Poisson Process Infinite Relational Model: a Bayesian nonparametric model for transactional data

by

Creagh Briercliffe

B.Sc. (Hons.), University of Manitoba, 2014

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Science

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES
(Statistics)

The University of British Columbia
(Vancouver)

August 2016

Abstract

Transactional data consists of instantaneously occurring observations made on ordered pairs of entities. It can be represented as a network—or more specifically, a directed multigraph—with edges possessing unique timestamps. This thesis explores a Bayesian nonparametric model for discovering latent class-structure in transactional data. Moreover, by pooling information within clusters of entities, it can be used to infer the underlying dynamics of the time-series data. Blundell, Beck, and Heller (2012) originally proposed this model, calling it the Poisson Process Infinite Relational Model; however, this thesis derives and elaborates on the necessary procedures to implement a fully Bayesian approximate inference scheme. Additionally, experimental results are used to validate the computational correctness of the inference algorithm. Further experiments on synthetic data evaluate the model's clustering performance and assess predictive ability. Real data from historical records of militarized disputes between nations test the model's capacity to learn varying degrees of structured relationships.

Preface

This dissertation is solely authored, unpublished work by the author, Creagh Briercliffe. The research topic and experiments were jointly designed with Profs. Alexandre Bouchard-Côté and Paul Gustafson. The Poisson Process Infinite Relational Model was first presented by Blundell, Beck, and Heller [6].

Table of Contents

Abstract . . . . . **ii**

Preface . . . . . **iii**

Table of Contents . . . . . **iv**

List of Tables . . . . . **vi**

List of Figures . . . . . **vii**

Glossary . . . . . **ix**

Acknowledgments . . . . . **x**

1 Introduction **1**

1.1 Related Work 2

2 Background **4**

2.1 Chinese Restaurant Process 4

2.2 Infinite Relational Model 6

3 Poisson Process Infinite Relational Model **11**

3.1 Problem Formulation 12

3.2 Generative Process 13

3.3 Inference 15

3.3.1 Sample Space of Partitions 16

3.3.2 Posterior Over Partitions 16
3.3.3 Posterior Over Rates 19
3.3.4 Implementation 20
3.3.5 Sampling Hyperparameters 21
3.4 Note on the Superposition Property 22

4 Experiments 24
4.1 Synthetic Data 24
4.1.1 Validation of Sampler 25
4.1.2 Convergence Diagnostics 28
4.1.3 Clustering Evaluation 32
4.1.4 Prediction 38
4.2 Real Data 39

5 Conclusion 47
5.1 Future Work 47

Bibliography 49

List of Tables

Table 3.1 The n^{th} Bell number corresponding to various values of n. B_n counts the number of distinct partitions of a set of size n. Thus, B_n is the size of the sample space for π when there are n entities. 17

Table 4.1 Countries belonging to some of the clusters inferred from the MAP partition of the MID dataset. 45

Table 4.2 The clusters and countries corresponding to the 6 largest average Poisson process rates recovered from $1,000$ posterior samples. Samples were based on MAP estimates of the model parameters. Clusters for the rates of transactions are shown in the order *sender* $\rightarrow$ *recipient*. 46

List of Figures

Figure 4.1 Multidigraph representation of randomly generated data, from 5 entities, at various stopping times, T. Vertex colours represent the three different clusters. Edges are coloured to signify that each transaction occurs at a unique time point. 26

Figure 4.2 95% confidence intervals for the posterior probabilities of each possible partition. Estimates are based on 10,000 MCMC samples with Monte Carlo standard errors calculated by a batch means method. Each datapoint shows the true posterior probability, and points in red are those not captured by the confidence interval. 28

Figure 4.3 ℓ_1 norm, measured at various iterations, between: (i) the vector of true posterior probabilities of each possible partition, and (ii) the steady-state distribution of the empirical transition matrix of MCMC samples. 30

Figure 4.4 Trace plots of MCMC samples of partitions, given different initializations. Data is comprised of 336 randomly generated transactions from 30 entities. The red lines depict the true partition. Numbering along the vertical axis is arbitrary. 31

Figure 4.5 Trace plots of MCMC samples for each of the hyperparameters, given different initializations. Data is comprised of 336 randomly generated transactions from 30 entities. The red lines depict the true parameter values. 33

Figure 4.6 Mean number of clusters in $10,000$ MCMC samples as the stopping time is increased. The red line depicts the number of clusters in the ground-truth partition used to generate the synthetic data. 34
Figure 4.7 Mean number of clusters in the MCMC samples as the stopping time is increased. Red points are an average of all $10,000$ samples, while blue points drop the first 10% as burn-in. The red line depicts the number of clusters in the ground-truth partition used to generate the synthetic data. 35
Figure 4.8 Adjusted Rand index between a partition estimate and the true partition as the stopping time is increased. Each datapoint is an average across 10 randomly generated datasets, and error bars show one standard deviation. Each estimator is computed from 10,000 MCMC samples. 37
Figure 4.9 Estimated log posterior predictive densities on a test set as the amount of training data (stopping time) is increased. Each estimate is an average based on 10,000 MCMC samples. The red line depicts the likelihood of the test set when the true partition and true rates are given. 40
Figure 4.10 Multidigraph representation of the MID dataset consisting of 138 countries (vertices) and 508 disputes (edges). Vertices are both coloured and numbered according to the 18 clusters inferred from the MAXPEAR partition. 43
Figure 4.11 Multidigraph representation of the MID dataset consisting of 138 countries (vertices) and 508 disputes (edges). Vertices are both coloured and numbered according to the 58 clusters inferred from the MAP partition. 44

Glossary

CRP Chinese Restaurant Process

IRM Infinite Relational Model

MAP maximum a posteriori, the mode of the posterior distribution

MAXPEAR An estimator of partitions that maximizes the posterior expected adjusted Rand index with the true partition

MCMC Markov chain Monte Carlo

MEDV An estimator of partitions based on an ad-hoc approach by Medvedovic et al. [21]

MID Militarized Interstate Dispute

MINBINDER An estimator of partitions that minimizes Binder's loss function

PPIRM Poisson Process Infinite Relational Model

SBM Stochastic Blockmodel

Acknowledgments

I have immensely enjoyed these last two years in Vancouver—and not just for the fact that I escaped the Winnipeg winters. I have been incredibly fortunate to be surrounded by dedicated mentors, talented colleagues, encouraging friends, and loving family. Through their constant support, I have been able to grow professionally, and feel at home in a new city. I am truly grateful to have been afforded the opportunity to pursue my interests, and humbled by the support I received.

First, I want to express my gratitude towards my academic co-supervisors, Profs. Alexandre Bouchard-Côté and Paul Gustafson.[1] Alex and Paul are the Bayesian-dream-team of unrelenting support, unbounded wisdom, and the true driving force behind my research. They possess the uncanny ability to take any of my seemingly unsurmountable research-related struggles, and boil them down to triviality. Alex introduced me to the field of Bayesian nonparametrics, and was a fountain of insightful comments, no matter the topic. Paul's ability to simplify complicated problems, and lead engaging meetings cannot be overstated. Thank you both for your tireless dedication to my continued academic growth.

I have many brilliant mentors and gifted colleagues to thank for helping me to achieve my academic goals. Thanks to Prof. Saman Muthukumarana for your enthusiastic encouragement. Thanks to Andrea Sollberger for your incredible sense of humour, and for tolerating my endless questions. Thanks to my officemates Derek, Jonathan, Sohrab and Yijun for your camaraderie. Thanks to my friends and colleagues, Gal, Jeff and Lena, whose solidarity carried me through our long bout with 551-consulting. I also want to acknowledge the financial support of the Natural Sciences and Engineering Research Council of Canada.

[1] Listed names are in alphabetical order, and not necessarily in order of gratitude.

A special thanks to my friends and former officemates, Sean Jewell and Neil Spencer. Sean and Neil's incredible cleverness and passion for learning drove me to continue my academic pursuits. I am grateful to Sean for convincing me to come to UBC, getting me engaged in the reading group, and helping me to adapt to the Kitsilano life. I am grateful to Neil for teaching me exploit A&W coupons, for his ever-insightful views, and for secretly acting as assistant coach to my fantasy hockey team. Together, their presence made coming to the office a delightful and rewarding experience.

When I needed a break from my studies, I was extremely fortunate to be able to call on an amazing group of friends. In particular, thanks to Andrew, Ryan, Spencer and Steve for making me feel like a pro at pitch-and-putt, and for communally chastising Steve's awful choice of Chinese food. A special thank you to my close friends Micah Goldberg and James Struthers, who value my friendship so much that they followed me to Vancouver. Micah's unique ability to unabashedly insert himself into situations where he is unwelcome, makes him a truly entertaining person to be around. James' unmatched charisma and propensity for self-destruction have made the last two years of domestic life some of the best. Thank you, James and Micah, for helping to make Vancouver feel like home.

Finally, a heartfelt thank you to my parents, Susan and Randy Briercliffe, for their endless support and unconditional encouragement. I cannot express enough gratitude for the amount of love and patience you have given me on this journey.

Chapter 1

Introduction

Often, the primary objectives of probabilistic unsupervised learning are to unveil implicit or hidden structure in data, and to harness this structure to learn the data generating process. To address these objectives, a common statistical technique is to use a mixture model, which boasts a unifying approach for structure discovery and probabilistic modelling. Most mixture models and related unsupervised learning algorithms are designed for feature-based data, expressed as vectors in Euclidean space [17]. This thesis focusses on data of a different form, where observations are made on ordered pairs of objects or entities, and occur in continuous-time—we call this *transactional data*. Moreover, this thesis elaborates on and explores a Bayesian nonparametric mixture model for transactional data, called the Poisson Process Infinite Relational Model (PPIRM).

Transactional data consists of a set of entities, $\mathcal{X} = \{x_1, \ldots, x_n\}$, on which a set of observations are made for ordered triples (x_i, x_j, t), where $t \in \mathbb{R}_{>0}$ records the event time. Each observation is called a *transaction* to evoke the notion that it represents a directed event, or exchange between two entities: one sender and one recipient, at a specific time. Similar data types are typically represented as a graph, where the entities mark the vertices, and the events are depicted as edges. Transactions are akin to dyadic events, where each transaction is an instantaneous directed edge between a pair of vertices. Thus, a full transactional dataset can be represented as a directed multigraph, with edges occurring instantaneously in time. Our goal is to use the PPIRM to learn the implicit class-structure of the entities, and

infer the dynamics of the time-series data.

The PPIRM was originally proposed by Blundell et al. [6] as a model for discovering the underlying social structure from a set of interactions between people. As it was not the main focus of [6], there is little detail regarding an inference procedure and subsequent assessment of its performance. To address this deficiency in the literature, we derive a fully Bayesian approximate inference scheme for the PPIRM, and evaluate its performance on synthetic and real data. Furthermore, Blundell et al. [6] posit that the superposition property of Poisson processes can be used to maintain efficient posterior inference, but we show that this approach can be misguided.

Section 1.1 discusses other work related to the PPIRM, while the remainder of this thesis is organized as follows. Chapter 2 provides the necessary background material on a Bayesian nonparametric distribution over partitions of a set. Chapter 2 concludes with the details of the Infinite Relational Model (IRM), of which the PPIRM is an extension. The PPIRM and a full derivation of an inference procedure are presented in Chapter 3. Experimental results on synthetic and real data are discussed in Chapter 4. Finally, conclusions and future directions are reported in Chapter 5.

1.1 Related Work

Blundell et al. [6] proposed the PPIRM by relating the work of DuBois and Smyth [9] to the IRM [20, 32]. While, the model proposed in [9] also uses Poisson processes to model event times, it requires the number of latent classes to be fixed in advance. Conversely, the IRM and PPIRM exploit a flexible nonparametric method, allowing the number of classes to be data-dependent and inferred with the class memberships. Furthermore, the work in [9] models latent classes of events, rather than latent classes of entities.

The IRM is a Bayesian nonparametric model for dyadic data. It can be viewed as a generalization of the parametric Stochastic Blockmodel (SBM) [18, 30], which models the regular block structure of an adjacency matrix. The dyadic data applicable to the IRM, which Kemp et al. [20] call relations, lack the time-varying structure of transactional data. Thus, transactional data are strictly more rich than

the relations that can be modelled with the IRM.

The model that was the main focus of [6] uses mutually-exciting Hawkes processes, where the rates of events depend upon those of other processes in an excitatory fashion. Similarly, Simma and Jordan [28] use a cascade of Poisson processes, which also form a marked Hawkes Process. These models allow for time-varying rates, which are aimed at capturing the reciprocating nature of many social interactions. However, they lack the conjugacy that affords the PPIRM the ability to maintain efficient posterior inference.

Recently, Xin et al. [31] proposed a continuous-time extension of the SBM that uses inhomogeneous Poisson processes to model transactions among basketball players. While their model allows for time-varying rates, it also requires the number of latent classes to be fixed in advance.

Finally, we want to emphasize that the domain of applications for the PPIRM and related work is far-reaching. Examples include social interactions [2, 6, 9, 18, 30], biological networks [2, 16], sports [31], and ontologies of semantic knowledge [20]. Subsequently, we aim to keep our model descriptions as general and domain-independent as possible. Then, in Section 4.2, we display the PPIRM's ability to model a real dataset of historical military conflicts between nations.

Chapter 2

Background

The PPIRM was presented by Blundell et al. [6] as an extension to the IRM. The IRM [20, 32] is a Bayesian model for relational data, which aims to learn a clustering of the data. In doing so, the IRM relies on a specialized prior distribution over all possible clusterings, called the Chinese Restaurant Process (CRP) [3].

This chapter is divided into two parts. Section 2.1 reviews the details of the CRP, and its features as a prior distribution. Section 2.2 discusses the IRM, starting from its most simple form with dyadic data, and then moving to the more general version. Together, these sections provide the necessary notation and background to discuss the PPIRM in Chapter 3.

2.1 Chinese Restaurant Process

Let S be a finite set of size $n = |S|$, and let $k \in \mathbb{N}$ such that $k \leq n$. A *partition* of S into k *blocks* is an unordered collection of non-empty disjoint sets $\{A_1, \ldots, A_k\}$ whose union is S [26]. In cluster analysis, blocks are referred to as *clusters*. Here, the task is to construct a partition of a dataset, such that the data within the same cluster are more similar to each other than to those in other clusters. Taking a Bayesian approach to clustering, one might wish to place a prior distribution over partitions of a set. Moreover, when the number of clusters, k, is unknown, a sensible prior should assign some probability mass to all possible partitions; that is, for all possible values of k. The CRP does just that: it induces a probability distribution

over partitions.

The CRP gets its name from the metaphorical seating plan that describes the stochastic process [3]. In this metaphor, customers arrive sequentially at a Chinese restaurant with an infinite number of tables. Each table can seat an infinite number of customers. The first customer enters the restaurant and sits at the first table. The second customer enters and either joins the first customer, or sits at a new table. In general, each subsequent customer either joins an already occupied table with probability proportional to the number of customers already sitting there, or sits at a new table with probability proportional to a fixed parameter α. At any point in this process, the configuration of customers at tables defines a random partition.

To be more precise, let π_i denote the table assignment of the i^{th} customer. The CRP with parameter α draws each π_i sequentially, such that

$$\mathbb{P}(\pi_i = k | \pi_1, \ldots, \pi_{i-1}) = \begin{cases} \frac{n_k}{i-1+\alpha} & \text{if } n_k > 0, \\ \frac{\alpha}{i-1+\alpha} & \text{if } k \text{ is a new table,} \end{cases}$$

where n_k denotes the number of customers seated at table k. Together, the table assignments, π_i, can be used to describe the full partition, π. Thus, a partition can be represented alternatively as a vector, with components representing the cluster memberships of each datum. The parameter, $\alpha \in \mathbb{R}_{>0}$, is called the *concentration parameter*. Intuitively, larger values of α tend to produce partitions with more occupied tables and fewer customers per table [11].

Though this process is defined sequentially, the distribution induced by the CRP is, in fact, exchangeable [3]. The probability of a particular partition, π, is invariant to the order in which the customers were seated. Hence, the order in which the data are assigned to clusters can be permuted without changing the probability of the partition. Given this property, the probability of any single partition can be expressed succinctly as

$$p(\pi) = \frac{\alpha^K \prod_{k=1}^{K} (n_k - 1)!}{\prod_{i=1}^{n} (i - 1 + \alpha)}, \tag{2.1}$$

where K denotes the number of tables. Note that Equation 2.1 depends on the number and sizes of the tables, but does not depend on the order in which the

customers were seated.

If we represent the customers by the elements of the set $[n] := \{1, 2, \ldots, n\}$, then after n customers have been seated, the tables define a partition of $[n]$. Furthermore, if we add slightly to the above seating metaphor and specify that each table is circular, then the CRP can also define a distribution over permutations of $[n]$. In this modified metaphor, when the $j+1^{\text{st}}$ customer enters the restaurant they choose with equal probability to sit at any of the following $j+1$ places: to the left of customer i for some $i \in [j]$, or alone at a new table. After the n customers have been seated, the tables represent the cycles of the permutation, whose product is a uniformly distributed random permutation of $[n]$ [26]. Since this feature of the CRP is rarely used in cluster analysis or Bayesian mixture modelling, we refer the interested reader to the works of Aldous [3] and Pitman [26] for further details.

In terms of a prior over partitions of a dataset, the CRP has several potentially desirable features. Namely, a sensible prior should encourage only as many clusters as warranted by the data [20]. Since new customers can always be assigned to new tables, the CRP permits the number of tables to grow with the data. Specifically, the expected number of tables grows logarithmically with the number of customers [11]. Furthermore, the CRP exhibits a rich-gets-richer phenomenon, where a customer is more likely to join a table occupied by a large number of customers. That is, the larger n_k is, the higher the probability that table k will grow [29]. As we shall see in Section 2.2, these features also make the CRP a useful prior for the partitions of relational data.

2.2 Infinite Relational Model

The IRM [20, 32] is a Bayesian model of binary-valued relationships amongst a set of entities. For example, the entities might consist of a set *people*, and a relation *likes* describes whether one person likes another. In this way, the relation can be defined as a function, *likes*: $people \times people \to \{0, 1\}$, where $likes(u, v)$ indicates whether or not person u likes person v. The goal here is to partition entities into clusters, where a good partition allows relationships between entities to be predicted by their cluster assignments. The IRM assumes that an entity's tendency to participate in relations is determined entirely by its cluster assignment.

The IRM can be seen as a nonparametric generalization of the parametric SBM introduced by Holland et al. [18], Wasserman and Anderson [30]. By utilizing the CRP as a prior over partitions, the IRM does not require the number of clusters to be fixed in advance, unlike the SBM. Rather, the IRM infers this number simultaneously with the cluster memberships for each entity. Given that the CRP permits each new entity to be assigned to a new cluster, the IRM effectively has access to a countably infinite collection of clusters. Hence, Kemp et al. [20] coined the name *Infinite Relational Model* for their approach to modelling relational data.

Like the SBM, the IRM can be presented in terms of modelling the graph structure of relational data. To describe the generative process, we adopt this approach and list the features of the data in terms of their analogs on a unweighted, directed graph (or *digraph*). We define this graph, $G = (V, E)$, by its vertices and directed edges. Let V denote the set of vertices of the graph, where each vertex represents a single entity. For vertices $u, v \in V$, let $e_{uv} \in \{0, 1\}$ denote the presence (1) or absence (0) of a directed edge from vertex u to v in the graph; edges correspond to the relationships between entities. Let E denote the edge set consisting of all ordered triples (u, v, e_{uv}) for all vertices $u, v \in V$. Note that for asymmetric relations, it is not necessary that $e_{uv} = e_{vu}$, thereby creating a directed graph.

The generative process of the IRM is as follows:

$$\pi | \alpha \sim \text{CRP}(\alpha) \tag{2.2}$$

$$\lambda_{kl} | \gamma \sim \text{Beta}(\gamma, \gamma) \qquad \forall k, l \in \text{range}(\pi) \tag{2.3}$$

$$e_{uv} | \lambda, \pi \sim \text{Bernoulli}(\lambda_{\pi_u \pi_v}) \qquad \forall u, v \in V \tag{2.4}$$

where π is a partition of the vertices V, drawn from the CRP with concentration parameter α, as described in Section 2.1. Adopting the notation from [6], we use $\text{range}(\pi)$ to denote the set of all unique clusters in π, which we index with k and l. Note that while the CRP is a distribution over a countably infinite number of clusters, $|\text{range}(\pi)|$ is restricted to a natural number between 1 and $|V|$, inclusive, for fixed vertex sets. That is, when $|\text{range}(\pi)| = 1$, all vertices belong to a single cluster; when $|\text{range}(\pi)| = |V|$, each vertex belongs to its own cluster. We write that vertex $u \in V$ belongs to the cluster given by π_u. The probability of a directed edge from vertex u to v is given by the parameter $\lambda_{\pi_u \pi_v}$. Finally, we use λ to denote

the matrix of Bernoulli parameters between all $|\text{range}(\pi)| \times |\text{range}(\pi)|$ pairs of clusters. Kemp et al. [20] choose to place symmetric conjugate priors, with single hyperparameter γ, on each entry of the λ matrix.

Since the generative process uses a conjugate prior on the entries of λ, it is straightforward to derive the marginal likelihood $p(E|\pi)$, integrating out λ:

$$\begin{aligned}
p(E|\pi) &= \int p(\lambda|\pi)p(E|\lambda,\pi)d\lambda \\
&= \int \left[\prod_{k,l\in\text{range}(\pi)} \text{Beta}(\lambda_{kl};\gamma,\gamma)\right]\left[\prod_{u,v\in V} \text{Bernoulli}(e_{uv};\lambda_{\pi_u\pi_v})\right] d\lambda \\
&= \int \prod_{k,l\in\text{range}(\pi)} \frac{\lambda_{kl}^{\gamma-1}(1-\lambda_{kl})^{\gamma-1}}{B(\gamma,\gamma)} \prod_{u:\pi_u=k}\prod_{v:\pi_v=l} \lambda_{kl}^{e_{uv}}(1-\lambda_{kl})^{1-e_{uv}} d\lambda \\
&= \prod_{k,l\in\text{range}(\pi)} \frac{1}{B(\gamma,\gamma)} \int \lambda_{kl}^{\gamma-1}(1-\lambda_{kl})^{\gamma-1}\lambda_{kl}^{m_{kl}}(1-\lambda_{kl})^{\bar{m}_{kl}} d\lambda_{kl} \\
&= \prod_{k,l\in\text{range}(\pi)} \frac{B(m_{kl}+\gamma,\bar{m}_{kl}+\gamma)}{B(\gamma,\gamma)},
\end{aligned} \tag{2.5}$$

where m_{kl} denotes the number of directed edges from all entities in cluster k to any entity in cluster l. That is, m_{kl} is the number of pairs (u,v) where $\pi_u = k$, $\pi_v = l$ and $e_{u,v} = 1$. $\bar{m}_{kl}$ is the number of pairs (u,v) where $\pi_u = k$, $\pi_v = l$ and $e_{u,v} = 0$. $B(a,b)$ denotes the Beta function, $B(a,b) = [\Gamma(a)+\Gamma(b)]/\Gamma(a+b)$. Lastly, note that for simplicity of notation, conditioning on the parameter γ is implicit.

The marginal likelihood in Equation 2.5 is, in fact, quite simple to compute. If we let n_k denote the number of entities belonging to cluster k, then $\bar{m}_{kl}$ can be calculated from $\bar{m}_{kl} = n_k n_l - m_{kl}$. Thus, we need only to record the counts n_k for all clusters, and m_{kl} for all cluster pairs. Using this marginal likelihood, inference can be easily carried out using Markov chain Monte Carlo (MCMC) methods to sample from the posterior on cluster assignments $p(\pi|E) \propto p(E|\pi)p(\pi)$. Kemp et al. [20], employ a fully Bayesian approach, using an exponential prior $p(\alpha) = e^{-\alpha}$ and improper prior $p(\gamma) \propto \gamma^{-5/2}$ when sampling the hyperparameters. Finally, we note that although λ is integrated out, it is simple to recover these parameters by drawing values independently from the posterior $\lambda_{kl}|E,\pi \sim \text{Beta}(m_{kl}+\gamma,\bar{m}_{kl}+\gamma)$ for all $k,l \in \text{range}(\pi)$. Moreover, the maximum a posteriori (MAP) value of λ_{kl} given π

and E is actually:

$$\frac{m_{kl}+\gamma-1}{m_{kl}+\bar{m}_{kl}+2\gamma-2}=\frac{m_{kl}+\gamma-1}{n_k n_l+2\gamma-2} \qquad \text{when } m_{kl},\bar{m}_{kl}>1-\gamma,$$

contrary to what is stated in [20].

The IRM can be extended to model multiple types of entities, and polyadic relations of any arbitrary arity [20]. For example, suppose there are several relations defined over the domain *people* $\times$ *people*: *likes*$(\cdot,\cdot)$, *respects*$(\cdot,\cdot)$ and *hates*$(\cdot,\cdot)$. These relations can then be grouped together as a single type referred to as predicates. Next, we can define a ternary relation *applies*(u,v,p), which is 1 if predicate p applies to the pair (u,v), of people u and v. Thus, we have a ternary relation over two entity types: people and predicates. The aim is to now simultaneously cluster the people and the predicates. When the observed data is viewed as a third-order tensor (*people* $\times$ *people* $\times$ *predicate*), the IRM finds a partition of people and predicates such that a good clustering ensures each three-dimensional sub-block of the data includes mostly ones or mostly zeroes.

To specify the generative process of this more general version of the IRM, consider an r-dimensional relation involving q different entity types, with $r \geq q$. Now, V_q denotes the entity set of the q^{th} entity type, and let $d_p \leq q$ denote the index of the entity type occupying dimension $p \leq r$. Then,

$$\begin{aligned}
\pi^{(i)}|\alpha &\sim \text{CRP}(\alpha) \quad \forall i=1,\ldots,q \\
\lambda_{k_1\ldots k_r}|\gamma &\sim \text{Beta}(\gamma,\gamma) \quad \forall k_1 \in \text{range}\left(\pi^{(d_1)}\right),\ldots,\forall k_r \in \text{range}\left(\pi^{(d_r)}\right) \\
e_{u_1\ldots u_r}|\lambda,\pi^{(1)},\ldots,\pi^{(q)} &\sim \text{Bernoulli}\left(\lambda_{\pi_{u_1}^{(d_1)}\ldots\pi_{u_r}^{(d_r)}}\right) \quad \forall u_1 \in V_{d_1},\ldots,\forall u_r \in V_{d_r}
\end{aligned}$$

where λ is now an r^{th}-order tensor (or r-dimensional array) of Bernoulli parameters between all r-tuples of clusters. Kemp et al. [20] further generalize the IRM by allowing multiple relations and introducing a parameter array $\lambda^{(j)}$ for each relation and corresponding edge set $E^{(j)}$. In this model, they assume that relations are conditionally independent given the cluster assignments, and that the cluster assignments for each entity type are independent.

Due to the flexibility of the IRM, it has many potential applications. It has been

used to discover structure in object-feature data [20], where the features can be viewed as unary predicates and the IRM can be used as a means for biclustering. In this case, the IRM was applied to an animal-feature matrix, and the clustering results were compared to solutions given by human subjects. Kemp et al. [20] further demonstrated its use by learning systems of related concepts as a means to acquire an ontology of semantic knowledge—an important task in AI development, where the ontology can be applied to further problem solving. They showed that the IRM can be used to discover a simple biomedical ontology from a semantic network comprised of concepts and binary predicates. The concepts included entities like 'disease or syndrome,' 'diagnostic procedure,' and 'mammal.' The predicates included verbs like *complicates, affects* and *causes*. Furthermore, by computing the MAP values of λ_{kl}, Kemp et al. [20] were able to identify the pairs of clusters (k,l) that are most strongly linked, and the predicates that linked them.

Chapter 3

Poisson Process Infinite Relational Model

One limitation of the IRM is the inability to model multiple instances of a single relation between the same pair of entities. For example, consider an entity set *players* and a relation *passes*, both referring to a soccer match. With the IRM, we are limited to binary-valued relations, where $passes(u, v)$ indicates whether or not player u passes the ball to player v; however, in reality, any given player could pass the ball to another several times within a given match. Thus, using the notation from Section 2.2, we want to capture this information by letting $e_{uv} \in \{0\} \cup \mathbb{N}$. In terms of the graph structure of the data, we want to model a weighted digraph.

Blundell et al. [6] suggest a straightforward modification to account for this, replacing the Beta-Bernoulli model in Expressions 2.3 and 2.4 with a Gamma-Poisson model. This model permits Poisson-distributed counts on the directed edges, while retaining the convenient conjugacy in the generative process. However, Blundell et al. [6] note that a Gamma-Poisson observation model does not allow for the prediction of events into the future, outside of the observed time window. Continuing with the previous example, we may be interested in predicting passes between players over the course of a season, beyond just a single match. Therefore, Blundell et al. [6] suggest using instead a Poisson process.

This chapter presents the PPIRM, which extends the IRM by modelling a richer structure of data, while retaining the ability to perform efficient, approximate pos-

terior inference. Section 3.1 formalizes the problem statement, while Section 3.2 describes the generative process of the PPIRM. Blundell et al. [6] give little detail regarding their inference scheme for the PPIRM, but mention that it relies on the superposition property of Poisson processes. Instead, we derive our own approximate inference procedure in Section 3.3, and Section 3.4 discusses why their proposal to use the superposition property is misguided.

3.1 Problem Formulation

Consider a dataset consisting of a set of entities and a set of recorded transactions between them, each with a specific timestamp. To distinguish from the binary-valued "relations" of the IRM, we use the term *transaction* to define events between ordered pairs of entities that occur at a specific time point. Multiple transactions can occur between the same ordered pair of entities, but each event must occur at a unique time point. Thus, we use the term transaction in the sense of an exchange between two entities, one sender and one recipient, at a particular time. In this way, the events can be seen as asymmetric, or directed, interactions. Examples include:

- a team of soccer players and the times at which a pass was made between any two of them during the course of a match;
- employees of a company and the times at which an email was sent from one employee to another;
- the neurons in the human brain and the times at which a signal is transmitted from the axon of one neuron to the dendrite of another;
- a group of Twitter users and the times at which one user retweets the content of another;
- a collection of airports and the times at which flights depart from one city travelling towards another.

Each transaction can be represented as a triple consisting of a sender entity, a recipient entity, and a time of occurrence. Let N denote the full dataset; that is, the set of all observed transaction triples. There are two ways in which this data, N, are

strictly more rich than the relations, E, from the IRM: each transaction includes a time of occurrence, and multiple transactions can be observed between any ordered pair of entities. Note that both of these features exist in each of the above examples. For example, there can be multiple flights departing from Vancouver heading to Winnipeg, but each has a unique departure time. Thus, the IRM cannot sufficiently capture the structure in this data.

Given such a dataset, the problem is one of learning or inference, where we wish to learn the underlying dynamics of the network structure formed by the data. More explicitly, we wish to infer the parameters of a model that accurately describes the structure of the data, N. Using this inferred model, we can discover groups of similarly behaving entities, and predict unobserved transactions. To this end, we construct a Bayesian stochastic model with the objective of partitioning entities into clusters, where a good set of partitions allows transactions between entities to be predicted by their cluster assignments. Like the IRM, the PPIRM assumes that an entity's tendency to participate in transactions is determined entirely by its cluster assignment and the cluster assignment of the partner entity. Thus, our problem is inherently two-fold:

(i) learn a latent clustering of the entities—an unsupervised learning task;

(ii) model the time-series structure of the transaction data.

3.2 Generative Process

To describe the PPIRM, we revert to discussing the graph structure of the transactional data, N, which is both useful for visualizing the data, and common practice in related literature [2, 18, 30]. We define this graph, $\mathbb{G} = (V,N)$, by its vertices and transaction set. $\mathbb{G}$ represents a directed multigraph (also known as a *multidigraph* or *quiver*) where each edge has its own identity given by its timestamp. In comparison, recall that the relational data modelled by the IRM can be represented by an unweighted digraph.

Let V denote the set of vertices, where each vertex represents a single entity, and let $n = |V|$ denote the order of $\mathbb{G}$. Note that throughout we try to use n to represent counts related to vertices, and m for counts related to the edges or transactions.

Like before, π is a partition of the vertices in V, and it can be represented as both a set of sets and as a vector. Note that we chose to implement the latter for its relative simplicity to code; however, implementing a set of sets, using for example Java's `LinkedHashSet` class [1], could improve computational efficiency. In the vector representation, π is a vector of length n containing the cluster assignments for all vertices. That is, $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$. Thus, vertex $u \in V$ belongs to the cluster given by the u^{th} component of π, π_u. Again, we adopt the notation from [6] and use range(π) to denote the set of all distinct clusters in π.

Directed edges from a vertex u to a vertex v are modelled as events occurring randomly from a Poisson process with rate $\lambda_{\pi_u \pi_v}$. Specifically, we use a Poisson process on $[0, \infty)$, such that the number of events in any interval $[t, t')$ of the real-half line, denoted $N[t, t')$, is Poisson distributed with rate $\lambda(t' - t)$. The generative process of the PPIRM is:

$$\pi | \alpha \sim \text{CRP}(\alpha) \tag{3.1}$$

$$\lambda_{kl} | \delta, \beta \sim \text{Gamma}(\delta, \beta) \qquad \forall k, l \in \text{range}(\pi) \tag{3.2}$$

$$N_{uv}(\cdot) | \lambda, \pi \sim \text{PoissonProcess}(\lambda_{\pi_u \pi_v}) \qquad \forall u, v \in V \tag{3.3}$$

where $N_{uv}(\cdot)$ is the random counting measure of the Poisson process, and δ and β are respectively the shape and rate (i.e. inverse scale) parameters of the Gamma prior on the rate of the Poisson process, λ_{kl}. We note that this model permits transactions to occur between an entity and itself, thereby creating self-loops in the multidigraph representation.

As a critique of the PPIRM, Blundell et al. [6] suggest two potential deficiencies with the model:

(i) the rate of transactions between an ordered vertex pair, (u, v), is independent of the rate between the reverse pair, (v, u); this relates to a phenomenon known as *reciprocity* [18, 30].

(ii) Conditioned on the time interval containing all observed transactions, the times of these events are uniformly distributed. With regards to stochastic processes, this feature is called *time homogeneity*.

To address these concerns, they propose replacing the Poisson processes with mutually-

exciting Hawkes processes. However, in doing so, they sacrifice conjugacy in the model, as there is no conjugate prior for the Hawkes process likelihood. Consequently, they cannot integrate out the stochastic process rate parameters, and are forced to sample them using MCMC methods. Thus, the PPIRM sacrifices some model flexibility in order to maintain efficient posterior inference.

3.3 Inference

The primary objective is to infer a partition given the observed transaction data. Specifically, the aim is to infer both the number of clusters and the cluster memberships for all entities. In the context of social network analysis, this is often referred to as learning the latent community structure of the network [16]. Here, we exploit the conjugate prior for the Poisson likelihood and integrate out the Poisson process rate parameters.[1] Thus, the posterior mass function over partitions is given by

$$p(\pi|N) = \frac{p(\pi)p(N|\pi)}{\sum_{\pi} p(\pi)p(N|\pi)} \tag{3.4}$$

$$\propto p(\pi)p(N|\pi)$$

$$= p(\pi) \int p(N|\lambda,\pi)p(\lambda|\pi)d\lambda, \tag{3.5}$$

where λ denotes the matrix of Poisson process rates between clusters. If we let $c = |\text{range}(\pi)|$ denote the number of distinct clusters in the partition π, then λ can be represented as a matrix of size $c \times c$.

Subsection 3.3.1 discusses the need for approximate inference methods due to the combinatorially large sample space of distinct partitions. Subsection 3.3.2 presents the full derivation of the of the unnormalized posterior mass function over partitions, while Subsection 3.3.3 derives the posterior density over the Poisson process rates, λ. Subsection 3.3.4 mentions the implementation details of our MCMC sampler for partitions. Finally, Subsection 3.3.5 details the rest of our fully Bayesian model and the procedures for sampling the hyperparameters, which were not discussed in [6].

[1] Note that for simplicity of exposition, we view the parameters α, δ and β as fixed numbers in $\mathbb{R}_{>0}$. Later, in Subsection 3.3.5, we treat them as random variables.

3.3.1 Sample Space of Partitions

The size of the sample space of distinct partitions is a combinatorial problem, and thus suffers from a combinatorial explosion as the number of entities, n, grows. To be precise, the number of distinct partitions of a set of size n is given by the n^{th} Bell number,

$$B_n = \frac{1}{e} \sum_{k=1}^{\infty} \frac{k^n}{k!},$$

by Dobiński's formula [25]. This formula allows B_n to be interpreted also as the n^{th} moment of a Poisson distribution with rate parameter 1. For example, $B_1 = 1$ because the 1-element set $\{u\}$ can be partitioned only one way, as $\{\{u\}\}$. Similarly, $B_3 = 5$ because the 3-element set $\{u, v, w\}$ can be partitioned in 5 distinct ways:

1. $\{\{u, v, w\}\}$
2. $\{\{u\}, \{v, w\}\}$
3. $\{\{v\}, \{u, w\}\}$
4. $\{\{w\}, \{u, v\}\}$
5. $\{\{u\}, \{v\}, \{w\}\}$

Table 3.1 lists some of the first several Bell numbers.

Due to this combinatorial explosion, for most interesting problems it is computationally infeasible to calculate the exact posterior probability of π, which requires computing a sum over all B_n distinct partitions, as seen in Equation 3.4. Therefore, we rely on MCMC to perform approximate inference with Equation 3.5 as the target posterior.

3.3.2 Posterior Over Partitions

Below, we derive each component of the posterior distribution given by Equation 3.5. By definition of the CRP, the first term on the right-hand side becomes

$$p(\pi) = \frac{\alpha^c \prod_{k=1}^{c} (n_k - 1)!}{\prod_{u=1}^{n} (u - 1 + \alpha)}, \tag{3.6}$$

Table 3.1: The n^{th} Bell number corresponding to various values of n. B_n counts the number of distinct partitions of a set of size n. Thus, B_n is the size of the sample space for π when there are n entities.

n	B_n
1	1
2	2
3	5
4	15
5	52
6	203
7	877
8	4,140
9	21,147
10	115,975
⋮	⋮
100	$\approx 4.7585 \times 10^{115}$

where n_k denotes the number of vertices belonging to cluster k, such that $\sum_{k=1}^{c} n_k = n$. Next, the conditional probability of the Poisson process rates, $p(\lambda|\pi)$, comes from taking the product of Gamma densities over all clusters $k, l \in \text{range}(\pi)$. Thus,

$$\begin{aligned} p(\lambda|\pi) &= \prod_{k=1}^{c}\prod_{l=1}^{c} \text{Gamma}(\lambda_{kl};\delta,\beta) \\ &= \prod_{k=1}^{c}\prod_{l=1}^{c} \frac{\beta^{\delta}}{\Gamma(\delta)} \lambda_{kl}^{\delta-1} e^{-\beta\lambda_{kl}}. \end{aligned} \tag{3.7}$$

The conditional likelihood of the transaction data given all other information can be decomposed as follows. Since transactions between each ordered vertex pair, (u,v), are modelled by an independent Poisson process, we can decompose N into Poisson distributed counts, X_{uv}, with events distributed uniformly on the interval $[0,T)$. Thus, we have that each X_{uv} is Poisson distributed with rate parameter $\lambda_{\pi_u\pi_v}T$. Simply put, X_{uv} is a count summary of the data, with x_{uv} denoting the observed number of transactions from vertex u to vertex v by time T. Now, the

likelihood can be written as

$$
\begin{aligned}
p(N|\pi,\lambda) &= \prod_{u=1}^{n}\prod_{v=1}^{n} \text{Poisson}(x_{uv};\lambda_{\pi_u\pi_v}T)\cdot \text{Uniform}(0,T)^{x_{uv}} \\
&= \prod_{u=1}^{n}\prod_{v=1}^{n} \frac{(\lambda_{\pi_u\pi_v}T)^{x_{uv}}e^{-\lambda_{\pi_u\pi_v}T}}{x_{uv}!}\left(\frac{1}{T}\right)^{x_{uv}} \\
&= \prod_{k=1}^{c}\prod_{l=1}^{c}\prod_{u:\pi_u=k}\prod_{v:\pi_v=l} \frac{(\lambda_{kl}T)^{x_{uv}}e^{-\lambda_{kl}T}}{x_{uv}!}\left(\frac{1}{T}\right)^{x_{uv}} \\
&= \prod_{k=1}^{c}\prod_{l=1}^{c} \frac{(\lambda_{kl}T)^{m_{kl}}e^{-\lambda_{kl}Tn_kn_l}}{\prod_{u:\pi_u=k}\prod_{v:\pi_v=l}x_{uv}!}\left(\frac{1}{T}\right)^{m_{kl}}, \qquad (3.8)
\end{aligned}
$$

where m_{kl} denotes the total number of transactions from all vertices in cluster k towards any vertex in cluster l by time T. That is, $m_{kl} = \sum_{u:\pi_u=k}\sum_{v:\pi_v=l} x_{uv}$.

Combining Equations 3.7 and 3.8, and evaluating the integral, gives the following likelihood where the rates have been marginalized out:

$$
\begin{aligned}
p(N|\pi) &= \int p(N|\lambda,\pi)p(\lambda|\pi)d\lambda \\
&= \int \prod_{k=1}^{c}\prod_{l=1}^{c} \frac{(\lambda_{kl}T)^{m_{kl}}e^{-\lambda_{kl}Tn_kn_l}}{\prod_{u:\pi_u=k}\prod_{v:\pi_v=l}x_{uv}!}\left(\frac{1}{T}\right)^{m_{kl}} \frac{\beta^{\delta}}{\Gamma(\delta)}\lambda_{kl}^{\delta-1}e^{-\beta\lambda_{kl}}d\lambda \\
&= \int \prod_{k=1}^{c}\prod_{l=1}^{c} \left[\frac{T^{m_{kl}}\beta^{\delta}}{\Gamma(\delta)\prod_{u:\pi_u=k}\prod_{v:\pi_v=l}x_{uv}!}\right]\left(\frac{1}{T}\right)^{m_{kl}} \lambda_{kl}^{m_{kl}+\delta-1}e^{-(Tn_kn_l+\beta)\lambda_{kl}}d\lambda \\
&= \prod_{k=1}^{c}\prod_{l=1}^{c} \left[\frac{T^{m_{kl}}\beta^{\delta}}{\Gamma(\delta)\prod_{u:\pi_u=k}\prod_{v:\pi_v=l}x_{uv}!}\right]\left(\frac{1}{T}\right)^{m_{kl}} \int \lambda_{kl}^{m_{kl}+\delta-1}e^{-(Tn_kn_l+\beta)\lambda_{kl}}d\lambda_{kl} \\
&= \prod_{k=1}^{c}\prod_{l=1}^{c} \left[\frac{T^{m_{kl}}\beta^{\delta}}{\Gamma(\delta)\prod_{u:\pi_u=k}\prod_{v:\pi_v=l}x_{uv}!}\right]\left(\frac{1}{T}\right)^{m_{kl}} \frac{\Gamma(m_{kl}+\delta)}{(Tn_kn_l+\beta)^{m_{kl}+\delta}} \\
&= \prod_{k=1}^{c}\prod_{l=1}^{c} \frac{\Gamma(m_{kl}+\delta)}{\Gamma(\delta)\prod_{u:\pi_u=k}\prod_{v:\pi_v=l}x_{uv}!}\left(\frac{T}{Tn_kn_l+\beta}\right)^{m_{kl}}\left(\frac{\beta}{Tn_kn_l+\beta}\right)^{\delta}\left(\frac{1}{T}\right)^{m_{kl}} \\
&= \prod_{k=1}^{c}\prod_{l=1}^{c} \frac{(m_{kl}+\delta-1)!}{(\delta-1)!m_{kl}!}\left(\frac{Tn_kn_l}{Tn_kn_l+\beta}\right)^{m_{kl}}\left(1-\frac{Tn_kn_l}{Tn_kn_l+\beta}\right)^{\delta} \\
&\qquad \cdot \frac{m_{kl}!}{\prod_{u:\pi_u=k}\prod_{v:\pi_v=l}x_{uv}!}\left(\frac{1}{Tn_kn_l}\right)^{m_{kl}}
\end{aligned}
$$

$$= \prod_{k=1}^{c}\prod_{l=1}^{c} \text{NegBin}\left(m_{kl}; \delta, \frac{Tn_k n_l}{Tn_k n_l + \beta}\right) \frac{m_{kl}!}{\prod_{u:\pi_u=k}\prod_{v:\pi_v=l} x_{uv}!} \left(\frac{1}{Tn_k n_l}\right)^{m_{kl}}, \tag{3.9}$$

where $\text{NegBin}(\cdot; r, p)$ is shorthand for the negative binomial probability mass function with parameters r and p. Finally, the kernel of our target posterior is

$$\begin{aligned} p(\pi|N) &\propto \frac{\alpha^c}{\prod_{u=1}^{n}(u-1+\alpha)} \prod_{k=1}^{c}(n_k-1)! \prod_{l=1}^{c} \text{NegBin}\left(m_{kl}; \delta, \frac{Tn_k n_l}{Tn_k n_l + \beta}\right) \\ &\qquad \cdot \frac{m_{kl}!}{\prod_{u:\pi_u=k}\prod_{v:\pi_v=l} x_{uv}!} \left(\frac{1}{Tn_k n_l}\right)^{m_{kl}} \\ &\propto \alpha^c \prod_{k=1}^{c}(n_k-1)! \prod_{l=1}^{c} \text{NegBin}\left(m_{kl}; \delta, \frac{Tn_k n_l}{Tn_k n_l + \beta}\right) m_{kl}! \left(\frac{1}{n_k n_l}\right)^{m_{kl}}. \end{aligned} \tag{3.10}$$

3.3.3 Posterior Over Rates

With $p(\pi|N)$ as the target posterior, we can sample the partition directly, without the need to sample the rates. Nevertheless, it is simple to recover these rates to establish the relationships between clusters. Furthermore, as we discuss in Subsection 4.1.4, certain experiments require samples of λ to use in likelihood calculations. Thus, we derive the posterior for λ given the data and partition as

$$\begin{aligned} p(\lambda|N,\pi) &\propto p(N|\lambda,\pi)p(\lambda|\pi) \\ &= \prod_{k=1}^{c}\prod_{l=1}^{c} \frac{(\lambda_{kl}T)^{m_{kl}} e^{-\lambda_{kl} T n_k n_l}}{\prod_{u:\pi_u=k}\prod_{v:\pi_v=l} x_{uv}!} \left(\frac{1}{T}\right)^{m_{kl}} \frac{\beta^\delta}{\Gamma(\delta)} \lambda_{kl}^{\delta-1} e^{-\beta\lambda_{kl}} \\ &\propto \prod_{k=1}^{c}\prod_{l=1}^{c} {\lambda_{kl}}^{m_{kl}+\delta-1} e^{-(Tn_k n_l+\beta)\lambda_{kl}}, \end{aligned}$$

which we recognize as an unnormalized Gamma distribution. Thus, the posterior distribution for λ is

$$p(\lambda|N,\pi) = \prod_{k=1}^{c}\prod_{l=1}^{c} \text{Gamma}(\lambda_{kl}; m_{kl}+\delta, Tn_k n_l + \beta). \tag{3.11}$$

Given the transaction data and MCMC samples of $\pi|N$, we can reinstantiate samples of λ. That is, we can draw a rate λ_{kl}, independently from the Gamma distribution in Equation 3.11, for every pair of clusters $k, l \in \text{range}(\pi)$.

3.3.4 Implementation

As mentioned in Subsection 3.3.1, due to the large sample space for π—even for modest numbers of vertices—we rely on MCMC to perform approximate inference. Using Expression 3.10 as the unnormalized target posterior, we implement a Metropolis-Hastings algorithm with a simple, symmetric proposal distribution. The procedure for sampling from the proposal distribution executes as shown in Algorithm 1.

Algorithm 1 Sample a partition, π^*, from the proposal distribution

```
1: procedure SAMPLEPARTITION
2:     π* ← π, where π is the partition at the current iteration
3:     Pick a vertex u ∈ V uniformly at random
4:     if ∃v ∈ V such that π_v = π_u and v ≠ u then
5:         Pick any existing cluster, or a new one, uniformly at random
6:     else
7:         Pick any existing cluster uniformly at random
8:     Assign π*_u to the chosen cluster
9:     return π*
```

This pseudo-code presents a simple proposal distribution, whereby at most one vertex moves to a new cluster at each sweep of the Metropolis-Hastings algorithm. The `if-else` statement simply checks whether or not the chosen vertex is alone in its current cluster. If this vertex is alone in its cluster, then moving it to a new, unoccupied cluster would be equivalent to picking its current cluster. Since this proposal is symmetric, the acceptance ratio for the Metropolis-Hastings algorithm simplifies to calculating the ratio of $p(\pi^*|N)$ to $p(\pi|N)$. We note that more complicated proposals, like split-merge algorithms [8], have been shown to improve computational efficiency towards exploring multiple modes.

Unless otherwise noted, we initialize our sampler with a partition that places each vertex in its own cluster. This initialization is sensible when the prior favours a low number of clusters, thus allowing the chain to generally explore more of

the sample space. Note that for a CRP with concentration parameter α, the expected number of occupied clusters grows as $O(\alpha \ln n)$, when there are n vertices (customers) [11]. Other sensible initializations include placing each vertex in the same cluster, and drawing a random partition from the prior. However, both may limit the mixing of the chain when the observed data is sparse. Subsection 4.1.2 examines the effect of different initializations in further detail.

3.3.5 Sampling Hyperparameters

Blundell et al. [6] did not discuss the choice of priors or sampling procedures for the hyperparameters in the PPIRM, so we choose to proceed as follows. We place an exponential prior on the concentration parameter, α, such that $p(\alpha) = e^{-\alpha}$. A rate parameter of one enforces a relatively small mean on α; consequently, the CRP with parameter α tends to produce partitions with a smaller number of clusters. This helps to capture the intuition that the prior should favour partitions with small numbers of clusters [20].

Recall that the Poisson process rate parameters are drawn independently from a Gamma distribution with shape δ and rate β. We, in turn, place Gamma priors on δ and β, and set their respective shape and rate parameters all to 0.01. This imposes a relatively small mean and large variance.

We again implement a Metropolis-Hastings algorithm to sample new values of the hyperparameters. Sampling is done separately for each hyperparamter in a deterministic order. We use a Normal distribution, truncated at zero, for the proposal distribution. Specifically, for a hyperparameter, θ, we propose a new value, θ^*, from the proposal distribution

$$q(\theta^*|\theta) = \frac{\phi(\theta^* - \theta)}{\Phi(\theta)}, \qquad \theta^* > 0,$$

where $\phi(\cdot)$ denotes the standard Normal density and $\Phi(\cdot)$ denotes the standard Normal cumulative distribution function. Then, the acceptance ratio for the Metropolis-

Hastings algorithm becomes

$$\frac{p(\theta^*)q(\theta|\theta^*)}{p(\theta)q(\theta^*|\theta)} = \frac{p(\theta^*)\phi(\theta-\theta^*)\Phi(\theta)}{p(\theta)\phi(\theta^*-\theta)\Phi(\theta^*)} = \frac{p(\theta^*)\Phi(\theta)}{p(\theta)\Phi(\theta^*)}.$$

3.4 Note on the Superposition Property

Blundell et al. [6] give little detail regarding their inference procedure for the PPIRM. Yet they note that conjugacy in the model can be maintained due to the superposition property of Poisson processes. It may be tempting to employ this idea when deriving the likelihood, $p(N|\lambda,\pi)$, by combining the transactions from all vertices in one cluster towards any vertex in a second cluster. In this way, the rates of each Poisson process would be additively combined via the superposition property. Furthermore, the posterior over partitions, as shown in Expression 3.10, is independent of the transaction counts x_{uv} between each vertex pair; rather, it depends on the total counts m_{kl} between each cluster pair. While this may seem like a compelling reason to combine the rates via the superposition property, we show that doing so would be misguided. In particular, using the superposition property to derive the likelihood will forfeit relevant information about the individual, vertex-specific interaction data.

Recall that m_{kl} denotes the number of transactions from all vertices in cluster k towards any vertex in cluster l by time T. Suppose that cluster k contains only the vertices u and u', while cluster l contains only vertex v, and $k \neq l$. Recall that x_{uv} and $x_{u'v}$ denote the individual transaction counts between ordered pairs of vertices, and that $m_{kl} = x_{uv} + x_{u'v}$. Now, consider the event $\{m_{kl} = 2\}$, which occurs when either $\{x_{uv} = 2, x_{u'v} = 0\}$, $\{x_{uv} = 1, x_{u'v} = 1\}$, or $\{x_{uv} = 0, x_{u'v} = 2\}$. The probability of each event, disregarding the temporal component, is given by

the Poisson distribution and written as

$$\begin{aligned}
\mathbb{P}(X_{uv}=2, X_{u'v}=0) &= \mathbb{P}(X_{uv}=0, X_{u'v}=2) \\
&= \frac{(\lambda_{kl}T)^2 e^{-\lambda_{kl}T}}{2!} \cdot \frac{(\lambda_{kl}T)^0 e^{-\lambda_{kl}T}}{0!} \\
&= \frac{(\lambda_{kl}T)^2 e^{-2\lambda_{kl}T}}{2}; \qquad (3.12) \\
\mathbb{P}(X_{uv}=1, X_{u'v}=1) &= \frac{(\lambda_{kl}T)^1 e^{-\lambda_{kl}T}}{1!} \cdot \frac{(\lambda_{kl}T)^1 e^{-\lambda_{kl}T}}{1!} \\
&= (\lambda_{kl}T)^2 e^{-2\lambda_{kl}T}. \qquad (3.13)
\end{aligned}$$

Compare this to the probability of the event $\{m_{kl}=2\}$, which we can derive using the superposition property, to give

$$\begin{aligned}
\mathbb{P}(M_{kl}=2) &= \frac{(\lambda_{kl}n_k n_l T)^2 e^{-\lambda_{kl}n_k n_l T}}{2!} \\
&= 2(\lambda_{kl}T)^2 e^{-2\lambda_{kl}T}. \qquad (3.14)
\end{aligned}$$

Comparing Equations 3.12 and 3.13, it is obvious that different vertex-specific counts occur with different probabilities. Furthermore, neither scenario occurs with the same probability as the combined Poisson count in Equation 3.14. Thus, conditioning solely on the event $\{m_{kl}=2\}$ discards information contained in the full transaction dataset with vertex-specific counts. This simple example shows that using only the sum of transaction counts across clusters, m_{kl}, will in some cases forfeit information contained in the full data. Therefore, we refrain from using the superposition of Poisson processes in deriving the likelihood and posterior, despite the suggestion of Blundell et al. [6].

Chapter 4

Experiments

This chapter assesses the performance of the PPIRM and showcases its potential with real data. Section 4.1 uses synthetic data to first establish the computational correctness of the inference algorithm presented in Chapter 3. Next, the PPIRM's performance is evaluated on several experiments designed to assess clustering performance and prediction. Section 4.2 uses real data from militarized conflicts between nations to model the correlates of war. Inferred partitions reveal structured relationships between groups of countries, which are compared against historical records.

4.1 Synthetic Data

To establish the validity of our inference algorithm, and to assess the performance of the PPIRM, synthetic data can be generated to run several experiments. Synthetic data is simulated from the generative process described in Section 3.2. For a given seed, number of entities, and parameter values for α, δ and β, data can be generated up to a stopping time T. By increasing T, further transactions can be generated from the specified model.

Figure 4.1 displays one instance of synthetic data, at various stopping times, represented as the multidigraph described in Section 3.2. Each subfigure is a snapshot of the generated transactions produced by the continuous-time process, up to the given stopping time. For this particular instance, the parameters were set to

$\alpha = 1$, $\delta = 0.3$ and $\beta = 0.01$. This setting enforces a relatively large variance on the Poisson process rate parameters, λ. Consequently, this allows for better contrast between the rates of transactions. By $T = 0.05$, in Figure 4.1b, we already observe a relatively large number of transactions between the orange and green clusters, and a relatively low number of transactions from the green to blue cluster. This pattern is further solidified as the stopping time is increased in Figures 4.1c and 4.1d. Note that we omit the times associated with each edge for the sake of clarity; instead, the edges are given different shades of colours to signify that each transaction occurs at a unique time point.

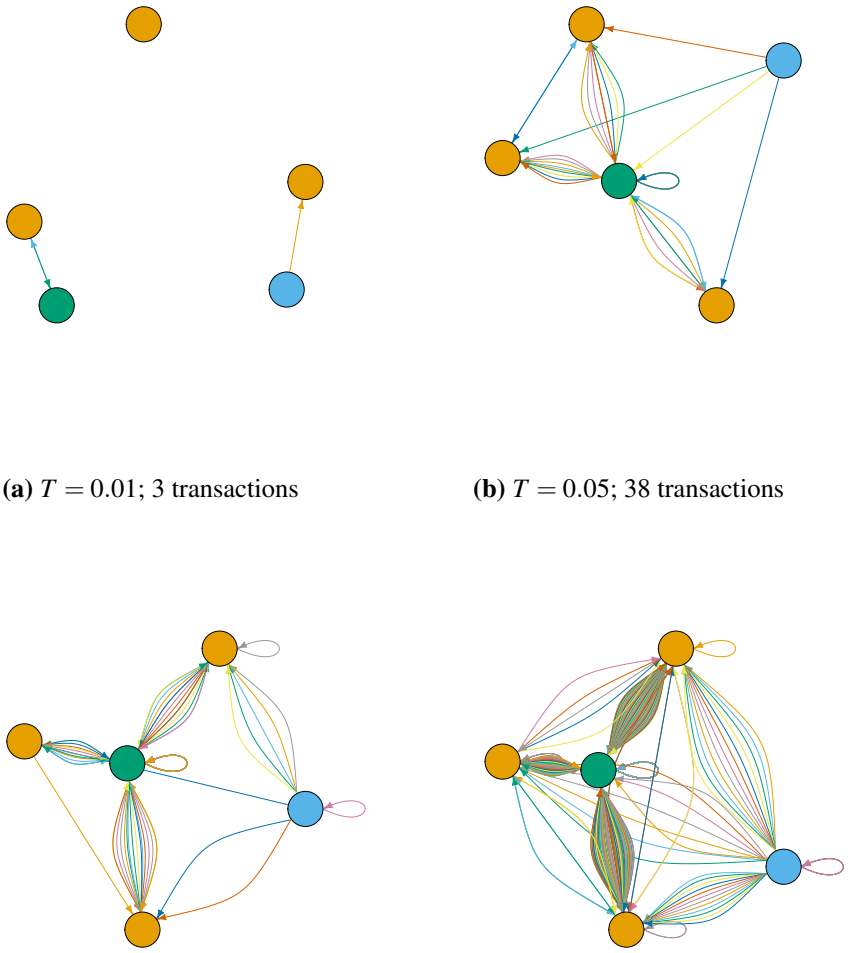
Subsection 4.1.1 discusses an approach that uses synthetic data to establish the computational correctness of our MCMC sampler for partitions. Subsection 4.1.2 examines two diagnostic techniques for assessing convergence of the Markov Chains used for approximate inference, and compares the effects of using various initializations. Finally, Subsections 4.1.3 and 4.1.4 present several experiments for assessing the clustering and predictive performance of the PPIRM using synthetic data.

4.1.1 Validation of Sampler

As we have designed our own inference scheme for the PPIRM, we must implement these procedures from scratch. In particular, we rely on the MCMC algorithms discussed in Section 3.3.4, to perform approximate inference on this Bayesian model. Consequently, a critical precursory step to further experimentation is to verify the computational correctness of the implemented sampler. We are particularly interested in verifying the correctness of the sampler for partitions.

Two popular approaches for validating Bayesian software are those of Cook et al. [7] and Geweke [12]. For non-scalar parameters, like partitions, the approach by Cook et al. [7] is not directly applicable. This is because their validation methods are based on the posterior quantiles of scalar parameters.

Geweke [12] presents an alternative strategy for testing Bayesian software, which relies on a two-stage process. First, data is drawn from the so-called *marginal-conditional simulator*, which is presumed to be error-free. Independent samples from the full joint distribution of $p(\pi, \lambda, N)$ are created by sampling each compo-



(a) $T = 0.01$; 3 transactions

(b) $T = 0.05$; 38 transactions

(c) $T = 0.1$; 69 transactions

(d) $T = 0.5$; 376 transactions

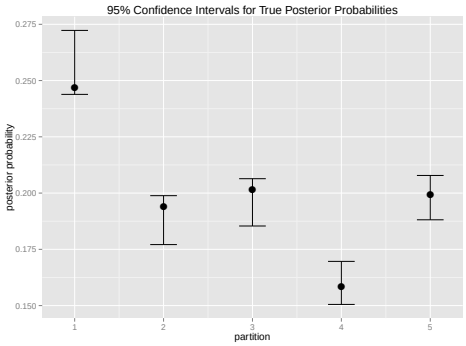
Figure 4.1: Multidigraph representation of randomly generated data, from 5 entities, at various stopping times, T. Vertex colours represent the three different clusters. Edges are coloured to signify that each transaction occurs at a unique time point.

nent, marginally, from the generative process, as described in Section 3.2. Second, the *successive-conditional simulator* creates alternative samples from $p(\pi, \lambda, N)$. This is achieved by generating samples from $p(N|\lambda, \pi)$ for each MCMC sample of (π, λ). Finally, a function f, such that $f(\pi, \lambda, N) \in \mathbb{R}$, is applied to each element of these two sequences from $p(\pi, \lambda, N)$, and a Student's t-test or Kolmogorov-Smirnov test is used to evaluate the results.

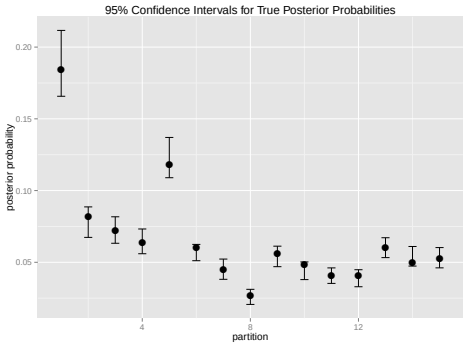
In order to use either the Cook et al. [7] or Geweke [12] approach, an appropriate function must be chosen that maps samples containing π to a real number. As an example, one could use the entropy of the cluster sizes as a function to summarize the partitions. However, such a function is not injective, and thus discards some information in the samples. Additionally, the Geweke [12] approach requires sampling λ at each step, in order to sample N, which we aimed to avoid in the first place by marginalizing out λ. For these reasons, we instead employ a simpler approach to validate our sampler for partitions.

Since the sample space of partitions is discrete, we can enumerate all possible partitions when the number of entities is reasonably small. Thus, we can calculate the posterior probabilities, up to a normalizing constant, directly from Expression 3.10 for all possible partitions. Then, we normalize these values to find the true posterior probabilities. Using the true probabilities as targets, we can calculate confidence intervals based on the MCMC samples by considering the proportion of iterations that each partition was sampled. Standard errors are calculated using a batch means method, where the batch size is taken to be the square root of the number of iterations.

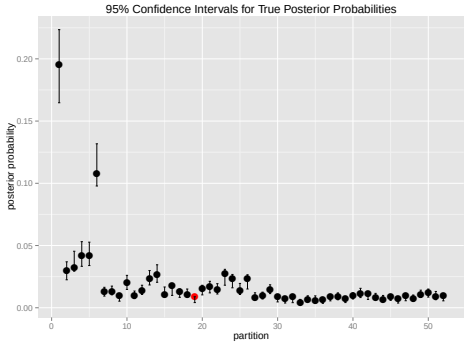
Figure 4.2 displays these 95% confidence intervals on three different datasets of varying numbers of entities. In order to isolate any issues with the sampler for partitions, all other parameters are given and only the partitions were sampled. Note that the numbering of the partitions along the horizontal axis is constructed in an arbitrary order. In Figures 4.2a and 4.2b, we see that all of the true posterior probabilities were captured by the confidence intervals. In Figure 4.2c, we observe that all but one of the 52 true posterior probabilities were captured by the confidence intervals. We conclude that our sampler for partitions appears to be working correctly.



(a) 3 entities; 5 partitions



(b) 4 entities; 15 partitions



(c) 5 entities; 52 partitions

Figure 4.2: 95% confidence intervals for the posterior probabilities of each possible partition. Estimates are based on 10,000 MCMC samples with Monte Carlo standard errors calculated by a batch means method. Each datapoint shows the true posterior probability, and points in red are those not captured by the confidence interval.

4.1.2 Convergence Diagnostics

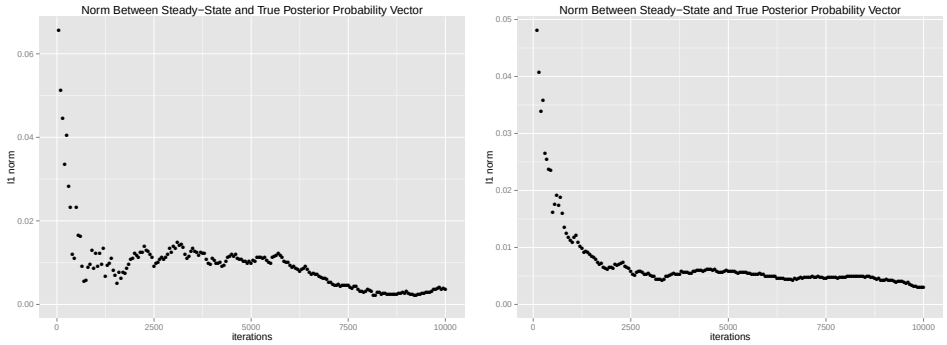
After establishing that the MCMC sampler is targeting the correct posterior distribution, a natural next step is to consider the rate at which the Markov Chain converges to the targeted posterior. Furthermore, it is common practice to use diagnostic tools to assess MCMC sampler convergence on full-scale problems. Thus, we examine two approaches for assessing the convergence of our samplers: an original approach that scores the stationary distribution of the Markov Chain over

various iterations, and a second approach that looks at the trace plots of samples generated from a larger synthetic dataset.

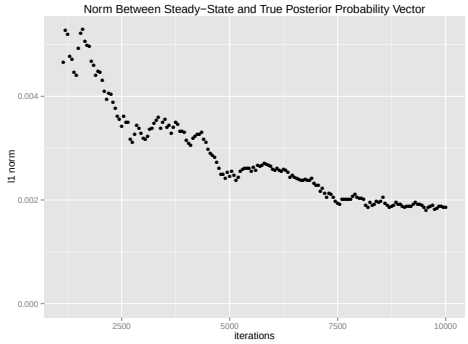
Utilizing the fact that we can calculate the true posterior probabilities, for all possible partitions, when the number of entities is reasonably small, we develop the following original experiment. We measure the distance between the vector of true posterior probabilities and the estimated posterior probabilities from the MCMC samples of partitions. At any given iteration, n, of the MCMC sampler, we can calculate an empirical transition matrix, $M^{(n)}$. The entry $M^{(n)}_{kl}$ is the proportion of iterations, up to n, that partition l was sampled, given that partition k was the previous state. From this transition matrix, a number of methods can be used to solve for the steady-state distribution. We use the eigendecomposition to solve for the eigenvector corresponding to an eigenvalue of one. We then measure the ℓ_1 norm (or *Manhattan distance*) between the steady-state vector and the vector of true posterior probabilities. Formally, the ℓ_1 norm between two vectors $\mathbf{x}$ and $\mathbf{y}$, both of length m, is calculated as $\sum_{i=1}^{m} |x_i - y_i|$. We choose the ℓ_1 norm, as we believe it to be a sensible choice; however, we recognize that other metrics, like the ℓ_2 norm, could be equally as valid. Finally, we examine the sequence of norm values, over various numbers of MCMC iterations. The rate at which this sequence approaches zero is indicative of the rate at which the Markov Chain targets the correct posterior distribution.

Figure 4.3 displays three instances of the aforementioned experiment on various numbers of entities. In Figures 4.3a and 4.3b, we observe that for 3 and 4 entities, the sequences tend to flatten out by about $2,500$ iterations. Pay special attention to the scale in Figure 4.3c. While for 5 entities the sequence continues to decrease, the norm values are just as close to zero. Thus, we conclude that in each case the estimated posterior probabilities quickly approach the true values; however, for larger numbers of entities, where there are considerably more partitions, longer chains will likely more accurately estimate all probabilities.

As a further diagnostic check, trace plots permit a visual assessment of the MCMC sampler convergence. A trace plot displays the sampled parameter values over all iterations. One method for assessing whether the sampler has converged to the targeted posterior distribution is to compare the trace plots from different chains, started from different initializations. Ideally, the chains should reach the



(a) 3 entities **(b)** 4 entities

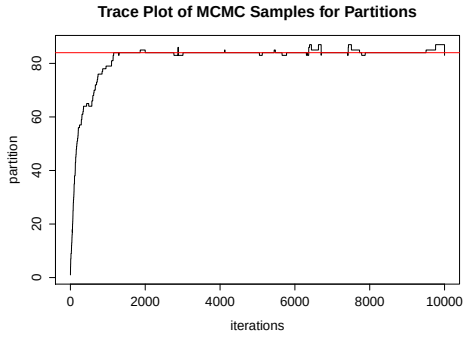


(c) 5 entities

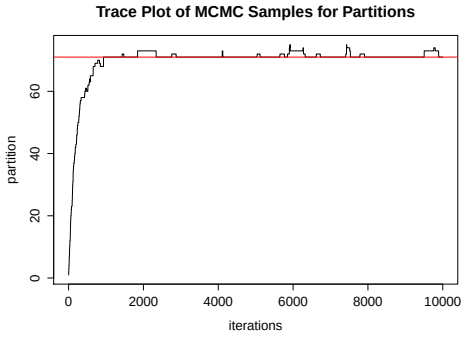
Figure 4.3: ℓ_1 norm, measured at various iterations, between: (i) the vector of true posterior probabilities of each possible partition, and (ii) the steady-state distribution of the empirical transition matrix of MCMC samples.

same (and correct) posterior distribution, despite different initializations. Furthermore, with sufficient data and identifiability of the model, we might expect this distribution to have a mode close to the true parameter value.

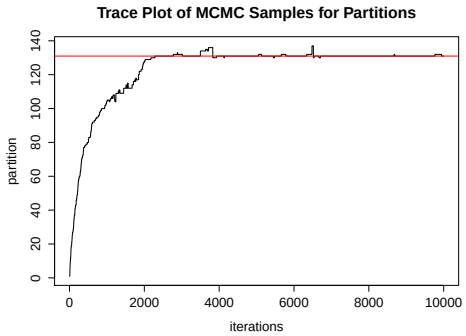
We generate synthetic data from 30 entities and run the MCMC samplers for all parameters in a deterministic order. Figure 4.4 shows the trace plots for the sampled partitions, having started from three different initializations: (a) all entities placed in different clusters, so that there are 30 clusters; (b) all entities placed in the same cluster, so that there is only one cluster; (c) a random draw from the CRP prior. We number the partitions arbitrarily, based simply on the order in which they were



(a) Initialization: all different clusters



(b) Initialization: all same cluster



(c) Initialization: random draw from prior

Figure 4.4: Trace plots of MCMC samples of partitions, given different initializations. Data is comprised of 336 randomly generated transactions from 30 entities. The red lines depict the true partition. Numbering along the vertical axis is arbitrary.

sampled. In each case, the chains appear to converge to a mode around the true partition, depicted by the red line, by about $2,000$ iterations. We note that the chain started from the initialization that placed all entities in the same cluster, shown in Figure 4.4b, sampled the fewest number of distinct partitions. Conversely, the chain started from a random partition from the prior (shown in Figure 4.4c) visited more states before reaching the mode; however, these visits were largely transient.

Using the same synthetic dataset from 30 entities, we compare the MCMC samples for the hyperparameters of the PPIRM. Figure 4.5 shows the trace plots for

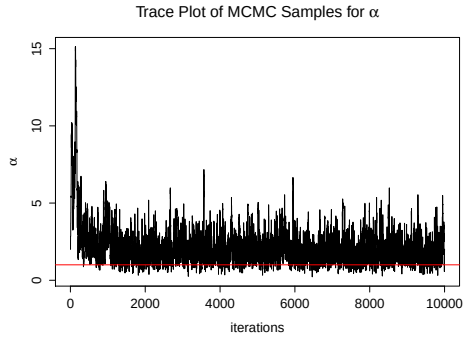
the sampled hyperparameters, α, δ and β, given various initial values. In each case, the two chains appear to converge towards the same mode around the true parameter value, after relatively few iterations. Note that for this dataset, there are 7 true clusters and hence a total of 49 different Poisson process rate parameters. This means that there are less than 7 observed transactions per the 49 different rate parameters that comprise λ. Thus, despite a large number of entities and with relatively few observed transactions, the samplers for the hyperparameters quickly converge to a distribution around the true parameter values.

4.1.3 Clustering Evaluation

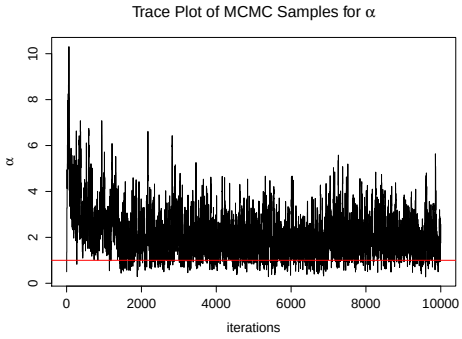
As mentioned in Section 3.1, the first main problem that we wish to address is to learn a latent clustering of the entities. With synthetic data we can evaluate the PPIRM's performance compared to a ground-truth partition, on what is otherwise an unsupervised learning task. Specifically, we can evaluate the PPIRM's ability to infer the number of clusters, and compare the average number of clusters in the MCMC samples as the amount of data is increased. Figure 4.6 displays two such examples with the same parameter settings and the same number of clusters in the ground-truth partition, but different numbers of entities. In both cases, all model parameters were sampled, except λ, which was marginalized out. Only the generated data (up to the specific stopping time) was used during inference.

In Figure 4.6a, we observe that the PPIRM accurately recovers the true number of clusters when the number of entities is reasonably small, and after only a moderate amount of data is observed. In this simulation, 128 transactions were observed by the stopping time $T = 0.05$. We note that, here, 128 transactions is less than 4 transactions per the 36 different rate parameters that comprise λ, given that there were 6 true clusters. Furthermore, there are a total of 10^2 possible ordered pairs of entities, given that there are 10 entities and self-loops are permitted.

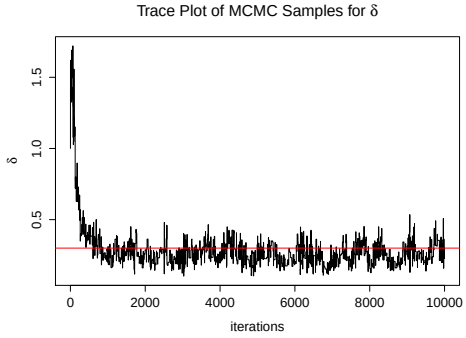
As discussed in Subsection 3.3.1, increasing the number of entities leads to a combinatorial explosion in the number of possible partitions. In Figure 4.6b, we triple the number of entities, but leave fixed the number of true clusters and the number of MCMC samples. We observe that even with increasing data, the PPIRM tends to slightly over-estimate the number of clusters. This phenomenon has been



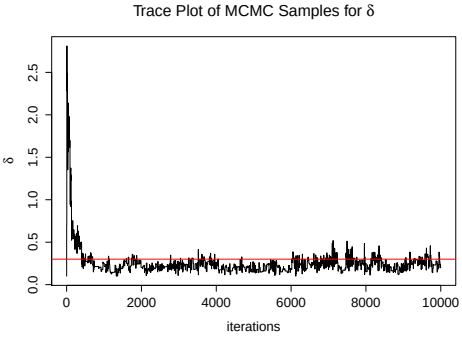
(a) Initial $\alpha = 2$



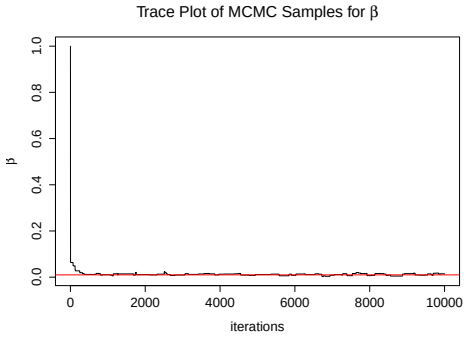
(b) Initial $\alpha = 0.5$



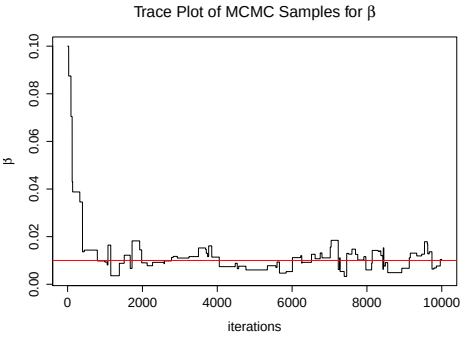
(c) Initial $\delta = 1$



(d) Initial $\delta = 0.1$

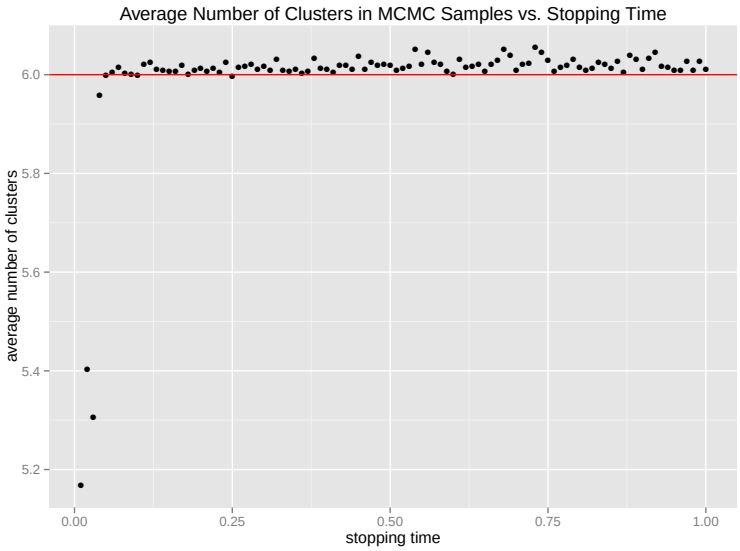


(e) Initial $\beta = 1$

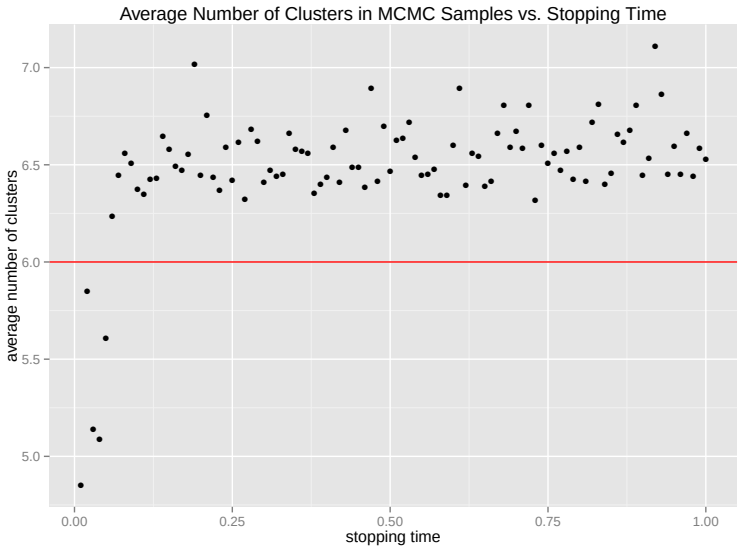


(f) Initial $\beta = 0.1$

Figure 4.5: Trace plots of MCMC samples for each of the hyperparameters, given different initializations. Data is comprised of 336 randomly generated transactions from 30 entities. The red lines depict the true parameter values.



(a) 10 entities; 6 true clusters



(b) 30 entities; 6 true clusters

Figure 4.6: Mean number of clusters in $10,000$ MCMC samples as the stopping time is increased. The red line depicts the number of clusters in the ground-truth partition used to generate the synthetic data.

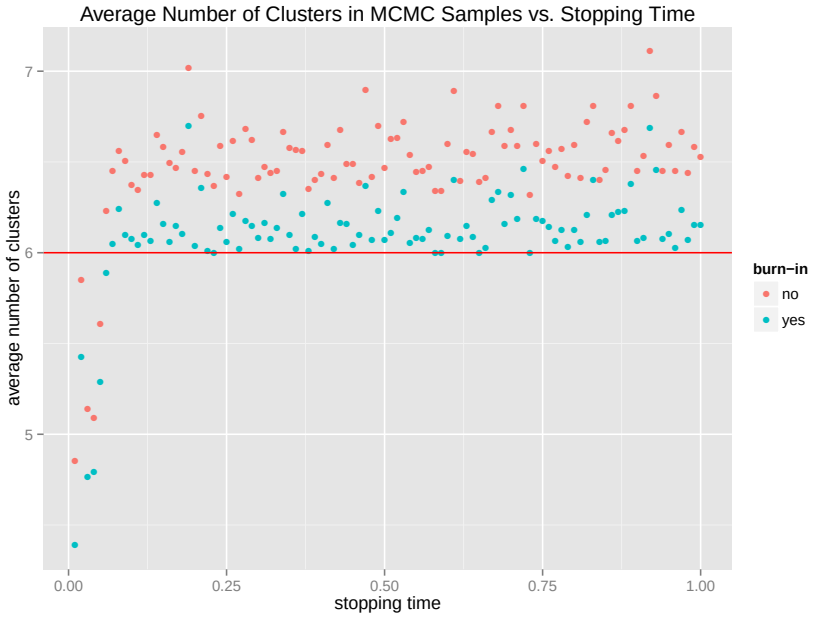


Figure 4.7: Mean number of clusters in the MCMC samples as the stopping time is increased. Red points are an average of all $10,000$ samples, while blue points drop the first 10% as burn-in. The red line depicts the number of clusters in the ground-truth partition used to generate the synthetic data.

studied for Dirichlet process mixture models by Miller and Harrison [22].

Given that our initialization for the partition sampler places each entity in its own cluster, there are as many clusters as entities when the sampler starts. Furthermore, the sampler moves only, at most, one entity per iteration; thus, it takes several iterations before the chain can reach a partition with only 6 clusters. For these reasons, we rerun the previous experiment with 30 entities and drop the first $1,000$ MCMC samples as burn-in. Figure 4.7 shows that when we exclude the first 10% of samples, the mean number of clusters is closer to the true value of 6.

To further evaluate the PPIRM's clustering performance, we assess the quality of the learned clusters based on various point estimates from the MCMC samples. We measure quality of the learned partition using the adjusted Rand index [19]. The adjusted Rand index measures the similarity between two partitions and ad-

justs for the expected number of chance agreements under uniform random sampling. Compared to a ground-truth, a randomly generated partition has an expected score of zero, while the correct partition has a score of one.

We consider four different point estimates from the MCMC samples:

(i) the MAP as observed from the MCMC samples. Note that as an estimator, the MAP is equivalent to the sequence of Bayes estimators that minimize the posterior expectation of the 0-1 loss function [10].

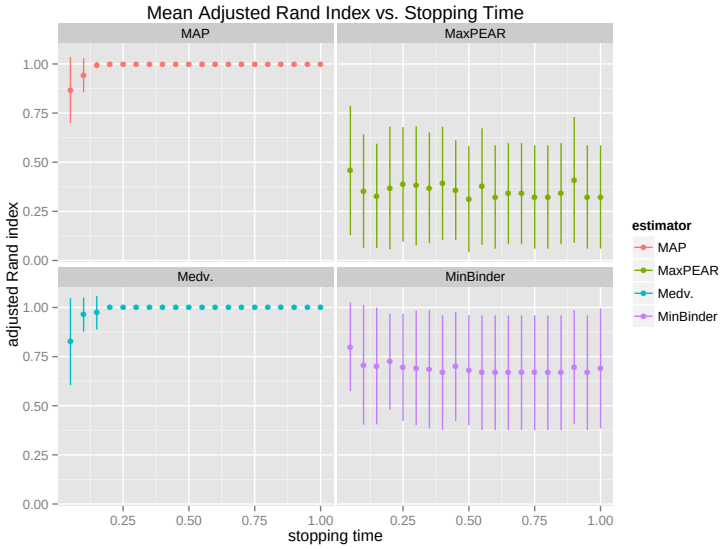
(ii) The partition based on an ad-hoc approach by Medvedovic et al. [21] that uses hierarchical clustering. We refer to this estimator as MEDV for short.

(iii) The partition that minimizes Binder's loss function [4]. This is also equivalent to maximizing the posterior expected Rand index with the true partition [10]. We refer to this estimator as MINBINDER.

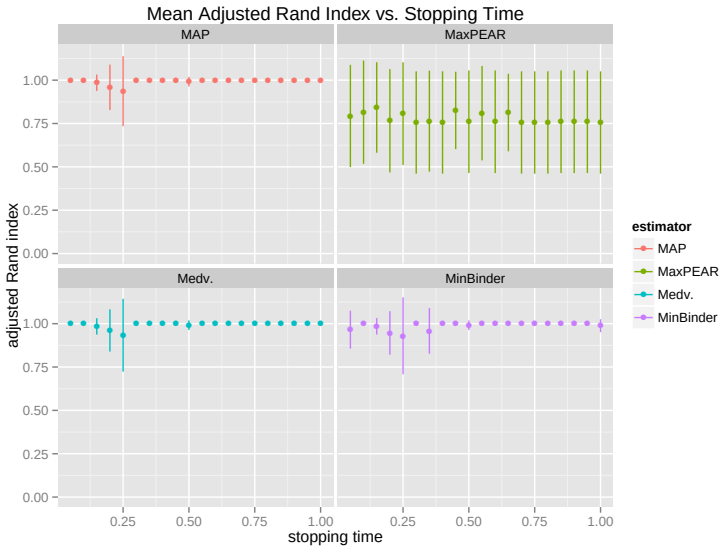
(iv) The partition that maximizes the posterior expected adjusted Rand index with the true partition. We refer to this estimator as MAXPEAR.

Since Fritsch and Ickstadt [10] explain and examine each estimator in detail, we omit these details here and simply compare their results with the PPIRM. Figure 4.8 illustrates two examples comparing these four estimators as the amount of data is increased. In both subfigures, ten datasets were randomly generated using the same parameter settings, but with different number of entities. From 10,000 MCMC samples, each estimator was computed and its adjusted Rand index with the true partition was averaged across the ten replicates.

In Figure 4.8a, we see that the MAP and MEDV estimators effectively recover the true partition when the number of entities is reasonably small, and after only a moderate amount of data is observed. Recall that an adjusted Rand index score of one indicates perfect recovery of the true partition. On the other hand, the MINBINDER and MAXPEAR estimators seem to be highly variable and, on average, show no improvement with increasing amounts of observed data. In Figure 4.8b, we see that when the number of entities is tripled the MAP and MEDV estimators still accurately recover the true partition; however, their accuracy is subject to some noise variations in the data. Furthermore, the MINBINDER and MAXPEAR estimators actually show improved accuracy with a greater number of entities; although,



(a) 10 entities



(b) 30 entities

Figure 4.8: Adjusted Rand index between a partition estimate and the true partition as the stopping time is increased. Each datapoint is an average across 10 randomly generated datasets, and error bars show one standard deviation. Each estimator is computed from 10,000 MCMC samples.

the MAXPEAR estimator is still quite variable. In sum, we see that the PPIRM accurately recovers the true partition, provided that an appropriate estimator is used.

4.1.4 Prediction

The second objective mentioned in Section 3.1 is to model the time-series structure of the transaction data. This requires inferring both a partition of the entities, as well as the rate parameters between each ordered pair of clusters. Together, the inferred parameters can be used to model the stochastic processes that govern the transactions, and make predictions about unobserved transactions.

With synthetic data we evaluate the PPIRM's ability to model a set of held-out data, given a set of training data used for inference. From the training data, the PPIRM produces MCMC samples of the partition using the scheme described in Section 3.3.4. For each sampled partition, we sample the rates, λ, by the procedure described in Section 3.3.3. In order to quantify how well the PPIRM explains held-out data, we estimate the log posterior predictive density of test data, N^*, given training data, N. The log posterior density is

$$\begin{aligned}
\log p(N^*|N) &= \log \int \sum_{\pi} p(N^*|N,\lambda,\pi)p(\lambda,\pi|N)d\lambda \\
&= \log \int \sum_{\pi} p(N^*|\lambda,\pi)p(\lambda,\pi|N)d\lambda \\
&= \log \mathbb{E}_{\lambda,\pi|N}\left[p(N^*|\lambda,\pi)\right], \qquad (4.1)
\end{aligned}$$

where the likelihood on the test data, $p(N^*|\lambda,\pi)$, can be evaluated using Equation 3.8. We can approximate the expectation in Equation 4.1 using MCMC samples from the joint density $\lambda,\pi|N$. That is,

$$\log \mathbb{E}_{\lambda,\pi|N}\left[p(N^*|\lambda,\pi)\right] \approx \log \left[\frac{1}{n}\sum_{i=1}^{n} p(N^*|\lambda^{(i)},\pi^{(i)})\right], \qquad (4.2)$$

where $(\lambda^{(i)},\pi^{(i)})$ are approximate samples from $\lambda,\pi|N$.

In our experiments, we compute the righthand-side of Expression 4.2 using the following procedure:

1. Generate synthetic data, with known parameter values, until stopping time

T.

2. Transactions occurring within the last $x\%$ of time become the test set; the remaining transactions are the training set. Note that this is not necessarily equivalent to using the last $x\%$ of all observed transactions as the test set.

3. Generate n approximate samples of $\pi|N$ by the MCMC methods discussed in Subsection 3.3.4.

4. Generate a sample of $\lambda|N,\pi^{(i)}$ for each MCMC sample, $\pi^{(i)}$, from $i = 1,2,\ldots,n$. Use the posterior distribution derived in Subsection 3.3.3.

5. For all n sample pairs, $\{(\pi^{(i)},\lambda^{(i)})\}_{i=1}^{n}$, calculate the likelihood on the test data. That is, calculate $p(N^*|\lambda^{(i)},\pi^{(i)})$, for $i = 1,2,\ldots,n$.

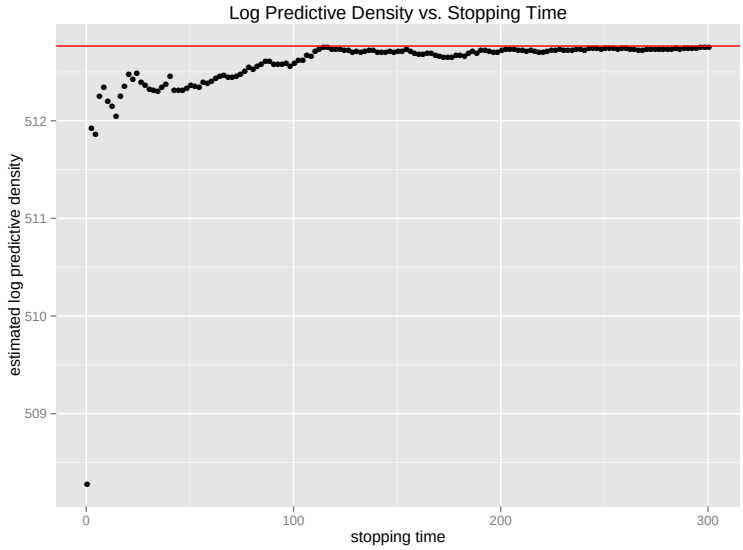
6. Compute $\log\left[\frac{1}{n}\sum_{i=1}^{n} p(N^*|\lambda^{(i)},\pi^{(i)})\right]$.

Figure 4.9 displays two examples that used this procedure with the same parameter settings, but with different numbers of entities. In both cases, all model parameters were sampled, including λ.

In Figure 4.9a, the test set is comprised of 270 transactions between 10 entities; by stopping time $T = 114$, there are $310,332$ transactions in the training set. We observe that after this point, the PPIRM has effectively learned the model parameters with reasonably sufficient accuracy for the model to fit the test data. Specifically, the training set is large enough for the inference algorithm to accurately recover the true parameters of the model, to a degree of accuracy dictated by this test set. Notice that, at this point, the estimated log predictive densities show a pattern of convergence to the likelihood of the test set given the true partition and true rates (as depicted by the red line). To be clear, the red line shows the value of $\log p(N^*|\lambda_t,\pi_t)$, where λ_t and π_t denote the true model parameters used to generate the synthetic data. Figure 4.9b shows a similar pattern when the number of entities is tripled.

4.2 Real Data

Through learning a latent clustering of entities, the PPIRM can be used to glean insight from real data that may otherwise be too massive or too convoluted for the



(a) 10 entities; 270 transactions in the test set

(b) 30 entities; 2,342 transactions in the test set

Figure 4.9: Estimated log posterior predictive densities on a test set as the amount of training data (stopping time) is increased. Each estimate is an average based on 10,000 MCMC samples. The red line depicts the likelihood of the test set when the true partition and true rates are given.

human eye to decipher. By inferring both the cluster memberships and the rates of transactions between clusters, the PPIRM can reveal varying degrees of structured relationships between groups of interacting pairs. We display this feature of the PPIRM on a dataset comprised of a group of countries and incidents where one side threatens, displays, or uses force against another. In keeping with the experiments of Blundell et al. [6], we use version 3.02 of the Militarized Interstate Dispute (MID) dataset, compiled by the Correlates of War Project [15]. Specifically, we extract transactions from the dyadic MID dataset [14], which is comprised of disputing pairs of countries.

MIDs are defined as cases of conflict between states in which the threat or use of military force, short of war, is explicitly directed towards the government or territory of another state [15]. A conflict can be classified as a MID if it results in fewer than $1,000$ deaths, and some military force is used [27]. Version 3.02 of the MID dataset spans the years 1993–2001. Each dyad in the dataset is ordered to identify which state took the first militarized action against the other. We use the start date of each dispute as the observed time of transaction. For MIDs involving multiple countries, a dyad is recorded for each ordered pair of countries involved in the conflict. Ghosn et al. [15] provides a codebook describing the full dataset, as well as a detailed, historical description of each conflict.

The dyadic MID dataset consists of 138 countries—these are the entities of the PPIRM—and 508 recorded disputes between the years 1993 and 2001. We fit the PPIRM to this data by using the MCMC methods described in Chapter 3. We ran our inference algorithm for $10,000$ iterations, starting from the initial partition that places each country in its own cluster, and discarded the first 500 burn-in samples. From the MCMC samples of partitions, we compute the point-estimates discussed in Section 4.1.3. We find that different estimates produce partitions of various coarseness; however, overall, the inferred clusterings are in keeping with that discussed by Ghosn et al. [15].

For illustrative purposes, we show two examples of the learned clusterings, and discuss how these match the historical narrative. Figure 4.10 displays the MAXPEAR partition of the 138 countries, represented as a multidigraph. Each country is a vertex of the graph, both coloured and numbered according to its respective cluster assignment. The 508 disputes are represented as the directed

edges of the graph, coloured in grey. We found that the MAXPEAR partition is the coarsest of the four partition estimates, in the sense that it has the fewest number of clusters, and hence the largest cluster sizes. The MAXPEAR partition has 18 clusters, whereas the MAP has 58, and is the finest partition. For comparison, Figure 4.11 shows the MAP partition on the same data, with a slightly different spatial arrangement for clarity.

Examining the clustering produced by the MAXPEAR, one striking feature is that cluster 8 (grey) is surrounded by and heavily connected to the countries in cluster 1 (purple). As it turns out, cluster 8 is comprised of only Yugoslavia, while cluster 1 contains 27 of the 28 NATO countries, excluding Latvia. Indeed, the dispute narrative [15] confirms that Yugoslavia entered into many conflicts with NATO countries, spurred by its involvement in the Bosnian civil war and general hostility over Kosovo dating back to before 1993. This hostility erupted into a series of incidents including a blockade imposed on Yugoslavia by Western countries, and NATO airstrikes resulting from the strife in Kosovo. The fact that Latvia is not part of cluster 1 is not at all surprising, as it did not officially join NATO until 2004, well after the coverage of this dataset. Furthermore, during the time period covered, Latvia had several of its own border disputes with Russia.

For the finer partition produced by the MAP, the PPIRM is able to identify several dispute threads, for which we highlight the relevant cluster memberships in Table 4.1. In particular, we note that cluster 19 identifies the countries who joined forces after the Afghan incursions into Tajikistan, circa 1994. The dispute narrative [15] lists repeated conflicts along the Tajik-Afghan border, accounting for the many edges between clusters 19 and 51 (Afghanistan). We note that the countries in cluster 19 took more of a secondary role in these conflicts, often led by Russia. Consequently, the MAP partitions Russia into its own cluster.

Cluster 43 identifies the dispute thread between two geographically contiguous groups of African countries. These disputes led to several incidents, most stemming from the Democratic Republic of Congo conflict, and eventually ending in a peace agreement between countries from both groups in 1999. Similarly, cluster 54 pinpoints the East Asian countries that experienced several disputes involving the seizure of patrol boats and fishing vessels along the Pacific coast.

Using the PPIRM to extract further insights, we recovered the Poisson process

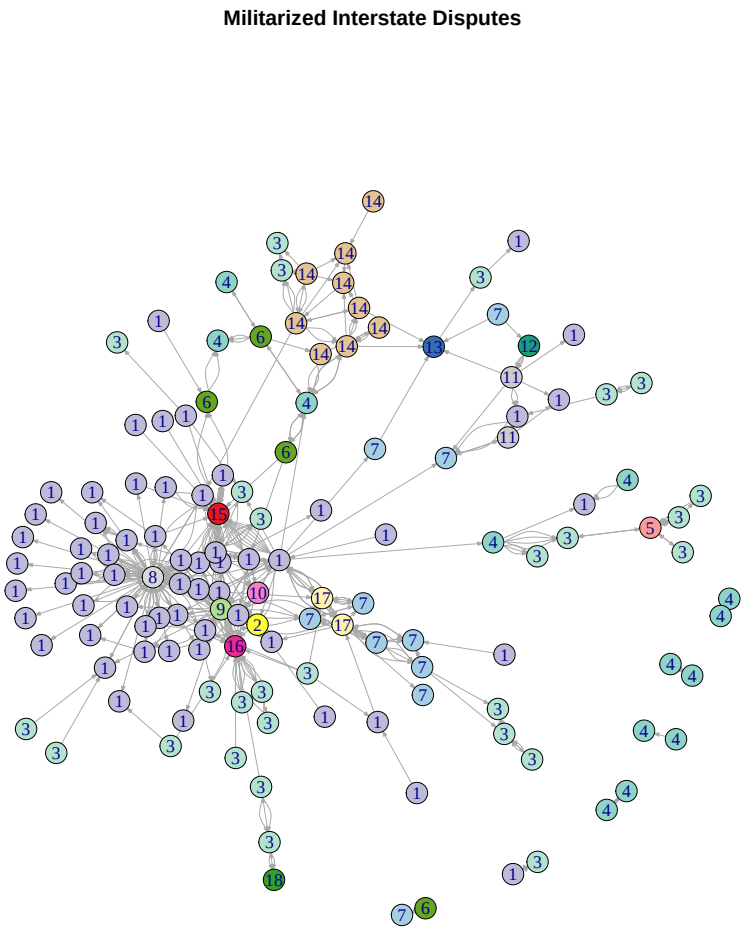


Figure 4.10: Multidigraph representation of the MID dataset consisting of 138 countries (vertices) and 508 disputes (edges). Vertices are both coloured and numbered according to the 18 clusters inferred from the MAXPEAR partition.

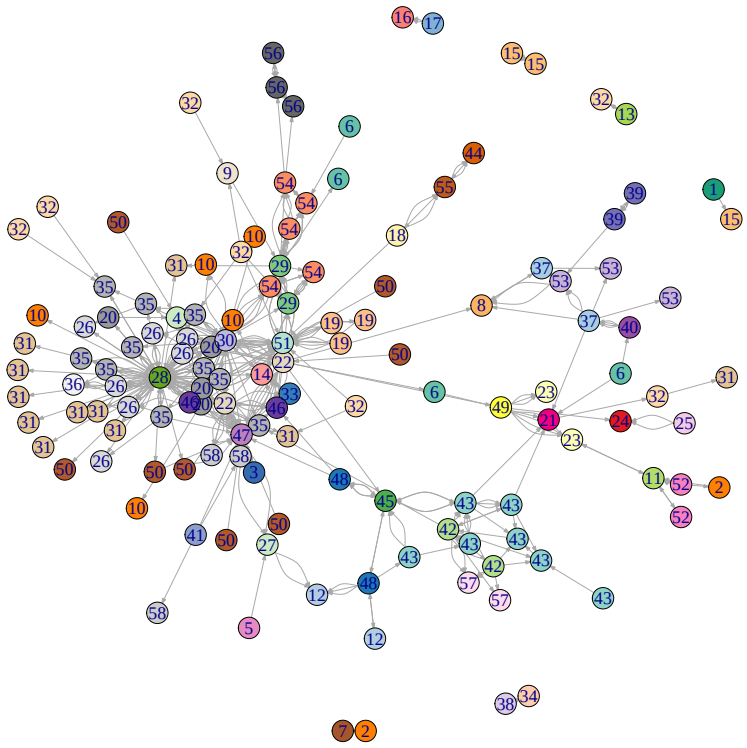


Figure 4.11: Multidigraph representation of the MID dataset consisting of 138 countries (vertices) and 508 disputes (edges). Vertices are both coloured and numbered according to the 58 clusters inferred from the MAP partition.

Table 4.1: Countries belonging to some of the clusters inferred from the MAP partition of the MID dataset.

MAP Cluster Label	Countries Belonging to Cluster
3	Kuwait
14	Azerbaijan
19	Tajikistan, Kyrgyzstan, Uzbekistan
22	U.S.A., Turkey
28	Yugoslavia
30	Russia
33	Armenia
35	Netherlands, Belgium, Luxembourg, Germany, Hungary, Italy, Lithuania, Ukraine, Norway, Denmark
43	Uganda, Kenya, Rwanda, Ethiopia, Zimbabwe, Namibia, Botswana
47	Iraq
51	Afghanistan
54	Taiwan, South Korea, Japan, Vietnam, Philippines

rates by sampling directly from the posterior. That is, with the MAP estimates of the partition and hyperparameters, δ and β, we sampled estimates of the rates, λ, using Equation 3.11. We produced $1,000$ samples for the 58×58 matrix λ, and took an entry-wise average over all samples. We note that, alternatively, one could compute the MAP of each entry of λ directly, as opposed to sampling. Table 4.2 identifies the clusters corresponding to the six largest average rates, as a means for discovering which groups of countries were most heavily involved in MIDs during the timeframe.

Again, these results closely adhere to the dispute narratives presented by Ghosn et al. [15], and correspond to the groups of countries most predominantly involved in MIDs. Furthermore, the PPIRM discovers patterns of similarly behaving countries. In particular, U.S.A. and Turkey are both heavily involved in conflicts with Iraq and Yugoslavia, separately. Consequently, the MAP partition clusters U.S.A. and Turkey together, and the recovered rates reveal their historical propensity for conflict with Iraq and Yugoslavia. Finally, we note that the rates affirm the afore-

Table 4.2: The clusters and countries corresponding to the 6 largest average Poisson process rates recovered from $1,000$ posterior samples. Samples were based on MAP estimates of the model parameters. Clusters for the rates of transactions are shown in the order *sender* $\rightarrow$ *recipient*.

Rank	Clusters	Countries Belonging to Clusters
1	$22 \rightarrow 47$	U.S.A., Turkey $\rightarrow$ Iraq
2	$3 \rightarrow 47$	Kuwait $\rightarrow$ Iraq
3	$33 \rightarrow 14$	Armenia $\rightarrow$ Azerbaijan
4	$30 \rightarrow 51$	Russia $\rightarrow$ Afghanistan
5	$22 \rightarrow 28$	U.S.A., Turkey $\rightarrow$ Yugoslavia
6	$19 \rightarrow 51$	Tajikistan, Kyrgyzstan, Uzbekistan $\rightarrow$ Afghanistan

mentioned narrative of a series of disputes involving Russia, Tajikistan and its neighbouring countries, towards Afghanistan.

Chapter 5

Conclusion

This thesis takes the PPIRM, originally proposed by Blundell et al. [6], and derives the necessary posterior distributions to implement a fully Bayesian inference scheme. The implementation of the MCMC sampler and choices of hyperpriors are discussed in full detail. Furthermore, a simple example is presented to show that the superposition of Poisson processes should not be used in deriving the posterior, despite the suggestion of Blundell et al. [6]. Experimental results are presented to validate the correctness of the implementation, which is a non-standard task when sampling partitions of a set. Synthetic data are used to assess the clustering and predictive performance of the PPIRM. Real data from a history of militarized conflicts between nations are used to showcase the PPIRM's ability to reveal varying degrees of structured relationships. Finally, Section 5.1 discusses some prospective directions for future work, building upon the PPIRM.

5.1 Future Work

There are several future directions that could be explored to improve the performance and extend the applicability of the PPIRM. As far as extending applicability, one could imagine any of the following additions to the transactional data. There can be

(i) entity-specific attributes;

(ii) different types of transactions;

(iii) additional side-channel information that can be used to describe patterns, like periodicity, in the event times.

In a Bayesian context, both (i) and (ii) may require an extra model layer that draws attributes or event types from a given distribution. As an example of (iii), information like the time of day could potentially reveal new patterns in the transactional data. This could be modelled via an inhomogeneous Poisson process, whose rate is dependent on both time and an intensity function over the side-channel information.

More direct alterations to the PPIRM include modifying the Gamma distribution in Expression 3.2 to be zero-inflated. Subsequently, this would permit some of the Poisson process rates to be zero, thereby preventing certain clusters from interacting with each other. As mentioned in Section 3.3.4, implementing a split-merge proposal distribution could help to improve the computational efficiency of the inference algorithm towards exploring multiple modes. An example of a split-merge algorithm can be found in [8].

Different choices for the prior over partitions could be used with the PPIRM. In theory, one may wish to cluster together entities that interact closely in time. A *distance dependent Chinese Restaurant Process* [5] allows the random cluster assignments of the entities to be based on the distances between them. These distances can be based on time, space, or other characteristics—like entity-specific attributes. To incorporate time dependency, distances between vectors of transaction times can be calculated between each ordered pair of entities. However, in doing so, the data is effectively used in shaping the prior.

Alternatively, the *Indian Buffet Process* [13] could be used to model vectors of latent features, as opposed to simply latent classes. This would relate the work of Miller et al. [23] and Palla et al. [24] to the PPIRM, thereby allowing each entity to potentially belong to any number of clusters. These latent features could be combined through a weight matrix, and transformed to values in $\mathbb{R}_{>0}$ that can be used as Poisson process rates. Thus, the rate of transactions between any ordered pair of entities would be determined by the the combined effect of all pairwise feature interactions.

Bibliography

[1] Class LinkedHashSet. https://docs.oracle.com/javase/7/docs/api/java/util/LinkedHashSet.html. Accessed: 2016-07-21. → pages 14

[2] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9(Sep):1981–2014, 2008. → pages 3, 13

[3] D. J. Aldous. Exchangeability and related topics. In *École d'Été de Probabilités de Saint-Flour XIII1983*, pages 1–198. Springer, 1985. → pages 4, 5, 6

[4] D. A. Binder. Bayesian cluster analysis. *Biometrika*, 65(1):31–38, 1978. → pages 36

[5] D. M. Blei and P. I. Frazier. Distance dependent chinese restaurant processes. *Journal of Machine Learning Research*, 12(Aug):2461–2488, 2011. → pages 48

[6] C. Blundell, J. Beck, and K. A. Heller. Modelling reciprocating relationships with hawkes processes. In *Advances in Neural Information Processing Systems*, pages 2600–2608, 2012. → pages ii, iii, 2, 3, 4, 7, 11, 12, 14, 15, 21, 22, 23, 41, 47

[7] S. R. Cook, A. Gelman, and D. B. Rubin. Validation of software for bayesian models using posterior quantiles. *Journal of Computational and Graphical Statistics*, 2012. → pages 25, 27

[8] D. B. Dahl. Sequentially-allocated merge-split sampler for conjugate and nonconjugate dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 11, 2005. → pages 20, 48

[9] C. DuBois and P. Smyth. Modeling relational events via latent classes. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 803–812. ACM, 2010. → pages 2, 3

[10] A. Fritsch and K. Ickstadt. Improved criteria for clustering based on the posterior similarity matrix. *Bayesian analysis*, 4(2):367–391, 2009. → pages 36

[11] S. J. Gershman and D. M. Blei. A tutorial on bayesian nonparametric models. *Journal of Mathematical Psychology*, 56(1):1–12, 2012. → pages 5, 6, 21

[12] J. Geweke. Getting it right: Joint distribution tests of posterior simulators. *Journal of the American Statistical Association*, 99(467):799–804, 2004. → pages 25, 27

[13] Z. Ghahramani and T. L. Griffiths. Infinite latent feature models and the indian buffet process. In *Advances in neural information processing systems*, pages 475–482, 2005. → pages 48

[14] F. Ghosn and S. Bennett. Codebook for the dyadic militarized interstate incident data, version 3.10. *Online: http:// correlatesofwar.org*, 2003. → pages 41

[15] F. Ghosn, G. Palmer, and S. A. Bremer. The mid3 data set, 1993–2001: Procedures, coding rules, and description. *Conflict Management and Peace Science*, 21(2):133–154, 2004. → pages 41, 42, 45

[16] M. Girvan and M. E. Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99 (12):7821–7826, 2002. → pages 3, 15

[17] T. Hofmann, J. Puzicha, and M. I. Jordan. Learning from dyadic data. *Advances in neural information processing systems*, pages 466–472, 1999. → pages 1

[18] P. W. Holland, K. B. Laskey, and S. Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983. → pages 2, 3, 7, 13, 14

[19] L. Hubert and P. Arabie. Comparing partitions. *Journal of classification*, 2 (1):193–218, 1985. → pages 35

[20] C. Kemp, J. B. Tenenbaum, T. L. Griffiths, T. Yamada, and N. Ueda. Learning systems of concepts with an infinite relational model. In *AAAI*, volume 3, page 5, 2006. → pages 2, 3, 4, 6, 7, 8, 9, 10, 21

[21] M. Medvedovic, K. Y. Yeung, and R. E. Bumgarner. Bayesian mixture model based clustering of replicated microarray data. *Bioinformatics*, 20(8): 1222–1232, 2004. → pages ix, 36

[22] J. W. Miller and M. T. Harrison. A simple example of dirichlet process mixture inconsistency for the number of components. In *Advances in neural information processing systems*, pages 199–206, 2013. → pages 35

[23] K. Miller, M. I. Jordan, and T. L. Griffiths. Nonparametric latent feature models for link prediction. In *Advances in neural information processing systems*, pages 1276–1284, 2009. → pages 48

[24] K. Palla, D. A. Knowles, and Z. Ghahramani. Relational learning and network modelling using infinite latent attribute models. *IEEE transactions on pattern analysis and machine intelligence*, 37(2):462–474, 2015. → pages 48

[25] J. Pitman. Some probabilistic aspects of set partitions. *The American Mathematical Monthly*, 104(3):201–209, 1997. → pages 16

[26] J. Pitman. Combinatorial stochastic processes. Technical Report 621, Lecture notes for St. Flour Summer School, 2002. → pages 4, 6

[27] M. R. Sarkees. The cow typology of war: Defining and categorizing wars (version 4 of the data). *Note with version 4 of the Correlates of War Data*, 2010. → pages 41

[28] A. Simma and M. I. Jordan. Modeling events with cascades of poisson processes. *Uncertainity in Artificial Intelligence (UAI)*, 2010. → pages 3

[29] Y. W. Teh. Dirichlet process. In *Encyclopedia of machine learning*, pages 280–287. Springer, 2011. → pages 6

[30] S. Wasserman and C. Anderson. Stochastic a posteriori blockmodels: Construction and assessment. *Social Networks*, 9(1):1–36, 1987. → pages 2, 3, 7, 13, 14

[31] L. Xin, M. Zhu, and H. Chipman. A continuous-time stochastic block model for basketball networks. *arXiv preprint arXiv:1507.01816*, 2015. → pages 3

[32] Z. Xu, V. Tresp, K. Yu, and H.-P. Kriegel. Learning infinite hidden relational models. *Uncertainity in Artificial Intelligence (UAI2006)*, 2006. → pages 2, 4, 6