}, \quad Z^* := \{Z^*_H\}_{H \in \mathcal{H}}, \quad \lambda^* := \{\lambda^*_t\}_{t \in \mathbb{Z}^{[1, L]}}, \quad \pi^* := \{\pi^*_t\}_{t \in \mathbb{Z}^{[1, L]}}, \quad \pi_K := \{\{\pi^{(j)}(\omega)\}_{\omega \in \mathbb{Z}^{[0, c_0-1]}}\}_{j \in \mathbb{Z}^{[1, q]}},
\]

we can have the conditional independence structure, and the full conditionals can be written as

\[
p(y, Z, Z^*, W, \lambda^*, \pi^*, \pi_K) = p(y, Z, Z^*, W, \lambda^*, \pi^*, \pi_K) \cdot p(W, \lambda^*, \pi^*, \pi_K).
\]

Then, we are ready to describe our samplers to update the rest of random variables. Iterations among these random variables are shown as follows. The proof can be found in the appendix of this chapter.

Note that \( p(y \mid Z, Z^*, \lambda^*) \) follows the factorisation

\[
p(y \mid Z, Z^*, \lambda^*) = \prod_{t \in \mathbb{Z}^{[q+1, T]}} p(y_t \mid \lambda^*_{Z^*_t}).
\]

1. Sample \( Z^*_H \) for each \( H \in \mathcal{H} \) from

\[
p(Z^*_H = l \mid \ldots) \propto \pi^*_t(\lambda^*_t)^{n^*_H} \exp(-n_H \lambda^*_t).
\]

where

\[
n^*_H = \sum_{t \in \mathbb{Z}^{[q+1, T]}} 1\{Z_t = H\}y_t
\]
and
\[ n_H = \sum_{t \in \mathbb{Z}_{[q+1,T]}} \mathbb{1}\{Z_t = H\}. \]

2 Sample \( V_l \) for \( l \in \mathbb{Z}_{[1,L]} \) from
\[ V_l | \cdots \sim \text{Beta} \left( 1 + N^*_l, \alpha_0 + \sum_{l' > l} N^*_l \right) \]
where \( N^*_l = \sum_{H \in \mathcal{H}} \mathbb{1}\{Z^*_H = l\} \), and update \( \pi^*_l \) accordingly.

3 Sample each \( \lambda^*_l \) with \( l \in \mathbb{Z}_{[1,L]} \) from
\[ \lambda^*_l | \cdots \sim \text{Gamma} \left( a + N^*_H(l), b + N_H(l) \right), \]
where
\[ N^*_H(l) = \sum_{H \in \mathcal{H}} \mathbb{1}\{Z^*_H = l\} n^*_H \]
and
\[ N_H(l) = \sum_{H \in \mathcal{H}} \mathbb{1}\{Z^*_H = l\} n_H. \]

4 For \( j \in \mathbb{Z}_{[1,q]} \) and \( \omega \in \mathbb{Z}_{[0,c_0-1]} \), sample
\[ \left\{ \pi^{(j)}_1(\omega), \ldots, \pi^{(j)}_{k_j}(\omega) \right\} | \cdots \sim \text{Dirichlet} \left\{ \gamma_j + n_{j,\omega}(1), \ldots, \gamma_j + n_{j,\omega}(k_j) \right\} \]
where \( n_{j,\omega}(h_j) = \sum_{t \in \mathbb{Z}_{[q+1,T]}} \mathbb{1}\{z_{j,t} = h_j, y_{t-j} = \omega\} \).

5 Sample \( z_{j,t} \) for \( j \in \mathbb{Z}_{[1,q]} \) and \( t \in \mathbb{Z}_{[q+1,T]} \) from
\[ p(z_{j,t} = h | z_{j',t} = h', j' \neq j, \ldots) \propto \pi^{(j)}_h(y_{t-j}) \left( \lambda^*_H \right)^{y_{t-j}} \exp \left( -\lambda^*_H \right), \]
where \( H_{-j=h} \) is equal to \( H \) at all position except the \( j \)-th position taking the value \( h \).
CHAPTER 5. BAYESIAN TENSOR FACTORISATIONS FOR UNIVARIATE TIME SERIES OF COUNTS

3 Simulation Experiments

Details of Datasets

We design a series of experiments with various scenarios to evaluate the predictive ability of our model and its competitors. Important lags that generate the response $y_t$ vary from scenario to scenario. For each scenario, we simulate 10 datasets and 5,000 data points for each dataset. Sarkar and Dunson [2016] considered a sample size of around 1,000 to evaluate models on time series of categorical data with 4 categories. With the increase of the range of data, it is reasonable to enlarge the size of data for better fitting of models.

We have tried several samples sizes for the training process, and a size of 5,000 is sufficient for fitting our BTF model as well as its competitors. Two types of datasets are generated in our experiments, which are log-linear ones and their non-linear counterparts, for further comparison.

Criteria and Competitors

We divide datasets into two training: testing ratios, which are 8 : 2 and 9 : 1, respectively. We are curious about two criteria for comparison: the average log predictive score (ALPS) and mean squared error (MSE). Therefore, we generate two tables comparing our BTF model with both its Bayesian and frequentist counterparts with maximum likelihood estimation (MLE). The details of the two tables are as follows.

- Criterion: ALPS; Competitor: Bayesian Poisson autoregressive (BaPAR) model. We follow the idea from Czado et al. [2009] to define ALPS used for comparison in this thesis.

**Definition 3.1** (Average log predictive score). The mathematical definition of average log predictive score can be expressed as

$$\frac{-\sum_{t \in T'} \sum_{i=1}^{N} \log \hat{p}^{(i)}(y_t)}{T'N},$$

where $T'$ represents the testing data points, $T'$ is the size of $T'$ and $\hat{p}^{(i)}(\cdot)$ is the one-step ahead estimated transition probability by each model at the $i$-th iteration of MCMC with total $N$ iterations.

- Criterion: MSE; Competitors: Poisson autoregressive (PAR) model and integer-valued generalised autoregressive conditionally heteroscedastic (INGARCH) model.
There are several different versions of MSE, and in this thesis, we use the definition as follows.

**Definition 3.2** (Mean squared error). The mathematical definition of mean squared error can be expressed as
\[
\sum_{t \in T} \frac{(y_t - \hat{y}_t)^2}{T'},
\]
where \(\hat{y}_t\) is the estimator by each model at time \(t\) in the testing dataset. We use the expression in (5.5) as the estimator for our BTF model.

ALPS is of great value when comparing models for dealing with times series of count because it takes all testing data into consideration. More importantly, ALPS considers the whole corresponding predictive density for each testing data instead of a point prediction. MSE is of interest to frequentists who always regard MSE as a popular candidate for the loss function, and we respect their value in this thesis.

**Settings for Models**

The assumed most distant important lag \(q\) is set to be the actual most distant important lag plus two throughout simulation experiments, as in [Sarkar and Dunson, 2016]. For example, if the most distant lag that \(y_t\) depends on is \(y_{t-8}\), then \(q\) is set to be 10. Akaike information criterion (AIC) and Bayesian information criterion (BIC) are used for model selection for PAR, BaPAR and INGARCH models.

The MCMC runs of BaPAR model are bases on 5,000 burn-in samples and 10,000 posterior samples, respectively. Recall that a BaPAR model follows the expression in (3.4). With the absence of further prior information, we set the priors for parameters as \(\beta_0 \sim N(0, 10^{-6})\) and \(\beta_i \sim N(0, 10^{-4})\) for any \(i \in \mathbb{Z}_{[1,q]}\). All MCMC runs for our BTF model are based on the following burn-in and posterior samples respectively: For the first stage fo our simple BTF model, 1,000 and 2,000 for selecting the important lags and their corresponding number of inclusions; and, for the second stage, 2,000 and 5,000 for sampling the rest of the parameters. The thinning rate for the last stage is set to be 10.

We utilise rjags [Plummer et al., 2016] package in R for the MCMC inference of BaPAR model. PAR and INGARCH models follow the expressions in (3.4) and (3.8), respectively, and they are trained through the package tscount [Liboschik et al., 2017] in R.
3.1 Log-Linear Datasets

We generate log-linear datasets as follows, with six different scenarios (different values of $\beta$’s).

\[
y_t \sim \text{Poisson}(\lambda_t),
\]

\[
\log(\lambda_t) = \beta_0 + \sum_{i=1}^{q'} \beta_i \log(y_{t-i} + 1),
\]

where $q'$ represents the maximal $i$ that $\beta_i \neq 0$.

The intuition of generating these datasets is to use from one to many as well as close to distant important lags, in order to compare our BTF model with its competitors in a more comprehensive perspective. The number of important lags varies from one to three, and both close important lags, like $t - 1$, as well as their distant counterparts, like $t - 29$, are selected to generate time series datasets for comparison. The details of these six different log-linear scenarios are as follows. Such datasets can be easily generated by \texttt{tscount} package [Liboschik et al., 2017] in R.

(A). $\beta_0 = 1, \beta_1 = 0.5$;

(B). $\beta_0 = 1, \beta_7 = 0.5$;

(C). $\beta_0 = 1, \beta_{29} = 0.7$;

(D). $\beta_0 = 1, \beta_1 = -0.5, \beta_7 = 0.5$;

(E). $\beta_0 = 1, \beta_{19} = -0.5, \beta_{29} = 0.5$;

(F). $\beta_0 = 1, \beta_1 = -0.5, \beta_7 = -0.5, \beta_{19} = 0.5$;

For example, in (D), we set all non-zero $\beta$’s to be $\beta_0 = 1$, $\beta_1 = -0.5$ and $\beta_7 = 0.5$, then the time series datasets are generated by

\[
y_t \sim \text{Poisson}(\lambda_t),
\]

\[
\log(\lambda_t) = 1 - 0.5 \log(y_{t-1} + 1) + 0.5 \log(y_{t-7} + 1).
\]

Table 5.1 shows that, according to ALPS, a BaPAR model has better performance against our BTF model with not distant important lags, as shown in Scenario (A), (B) and (D). There are no or not many unimportant intermediate lags leading unsatisfactory predictive
Table 5.1: Mean of average log predictive score (ALPS) (with standard deviations in brackets) between Bayesian Poisson autoregression model and our Bayesian tensor factorisations model (BTF) based on 10 Poisson autoregression generated datasets for each one of 6 Scenarios. AIC and BIC columns indicate that the best model has been chosen with the corresponding criterion. Models with the best performance are highlighted bold.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Data Sizes</th>
<th>ALPS</th>
<th>Bayesian Poisson autoregression</th>
<th>BTF</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A) :</td>
<td>4000 : 1000</td>
<td>2.436(0.024)</td>
<td>2.436(0.024)</td>
<td>2.443(0.022)</td>
</tr>
<tr>
<td></td>
<td>4500 : 500</td>
<td>2.441(0.031)</td>
<td>2.441(0.031)</td>
<td>2.450(0.030)</td>
</tr>
<tr>
<td>(B) :</td>
<td>4000 : 1000</td>
<td>2.450(0.019)</td>
<td>2.449(0.019)</td>
<td>2.458(0.022)</td>
</tr>
<tr>
<td></td>
<td>4500 : 500</td>
<td>2.454(0.028)</td>
<td>2.452(0.031)</td>
<td>2.463(0.030)</td>
</tr>
<tr>
<td>(C) :</td>
<td>4000 : 1000</td>
<td>3.126(0.018)</td>
<td>3.126(0.018)</td>
<td>3.108(0.014)</td>
</tr>
<tr>
<td></td>
<td>4500 : 500</td>
<td>3.123(0.024)</td>
<td>3.123(0.024)</td>
<td>3.106(0.021)</td>
</tr>
<tr>
<td>(D) :</td>
<td>4000 : 1000</td>
<td>1.870(0.016)</td>
<td>1.870(0.016)</td>
<td>1.882(0.024)</td>
</tr>
<tr>
<td></td>
<td>4500 : 500</td>
<td>1.876(0.020)</td>
<td>1.876(0.020)</td>
<td>1.885(0.017)</td>
</tr>
<tr>
<td>(E) :</td>
<td>4000 : 1000</td>
<td>1.873(0.015)</td>
<td>1.873(0.015)</td>
<td>1.857(0.017)</td>
</tr>
<tr>
<td></td>
<td>4500 : 500</td>
<td>1.869(0.018)</td>
<td>1.869(0.018)</td>
<td>1.852(0.020)</td>
</tr>
<tr>
<td>(F) :</td>
<td>4000 : 1000</td>
<td>1.683(0.013)</td>
<td>1.683(0.013)</td>
<td>1.631(0.009)</td>
</tr>
<tr>
<td></td>
<td>4500 : 500</td>
<td>1.689(0.017)</td>
<td>1.689(0.017)</td>
<td>1.635(0.012)</td>
</tr>
</tbody>
</table>

Table 5.2: Mean of mean squared error (MSE) (with standard deviations in brackets) among Poisson autoregression, integer-valued generalised autoregressive conditionally heteroscedastic (INGARCH) model and our Bayesian tensor factorisations model (BTF) based on 10 Poisson autoregression generated data sets for each one of 6 Scenarios. AIC and BIC columns indicate that the best model has been chosen with the corresponding criterion. Models with the best performance are highlighted bold.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Data Sizes</th>
<th>MSE</th>
<th>Poisson autoregression</th>
<th>BTF</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A) :</td>
<td>4000 : 1000</td>
<td>7.892(0.442)</td>
<td>7.892(0.442)</td>
<td>7.903(0.439)</td>
</tr>
<tr>
<td></td>
<td>4500 : 500</td>
<td>8.060(0.649)</td>
<td>8.060(0.649)</td>
<td>8.078(0.648)</td>
</tr>
<tr>
<td>(B) :</td>
<td>4000 : 1000</td>
<td>7.984(0.476)</td>
<td>7.983(0.477)</td>
<td>8.001(0.472)</td>
</tr>
<tr>
<td></td>
<td>4500 : 500</td>
<td>7.887(0.698)</td>
<td>7.882(0.702)</td>
<td>7.892(0.710)</td>
</tr>
<tr>
<td>(C) :</td>
<td>4000 : 1000</td>
<td>29.353(0.840)</td>
<td>29.353(0.840)</td>
<td>29.360(0.851)</td>
</tr>
<tr>
<td></td>
<td>4500 : 500</td>
<td>29.046(1.246)</td>
<td>29.046(1.246)</td>
<td>29.049(1.245)</td>
</tr>
<tr>
<td>(D) :</td>
<td>4000 : 1000</td>
<td>2.965(0.158)</td>
<td>2.965(0.158)</td>
<td>2.965(0.165)</td>
</tr>
<tr>
<td></td>
<td>4500 : 500</td>
<td>2.984(0.111)</td>
<td>2.984(0.111)</td>
<td>2.982(0.112)</td>
</tr>
<tr>
<td>(E) :</td>
<td>4000 : 1000</td>
<td>2.977(0.089)</td>
<td>2.977(0.089)</td>
<td>2.976(0.089)</td>
</tr>
<tr>
<td></td>
<td>4500 : 500</td>
<td>2.993(0.066)</td>
<td>2.993(0.066)</td>
<td>2.988(0.063)</td>
</tr>
<tr>
<td>(F) :</td>
<td>4000 : 1000</td>
<td>2.233(0.091)</td>
<td>2.233(0.091)</td>
<td>2.232(0.093)</td>
</tr>
<tr>
<td></td>
<td>4500 : 500</td>
<td>2.134(0.132)</td>
<td>2.134(0.132)</td>
<td>2.135(0.132)</td>
</tr>
</tbody>
</table>

performance for the BaPAR model in these three cases. While for scenarios with more distant important lags like \( y_{t-19} \) or \( y_{t-29} \) in Scenario (C), (E) and (F), our BTF model has lower ALPS, or equivalently, better predictive performance, against the BaPAR model. Table 5.2 tells a similar story except for the last scenario, where our BTF model outperforms BaPAR in Table 5.1 while fails to do so against PAR and INGARCH models in Table 5.2. Both PAR and INGARCH models have leading predictive performance, in terms of the criterion of MSE, in scenarios where important lags are not distant. On the contrary, with important lags to be relatively distant, our BTF outperforms its two competitors in Scenario (C) and (E). Therefore, a conclusion can be drawn that our BTF model can predict as accurately as, or even better than BaPAR, PAR and INGARCH models when modelling our simulated datasets of time series that tends to be linear or log-linear.
3.2 Non-Linear Datasets

We generate non-linear datasets by the following rules, with six different scenarios for selecting important lags. There are $K$ important lag(s) $\{y_{t-i_1}, \ldots, y_{t-i_K}\}$; the first 10 data points in each time series are generated by Poisson($\nu_-$); for $t > 10$, if $\sum_{j=1}^{K} y_{t-i_j} \geq K\nu_-$, then $y_t \sim \text{Poisson}(\nu_+)$; else $y_t \sim \text{Poisson}(\nu_-)$.

Following the intuition mentioned in the previous subsection, we consider different numbers of important lags as well as close to distant important time lags to generate datasets of time series. We believe that this provides a comprehensive understanding of the predictive performance of each model. The details of these six different non-linear scenarios are as follows.

(A). Important lag: $y_{t-1}, \nu_+=5, \nu_- = 10$;

(B). Important lag: $y_{t-7}, \nu_+=5, \nu_- = 10$;

(C). Important lags: $y_{t-1}$ & $y_{t-3}, \nu_+=5, \nu_- = 10$;

(D). Important lags: $y_{t-4}$ & $y_{t-7}, \nu_+=5, \nu_- = 10$;

(E). Important lags: $y_{t-1}$ & $y_{t-4}$ & $y_{t-7} & y_{t-9}, \nu_+=5, \nu_- = 10$;

(F). Important lags: $y_{t-1}$ & $y_{t-4}$ & $y_{t-7} & y_{t-9}, \nu_+=5, \nu_- = 10$.

For example, in (E), there exist 3 important lag $\{y_{t-1}, y_{t-4}, y_{t-7}\}; (\nu_+, \nu_-) = (5, 10)$; $\{y_t\}_{t \in \mathbb{Z}_{[1,10]}}$ are generated by Poisson(5); for $t > 10$, if $y_{t-1} + y_{t-4} + y_{t-7} \geq 30$, then $y_t \sim \text{Poisson}(5)$; else $y_t \sim \text{Poisson}(10)$.

The outcome of comparison between two Bayesian models, BaPAR and our BTF models, is shown in Table 5.3. According to the table, our BTF has better predictive performance against BaPAR model, due to its lower corresponding ALPS in all the scenarios. Our BTF model also outperforms PAR and INGARCH models when MSE is considered the criterion for comparison, as shown in Table 5.4. Hence, one can conclude that our BTF model can be as accurate as BaPAR, PAR and INGARCH models in terms of prediction when dealing with our simulated datasets with non-linear time series of count.

3.3 MCMC Diagnostics

Figure 5.1 shows the information of MCMC diagnostics for Scenario (C) in the non-linear comparison. We use 5,000 data points generated by the rules as i. $y_{t-1}$ and $y_{t-3}$ are
important lags; ii. If $y_{t-1} + y_{t-3} \geq 20$, $y_t$ is generated by Poisson(5), otherwise, generated by Poisson(10). We use the expectation of our BTF model, in (5.5), to test whether the MCMC of our simple BTF model is convergent. In Figure 5.1, the running means and quantiles are very stable and show convergence in all cases. Also, there is a fair agreement between true conditional values and the running posterior means with the true conditional values shown in the lower-left corners.

## 4 Applications

Two real-world applications are used to comparing the predictive performance of our simple Bayesian tensor factorisations (BTF) model, Poisson autoregressive (PAR) model, integer-valued generalised autoregressive conditionally heteroscedastic (INGARCH) model...
CHAPTER 5. BAYESIAN TENSOR FACTORISATIONS FOR UNIVARIATE TIME SERIES OF COUNTS

Figure 5.1: Results of simulation experiments for the case with $\nu_+ = 5$, $\nu_- = 10$, true important lags $\{1, 3\}$. In each panel, $\lambda(i_1, i_3)$ represents the conditional mean of $y_t$ given $y_{t-1} = i_1, y_{t-3} = i_3$, and the true value is shown in the lower left corner. In each panel, the conditional mean of $y_t|y_{t-1}, y_{t-3}$ is drawn in cyan line. The blue solid line shows the running mean, and the black dashed lines are the 5% and 95% quantiles for each case. $y$-axis represents the mean of Poisson parameter and $x$-axis does MCMC samplers after burn-in. Thinning rate is set to be 10.

4.1 Application 1

We use the dataset Death from Drug Dependence (Toxicomania) obtained from the Cause of Death Register maintained by Statistics Norway. The dataset is about the monthly death from drug dependence taking place from January 1969 to June 2009, with 486 points in all. Figure 5.2 is the trace plot of this dataset. We consider two ratios of training: testing datasets as $8:2$ and $9:1$, or namely, the first 388 and 437 data points for the training of these models, and leave the rest 98 and 49 points respectively for testing.

We assume the maximal distant important lag to be 10 for our simple BTF model, PAR INGARCH and BaPAR models. The midpoint of training data is around 20, therefore, we...
set $a$ and $b$ to be 20 and 1 respectively.

Outcomes for comparison are shown in Table 5.5 and 5.6. With the first 388 data as the training dataset, BaPAR(7), PAR(7), INGARCH(1, 6) and INGARCH(1, 2) are selected for comparison due to the least corresponding AIC and BIC criteria. Our BTF has the lower ALPS against the BaPAR model, while INGARCH(1, 2) leads the predictive performance when we take MSE as the criterion for comparison. Such circumstance does not appear when a larger size of the training dataset is considered. With 437 training data points, BaPAR(8), BaPAR(5), PAR(8), PAR(5), INGARCH(1,3) and INGARCH(1,1) are selected for comparing with our BTF model. The outcome under both ALPS and MSE criteria shows that our BTF has the best predictive performance due to the least scores. The relative frequency distributions (RFD) of selecting important lags by our BTF model are shown in Figure 5.3. According to both panels of Figure 5.3, the first, second and fifth lags, $y_{t-1}$, $y_{t-2}$ and $y_{t-5}$, are indicated as important because each of their corresponding RFD is higher than 0.5.

### 4.2 Application 2

The second application focuses on the number of stocks with a positive daily return listed in NASDAQ100 Index. Three sizes of datasets are considered as follows.
### Table 5.5: Average log predictive score (ALPS) (with models selected in brackets) between Bayesian Poisson autoregressive model and our simple Bayesian tensor factorisations (BTF) model. AIC and BIC columns indicate that the best model has been chosen with the corresponding criterion. Training and testing data sizes appear in the first column. Models with best performance are highlighted bold.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Data Sizes</th>
<th>Bayesian Poisson autoregression</th>
<th>BTF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Toxicomania</td>
<td>388 : 98</td>
<td>2.715(7)</td>
<td>2.617</td>
</tr>
<tr>
<td></td>
<td>437 : 49</td>
<td>1.752(5)</td>
<td>1.722</td>
</tr>
</tbody>
</table>

### Table 5.6: Mean squared error (MSE) (with models selected in brackets) among Poisson autoregression, integer-valued generalised autoregressive conditionally heteroscedastic (INGARCH) model and our Bayesian tensor factorisations (BTF) model. AIC and BIC columns indicate that the best model has been chosen with the corresponding criterion. Training and testing data sizes appear in the first column. Models with best performance are highlighted bold.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Data Sizes</th>
<th>Poisson autoregression</th>
<th>INGARCH</th>
<th>BTF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Toxicomania</td>
<td>388 : 98</td>
<td>24.306(7)</td>
<td>25.009(1, 2)</td>
<td>28.867</td>
</tr>
<tr>
<td></td>
<td>437 : 49</td>
<td>2.674(5)</td>
<td>2.442(1, 1)</td>
<td>2.116</td>
</tr>
</tbody>
</table>

Figure 5.3: Result of important lag selection for the Toxicomania dataset. Left panel: the inclusion proportions of different lags for the scenario with 388 training data points and 98 testing data points. Right panel: the inclusion proportions of different lags for the scenario with 437 training data points and 49 testing data points. The orders of past lags are in the horizontal and their corresponding relative frequency distributions to be important are in the vertical.

- 3 years (01-May-2016 to 30-Apr-2019), 100 stocks and 753 data points in total;
- 7 years (01-May-2012 to 30-Apr-2019), 93 stocks and 1,759 data points in total;
- 10 years (01-May-2009 to 30-Apr-2019), 89 stocks and 2,515 data points in total.

The trace plots of these three datasets can be found in Figure 5.4.

For the prior settings, we set the assumed most distant lag \( q \) to be 10, \( a \) and \( b \) to be 50 and 1 respectively. ALPS and MSE comparisons are in Table 5.7 and 5.8, where models are selected according to the AIC and BIC criteria. For further comparison, we consider ratios between training and test data of 8 : 2 and 9 : 1 respectively. The result indicates that our BTF model has better predictive performance since it has lower scores in 11 out
of 12 scenarios. Lags selected as important by our BTF model for each dataset are shown in Figure 5.5. According to Figure 5.5, all of the lags selected as important for these three datasets gather within the sixth lag, or in other words, $y_{t-6}$.

<table>
<thead>
<tr>
<th>ALPS</th>
<th>Bayesian Poisson autoregression</th>
<th>BTF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset</td>
<td>Data Sizes</td>
<td>AIC</td>
</tr>
<tr>
<td>3 years</td>
<td>602 : 151</td>
<td>10.293(9)</td>
</tr>
<tr>
<td></td>
<td>677 : 76</td>
<td>6.964(9)</td>
</tr>
<tr>
<td>7 years</td>
<td>1407 : 352</td>
<td>9.383(10)</td>
</tr>
<tr>
<td></td>
<td>1583 : 176</td>
<td>9.345(10)</td>
</tr>
<tr>
<td>10 years</td>
<td>2012 : 503</td>
<td>8.235(10)</td>
</tr>
<tr>
<td></td>
<td>2263 : 252</td>
<td>8.617(10)</td>
</tr>
</tbody>
</table>

Table 5.7: Average log predictive score (ALPS) (with models selected in brackets) between Bayesian Poisson autoregressive model and our simple Bayesian tensor factorisations model (BTF). AIC and BIC columns indicate that the best model has been chosen with the corresponding criterion. Training and testing data sizes appear in the first column. Models with best performance are highlighted bold.
<table>
<thead>
<tr>
<th>Dataset</th>
<th>Data Sizes</th>
<th>Poisson autoregression</th>
<th>INGARCH</th>
<th>BTF</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 years</td>
<td>602 : 151</td>
<td>AIC 639.858(9) BIC 639.685(8)</td>
<td>AIC 661.392(9) BIC 661.392(6.9)</td>
<td>MSE 612.978</td>
</tr>
<tr>
<td>7 years</td>
<td>677 : 76</td>
<td>AIC 404.060(9) BIC 404.060(9)</td>
<td>AIC 483.518(10) BIC 483.518(10)</td>
<td>MSE 428.815</td>
</tr>
<tr>
<td>10 years</td>
<td>1407 : 352</td>
<td>AIC 537.946(10) BIC 537.946(10)</td>
<td>AIC 547.346(2.9) BIC 547.346(2.9)</td>
<td>MSE 515.900</td>
</tr>
<tr>
<td></td>
<td>1583 : 176</td>
<td>AIC 538.106(10) BIC 538.106(10)</td>
<td>AIC 551.670(10) BIC 551.670(10)</td>
<td>MSE 522.618</td>
</tr>
<tr>
<td></td>
<td>2012 : 503</td>
<td>AIC 426.443(10) BIC 427.706(8)</td>
<td>AIC 454.387(10) BIC 454.387(10)</td>
<td>MSE 417.212</td>
</tr>
<tr>
<td></td>
<td>2263 : 252</td>
<td>AIC 465.489(10) BIC 464.940(7)</td>
<td>AIC 468.537(5) BIC 481.260(5)</td>
<td>MSE 438.748</td>
</tr>
</tbody>
</table>

Table 5.8: Mean squared error (MSE) (with models selected in brackets) among Poisson autoregression, integer-valued generalised autoregressive conditionally heteroscedastic (INGARCH) model and our Bayesian tensor factorisations model (BTF). AIC and BIC columns indicate that the best model has been chosen with the corresponding criterion. Training and testing data sizes appear in the first column. Models with best performance are highlighted bold.
Figure 5.5: Result of important lag selection for the NASDAQ100 datasets. (a): the inclusion proportions of different lags for the scenario with 602 training data points and 151 data points (3 years); (b): the inclusion proportions of different lags for the scenario with 677 training data points and 76 data points (3 years); (c): the inclusion proportions of different lags for the scenario with 1407 training data points and 352 data points (7 years); (d): the inclusion proportions of different lags for the scenario with 1583 training data points and 176 data points (7 years); (e): the inclusion proportions of different lags for the scenario with 2012 training data points and 503 data points (10 years); (f): the inclusion proportions of different lags for the scenario with 2263 training data points and 252 data points (10 years). The orders of past lags are in the horizontal and their corresponding relative frequency distributions to be important are in the vertical.
5 Appendix

Detail of the Gibbs samplers in Section 2.2 are as follows.

- **Step 1**

\[
p(Z_H = l \mid \ldots) \propto p(z_H = l)p(y \mid \lambda^*_Z)
\]
\[
= \pi_l \prod_{t \in Z_{q+1,T}} [ (\lambda^*_t)^y_t \exp(-\lambda^*_t)] \mathbb{1}_{\{Z_t = l\}}
\]
\[
= \pi_l (\lambda^*_l)^{n_H^*} \exp(-n_H \lambda^*_l),
\]
where
\[
n_H^* = \sum_{t \in Z_{q+1,T}} \mathbb{1}_{\{Z_t = H\}} y_t
\]
and
\[
n_H = \sum_{t \in Z_{q+1,T}} \mathbb{1}_{\{Z_t = H\}}.
\]

- **Step 2**

\[
p(V_l \mid \ldots) \propto p(V_l)p(z^* \mid V_l)
\]
\[
= (V_l)^{a-1} (1 - V_l)^{\alpha_0 - 1} (V_l)^{N^*_l} (1 - V_l)^{\sum_{t'} N^*_t'}
\]
\[
V_l \mid \ldots \sim \text{Beta} \left(1 + N^*_l, \alpha_0 + \sum_{t' > t} N^*_t'\right),
\]
where
\[
N^*_l = \sum_{H \in \mathcal{H}} \mathbb{1}_{\{Z^*_H = l\}}.
\]

- **Step 3**

\[
p(\lambda^*_l \mid \ldots) \propto p(\lambda^*_l)p(y_t \mid \lambda^*_Z)
\]
\[
= (\lambda^*_l)^{a-1} \exp(-b \lambda^*_l)(\lambda^*_l)^{N_H(l)} \exp(-N_H(l))
\]
\[
\lambda^*_l \mid \ldots \sim \text{Gamma} \left(a + N^*_H(l), b + N_H(l)\right),
\]
where
\[
N^*_H(l) = \sum_{H \in \mathcal{H}} \mathbb{1}_{\{Z^*_H = l\}} n^*_H,
\]
and
\[ N_H(l) = \sum_{H \in \mathcal{H}} 1\{Z_H^* = l\} n_H. \]

- Step 4

\[
p(\pi_1^{(j)}(\omega), \ldots, \pi_{k_j}^{(j)}(\omega) | \ldots) \propto p(\pi_1^{(j)}(\omega), \ldots, \pi_{k_j}^{(j)}(\omega)) p(Z | \pi_1^{(j)}(\omega), \ldots, \pi_{k_j}^{(j)}(\omega))
\]
\[
= \prod_{i=1}^{k_j} \pi_i^{\gamma_j-1} \prod_{t \in Z_{[q+1, T]}} \prod_{i=1}^{k_j} \pi_i^{1\{z_{j,t} = h_j, y_{t-j} = \omega\}}
\]
\[
= \prod_{i=1}^{k_j} \pi_i^{\gamma_j + \sum_{t \in Z_{[q+1, T]}} 1\{z_{j,t} = h_j, y_{t-j} = \omega\} - 1}
\]
\[
\{\pi_1^{(j)}(\omega), \ldots, \pi_{k_j}^{(j)}(\omega)\} | \ldots \sim \text{Dirichlet}\{\gamma_j + n_{j,\omega}(1), \ldots, \gamma_j + n_{j,\omega}(k_j)\},
\]

where
\[
n_{j,\omega}(h_j) = \sum_{t \in Z_{[q+1, T]}} 1\{z_{j,t} = h_j, y_{t-j} = \omega\}.
\]

- Step 5

\[
p(z_{j,t} = h | \ldots) \propto p(z_{j,t} = h | \pi_h^{(j)}(y_{t-j})) p(y_t | Z_t = H_{.../j=h})
\]
\[
= \pi_h^{(j)}(y_{t-j}) \left( \lambda_{Z_H^*}^{*} \right)^y \exp \left( -\lambda_{Z_H^*}^{*} \right).
\]
Chapter 6

Bayesian Tensor Factorisations for Univariate Time Series of Large Counts

Our simple BTF model for univariate time series of count data is introduced in the previous chapter. It shows the outstanding predictive performance when dealing with either log-linear or non-linear datasets. However, the flaw of our simple BTF model is obvious. The range of observed values of \( \{y_t\}_{t=1}^T \) determines our model’s complexity. When the range is large, the total number of cluster in the \( j \)-th lag, \( k_j \), will not be necessarily large. The Poisson parameters in the kernels in (5.2) are still limited since we apply the Dirichlet process to reduce parameters. However, the weights of tensor factorisation in (5.3) are the factor that causes trouble for computation. For each \( j \in \mathbb{Z}_{[1,q]} \), there will be \( c_0(k_j - 1) \) independent parameters for the weights, where \( c_0 \to \infty \). With the increase of \( q \) and \( c_0 \), \( \sum_{j=1}^{q} c_0(k_j - 1) \) will be inevitably large. Therefore, when the expected range of \( \{y_t\}_{t=1}^T \) is large, our simple BTF model will be time-costing and inefficient due to enormous parameters.

In order to solve the problem, we introduce a BTF model for univariate time series of count data when the counts are large based on the model we have specified in the previous chapter. The idea is that we first split the data into two sets. The first one is for data clustering and the second one is for model training. This chapter consists of model specification, estimation and inference, simulation experiments and applications.
1 Model Specification

Our idea is to use some data points in \( \{y_t\}_{t \in Z_{[1,T]}} \) and the mixture of Poisson distributions to estimate the cluster that a data point belongs to, and through this approach, our general Bayesian tensor factorisation model can then solve the problem.

We separate \( \{y_t\}_{t=1}^T \) into two segments of size \( T_1 \) and \( T_2 \), representing the size of pre-training dataset and training dataset, respectively. Specifically, \( \{y_t\}_{t \in Z_{[1,T_1]}} \) is the pre-training dataset, with its counterpart training dataset \( \{y_t\}_{t \in Z_{[T_1+1,T_1+T_2]}} \).

We have \( T = T_1 + T_2 \) realisations of such random variables and the statistical problem is that of using the first \( T_1 + T_2 \) observations to build a predictive model that estimates the predictive densities for the testing data. Our methodology is based on the following model fitting mechanism that we will fit a statistical model based on the pre-training data that will be used for fitting our predictive model in the training dataset.

For the first step, we define a collection of latent variables \( \{w_{1:c-1}, \mu_{1:c}, c\} \) that models the pre-training data as

\[
p(y_t \mid w_{1:c-1}, \mu_{1:c}, c) = \sum_{i=1}^{c} w_i \text{PD}(y_t; \mu_i),
\]

for any \( t \in Z_{[1,T_1]} \), where \( \text{PD}(y_t; \mu_i) \) denotes the Poisson density of \( y_t \) with parameter \( \mu_i \), \( 0 < w_i < 1 \), \( \sum_{i=1}^{c} w_i = 1 \), \( \mu_i \geq 0 \) and \( \iota \in Z_{[1,c]} \) represents the label of component \( y_t \) belongs to in the pre-training dataset. Thus, (6.1) assumes that any \( y_t \) in the pre-training dataset is distributed as a finite mixture of Poisson distributions with \( c \) components, weights \( w_i \) and intensities \( \mu_i \).

If new observations of \( y_t \) for all in the training dataset follow the finite mixture of Poisson distributions in (6.1), then we can have a latent variable \( d_t \) representing the estimated label of component that \( y_t \) belongs to. The latent variable \( d_t \) can be generated by a predictive probability mass function as, for any \( t > T_1 \),

\[
p(d_t = i) = w_i
\]

for any \( t > T_1 \), with \( i \in Z_{[1,c]} \). We set \( d_{j,t} = d_{t-j} \) for all \( j \in Z_{[1,q]} \) and \( t \in Z_{[T_1+1+q,T_1+T_2]} \). We define another collection of latent variables as \( D_t = \{d_{j,t}\}_{j \in Z_{[1,q]}} \).

For the next step, we follow the idea from the previous chapter to build a probabilistic model for the training dataset by assuming that the transition probability law of the
sequence \( \{ y_t \} \in \mathbb{Z}_{[T_1 + 1 + q, T_1 + T_2]} \) conditional on \( F_{t-1} \) is that of a probabilistic model of this target sequence conditional on \( D_t \). That is, we have

\[
p(y_t \mid F_{1:t-1}) = p(y_t \mid D_t).
\] (6.3)

Similar to Chapter 5, we define time-dependent random variables \( Z_t := \{ z_{j,t} \}_{j \in \mathbb{Z}_{[1,q]}} \) together with a collection of indices \( H := \{ h_j \}_{j \in \mathbb{Z}_{[1,q]}} \). Furthermore, we define \( H := \{ h_j \}_{j \in \mathbb{Z}_{[1,q]}} \). The connection among \( Z_t, H \) and \( H \) is that for any \( t \in \mathbb{Z}_{[T_1 + 1 + q, T_1 + T_2]} \), \( Z_t \) is sampled with the value \( H \in H \). The conditional transition probability law (6.3) can then be written as a Bayesian hierarchical model, for \( j \in \mathbb{Z}_{[1,q]} \) and \( t \in \mathbb{Z}_{[T_1 + 1 + q, T_1 + T_2]} \), as

\[
y_t \mid Z_t = H \sim \text{Poisson}(\lambda_H),
\] (6.4)

\[
 z_{j,t} \mid d_{j,t} \sim \text{Multinomial} \left( \Omega_{[1,k_j]}, \left\{ \pi_{h_j}^{(j)}(d_{j,t}), \ldots, \pi_{k_j}^{(j)}(d_{j,t}) \right\} \right).
\] (6.5)

(6.4) and (6.5) immediately imply that

\[
p(y_t \mid D_t) = \sum_{H \in H} \text{PD}(y_t; \lambda_H) \prod_{j \in \mathbb{Z}_{[1,q]}} \pi_{h_j}^{(j)}(d_{j,t}).
\]

with constraints

\[
\lambda_H \geq 0, \quad \text{for any } H \in H,
\]

\[
\sum_{h_j=1}^{k_j} \pi_{h_j}^{(j)}(d_{j,t}) = 1, \quad \text{for each combination of } (j, d_{j,t}).
\]

Exact as what has been done in the previous chapter, Chapter 5, we use the stick-breaking construction for \( \lambda_H \) as

\[
\lambda_H \sim \sum_{l=1}^{\infty} \pi_l^* \delta(\lambda_l^*),
\]

where \( \delta(.) \) is an indicator function, and independently, for \( l \in \mathbb{Z}_{[1,\infty]} \), \( \lambda_l^* \sim \text{Gamma}(a, b) \), \( \pi_l^* = V_l \prod_{s=1}^{l-1} (1 - V_s) \), \( V_l \sim \text{Beta}(1, \alpha_0) \) representing a label-clustered Poisson rate, which is generated by a Gamma distribution. Similar to the last chapter, we can assign independent priors on \( \pi^{(j)}(d_{j,t}) \) as

\[
\pi^{(j)}(d_{j,t}) = \left\{ \pi_{1}^{(j)}(d_{j,t}), \ldots, \pi_{k_j}^{(j)}(d_{j,t}) \right\} \sim \text{Dirichlet}(\gamma_j, \ldots, \gamma_j),
\]
and priors on $k_j$ as
\[
p(k_j = \kappa) \propto \exp(-\varphi j \kappa),
\]
where $j \in \mathbb{Z}_{[1,q]}$, $\kappa \in \mathbb{Z}_{[1,c]}$. The prior settings can be found in Section 1 of Chapter 5.

2 Estimation and Inference

We propose a three-stage procedure for sampling variables in our BTF model. The first stage is to estimate the label of group that a count on time series belongs to, or in other words, to estimate $d_t$ for any $t > T_1$. Similar to the estimation and inference section in the last chapter, the second stage is to estimate $K$ by using the training dataset and estimated $D$. This stage can eliminate the uncertainty of $K$, and thus improve the efficiency of our model. The third stage is to sample all the other random variables through Gibbs sampling, by data from the training dataset as well as estimated $K$ and $D$.

2.1 First Stage – Estimation of $D$

Assume that we follow the procedure in Marin et al. [2005]. \{y_t\}_{t \in \mathbb{Z}_{[1,T_1]}} is a mixture of $c$ univariate Poisson distributions with density $\sum_{i=1}^c w_i \text{PD}(y_t; \mu_i)$. \{w_i\}_{i \in \mathbb{Z}_{[1,c]}} are some weights with $\sum_{i=1}^c w_i = 1$ and \{\mu_i\}_{i \in \mathbb{Z}_{[1,c]}} are some Poisson rates. By setting the priors as, for $i \in \mathbb{Z}_{[1,c]}$,
\[
\mu_i \sim \text{Gamma}(1, 1), \quad \{w_i\}_{i \in \mathbb{Z}_{[1,c]}} \sim \text{Dirichlet}(1, \ldots, 1),
\]
we can derive the corresponding Gibbs sampler as follows:

1 Generate the label of $y_t$, $\iota_t$, for $t \in \mathbb{Z}_{[1,T_1]}$ from $i \in \mathbb{Z}_{[1,c]}$ as
\[
p(\iota_t = i) \propto w_i (\theta_i)^{y_t} \exp(-\theta_i) .
\]
Set $n_i = \sum_{t \in \mathbb{Z}_{[1,T_1]}} 1_{\iota_t = i}$ and $I_i = \sum_{t \in \mathbb{Z}_{[1,T_1]}} 1_{\iota_t = i} y_t$.

2 Generate
\[
\{w_i\}_{i \in \mathbb{Z}_{[1,c]}} \sim \text{Dirichlet}(1 + n_1, \ldots, 1 + n_c).
\]

3 For $i \in \mathbb{Z}_{[1,c]}$, generate
\[
\mu_i \sim \text{Gamma}(1 + I_i, 1 + n_i).
\]
For simplicity and convenience, instead of sampling \( d_t \) from (6.2), we estimate such collection of latent variables as, for any \( t > T_1 \),

\[
d_t = \text{arg}\max_i \text{PD}(y_t, \mu_i).
\]

### 2.2 Second Stage – Estimation of \( K \)

We have explained the benefits of sampling \( K \) through an approximate sampler before estimating the other random variables in the previous chapters, and similarly in this chapter, we use the estimated value of \( k_j \) to infer whether the predictor \( d_{j,t} \) is important or not. That is, for any \( j \in \mathbb{Z}_{[1,q]} \), when \( k_j = 1 \), \( d_{j,t} \) is not important for \( y_t \) and when \( k_j > 1 \), \( d_{j,t} \) is important for \( y_t \). Section 2.1 in Chapter 5 describes the approximate sampler for estimating \( K \), and we can replicate (5.13) by substituting \( d_{j,t} \) for \( y_{t-j} \) and limiting \( t \in \mathbb{Z}_{[T_1+1+q,T_1+T_2]} \) for estimating \( K \) in this section.

### 2.3 Third Stage – Posteriors of Other Random Variables

The last stage is to sample the other random variables through posterior computations conditional on the estimated \( K \) and \( D \). A similar procedure of updating the rest of random variables, described in Section 2.2 and the appendix of Chapter 5, can be utilised with an extra step to replace \( y_{t-j} \) by \( d_{j,t} \) and set \( t \in \mathbb{Z}_{[T_1+1+q,T_1+T_2]} \).

### 3 Simulation Experiments

#### Criteria and Competitors

We apply the same criteria and competitors as what we have done in the previous chapter for comparing the predictive performance of our BTF model. Recall that the ALPS is the criterion when comparing against the Bayesian autoregressive (BaPAR) model and the MLE is the one for comparing with the Poisson autoregressive (PAR) and integer-valued generalised autoregressive conditionally heteroscedastic (INGARCH) models. We divide simulated datasets into two training: testing ratios, which are 8 : 2 and 9 : 1, respectively, for further comparison.

#### Settings for Models
Similar to the last chapter, the assumed most distant important lag $q$ is set to be the actual most distant important lag plus two throughout simulation experiments. We assume the number of clusters $c$ to be 10. PAR, BaPAR and INGARCH models are selected by the AIC and BIC.

The MCMC runs of BaPAR model are bases on 5,000 burn-in samples and 10,000 posterior samples, respectively. A BaPAR model follows the expression in (3.4). Without further prior information, we set the priors for parameters as $\beta_0 \sim N(0, 10^{-6})$ and $\beta_i \sim N(0, 10^{-4})$ for any $i \in \mathbb{Z}_{[1,q]}$. Exact as Section 3 of Chapter 5, all MCMC runs for our BTF model are based on the following burn-in and posterior samples respectively: For the first stage for our simple BTF model, 1,000 and 2,000 for selecting the important lags and their corresponding number of inclusions; and, for the second stage, 2,000 and 5,000 for sampling the rest of the parameters. The thinning rate for the last stage is set to be 10.

Recall that we utilise \texttt{rjags} [Plummer et al., 2016] package in R for the MCMC inference of BaPAR model. PAR and INGARCH models follow the expressions in (3.4) and (3.8), respectively, and they are trained through the package \texttt{tscount} [Liboschik et al., 2017] in R.

### Details of Datasets

Similar to the last chapter, we design a series of experiments with various non-linear scenarios to evaluate the predictive ability of our model and its competitors. Important lags that generate the response $y_t$ vary from scenario to scenario. For each scenario, we simulate 10 datasets and 5,000 data points for each dataset.

We generate non-linear datasets by the following rules, with six different scenarios for selecting important lags. There are $K$ important lag(s) $\{y_{t-i_1}, \ldots, y_{t-i_K}\}$; the first 10 data points in each time series are generated by Poisson($\nu_-$); for $t > 10$, if $\sum_{j=1}^{K} y_{t-i_j} \geq K\nu_+$, then $y_t \sim \text{Poisson}(\nu_+)$; else $y_t \sim \text{Poisson}(\nu_-)$.

Again, we consider datasets with different numbers of important lags, and both close and distant important lags are used for generating these time series. We believe the variety of datasets can result in a more comprehensive view on model comparisons. The details of these six different non-linear scenarios are as follows.

(A). Important lag: $y_{t-1}$, $\nu_+ = 30$, $\nu_- = 50$;

(B). Important lag: $y_{t-7}$, $\nu_+ = 30$, $\nu_- = 50$;

(C). Important lags: $y_{t-3}$ & $y_{t-7}$, $\nu_+ = 20$, $\nu_- = 100$;
(D). Important lags: \(y_{t-7} \& y_{t-9}, \nu_+ = 20, \nu_- = 100\);

(E). Important lags: \(y_{t-3} \& y_{t-7} \& y_{t-9}, \nu_+ = 20, \nu_- = 100\);

(F). Important lags: \(y_{t-7} \& y_{t-8} \& y_{t-9}, \nu_+ = 20, \nu_- = 100\).

For example, in (C), there exist 2 important lags \(y_{t-3}\) and \(y_{t-7}\); \((\nu_+, \nu_-) = (20, 100)\); \(\{y_t\}_{t \in \mathbb{Z}_{1,10}}\) are generated by Poisson(20); for \(t > 10\), if \(y_{t-3} + y_{t-7} \geq 40\), then \(y_t \sim \text{Poisson}(20)\); else \(y_t \sim \text{Poisson}(100)\).

The outcome of comparison between two Bayesian models, BaPAR and our BTF models, is shown in Table 6.1. According to the table, our BTF has better predictive performance against BaPAR model, due to its lower corresponding ALPS in all the scenarios except the first one. Our BTF model also outperforms PAR and INGARCH models when MSE is considered the criterion for comparison, as shown in Table 6.2. Hence, our general BTF model can be as accurate as, or more accurate than BaPAR, PAR and INGARCH models in terms of prediction when dealing with non-linear time series of large count.

<table>
<thead>
<tr>
<th>ALPS</th>
<th>Bayesian Poisson autoregression</th>
<th>BTF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scenario</td>
<td>Data Sizes</td>
<td>AIC</td>
</tr>
<tr>
<td>(A) :</td>
<td>4000 : 1000</td>
<td>3.860(0.032)</td>
</tr>
<tr>
<td></td>
<td>4500 : 500</td>
<td>3.860(0.029)</td>
</tr>
<tr>
<td>(B) :</td>
<td>4000 : 1000</td>
<td>3.892(0.056)</td>
</tr>
<tr>
<td></td>
<td>4500 : 500</td>
<td>3.897(0.064)</td>
</tr>
<tr>
<td>(C) :</td>
<td>4000 : 1000</td>
<td>3.615(0.207)</td>
</tr>
<tr>
<td></td>
<td>4500 : 500</td>
<td>3.665(0.222)</td>
</tr>
<tr>
<td>(D) :</td>
<td>4000 : 1000</td>
<td>3.857(0.172)</td>
</tr>
<tr>
<td></td>
<td>4500 : 500</td>
<td>3.822(0.192)</td>
</tr>
<tr>
<td>(E) :</td>
<td>4000 : 1000</td>
<td>3.426(0.030)</td>
</tr>
<tr>
<td></td>
<td>4500 : 500</td>
<td>3.440(0.023)</td>
</tr>
<tr>
<td>(F) :</td>
<td>4000 : 1000</td>
<td>5.338(0.092)</td>
</tr>
<tr>
<td></td>
<td>4500 : 500</td>
<td>5.270(0.120)</td>
</tr>
</tbody>
</table>

Table 6.1: Mean of average log predictive score (ALPS) (with standard deviations in brackets) between Bayesian Poisson autoregressive model and our Bayesian tensor factorisations model (BTF) based on 10 Poisson autoregression generated datasets for each one of 6 Scenarios. AIC and BIC columns indicate that the best model has been chosen with the corresponding criterion. Models with the best performance are highlighted bold.

4 Applications

In this section, we use two real-world applications to show that our BTF model can have fair performance. We compare our Bayesian tensor factorisation model to other three models with two datasets from Google Flu Trends (google.org/flutrends) that refer to weekly flu counts in Norway and Castilla–La Mancha (Spain); see Figure 6.1. Both of the datasets are with 514 data points, respectively. Large assumed maximum lag \(q\) is not
SRoMSE as the criterion, Poisson autoregression leads the predictive performance in the
Table 6.3 indicates that BTF models outperform, in terms of ALPS, Bayesian Poisson
we can still retrieve the signal from the pre-training dataset, even though some of the
Therefore, we choose the maximum lag
Table 6.2: Mean of mean squared error (MSE) (with standard deviations in brackets) among Poisson autoregression,
autoregression models in all scenarios. When taking the square root of mean squared error
in both scenarios are indicated as important. For the Castilla-La Mancha dataset, the first and second lags
first lag in the second scenario (with 462 training data points and 52 testing data points), and the
frequency distribution is higher than 0.5. For the Norway dataset, the first and second lags
Figure 6.2 and 6.3. Note that a lag is selected as important when its corresponding relative
For both datasets, the resulting important lags selected by our BTF model can be found in
labels will be left empty.
recommended for datasets with relatively small length, like the two datasets in this section.
Since with the increase of q, more predictors will be considered and fewer data points will
be used for model training. Under such circumstances, there is a high risk of overfitting.
Therefore, we choose the maximum lag q to be ten for all models we apply to the data.
Our BTF model considers a pre-training data size of 154 for each dataset and best models
for Bayesian Poisson autoregression, Poisson regressive and INGARCH models are based
on AIC and BIC criteria. The last 103 and 52 data points are chosen for comparison for
each dataset.
Both of the training datasets are with the range of around 800, and for the simplicity of
our BTF model, we initially assume the number of clusters c to be 10 for both cases. For
generality, it is reasonable to assume a relatively large number of clusters, like 10, so that
we can still retrieve the signal from the pre-training dataset, even though some of the
labels will be left empty.
For both datasets, the resulting important lags selected by our BTF model can be found in
Figure 6.2 and 6.3. Note that a lag is selected as important when its corresponding relative
frequency distribution is higher than 0.5. For the Norway dataset, the first and second lags
in the first scenario (with 411 training data points and 103 testing data points), and the
first lag in the second scenario (with 462 training data points and 52 testing data points)
are indicated as important. For the Castilla-La Mancha dataset, the first and second lags
in both scenarios are indicated as important.
Table 6.3 indicates that BTF models outperform, in terms of ALPS, Bayesian Poisson
autoregression models in all scenarios. When taking the square root of mean squared error
(SRoMSE) as the criterion, Poisson autoregression leads the predictive performance in the

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Data Sizes</th>
<th>Poisson autoregression</th>
<th>INGARCH</th>
<th>BTF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>AIC</td>
<td>BIC</td>
<td>AIC</td>
</tr>
<tr>
<td>(A) :</td>
<td>4000 : 1000</td>
<td>89.296(3.673)</td>
<td>89.296(3.673)</td>
<td>88.984(3.553)</td>
</tr>
<tr>
<td>(B) :</td>
<td>4000 : 1000</td>
<td>94.212(4.975)</td>
<td>94.247(5.028)</td>
<td>94.639(5.463)</td>
</tr>
<tr>
<td>(C) :</td>
<td>4500 : 500</td>
<td>92.088(6.040)</td>
<td>92.088(6.040)</td>
<td>92.182(6.470)</td>
</tr>
<tr>
<td>(D) :</td>
<td>4000 : 1000</td>
<td>128.918(52.725)</td>
<td>128.918(52.725)</td>
<td>77.461(6.564)</td>
</tr>
<tr>
<td>(E) :</td>
<td>4000 : 1000</td>
<td>127.011(56.587)</td>
<td>128.217(53.440)</td>
<td>79.710(5.387)</td>
</tr>
<tr>
<td>(F) :</td>
<td>4500 : 500</td>
<td>176.141(38.399)</td>
<td>176.133(38.412)</td>
<td>90.255(10.663)</td>
</tr>
</tbody>
</table>

Table 6.2: Mean of mean squared error (MSE) (with standard deviations in brackets) among Poisson autoregression, integer-valued generalised autoregressive conditionally heteroscedastic (INGARCH) model and our Bayesian tensor factorisations model (BTF) based on 10 Poisson autoregression generated data sets for each one of 6 Scenarios. AIC and BIC columns indicate that the best model has been chosen with the corresponding criterion. Models with the best performance are highlighted bold.
Norway dataset among these three models, as shown in Table 6.4. However, our BTF model has significantly lower SRoMSE score against Poisson autoregressive and INGARCH models when modelling the Castilla-La Mancha time series of flu count.

Figure 6.1: Trace plot of 514 time-series data points counting flu cases in Norway and Castilla-La Mancha (CLM), Spain, counted by each week from 09-Oct-2005 to 09-Aug-2015.

Figure 6.2: Result of important lag selection for the Norway flu dataset. Left panel: the inclusion proportions of different lags for the scenario with 411 training data points and 103 testing data points. Right panel: the inclusion proportions of different lags for the scenario with 462 training data points and 52 testing data points. The orders of past lags are in the horizontal and their corresponding relative frequency distributions to be important are in the vertical.

<table>
<thead>
<tr>
<th>ALPS</th>
<th>Bayesian Poisson autoregression</th>
<th>BTF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Country/Region</td>
<td>Data Sizes</td>
<td>AIC</td>
</tr>
<tr>
<td>Norway</td>
<td>411 : 103</td>
<td>7.560(10)</td>
</tr>
<tr>
<td></td>
<td>462 : 52</td>
<td>7.805(9)</td>
</tr>
<tr>
<td>Castilla-La Mancha, Spain</td>
<td>411 : 103</td>
<td>12.295(10)</td>
</tr>
<tr>
<td></td>
<td>462 : 52</td>
<td>15.073(9)</td>
</tr>
</tbody>
</table>

Table 6.3: Average log predictive score (ALPS) between Bayesian Poisson autoregression and our Bayesian tensor factorisations model (BTF) for flu counts datasets in Norway and Castilla-La Mancha, Spain. AIC and BIC columns indicate that the best model has been chosen (in brackets) with the corresponding criterion. Training and testing data sizes appear in the second column. Models with best performance are highlighted bold.
Figure 6.3: Result of important lag selection for the Castilla-La Mancha flu dataset. Left panel: the inclusion proportions of different lags for the scenario with 411 training data points and 103 testing data points. Right panel: the inclusion proportions of different lags for the scenario with 462 training data points and 52 testing data points. The orders of past lags are in the horizontal and their corresponding relative frequency distributions to be important are in the vertical.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Data Sizes</th>
<th>Poisson autoregression</th>
<th>INGARCH</th>
<th>BTF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>AIC</td>
<td>BIC</td>
<td>AIC</td>
</tr>
<tr>
<td>Norway</td>
<td>411 : 103</td>
<td>33.507(10) 33.507(10)</td>
<td></td>
<td>62.610(10, 8) 62.610(10, 8)</td>
</tr>
<tr>
<td></td>
<td>462 : 52</td>
<td>48.896(9) 48.896(9)</td>
<td></td>
<td>53.219(10, 7) 53.219(10, 7)</td>
</tr>
<tr>
<td>Castilla-La Mancha, Spain</td>
<td>411 : 103</td>
<td>78.496(10) 76.380(5)</td>
<td></td>
<td>86.818(10, 10) 86.818(10, 10)</td>
</tr>
<tr>
<td></td>
<td>462 : 52</td>
<td>85.340(9) 85.340(9)</td>
<td></td>
<td>120.296(9, 8) 120.296(9, 8)</td>
</tr>
</tbody>
</table>

Table 6.4: Square root of mean squared error (SRoMSE) (with models selected in brackets) among Poisson autoregression, integer-valued generalised autoregressive conditionally heteroscedastic (INGARCH) model and our Bayesian tensor factorisations model (BTF). AIC and BIC columns indicate that the best model has been chosen with the corresponding criterion. Training and testing data sizes appear in the first column. Models with best performance are highlighted bold.
Chapter 7

Bayesian Tensor Factorisations for Multivariate Time Series of Count Data

In the previous chapters, we introduce our methodology for modelling univariate time series of count data. In the last chapter, we successfully adopt the mixture of Poisson distributions for the data clustering. As an extension and generalisation, we discuss our BTF methodology for modelling multivariate time series of count data in this chapter. We specify the model as a beginning, followed by estimations and inferences. Outcomes of simulation experiments and applications are provided in the end.

1 Model Specification

In this chapter, we consider an $M$-dimensional time-indexed sequence of random variables with size $T$, $\{Y_t\}_{t=1}^T$, where $Y_t = (y_{1,t}, \ldots, y_{M,t})^T$ taking values in $\mathbb{N}$ including zero. Furthermore, we define the filtration of $Y_t$ as $\mathcal{F}_t = \{Y_1:t\}$. Exact as what we have done in the previous chapter, we separate $\{Y_t\}_{t=1}^T$ into two segmentations as $\{Y_t\}_{t \in \mathbb{Z}_{[1,T_1]}}$ the pre-training dataset and $\{Y_t\}_{t \in \mathbb{Z}_{[T_1+1,T_1+T_2]}}$ the training dataset.

Similar to the previous chapter, the pre-training dataset is designed to estimate labels of counts that will be used in model fitting and testing. Define a collection of latent variables
\{w_{m,1:c_m-1}, \mu_{m,1:c_m}, c_m \}_{m \in \mathbb{Z}_{[1,M]}}\},\) we can model the pre-training data as
\[
p(y_{m,t} \mid w_{m,1:c_m-1}, \mu_{m,1:c_m}, c_m) = \sum_{i=1}^{c_m} w_{m,i} \text{PD}(y_{m,t} \mid \mu_{m,i}),
\]
\[
p(t_{m,t} = i) = w_{m,i},
\]
for any \(m \in \mathbb{Z}_{[1,M]}\), \(t \in \mathbb{Z}_{[1,T_1]}\), where \(\text{PD}(y_{m,t} \mid \mu_{m,i})\) denotes the Poisson density of \(y_{m,t}\) with parameter \(\mu_{m,i}\), \(0 < w_{m,i} < 1\), \(\sum_{i=1}^{c_m} w_{m,i} = 1\), \(\mu_{m,i} \geq 0\) and \(t_{m,t} \in \mathbb{Z}_{[1,c_m]}\) represents the label of component \(y_{m,t}\) belongs to in the pre-training dataset.

Let \(d_{m,t}\) denote the estimated label of component that \(y_{m,t}\) belongs to in the training dataset, the latent variable \(d_{m,t}\) can then be generated by a predictive probability mass function as, for any \(t > T_1\),
\[
p(d_{m,t} = i) = w_{m,i}
\]
with \(i \in \mathbb{Z}_{[1,c_m]}\). We set \(d_{m,j,t} = d_{m,t-j}\) for all \(m \in \mathbb{Z}_{[1,M]}\), \(j \in \mathbb{Z}_{[1,q]}\) and \(t \in \mathbb{Z}_{[T_1+1+q,T_1+T_2]}\), and we define another collection of latent variables as \(D_t := \{\{d_{m,j,t}\}_{m \in \mathbb{Z}_{[1,M]}}\}_{j \in \mathbb{Z}_{[1,q]}}\).

For the next step, we build a probabilistic model for the training dataset by assuming that the transition probability law of a target sequence \(\{y_{\tau,t}\}_{t \in \mathbb{Z}_{[T_1+1+q,T_1+T_2]}}\) with \(\tau \in \mathbb{Z}_{[1,M]}\) conditional on \(\mathcal{F}_{t-1}\) is that of a probabilistic model of this target sequence conditional on \(D_t\). That is, for any \(m \in \mathbb{Z}_{[1,M]}\), we have
\[
p(y_{\tau,t} \mid \mathcal{F}_{1:t-1}) = p(y_{\tau,t} \mid D_t).
\]

Similar to what we have done in the previous, we introduce further latent variables \(k_{m,j}\) with values in \(\mathbb{Z}_{[1,c_m]}\) that determine two things: first, if \(k_{m,j} = 1\) then \(d_{m,j,t}\) does not affect \(y_{\tau,t}\); any other value of \(k_{m,j}\) determines how many values of \(d_{m,j,t}\) affect \(y_{\tau,t}\). Thus, conditional on \(k_{m,j}\), the tensor factorisation parameter space is reduced to \(\prod_{m \in \mathbb{Z}_{[1,M]}} \prod_{j \in \mathbb{Z}_{[1,q]}} k_{m,j}\).

We define other three collections of random variables, that are used in our model, as a time-dependent random variables collection \(Z_t := \{\{z_{m,j,t}\}_{j \in \mathbb{Z}_{[1,q]}}\}_{m \in \mathbb{Z}_{[1,M]}}\) as well as a time-independent random variables collection \(K := \{\{k_{m,j}\}_{j \in \mathbb{Z}_{[1,q]}}\}_{m \in \mathbb{Z}_{[1,M]}}\) and a collection of indices \(H := \{\{h_{m,j}\}_{j \in \mathbb{Z}_{[1,q]}}\}_{m \in \mathbb{Z}_{[1,M]}}\). Furthermore, we define \(\mathcal{H} := \{h_{m,j} \in \mathbb{Z}_{[1,k_{m,j}]}, m \in \mathbb{Z}_{[1,M]}, j \in \mathbb{Z}_{[1,q]}\}\).

The conditional transition probability law (7.1) can then be written as a Bayesian hierar-
Bayesian tensor factorisations for multivariate time series of count data

A statistical model, for \( j \in \mathbb{Z}_{[1,q]} \), \( m \in \mathbb{Z}_{[1,M]} \), \( H \in \mathcal{H} \) and \( t \in \mathbb{Z}_{[T_1+1+q,T_1+T_2]} \), as

\[
y_{\tau,t} \mid Z_t = H \sim \text{Poisson}(\lambda_H), \tag{7.2}
\]

\[
z_{m,j,t} \mid d_{m,j,t} \sim \text{Multinomial}\left(\mathbb{Z}_{[1,k_{m,j}]}, \{\pi^{(m,j)}_1(d_{m,j,t}), \ldots, \pi^{(m,j)}_{k_{m,j}}(d_{m,j,t})\}\right). \tag{7.3}
\]

(7.2) and (7.3) immediately imply that

\[
p(y_{\tau,t} \mid D_t) = \sum_{H \in \mathcal{H}} \text{PD}(y_{\tau,t}; \lambda_H) \prod_{m \in \mathbb{Z}_{[1,M]}} \prod_{j \in \mathbb{Z}_{[1,q]}} \pi^{(m,j)}_{k_{m,j}}(d_{m,j,t}).
\]

with constraints

\[
\lambda_H \geq 0, \quad \text{for any } H \in \mathcal{H},
\]

\[
\sum_{h_{m,j} = 1}^{k_{m,j}} \pi^{(m,j)}_{h_{m,j}}(d_{m,j,t}) = 1, \quad \text{for each combination of } (m, j, d_{m,j,t}).
\]

Exact as what has been done in the previous chapters, we use the stick-breaking construction for \( \lambda_H \) as

\[
\lambda_H \sim \sum_{l=1}^{\infty} \pi^*_l \delta(\lambda^*_l),
\]

where \( \delta(\cdot) \) is an indicator function, and independently, for \( l \in \mathbb{Z}_{[1,\infty)} \), \( \lambda^*_l \sim \text{Gamma}(a, b) \), \( \pi^*_l = V_l \prod_{s=1}^{l-1} (1 - V_s) \), \( V_l \sim \text{Beta}(1, \alpha_0) \) representing a label-clustered Poisson rate, which is generated by a Gamma distribution. Moreover, \( Z^*_t \) is defined to label \( Z_t \).

Similar to Chapter 5, we can assign independent priors on \( \pi^{(m,j)}(d_{m,j,t}) \) as

\[
\pi^{(m,j)}(d_{m,j,t}) = \{\pi^{(m,j)}_1(d_{m,j,t}), \ldots, \pi^{(m,j)}_{k_{m,j}}(d_{m,j,t})\} \sim \text{Dirichlet}(\gamma_{m,j}, \ldots, \gamma_{m,j}).
\]

and priors on \( k_{m,j} \) as

\[
p(k_{m,j} = \kappa_m) \propto \exp(-\varphi j \kappa_m),
\]

where \( j \in \mathbb{Z}_{[1,q]} \), \( \kappa_m \in \mathbb{Z}_{[1,c_m]} \). We follow the prior settings in Section 1 of Chapter 5.

## 2 Estimation and Inference

Again, we propose a three-stage procedure for sampling variables in our BTF model. The first stage is to estimate the label of group that a count on time series belongs to, or in other words, to estimate \( d_{m,j,t} \) for any \( m \in \mathbb{Z}_{[1,M]} \), \( j \in \mathbb{Z}_{[1,q]} \) and any \( t > T_1 \). The second
stage is to estimate \( K \) by using the training dataset and estimated \( D \). This stage can eliminate the uncertainty of \( K \), and thus improve the efficiency of our model. The third stage is to sample all the other random variables through Gibbs sampling, by data from the training dataset as well as estimated \( K \) and \( D \).

2.1 First Stage – Estimation of \( D \)

Similar to last chapter, we consider the \( m \)-th time series \( \{y_{m,t}\}_{t \in \mathbb{Z}[1,T_1]} \) a mixture of \( c_m \) univariate Poisson distributions with \( \sum_{i=1}^{c_m} w_{m,i} \text{PD}(y_{m,t}, \mu_{m,i}) \), where \( \sum_{i=1}^{c_m} w_{m,i} = 1 \) and \( \{\mu_{m,i}\}_{i \in \mathbb{Z}[1,c_m]} \) for \( m \in \mathbb{Z}[1,M] \). By using the Gibbs sampler described in Section 2.1 of Chapter 6, we can then estimate \( d_{m,t} \) as \( \text{arg}\max_i \text{PD}(y_{m,t}, \mu_{m,i}) \) for any \( t > T_1 \) and \( m \in \mathbb{Z}[1,M] \).

2.2 Second Stage – Estimation of \( K \)

We have introduced how to indicate whether a predictor is important or not according to the corresponding estimated value of its inclusion in the previous chapters. We generalise the idea for multivariate time series in this chapter, that is to use the value of \( k_{m,j} \) to infer whether the predictor \( d_{m,j,t} \) is important or not. For any \( m \in \mathbb{Z}[1,M] \) and \( j \in \mathbb{Z}[1,q] \), when \( k_{m,j} = 1 \), \( d_{m,j,t} \) is not important for \( y_{\tau,t} \); while when \( k_{m,j} > 1 \), \( d_{m,j,t} \) is important for \( y_{\tau,t} \). A similar Metropolis-Hastings algorithm is described as (5.13) in Section 2.1 of Chapter 5 for sampling \( K \).

2.3 Third Stage – Posteriors of Random Variables

The last stage is to sample the other random variables through posterior computations conditional on the estimated \( K \) and \( D \). A similar procedure of updating the rest of random variables conditional on \( D \) and \( K \) is described in Section 2.2 of Chapter 5.

3 Simulation Experiments

Criteria and Competitors

We apply the same criteria and competitors as what we have done in the previous chapters for comparing the predictive performance of our BTF model. Recall that the average log predictive score (ALPS) is the criterion when comparing against the Bayesian autoregressive
(BaPAR) model and the maximum likelihood estimation (MLE) is the one for comparing with the Poisson autoregressive (PAR) and integer-valued generalised autoregressive conditionally heteroscedastic (INGARCH) models. We divide simulated datasets into two training: testing ratios, which are 8 : 2 and 9 : 1, respectively, for further comparison.

**Settings for Models**

Similar to the last chapters, the assumed most distant important lag $q$ is set to be the actual most distant important lag plus two throughout simulation experiments. For example, if the most distant lag that $y_{1,t}$ depends on is $y_{2,t-8}$, then $q$ is set to be 10. We assume the number of clusters $c_m$ to be 10 for $m \in \mathbb{Z}_{[1,M]}$. Akaike information criterion (AIC) and Bayesian information criterion (BIC) are used for model selection for PAR, BaPAR and INGARCH models.

Recall that the MCMC runs of BaPAR model are bases on 5,000 burn-in samples and 10,000 posterior samples, respectively. A multivariate BaPAR model follows the expression in (3.12). For each $m \in M$, we set the priors for parameters as $\beta_{m,0} \sim N(0, 10^{-6})$, $\beta_{m,i} \sim N(0, 10^{-4})$ for any $i \in \mathbb{Z}_{[1,q]}$ and $\eta_{\ell} \sim N\left(\frac{1}{M-1}, \frac{1}{M-1}\right)$ for any $\ell \neq m$. All MCMC runs for our BTF model are based on the following burn-in and posterior samples respectively: For the first stage for our simple BTF model, 1,000 and 2,000 for selecting the important lags and their corresponding number of inclusions; and, for the second stage, 2,000 and 5,000 for sampling the rest of the parameters. The thinning rate for the last stage is set to be 10.

Recall that we utilise rjags [Plummer et al., 2016] package in R for the MCMC inference of BaPAR model. Multivariate PAR and INGARCH models follow the expression in (3.12) and (3.13), respectively, and they are trained through the package tscount [Liboschik et al., 2017] in R.

**Details of Datasets**

We design a series of experiments with various non-linear scenarios to evaluate the predictive ability of our model and its competitors. We assume that we are interested in predicting $y_{1,t}$, or in other words, $\tau = 1$. Important lags that generate the response $y_{1,t}$ vary from scenario to scenario. For each scenario, we simulate 10 datasets and 5,000 data points for each dataset.
We generate non-linear datasets by the following rules, with six different scenarios for selecting important lags. There are $K$ important lag(s) $\{y_{m,t-i_1}, \ldots, y_{m,K,t-i_K}\}$ for the target time series $\{y_{1,t}\}$; the first 10 data points in each time series are generated by $\text{Poisson}(\nu_-)$; for $t > 10$, if $\sum_{j=1}^{K} y_{m,j,t-i_j} \geq \nu_+$, then $y_{1,t} \sim \text{Poisson}(\nu_+)$; else $y_{1,t} \sim \text{Poisson}(\nu_-)$.

The details of these six different non-linear scenarios are as follows.

(A). $M = 2; \nu_- = 20, \nu_+ = 10$; Non-zero coefficient for $y_{1,t}$: $y_{2,t-1}$; No non-zero coefficient for $y_{2,t}$.

(B). $M = 2; \nu_- = 20, \nu_+ = 10$; Non-zero coefficients for $y_{1,t}$: $y_{1,t-3}$ & $y_{2,t-5}$; No non-zero coefficient for $y_{2,t}$.

(C). $M = 2; \nu_- = 20, \nu_+ = 10$; Non-zero coefficient for $y_{1,t}$: $y_{2,t-1}$; Non-zero coefficient for $y_{2,t}$: $y_{1,t-2}$.

(D). $M = 2; \nu_- = 20, \nu_+ = 10$; Non-zero coefficients for $y_{1,t}$: $y_{1,t-3}$ & $y_{2,t-4}$; Non-zero coefficients for $y_{2,t}$: $y_{1,t-1}$ & $y_{2,t-3}$ & $y_{2,t-5}$.

(E). $M = 3; \nu_- = 20, \nu_+ = 10$; Non-zero coefficient for $y_{1,t}$: $y_{2,t-1}$; Non-zero coefficients for $y_{2,t}$: $y_{3,t-2}$ & $y_{2,t-3}$ & $y_{2,t-5}$; Non-zero coefficient for $y_{3,t}$: $y_{1,t-3}$.

(F). $M = 3; \nu_- = 60, \nu_+ = 20$; Non-zero coefficients for $y_{1,t}$: $y_{1,t-3}$& $y_{2,t-4}$& $y_{3,t-1}$; Non-zero coefficients for $y_{2,t}$: $y_{1,t-1}$& $y_{2,t-2}$& $y_{3,t-5}$; Non-zero coefficients for $y_{3,t}$: $y_{1,t-3}$& $y_{2,t-2}$& $y_{3,t-5}$.

The outcome of comparison between two Bayesian models, BaPAR and our BTF models, is shown in Table 7.1. According to the table, our BTF has better predictive performance against BaPAR model, due to its lower corresponding ALPS in all the scenarios except the first one. Our BTF model also outperforms PAR and INGARCH models when MSE is considered the criterion for comparison, as shown in Table 7.2.

4 Application

In this section, we analyse flu datasets from five of the regions in south-eastern Spain, which are Andalusia, Castilla-La Mancha, Illes Balears, Region de Murcia and Comunitat Valenciana. Datasets are from Google Flu Trends (google.org/flutrends). We model each time series with covariate effects. Map of five regions is shown in Figure 7.1, and the
distance between any two of regions is shown in Table 7.3. Datasets are collected from 09-Oct-2005 to 09-Aug-2015 by each week, with 514 data points for each dataset. The trace plot of these five time series are shown in Figure 7.2.

<table>
<thead>
<tr>
<th>A</th>
<th>CLM</th>
<th>IB</th>
<th>RM</th>
<th>VC</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>350</td>
<td>809</td>
<td>373</td>
</tr>
<tr>
<td>CLM</td>
<td>350</td>
<td>0</td>
<td>565</td>
<td>213</td>
</tr>
<tr>
<td>IB</td>
<td>809</td>
<td>565</td>
<td>0</td>
<td>498</td>
</tr>
<tr>
<td>RM</td>
<td>373</td>
<td>213</td>
<td>498</td>
<td>0</td>
</tr>
<tr>
<td>VC</td>
<td>539</td>
<td>298</td>
<td>248</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 7.3: Distance between regions in kilometre. A = Andalusia, CLM = Castilla-La Mancha, IB = Illes Balears, RM = Región de Murcia and VC = Comunitat Valenciana. Information found on Open Street Map (openstreetmap.org).

Similar to the assumptions made in last chapter, we choose the maximum lag q to be 10
for all multivariate BTF models we apply to the data, and assume the number of clusters $c_m$ to be 10 for $m \in \mathbb{Z}_{[1,M]}$. The sizes of training against testing dataset are 411 : 103 and 462 : 52 respectively. Our BTF considered the first 154 data points as the pre-training dataset.

The results in Table 7.4 show that our Bayesian tensor factorisation model outperforms Bayesian Poisson autoregression significantly in terms of predicting by using the ALPS criterion in seven out of ten scenarios. In the second scenario of Illes Balears, both models selected by the Bayesian Poisson autoregression and our BTF generate close ALPS, hence they have almost equal performance in terms of prediction by modelling this dataset. For model comparison with the criterion of square root of mean squared error (SRoMSE) in Table 7.5, our Bayesian tensor factorisation model also shows fair and better predictive performance against Poisson autoregression and INGARCH model in six out of ten scenarios. Important lags selected by our BTF are in Table 7.6 and 7.7, respectively. Their corresponding relative frequency distributions of selecting important predictors for these five time series by our BTF model are shown in Figure 7.3 and 7.4, respectively.
CHAPTER 7. BAYESIAN TENSOR FACTORISATIONS FOR MULTIVARIATE TIME SERIES OF COUNT DATA

Figure 7.2: Trace plot of 514 time-series data points counting flu cases in Andalusia, Castilla-La Mancha, Illes Balears, Region de Murcia and Comunitat Valenciana counted by each week from 09-Oct-2005 to 09-Aug-2015.

<table>
<thead>
<tr>
<th>Region</th>
<th>Data Sizes</th>
<th>ALPS</th>
<th>Bayesian Poisson autoregression</th>
<th>BTF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Andalucia</td>
<td>411 : 103</td>
<td>9.097(10)</td>
<td>9.296(4)</td>
<td>17.342</td>
</tr>
<tr>
<td></td>
<td>462 : 52</td>
<td>17.550(4)</td>
<td>17.550(4)</td>
<td>14.019</td>
</tr>
<tr>
<td>Castilla-La Mancha</td>
<td>411 : 103</td>
<td>14.589(7)</td>
<td>14.467(5)</td>
<td>5.810</td>
</tr>
<tr>
<td></td>
<td>462 : 52</td>
<td>23.765(10)</td>
<td>23.708(9)</td>
<td>5.998</td>
</tr>
<tr>
<td>Illes Balears</td>
<td>411 : 103</td>
<td>14.334(10)</td>
<td>14.297(3)</td>
<td>4.349</td>
</tr>
<tr>
<td></td>
<td>462 : 52</td>
<td>6.788(10)</td>
<td>7.019(5)</td>
<td>5.331</td>
</tr>
<tr>
<td>Region de Murcia</td>
<td>411 : 103</td>
<td>25.593(10)</td>
<td>25.379(8)</td>
<td>10.680</td>
</tr>
<tr>
<td></td>
<td>462 : 52</td>
<td>5.771(10)</td>
<td>5.771(10)</td>
<td>15.092</td>
</tr>
<tr>
<td>Comunitat Valenciana</td>
<td>411 : 103</td>
<td>13.760(10)</td>
<td>14.601(8)</td>
<td>6.027</td>
</tr>
<tr>
<td></td>
<td>462 : 52</td>
<td>21.532(10)</td>
<td>21.532(10)</td>
<td>6.534</td>
</tr>
</tbody>
</table>

Table 7.4: Average log predictive score (ALPS) between Bayesian Poisson autoregressive model and Bayesian tensor factorisations model (BTF) for multivariate flu counts datasets (with models selected in brackets). Multiple datasets include flu counts in Andalusia, Castilla-La Mancha, Illes Balears, Region de Murcia and Comunitat Valenciana. AIC and BIC columns indicate that the best model has been chosen with the corresponding criterion. Models with best performance are highlighted bold.
Table 7.5: Square root of mean squared error (SRoMSE) (with models selected in brackets) among Poisson autoregression, integer-valued generalised autoregressive conditionally heteroscedastic (INGARCH) model and our Bayesian tensor factorisations model (BTF) for multivariate flu counts datasets. Multiple datasets include flu counts in Andalusia, Castilla-La Mancha, Illes Balears, Region de Murcia and Comunitat Valenciana. AIC and BIC columns indicate that the best model has been chosen with the corresponding criterion. Models with best performance are highlighted bold.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Data Sizes</th>
<th>AIC</th>
<th>BIC</th>
<th>AIC</th>
<th>BIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Andalucia</td>
<td>411 : 103</td>
<td>50.357(9)</td>
<td>49.796(4)</td>
<td>61.917(9.8)</td>
<td>61.917(9.8)</td>
</tr>
<tr>
<td>Castilla-La Mancha</td>
<td>411 : 103</td>
<td>104.974(7)</td>
<td>1003.809(5)</td>
<td>5766.902(9.6)</td>
<td>5766.902(9.6)</td>
</tr>
<tr>
<td>Illes Balears</td>
<td>411 : 103</td>
<td>298.491(10)</td>
<td>320.038(3)</td>
<td>363.076(10.7)</td>
<td>363.076(10.7)</td>
</tr>
<tr>
<td>Region de Murcia</td>
<td>411 : 103</td>
<td>42.948(10)</td>
<td>45.580(5)</td>
<td>70.488(9.5)</td>
<td>70.488(9.5)</td>
</tr>
<tr>
<td>Comunitat Valenciana</td>
<td>411 : 103</td>
<td>455.822(8)</td>
<td>455.822(8)</td>
<td>577.468(9.10)</td>
<td>1048.854(6.10)</td>
</tr>
</tbody>
</table>

Table 7.6: Important predictors selected for each time series by our BTF model for the scenario with 411 training data points and 52 testing data points. A = Andalusia, CLM = Castilla-La Mancha, IB = Illes Balears, RM = Region de Murcia and VC = Comunitat Valenciana.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Data Sizes</th>
<th>A</th>
<th>CLM</th>
<th>IB</th>
<th>RM</th>
<th>VC</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>$d_{1,t}$, $d_{2,t}$, $d_{2,2,t}$</td>
<td>$d_{1,t}$, $d_{2,t}$, $d_{2,2,t}$</td>
<td>$d_{4,1,t}$</td>
<td>$d_{5,1,t}$, $d_{5,2,t}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLM</td>
<td>$d_{1,t}$, $d_{2,t}$, $d_{2,2,t}$</td>
<td>$d_{1,t}$, $d_{2,t}$, $d_{2,2,t}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IB</td>
<td>$d_{1,t}$, $d_{2,t}$, $d_{2,2,t}$, $d_{3,1,t}$, $d_{3,2,t}$</td>
<td>$d_{4,1,t}$</td>
<td>$d_{5,1,t}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RM</td>
<td>$d_{1,t}$, $d_{2,t}$, $d_{2,2,t}$</td>
<td>$d_{4,1,t}$</td>
<td>$d_{5,1,t}$, $d_{5,2,t}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VC</td>
<td>$d_{1,t}$, $d_{2,t}$, $d_{2,2,t}$</td>
<td>$d_{5,1,t}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 7.7: Important predictors selected for each time series by our BTF model for the scenario with 462 training data points and 103 testing data points. A = Andalusia, CLM = Castilla-La Mancha, IB = Illes Balears, RM = Region de Murcia and VC = Comunitat Valenciana.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Data Sizes</th>
<th>A</th>
<th>CLM</th>
<th>IB</th>
<th>RM</th>
<th>VC</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>$d_{1,t}$, $d_{2,t}$, $d_{2,2,t}$</td>
<td>$d_{1,t}$, $d_{2,t}$, $d_{2,2,t}$</td>
<td>$d_{4,1,t}$</td>
<td>$d_{5,1,t}$, $d_{5,2,t}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLM</td>
<td>$d_{1,t}$, $d_{2,t}$, $d_{2,2,t}$</td>
<td>$d_{1,t}$, $d_{2,t}$, $d_{2,2,t}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IB</td>
<td>$d_{1,t}$, $d_{2,t}$, $d_{2,2,t}$, $d_{3,1,t}$, $d_{3,2,t}$</td>
<td>$d_{4,1,t}$</td>
<td>$d_{5,1,t}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RM</td>
<td>$d_{1,t}$, $d_{2,t}$, $d_{2,2,t}$</td>
<td>$d_{4,1,t}$, $d_{4,2,t}$</td>
<td>$d_{5,1,t}$, $d_{5,2,t}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VC</td>
<td>$d_{1,t}$, $d_{2,t}$, $d_{2,2,t}$</td>
<td>$d_{5,1,t}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure 7.3: Result of important lag selection for the south-eastern Spain flu dataset: the inclusion proportions of different lags for the scenario with 411 training data points and 103 testing data points. Vertical region: target region; Horizontal region: corresponding covariant region. For each panel, the orders of past lags are in the horizontal and their corresponding relative frequency distributions to be important are in the vertical. A: Andalusia; CLM: Castilla-La Mancha; IB: Illes Balears; RM: Region de Murcia; VC: Comunitat Valenciana.
Figure 7.4: Result of important lag selection for the south-eastern Spain flu dataset: the inclusion proportions of different lags for the scenario with 462 training data points and 52 testing data points. Vertical region: target region; Horizontal region: corresponding covariant region. For each panel, the orders of past lags are in the horizontal and their corresponding relative frequency distributions to be important are in the vertical. A: Andalusia; CLM: Castilla-La Mancha; IB: Illes Balears; RM: Region de Murcia; VC: Comunitat Valenciana.
Chapter 8

Bayesian Tensor Factorisations for Poisson Regression

In the previous chapters, we discuss our BTF model on both univariate and multivariate time series of count data. All simulated experiments as well as real-world applications indicate that our BTF model can have fair predictive performance for modelling both univariate and multivariate time series of count data. In this chapter, we develop our BTF model to deal with general regressive problems of count data. Model is specified in the first part, followed by estimation and inference. An application is provided in the end of this chapter.

1 Model Specification

Suppose there is a dataset with size $T$, and we consider some $y_t$ is the response with its corresponding $q$ predictors $x_t^{(1)}, \ldots, x_t^{(q)}$ for $t \in \mathbb{Z}_{[1,T]}$. As what we have done in the previous chapters, we separate the whole dataset into two segmentations, which are pre-training data and training data. The sizes of pre-training and training data are $T_1$ and $T_2$ respectively. We define $x_t := \{x_t^{(1)}, \ldots, x_t^{(q)}\}$, then $\{y_t, x_t\}_{t \in \mathbb{Z}_{[1,T_1]}}$ is the pre-training dataset, with its counterpart training dataset $\{y_t, x_t\}_{t \in \mathbb{Z}_{[T_1+1,T_1+T_2]}}$.

Similar to what we have done for time series in the previous chapters, for the first step, we define a collection of latent variables $\{u_{1,cj}^{(j)}, \mu_{1,cj}^{(j)}, c_j\}_{j \in \mathbb{Z}_{[1,q]}}$ that models the pre-training
data as
\[ p(x_t^{(j)} | w_{1,c_j-1}^{(j)}, \mu_1^{(j)}, c_j) = \sum_{i=1}^{c_j} w_i^{(j)} PD(x_t^{(j)}; \mu_i^{(j)}), \quad (8.1) \]
\[ p(x_t^{(j)} = i) = w_i^{(j)}, \]
for any \( j \in \mathbb{Z}_{[1,q]}, t \in \mathbb{Z}_{[1,T_1]} \), where \( PD(x_t^{(j)}; \mu_i^{(j)}) \) denotes the Poisson density of \( x_t^{(j)} \) with parameter \( \mu_i^{(j)} \), \( 0 < w_i^{(j)} < 1, \sum_{i=1}^{c_j} w_i^{(j)} = 1, \mu_i^{(j)} \geq 0 \) and \( t_i^{(j)} \in \mathbb{Z}_{[1,c_j]} \) represents the label of component \( x_t^{(j)} \) belongs to in the pre-training dataset. Thus, \((8.1)\) assumes that any \( x_t^{(j)} \) in the pre-training dataset is distributed as a finite mixture of Poisson distributions with \( c_j \) components, weights \( w_i^{(j)} \) and intensities \( \mu_i^{(j)} \).

Similarly, if new observations of \( x_t^{(j)} \) for all \( j \in \mathbb{Z}_{[1,q]} \) in the training dataset follow the finite mixture of Poisson distributions in \((8.1)\), then we can have a latent variable \( d_t^{(j)} \) representing the estimated label of component that \( x_t^{(j)} \) belongs to. The latent variable \( d_t^{(j)} \) can be generated by a predictive probability mass function as, for any \( t > T_1 \),
\[ p(d_t^{(j)} = i) = w_i^{(j)}, \quad (8.2) \]
with \( i \in \mathbb{Z}_{[1,c_j]} \). Furthermore, we define \( D_t := \{d_t^{(j)}\}_{j \in \mathbb{Z}_{[1,q]}} \).

Similarly, for the next step, we build a probabilistic model for the training dataset by assuming that the transition probability law of the response \( \{y_t\}_{t \in \mathbb{Z}_{[T_1+1,T_1+T_2]}} \) conditional on \( x_t \) is that of a probabilistic model of this response conditional on \( D_t \). In other words,
\[ p(y_t \mid x_t) = p(y_t \mid D_t). \quad (8.3) \]

Following the previous chapters, we introduce further latent variables \( k_j \) with values in \( \mathbb{Z}_{[1,c_j]} \) that determine two things: first, if \( k_j = 1 \), then \( y_t \) is independent of \( d_t^{(j)} \); any other values of \( k_j \) determines how many values of \( d_t^{(j)} \) affect \( y_t \).

Again, we define other three collections of random variables, that are used in our model, as a time-dependent random variables collection \( Z_t := \{z_t^{(j)}\}_{j \in \mathbb{Z}_{[1,q]}} \) as well as a time-independent random variables collection \( K := \{k_j\}_{j \in \mathbb{Z}_{[1,q]}} \) and a collection of indices \( H := \{h_j\}_{j \in \mathbb{Z}_{[1,q]}} \).

Furthermore, we define \( H := \{h_j \in \mathbb{Z}_{[1,k]}, j \in \mathbb{Z}_{[1,q]} \} \).

The conditional transition probability law \((8.3)\) can then be written as a Bayesian hierarchical model, for \( j \in \mathbb{Z}_{[1,q]} \), \( H \in \mathcal{H} \) and \( t \in \mathbb{Z}_{[T_1+1,T_1+T_2]} \), as
\[ y_t \mid Z_t = H \sim \text{Poisson}(\lambda_H), \quad (8.4) \]
\[
\begin{aligned}
z_t^{(j)} \mid d_t^{(j)} & \sim \text{Multinomial} \left( \theta_{[1,k]}, \{ \pi_1^{(j)}(d_t^{(j)}), \ldots, \pi_k^{(j)}(d_t^{(j)}) \} \right). \\
(8.5)
\end{aligned}
\]

(8.4) and (8.5) immediately imply that
\[
p(y_t \mid D_t) = \sum_{H \in \mathcal{H}} \text{PD}(y_t; \lambda_H) \prod_{j \in \mathbb{Z}[1,q]} \pi_{h_j}^{(j)}(d_t^{(j)}),
\]
with constraints
\[
\begin{align}
\lambda_H & \geq 0, \quad \text{for any } H \in \mathcal{H}, \\
\sum_{h_j=1}^{k_j} \pi_{h_j}^{(j)}(d_t^{(j)}) & = 1, \quad \text{for each combination of } (j, d_t^{(j)}).
\end{align}
\]

Independently, for each \( H \in \mathcal{H} \), we use the stick-breaking construction as
\[
\lambda_H \sim \sum_{l=1}^{\infty} \pi_l^{*} \delta(\lambda_l^{*}),
\]
where \( \delta(.) \) is an indicator function, and independently, for \( l \in \mathbb{Z}[1,\infty] \), \( \lambda_l^{*} \sim \text{Gamma}(a,b) \), \( \pi_l^{*} = V_l \prod_{s=1}^{l-1}(1 - V_s), V_l \sim \text{Beta}(1,\alpha_0) \) representing a label-clustered Poisson rate, which is generated by a Gamma distribution. The prior settings can be found in Section 1 of Chapter 5.

2 Estimation and Inference

Similar to the previous chapters, we propose a three-stage procedure for sampling variables in our BTF model. The first stage is to estimate the label of group that a count in the dataset belongs to, or in other words, to estimate \( d_t^{(j)} \) for any \( j \in \mathbb{Z}[1,q] \) and any \( t > T_1 \). The second stage is to estimate \( K \) by using the training dataset and estimated \( D \). This stage can eliminate the uncertainty of \( K \), and thus improve the efficiency of our model. The third stage is to sample all the other random variables through Gibbs sampling, by data from the training dataset as well as estimated \( K \) and \( D \).

2.1 First Stage – Estimation of \( D \)

Similarly, we consider the \( j \)-th variable \( \{x_t^{(j)}\}_{t \in \mathbb{Z}[1,T_1]} \) as a mixture of \( c_j \) univariate Poisson distributions with density \( \sum_{i=1}^{c_j} w_i^{(j)} \text{PD}(x_t^{(j)}; \mu_i^{(j)}) \). \( \{w_i^{(j)}\}_{i \in \mathbb{Z}[1,c_j]} \) are some weights with
\[ \sum_{i=1}^{c_j} w_i^{(j)} = 1 \text{ and } \{ \mu_i^{(j)} \}_{i \in Z_{c_j}} \text{ are some Poisson rates. A similar procedure can be found in Section 2.1, Chapter 6. For simplicity and convenience, instead of sampling } d_t^{(j)} \text{ from } (8.2), \text{ we estimate } d_t^{(j)} \text{ as } \arg \max_{i} PD(x_t^{(j)}, \mu_i^{(j)}) \text{ for any } t > T_1. \]

### 2.2 Second Stage – Estimation of K

Like what we have done in the previous chapters for modelling time series of count data, the step of estimating \( K \) before any other variables has two main advantage, including facilitating the efficiency of the model and the inference to which predictors are important. A similar approximate sampler for estimating \( K \) can be found in Section 2.1, Chapter 5. Replacing \( y_{t-j} \) by \( d_t^{(j)} \) and setting \( t \in Z_{[T_1+1+q,T_1+T_2]} \), we can apply the approach to approximate \( K \) that we need in this section.

### 2.3 Third Stage – Posteriors of Random Variables

The last stage is to sample the other random variables through posterior computations conditional on the estimated \( K \) and \( D \). A similar procedure of updating the rest of random variables, described in Section 2.2 of Chapter 5, can be utilised with an extra step to substitute \( d_t^{(j)} \) for \( y_{t-j} \) and set \( t \in Z_{[T_1+1+q,T_1+T_2]} \).

### 3 Application to COVID-19

In this application, we use the dataset of COVID-19 to test if our BTF model has better performance than other models for Poisson regression. We consider the response \( y_t \) as the death cases at one day in a country/territory/area, and the predictors \( \{ x_t^{(1)}, \ldots, x_t^{(q)} \} \) as the daily confirmed cases in the past \( q \) days. Comparison models are our BTF, the Bayesian generalised linear model (BaGLM) and generalised additive model (GAM).

European Centre for Disease Prevention and Control records the data of daily confirmed cases and deaths for almost every country/territory/area. We use the datasets in 4 countries, including USA, UK, Brazil and France, updated from 1st March 2020 to 25th July 2020. For the BTF model, data from 1st March to 30th April and that from 1st May to 15th July are considered as the pre-training and training dataset, respectively. On the contrary, the GLM and GAM models use data from 1st March to 15th July, as the training data and that from 16th July to 25th July as the testing data.
We consider the confirmed cases in the past ten days as predictors, or in other words, \( q = 10 \). For simplicity, the number of clusters is assumed to be 6, or in other words, \( c_j = 6 \) for \( j \in \mathbb{Z}_{[1,q]} \). We set the priors \( \alpha_0 = 1, b = 1 \) and \( a \) to be approximately equal to the midrange of observed dataset, according to the suggestion from Viallefont et al. [2002].

Suppose that we have the response \( y_t \) and \( q \) predictors \( \{x_t^{(1)}, \ldots, x_t^{(q)}\} \), a BaGLM with a log link function can be expressed as

\[
y_t = \text{Poisson}(\lambda_t) \\
\log(\lambda_t) = \beta_0 + \sum_{i=1}^{q} \beta_i \log(x_t^{(i)} + 1)
\]  

(8.6)

The MCMC runs of GLM model are based on 5,000 burn-in samples and 10,000 posterior samples, respectively. Death cases in these four countries vary from two to four digits, therefore, we can set the prior belief for the intercept parameter \( \beta_0 \sim N(5, 3^2) \) as \( \exp(2) \approx 10 \) and \( \exp(8) \approx 3,000 \). The other parameters are highly unlikely to be far from zero, then one can set the priors \( \beta_i \sim N(0, 10^{-4}) \) for any \( i \in \mathbb{Z}_{[1,q]} \). All MCMC runs for our BTF model are based on the following burn-in and posterior samples respectively: For the first stage for our simple BTF model, 1,000 and 2,000 for selecting the important lags and their corresponding number of inclusions; and, for the second stage, 2,000 and 5,000 for sampling the rest of the parameters. The thinning rate for the last stage is set to be 10.

ALPS is considered for Bayesian models comparison in this thesis. Recall that the mathematical definition of ALPS can be expressed as

\[
-\sum_{t \in T'} \sum_{i=1}^{N} \log \hat{p}^{(i)}(y_t) / T'N,
\]

where \( T' \) represents the testing data points, \( T' \) is the size of \( T' \) and \( \hat{p}^{(i)}(\cdot) \) is the one-step ahead estimated transition probability by each model at the i-th iteration of MCMC with total \( N \) iterations. \( -\log \hat{p}^{(i)}(y_t) \) can reach infinity when the \( t \)-th testing data point is too far away from its estimated counterpart, which leads ALPS \( \to \infty \), therefore a threshold needs to be set as

\[
-\log \hat{p}^{(i)}(y_t) = 500 \quad \text{if} \quad -\log \hat{p}^{(i)}(y_t) \geq 500.
\]

A generalised additive model (GAM) was initially invented in Hastie and Tibshirani [1990], and it can be regarded as a subsequent stage for generalisation. A GAM structure can be
written as
\[
y = \text{Poisson}(\lambda)
\]
\[
g(\lambda) = \beta_0 + s_1(x_1) + s_2(x_2) + \cdots + s_q(x_q) = \beta_0 + \sum_{i=1}^{q} s_i(x_i), \tag{8.7}
\]
where \(s_1(\cdot), \ldots, s_q(\cdot)\) denote the smooth functions. According to Larsen [2015], there are three advantages of GAMs: (i) It is easy to be interpreted, (ii) those flexible predictor functions can uncover hidden patterns in the data and (iii) regularisation of predictor functions helps avoid overfitting. Smooth functions can fit both linearity and non-linearity, and it is clear that (8.7) is a generalisation of (8.6). For a GAM, there are three main classes of smoothers, including the local regression, smoothing splines and regressive splines. In this section, we use the regressive splines for GAMs, due to they are computationally cheap and independent on the response.

We compare our BTF model with the GAM model in terms of MSE. Recall that the mathematical definition of MSE can be expressed as
\[
\sum_{t \in T'} (y_t - \hat{y}_t)^2 / T',
\]
where \(\hat{y}_t\) is the estimator by each model at time \(t\) in the testing dataset.

Recall that a BaGLM is analysed directly by \texttt{bayesglm} function in R. \texttt{mgcv} [Wood, 2017] is a comprehensive package in R for modelling data through GAM models. For both of the models, we select variables according to the AIC criterion.

<table>
<thead>
<tr>
<th></th>
<th>ALPS</th>
<th>BaGLM</th>
<th>BTF</th>
<th>SRoMSE</th>
<th>GAM</th>
<th>BTF</th>
</tr>
</thead>
<tbody>
<tr>
<td>USA</td>
<td>475.993</td>
<td>7.376</td>
<td>651.450</td>
<td>287.845</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UK</td>
<td>315.733</td>
<td>9.585</td>
<td>81.592</td>
<td>78.828</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Brazil</td>
<td>32.178</td>
<td>7.736</td>
<td>280.397</td>
<td>293.601</td>
<td></td>
<td></td>
</tr>
<tr>
<td>France</td>
<td>343.184</td>
<td>11.765</td>
<td>5212.052</td>
<td>50.212</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 8.1: Comparison of models including the Bayesian generalised linear model (BaGLM), generalised additive model (GAM) and our general Bayesian tensor factorisations (BTF) model for COVID-19 datasets. Left table: comparison between BaGLM and BTF in terms of average log predictive score (ALPS). Right table: comparison between GAM and BTF in terms of square root of mean squared error (SRoMSE). Models with best performance are highlighted bold. Countries for comparison include USA, UK, Brazil and France. Response: the daily death count. Predictors: the daily confirmed cases in the past 10 days. Training dataset: 1st March to 15th July 2020; Testing dataset: 16th July to 25th July 2020.

3.1 USA

Comparison among the BaGLM, GAM and our BTF model with COVID-19 data in the USA is in the first column of Table 8.1. The models with the least ALPS and SRoMSE are highlighted bold. According to the outcome, our BTF model has the best performance
for predicting death counts by considering confirmed cases in the past 10 days.

We consider variable selection for both BaGLM and GAM according to the AIC criterion. The BaGLM model selects all but the first predictor as important, i.e. the daily death count depends on the daily confirmed cases 2, 3, \ldots, 10 days ago. In contrast, the GAM model regards the smooth terms of all potential predictors as dependent for the response.

There is a three-stage scheme for the BTF. The first one is to model clusters of counts of daily confirmed cases in the training dataset, which is the standard approach for the mixture of Poisson distributions. The details of clustering the training dataset can be seen in Figure 8.1. According to Figure 8.1, it indicates that there are six unique clusters with different Poisson rates and their corresponding logarithmic weights, even though some Poisson rates are very close to each other. In Figure 8.1, the Poisson rates are around 2,000 to more than 33,000, and their lower counterparts vary from -2.415 to -1.114.

The second stage of the BTF is to select important variables for the response. The pool of variables for selection contains all the counts of daily confirmed cases in the past ten days, which are represented by $x_t$’s. The relative frequency distribution (RFD) for determining whether a predictor is important or not is shown in Panel (a) in Figure 8.2. The threshold RFD for a predictor indicated as important is 0.5. $x_t^{(1)}$, $x_t^{(5)}$, $x_t^{(6)}$, $x_t^{(7)}$ and $x_t^{(8)}$ are indicated as important variables, or in other words, the daily death count depends on the daily confirmed cases 1, 5, 6, 7 and 8 days ago. The outcome for selecting important variables makes sense for COVID-19 since the severe cases lead to death counts after a couple of days once confirmed. Also, some mild confirmed cases turn to severe ones in around a week and lead to death counts after a couple of days. Panel (b) in Figure 8.2 shows the RFD of $\prod_{j=1}^{q} k_j$, or the total number of possible combinations of $H \in \mathcal{H}$. According to Panel (b), almost all possible combinations of $H \in \mathcal{H}$ is under 200.

Stage three is to sample the rest of posteriors. Panel (c) in Figure 8.2 shows a part of MCMC information, which describes the RFD of the number of clusters of the probability kernel $\lambda_{h_1 \ldots h_q}$. It shows that the number of unique $\lambda_{h_1 \ldots h_q}$ in each sampler gathers majorly in the range 30 to 45. All three panels in Figure 8.2 can verify the convergence of the MCMC of stage two and stage three.

### 3.2 UK

Following the procedure of comparison among BaGLM, GAM and BTF model in the last section, we investigate the predictive performance of these three models on the COVID-19
data in the UK, which is shown in the second column of Table 8.1. The BaGLM selects all but the sixth predictor as important, whilst the GAM regards the smooth terms of all potential predictors as dependent for the response.
The details of clustering the training dataset are shown in Figure 8.3. It indicates that there are six unique clusters with different Poisson rates and their corresponding logarithmic weights. The Poisson rates vary from around 300 to more than 4,000, and their lower counterparts range from -3.117 to -1.044.

According to Panel (a) in Figure 8.4, the threshold RFD for a variable indicated as important is 0.5. $x^{(5)}_t$, $x^{(6)}_t$ and $x^{(7)}_t$ are indicated as important variables, or in other words, the daily death count depends on the daily confirmed cases 5, 6 and 7 days ago. Panel (b) in Figure 8.4 shows the RFD of $\prod_{j=1}^q k_j$, or in other words, the total number of possible combinations of $H \in \mathcal{H}$. According to Panel (b), the total number of possible combinations of $H \in \mathcal{H}$ is under 100 with a probability 98.5%.

Panel (c) in Figure 8.4 shows a part of MCMC information, which describes the RFD of the number of clusters of the probability kernel $\lambda_{h_1...h_q}$. It shows that the number of unique $\lambda_{h_1...h_q}$ in each sampler gathers in the range 8 to 12. All three panels in Figure 8.4 can verify the convergence of our MCMC of stage two and stage three.

![Figure 8.3: UK: Information of the clusterings. Upper panel: total number of clusters and their corresponding Poisson rates; Lower panel: weight for each cluster in logarithm. Both panels share the same x-axis.](image)

3.3 Other Countries

The predictive performances on daily death counts, in Brazil and France, of the BaGLM, GAM and BTF model are shown in the third and last columns of Table 8.1.
When modelling the Brazil dataset, both BaGLM and GAM classify all the predictors as important. Figure 8.5 and 8.6 show the MCMC information of all three stages of our BTF model. For the France dataset, the BaGLM selects all but the seventh predictor as important ones, whilst the GAM regards all the predictors as important. Figure 8.7 and 8.8 show the MCMC information of all three stages of our BTF model.
Figure 8.5: Brazil: Information of the clusterings. Upper panel: total number of clusters and their corresponding Poisson rates; Lower panel: weight for each cluster in logarithm. Both panels share the same x-axis.

Figure 8.6: Brazil: (a) The bar chart of relative frequency distribution for variables indicated as important. All the variables for selection is labelled on x-axis, and y-axis represents their corresponding relative frequency. The threshold for a variable to be important is 0.5, which is depicted in red. (b) The RFD of $\prod_{j=1}^{q} k_j$, or the total number of possible combinations of $(h_1, \ldots, h_q)$. (c) The RFD of the number of clusters of the probability kernel $\lambda_{h_1 \ldots h_q}$. y-axis in both (b) and (c) represent the number, and x-axis in both ones are the relative frequency.
Figure 8.7: France: Information of the clusterings. Upper panel: total number of clusters and their corresponding Poisson rates; Lower panel: weight for each cluster in logarithm. Both panels share the same x-axis.

Figure 8.8: France: (a) The bar chart of relative frequency distribution for variables indicated as important. All the variables for selection is labelled on x-axis, and y-axis represents their corresponding relative frequency. The threshold for a variable to be important is 0.5, which is depicted in red. (b) The RFD of $\prod_{j=1}^{q} k_j$, or the total number of possible combinations of $(h_1, \ldots, h_q)$. (c) The RFD of the number of clusters of the probability kernel $\lambda_{h_1 \ldots h_q}$. y-axis in both (b) and (c) represent the number, and x-axis in both ones are the relative frequency.
Chapter 9

Conclusion

This thesis developed a nonparametric model for time series of count data based on tensor factorisations. We have extended the approach by Sarkar and Dunson [2016] that developed a BTF model for categorical data.

Chapter 5 introduced our BTF model for modelling univariate times series of small count data. Through both simulated experiments and real-world datasets, our model shows satisfactory predictive performance. In the scenarios where times series are with relative distant important lags or trending to be non-linear, our BTF approach outperforms its competitors, including Poisson autoregressive and INGARCH models in terms of prediction. In this chapter, we also introduced and concluded that the application of tensor factorisations can help us select important lags of the time series, which benefits the statistical inference.

Chapter 6 generalised its previous chapter to deal with time series of large count data. Our extension involved solving a series of problems. The most important of these was the fact that the computational complexity of our model increase very fast when the number of counts is large. We have dealt with this problem in Chapter 6 and 7 by proposing a two-stage inferential framework. In the first stage, the observed counts are modelled as mixtures of Poisson densities and in the second stage, our BTF model is performed on the Poisson intensities rather than on the observed counts. This greatly facilitates the computational burden when the counts are large.

In Chapter 8 we have also proposed the use of our model in standard Poisson regression in which observations (counts) and covariates are observed in time. Our nonparametric model provides a rich alternative to standard Poisson regression models.
Future research follows here.

1 **Application on Continuous Data**: It is satisfactory that we can apply the BTF framework on a time series of count data and Poisson regression problems. This thesis shows that our BTF model can accept both count data and categorical data as the input for predicting the conditional probability density function. Interesting future research can be the acceptance of continuous data as the input in our BTF model. One can bin the continuous data into bins and then, use the models of this thesis by considering the observed data as the bin midpoints. Here, the width of the bins is an important factor to be investigated, because it will specify the trade-off between precision and computational complexity. For example, considering a time series with a stock’s daily price change, one can take a cent as the bin width, therefore, the past observations can be regarded as one-cent counts.

2 **More Applications on Real-World Datasets**: In this thesis, we have applied our BTF model on dealing with many real-world datasets, varying from the number of stocks with a positive daily return in NASDAQ 100 to the daily death counts caused by COVID-19. However, neither daily counts nor weekly counts can be classified as high-frequency data. It will be interesting to model some high-frequency time series of counts, for example, those of total transactions in a millisecond of stock to find those important lags.


Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 1045–1054.


