DISSERTATION

BAYESIAN MODELS AND STREAMING SAMPLERS FOR COMPLEX DATA WITH APPLICATION TO NETWORK REGRESSION AND RECORD LINKAGE

Submitted by

Ian M. Taylor

Department of Statistics

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Fall 2023

Doctoral Committee:

Advisor: Andee Kaplan
Co-Advisor: Bailey K. Fosdick

Kayleigh P. Keller
Matthew D. Koslovsky
Peter Jan van Leeuwen

ABSTRACT

BAYESIAN MODELS AND STREAMING SAMPLERS FOR COMPLEX DATA WITH APPLICATION TO NETWORK REGRESSION AND RECORD LINKAGE

Real-world statistical problems often feature complex data due to either the structure of the data itself or the methods used to collect the data. In this dissertation, we present three methods for the analysis of specific complex data: Restricted Network Regression, Streaming Record Linkage, and Generative Filtering.

Network data contain observations about the relationships between entities. Applying mixed models to network data can be problematic when the primary interest is estimating unconditional regression coefficients and some covariates are exactly or nearly in the vector space of node-level effects. We introduce the Restricted Network Regression model that removes the collinearity between fixed and random effects in network regression by orthogonalizing the random effects against the covariates. We discuss the change in the interpretation of the regression coefficients in Restricted Network Regression and analytically characterize the effect of Restricted Network Regression on the regression coefficients for continuous response data. We show through simulation on continuous and binary data that Restricted Network Regression mitigates, but does not alleviate, network confounding. We apply the Restricted Network Regression model in an analysis of 2015 Eurovision Song Contest voting data and show how the choice of regression model affects inference.

Data that are collected from multiple noisy sources pose challenges to analysis due to potential errors and duplicates. Record linkage is the task of combining records from multiple files which refer to overlapping sets of entities when there is no unique identifying field. In streaming record linkage, files arrive sequentially in time and estimates of links are updated after the arrival of each file. We approach streaming record linkage from a Bayesian perspective with estimates calculated

from posterior samples of parameters, and present methods for updating link estimates after the arrival of a new file that are faster than fitting a joint model with each new data file. We generalize a two-file Bayesian Fellegi-Sunter model to the multi-file case and propose two methods to perform streaming updates. We examine the effect of prior distribution on the resulting linkage accuracy as well as the computational trade-offs between the methods when compared to a Gibbs sampler through simulated and real-world survey panel data. We achieve near-equivalent posterior inference at a small fraction of the compute time.

Motivated by the streaming data setting and streaming record linkage, we propose a more general sampling method for Bayesian models for streaming data. In the streaming data setting, Bayesian models can employ recursive updates, incorporating each new batch of data into the model parameters' posterior distribution. Filtering methods are currently used to perform these updates efficiently, however, they suffer from eventual degradation as the number of unique values within the filtered samples decreases. We propose Generative Filtering, a method for efficiently performing recursive Bayesian updates in the streaming setting. Generative Filtering retains the speed of a filtering method while using parallel updates to avoid degenerate distributions after repeated applications. We derive rates of convergence for Generative Filtering and conditions for the use of sufficient statistics instead of storing all past data. We investigate properties of Generative Filtering through simulation and ecological species count data.

ACKNOWLEDGEMENTS

I would like to thank my advisors, Dr. Andee Kaplan and Dr. Bailey Fosdick, for their teaching, guidance, encouragement, and gentle prodding on the long road to completing this dissertation. Without you, this work could not exist. I also owe a great deal of thanks to Dr. Brenda Betancourt and Dr. Kayleigh Keller, for their collaboration on the chapters in this dissertation. Thanks also to my committee members past and present, Dr. Kayleigh Keller, Dr. Matthew Koslovsky, Dr. Peter Jan van Leeuwen, Dr. Julia Sharp, and Dr. Jay Breidt, for providing invaluable support at critical moments in the process of finishing my degree.

I would like to especially thank my family for their unwavering love and support from a distance. And thanks to the family I gained here – DeMoulin, Durakovich, Porter, Christopherson, Appleton, and Love – for making Colorado feel like home.

Finally, there is no one I would rather have been with through the last six years than my wife, Ginny. You have been a constant source of love, stability, and encouragement and you never let me give up on myself. The completion of this dissertation is a big change for us – you've only known me as a graduate student – but I know that wherever life takes us we'll go together. I love you.

TABLE OF CONTENTS

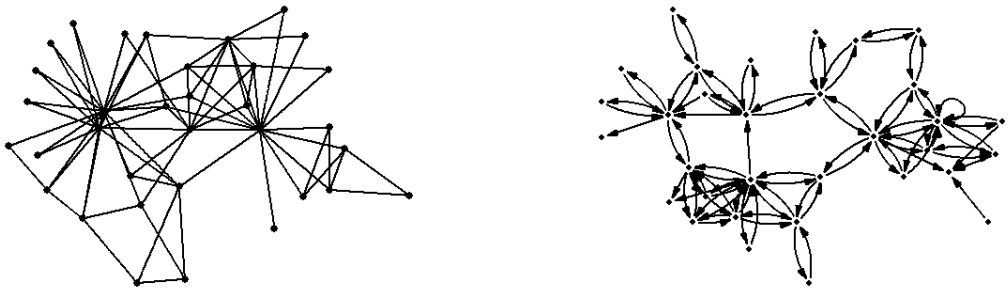
Chapter 1

Introduction

Complexity in data due to either the structure of the data itself or the methods used to collect the data presents challenges to analysis. In some cases, dependence in the data presents a challenge for otherwise standard statistical techniques. In other cases, data come from multiple sources containing errors and duplicates that must be accounted for before analysis. Data may also be streaming, i.e., generated from some ongoing process, and analysis cannot wait until data collection is finished to begin. In this dissertation, we present three methods for the analysis of complex data: Restricted Network Regression, Streaming Record Linkage, and Generative Filtering. In Chapter 2, we introduce restricted regression for network data to resolve collinearity between network-structured fixed and random effects in mixed models. In Chapter 3, we develop a model for record linkage in the streaming data setting, where noisy duplicates across files must be consolidated while files continue to arrive. Motivated by streaming record linkage, in Chapter 4, we introduce Generative Filtering, a Markov chain Monte Carlo (MCMC) sampler for performing recursive Bayesian updates in the streaming data setting. In Chapter 5, we close with a summary and discussion of future work that could build on the contributions in previous chapters. The remainder of this chapter contains background and exposition to the topics in the following chapters.

1.1 Network Data

Network data are measurements about the pairwise relationships between individuals or entities. Network data are often represented as a graph where entities are points, or nodes, and relationships are represented as lines connecting the points, or edges (Figure 1.1). The simplest kind of network data only contain the presence or absence of a relationship between two entities. For example, the Zachary karate club network (Zachary, 1977) contains data on the presence or absence of friendships between members of a karate club as determined by interactions between members outside of the club. As the friendships have no directional component, we call this kind



(A) (B)

Figure 1.1: Examples of undirected (A) and directed (B) networks depicted as graphs. (A) The Zachary karate club network (Zachary, 1977), showing friendships between members of a karate club. (B) Links between US domestic terrorist websites (Zhou et al., 2005), where the direction of an edge shows a link from one website to another. The network in (B) also contains self-edges, represented by loops.

of network undirected. By contrast, directed networks contain relationships with a directional component. For example, the presence or absence of hyperlinks between US domestic terrorist websites (Zhou et al., 2005) are directional – a link from one website to another is different from a link from the second website to the first, and one link may exist without the other. In addition to relationships between entities, networks may also contain relationships between entities and themselves, for example, a website linking to its own pages. These relationships are represented by loops on graphs.

More complicated networks contain not just the presence or absence of relationships, but also measurements about those relationships. These networks are often represented as $n \times n$ relational matrices, $\boldsymbol{Y}$, where n denotes the number of nodes. The value of the component y_{ij} is the measurement of the relationship between node i and node j. If the network is directed, then $\boldsymbol{Y} \neq \boldsymbol{Y}^\top$ in general, while if it is undirected, $\boldsymbol{Y} = \boldsymbol{Y}^\top$. Examples of such networks include international trade networks (Marrs et al., 2022) or friendship rank nomination networks (Hoff et al., 2013).

When analyzing relational matrices or networks, we are often interested in the effect of one or more covariates on a response network. Using a (possibly generalized) linear regression model to infer these relations is known as network regression. Network regression has been applied to

medical meta-analysis (e.g., Li et al., 2018; Gwon et al., 2020), analysis of international politics (e.g., Campbell et al., 2019), and social networks (e.g., Cantner and Graf, 2006). A general network regression model is,

$$y_{ij} = \boldsymbol{x}_{d,ij}^{\top}\boldsymbol{\beta}_d + \boldsymbol{x}_{s,i}^{\top}\boldsymbol{\beta}_s + \boldsymbol{x}_{r,j}^{\top}\boldsymbol{\beta}_r + \varepsilon_{ij},$$

where $\boldsymbol{x}_{d,ij}$ is a vector of dyadic covariates measured on the relationship of node i to node j, $\boldsymbol{x}_{s,i}$ are sender covariates measured on the sending node, i, and $\boldsymbol{x}_{r,j}$ are receiver covariates measured on the receiving node, j. Sender and receiver covariates are also called row and column covariates, respectively, due to the structure of the relational matrix, $\boldsymbol{Y}$. Because of the network structure of the response, $\boldsymbol{Y}$, the error structure accounts for dependence among the observations, either with random effects (Holland et al., 1983; Wang and Wong, 1987; Hoff et al., 2002; Li and Loken, 2002; Hoff, 2005) or with covariance of $\boldsymbol{\varepsilon}$ under an exchangeability assumption (Marrs et al., 2022). In Chapter 2, we characterize network confounding, which can occur when including both sender or receiver covariates and network-structured random effects in the model. We then introduce Restricted Network Regression as a model that mitigates network confounding and investigate its performance through theory, simulation when $\boldsymbol{Y}$ is continuous and discrete, and a case study of Eurovision Song Contest voting data. We show that Restricted Network Regression produces estimates of unconditional regression effects with less bias and properly calibrated credible interval coverage relative to other network regression models.

1.2 Record Linkage

Record linkage is the process of combining or matching records from multiple files, often when no unique identifying information is available. Records may contain errors (e.g., from manual entry) in addition to the absence of a unique identifier, which complicates the matching process. When there are unique identifiers and no errors, the matching process is trivial, and statistical record linkage methods are not necessary. The goal of record linkage is to identify which records

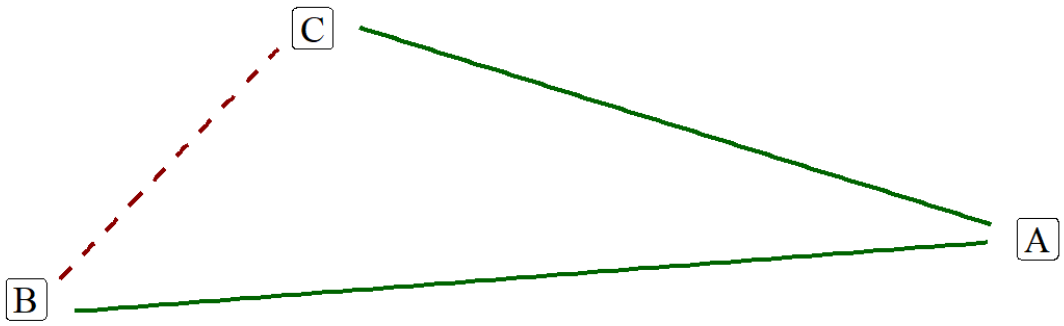


Figure 1.2: An example of a contradiction that may arise in a record linkage procedure, visualized as a graph. Here three records, represented as A, B, and C, are to be linked. A is estimated to be linked to B, and A is estimated to be linked to C, but B is not estimated to be linked to C. This is a contradiction because by transitivity, B and C must refer to the same entity but they are not linked.

are duplicates and refer to the same entity as other records. Duplicate records representing the same entity may appear within the same file or across multiple files. In the literature, the term "record linkage" typically refers to the case in which files are duplicate-free and duplicates appear across files, while "deduplication" refers to the case in which there may be duplicates within files. In Chapter 3, we focus on the record linkage case by assuming no duplicates within the same file.

Record linkage is used in cases when either the removal of duplicates in itself is a goal or when the consolidation of duplicates allows for downstream analysis. In the former case, researchers may be interested in total population counts when individuals from the population are counted in distinct, overlapping ways (e.g., Sadinle and Fienberg, 2013). In the latter case, a downstream analysis may involve relating information that is only in one file to information that is only in another (e.g., Fleming et al., 2012). When the uncertainty of each link is quantified, this uncertainty can be incorporated into a downstream analysis (Kaplan et al., 2023).

Two broad classes of record linkage or deduplication models are Fellegi-Sunter style models (Fellegi and Sunter, 1969), or latent entity models (e.g., Steorts, 2015). Fellegi-Sunter style models first compare pairs of potentially linked records, then categorize pairs of records as either matched or non-matched based on the strength of their similarity. Various restrictions may be placed on

this categorization to avoid contradictions in the links (Figure 1.2). For example, Sadinle (2017) disallow links from different records in one file to the same record in the other, preventing two records in the same file from being transitively linked in violation of the record linkage assumption. Latent entity models assume there exist unobserved true entities and simultaneously estimate the number of true entities, the values associated with the true entities, and the associations between records and entities. Latent entity models avoid potential link contradictions at the expense of a more computationally demanding estimation procedure.

A commonality in the existing statistical record linkage literature is that there are a fixed number of files, and linkage is performed in a single offline procedure. In Chapter 3 we develop a Fellegi-Sunter style record linkage model for the streaming data setting, where files are available sequentially in time, there is no predetermined number of files, and estimates of links are desired after the arrival of each file. Streaming record linkage arises in settings such as longitudinal surveys, electronic health records, and online events databases, among others. The challenge in streaming record linkage is to efficiently update parameter estimates as new data arrive. We define a link constraint to avoid undesired, implied links with many files. We introduce two streaming samplers to efficiently update model parameter estimates with the arrival of new data. We then apply our model to both simulated and real-world record linkage data.

1.3 Streaming Data

In the streaming data setting, data arrive either continuously or in frequent batches, with no predetermined amount of data. Estimates of model parameters are desired after each arrival of new data. The streaming data setting arises in areas such as social network sentiment analysis (e.g., Bifet and Frank, 2010), taxi-passenger demand prediction (e.g., Moreira-Matias et al., 2013), and real-time anomaly detection (e.g., Ahmad et al., 2017). The streaming data setting poses a computational challenge as obtaining model estimates becomes more time-consuming.

For Bayesian modeling, the streaming data setting fits naturally with recursive Bayesian updates, where after the arrival of a new batch of data $\boldsymbol{y}_t$, the posterior distribution of the pa-

rameters $\boldsymbol{\theta}$ is updated using the previous posterior, $p(\boldsymbol{\theta}|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{t-1})$ as a prior and estimating the updated posterior, $p(\boldsymbol{\theta}|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_t) \propto p(\boldsymbol{\theta}|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{t-1})p(\boldsymbol{y}_t|\boldsymbol{\theta})$. Often an additional parameter, $\boldsymbol{\phi}$, characterizes the distribution of $\boldsymbol{y}_t$, so the desired posterior distribution is instead $p(\boldsymbol{\theta}, \boldsymbol{\phi}|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_t) \propto p(\boldsymbol{\theta}|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{t-1})p(\boldsymbol{y}_t|\boldsymbol{\theta}, \boldsymbol{\phi})p(\boldsymbol{\phi})$.

When model parameters have conjugate prior distributions, recursive Bayesian updates can be performed analytically. Otherwise, recursive Bayesian updates can be performed using an approximation to the previous posterior distribution. Posterior distributions are frequently approximated using samples produced by Markov chain Monte Carlo. We focus on methods that perform recursive Bayesian updates by resampling the existing posterior samples, which we refer to as filtering methods, specifically Prior-Proposal-Recursive Bayes (PPRB, Hooten et al., 2021) and PPRB-within-Gibbs (Chapter 3). These filtering methods are fast but suffer from eventual degradation as the pool of samples is reduced through repeated application.

In Chapter 3, we first encounter the streaming data setting in the context of streaming record linkage, where each arrival of new data is a new file to be linked to previous files. Motivated by this application, in Chapter 4 we introduce Generative Filtering, a streaming sampler that extends those introduced in Chapter 3. We introduce Generative Filtering, provide theoretical bounds for its convergence to the posterior distribution, and demonstrate speed and accuracy through application to three different models via simulated datasets and species survey data.

Chapter 2

Restricted Regression in Networks

2.1 Introduction

Network data are measurements about the relationships between pairs of entities. These network measurements can be visualized as measurements on the edges between nodes of a graph (Becker et al., 1995). Examples of network data include relationships between potential borrowers on peer-to-peer lending platforms (Lee and Sohn, 2022) or annual migration between countries (Aleskerov et al., 2017). Data are typically represented as a matrix, $\boldsymbol{Y}$, where y_{ij} is the value in row i and column j, for $i, j = 1, \ldots, n$. The value y_{ij} is the measurement about the dyadic relationship between the sending node i and the receiving node j. If $y_{ij} = y_{ji}$ for all i, j, then the network is called undirected, otherwise it is called directed.

Network regression uses covariates measured on the node pairs to model the dyadic relationships. An example of such a model is

$$y_{ij} = g(z_{ij}), \tag{2.1}$$

$$z_{ij} = \boldsymbol{x}_{ij}^{\top}\boldsymbol{\beta} + \gamma_{ij}, \tag{2.2}$$

where y_{ij} is the observed network measure, $g(\cdot)$ is a function mapping latent continuous values, z_{ij}, to the observed y_{ij}, $\boldsymbol{x}_{ij}$ is a p-vector of covariates related to node i, node j or the relationship from node i to node j, $\boldsymbol{\beta}$ is a p-vector of regression coefficients, and γ_{ij} is random error. For a binary observation $y_{ij} \in \{0, 1\}$ indicating the presence or absence of an edge from node i to node j, a probit version of (2.1) is given by setting

$$g(z_{ij}) = I(z_{ij} > 0), \tag{2.3}$$

$$\gamma_{ij} \overset{\text{i.i.d}}{\sim} N(0, \sigma^2), \tag{2.4}$$

where $I(\cdot)$ is the indicator function (Albert and Chib, 1993). For the rest of this chapter, we will use z_{ij} to refer to a continuous latent variable resulting from the regression equation, and y_{ij} to refer to observations related to z_{ij} through a function, $g(\cdot)$, possibly the identity function.

Network regression has found applications in medical meta-analysis (e.g., Li et al., 2018; Gwon et al., 2020), analysis of international politics (e.g., Campbell et al., 2019), and social networks (e.g., Cantner and Graf, 2006). Other methods for modeling network data include stochastic block models (Holland et al., 1983) and exponential-family random graph models (Robins et al., 2007), which allow inference on latent aspects of the network structure such as clusters and density. While many different properties of a network may be of interest, our primary interest is inference for the regression coefficients, $\boldsymbol{\beta}$.

The network structure of the data creates dependence among the measurements. Observations $\{y_{ij}\}$ with one or more nodes in common, for example y_{ij} and $y_{ij'}$, can be dependent due to their common node, i. If not all of this dependence is captured by the covariates, latent random effects can be used to account for excess variation (Holland et al., 1983; Wang and Wong, 1987; Hoff et al., 2002; Li and Loken, 2002; Hoff, 2005). In this approach, the measurements are modeled as conditionally independent given the latent structure. Additive node random effects can be used to account for dependence in the observations,

$$\gamma_{ij} = a_i + b_j + \varepsilon_{ij}, \tag{2.5}$$

$$a_i \sim g_a(\boldsymbol{\theta}_a), \quad b_j \sim g_b(\boldsymbol{\theta}_b), \tag{2.6}$$

$$\varepsilon_{ij} \overset{\text{i.i.d}}{\sim} N(0, \sigma^2). \tag{2.7}$$

The terms a_i and b_j are called sender and receiver random effects, respectively, and are meant to capture variation due to unobserved node factors, for example, sociability and popularity in social networks. The random effects have distributions, g_a and g_b, parameterized by parameters $\boldsymbol{\theta}_a$ and $\boldsymbol{\theta}_b$. These effects first appeared in the social relations model of Warner et al. (1979). Node random effects have also been incorporated in more complex network models such as popularity-adjusted

block models (Sengupta and Chen, 2018) and additive and multiplicative effects network models (Hoff, 2021). An alternate approach is to model the correlation between measurements on dyads which share a node by the residual covariance matrix under an assumption of exchangeable errors (Marrs et al., 2022).

A key difference between the model in (2.2) with the error structure in (2.4) compared to the error structure in (2.5) is the interpretation of the regression parameters. With the error structure in (2.4), the regression parameters are called the *unconditional regression effects* because they capture the marginal effect of $\boldsymbol{X}$ on $\boldsymbol{z}$. With (2.5), the regression parameters are called the *conditional regression effects* because their value is interpreted as the effect of $\boldsymbol{X}$ conditioned on the random effects $\boldsymbol{a} = (a_1, \ldots, a_n)$ and $\boldsymbol{b} = (b_1, \ldots, b_n)$. For the rest of this chapter, we will distinguish between these interpretations by writing $\boldsymbol{\beta}$ for conditional regression effects and $\boldsymbol{\delta}$ for unconditional regression effects in (2.2).

In network-structured data, covariates can occupy the same linear space as the random effects a_i or b_j. We define this collinearity as *network confounding*. Despite potential impacts of bias due to network confounding, including random effects is typically desired to account for correlation between observations due to network dependence and to allow for more accurate uncertainty quantification for the regression effects. A method that mitigates network confounding would allow for accurate estimation of the unconditional regression effects, while using random effects to account for unobserved network-structured variability in the response.

Confounding between covariates and random effects has been of significant interest in the spatial statistics literature, where it is referred to as *spatial confounding* (Clayton et al., 1993; Reich et al., 2006; Hodges and Reich, 2010). A typical spatial model for continuous areal data is the Intrinsic Conditional Autoregressive (ICAR; Besag et al., 1991) model:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\eta} + \boldsymbol{\varepsilon}, \tag{2.8}$$

$$p(\boldsymbol{\eta}) \propto \tau_s^{n-G} \exp\left(-0.5\tau_s \boldsymbol{\eta}^\top \boldsymbol{Q}\boldsymbol{\eta}\right), \tag{2.9}$$

$$\boldsymbol{\varepsilon} \sim N(\boldsymbol{0}, \sigma^2 \boldsymbol{I}). \tag{2.10}$$

The random effect $\boldsymbol{\eta}$ is intended to capture spatial variation in the response $\boldsymbol{z}$ not accounted for by the predictors $\boldsymbol{X}$. The random effect is regularized to have spatial variation via the matrix $\boldsymbol{Q}$, the Laplacian of the graph of neighboring areas. Spatial confounding occurs when a covariate is smoothly spatially varying, as this creates a scenario when there is collinearity or near collinearity between the fixed covariates $\boldsymbol{X}$ and spatial random effects $\boldsymbol{\eta}$ such that both the random effect and the covariate are attempting to capture similar structure. Hodges and Reich (2010) noted that estimates of regression coefficients $\boldsymbol{\beta}$ can be dramatically affected by the inclusion or exclusion of random effects $\boldsymbol{\eta}$, and introduced restricted spatial regression to resolve the confounding of the random effects and covariates by orthogonalizing $\boldsymbol{\eta}$ against $\boldsymbol{X}$. The restricted spatial regression equation is expressed as

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\delta} + (\boldsymbol{I} - \boldsymbol{P}_X)\boldsymbol{\eta} + \boldsymbol{\varepsilon}, \tag{2.11}$$

where $\boldsymbol{P}_X = \boldsymbol{X}(\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top$ is the linear projection matrix onto the column space of $\boldsymbol{X}$. Restricted spatial regression has been studied extensively in the spatial statistics literature as a means to alleviate spatial confounding (e.g., Hodges and Reich, 2010; Hughes and Haran, 2013; Hanks et al., 2015; Khan and Calder, 2022; Zimmerman and Hoef, 2022). Hanks et al. (2015) show that in a Bayesian setting, inference on both $\boldsymbol{\beta}$ and $\boldsymbol{\delta}$ can be achieved simultaneously by calculating $\boldsymbol{\delta} = \boldsymbol{\beta} + (\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top\boldsymbol{\eta}$ for each posterior sample.

However, the dependence in network data is more complex than the dependence in spatial data. In the network setting, observations are made on the relationships between nodes, and shared nodes between two observations create dependence. This is in contrast to the areal spatial setting, where observations are made on discrete, disjoint areas and the neighbor relation between areas creates dependence (Figure 2.1, A & B). The additive random effects in (2.5), reflect two different ways in which network observations can be related: sharing a sender node (observations y_{ij} and $y_{ij'}$) and sharing a receiver node (observations y_{ij} and $y_{i'j}$). These two types of dependence result in a complex dependence structure (Figure 2.1, C & D). For each of these two types of dependence, the n^2 observations are divided into n groups of n observations which are all mutually dependent within a group.

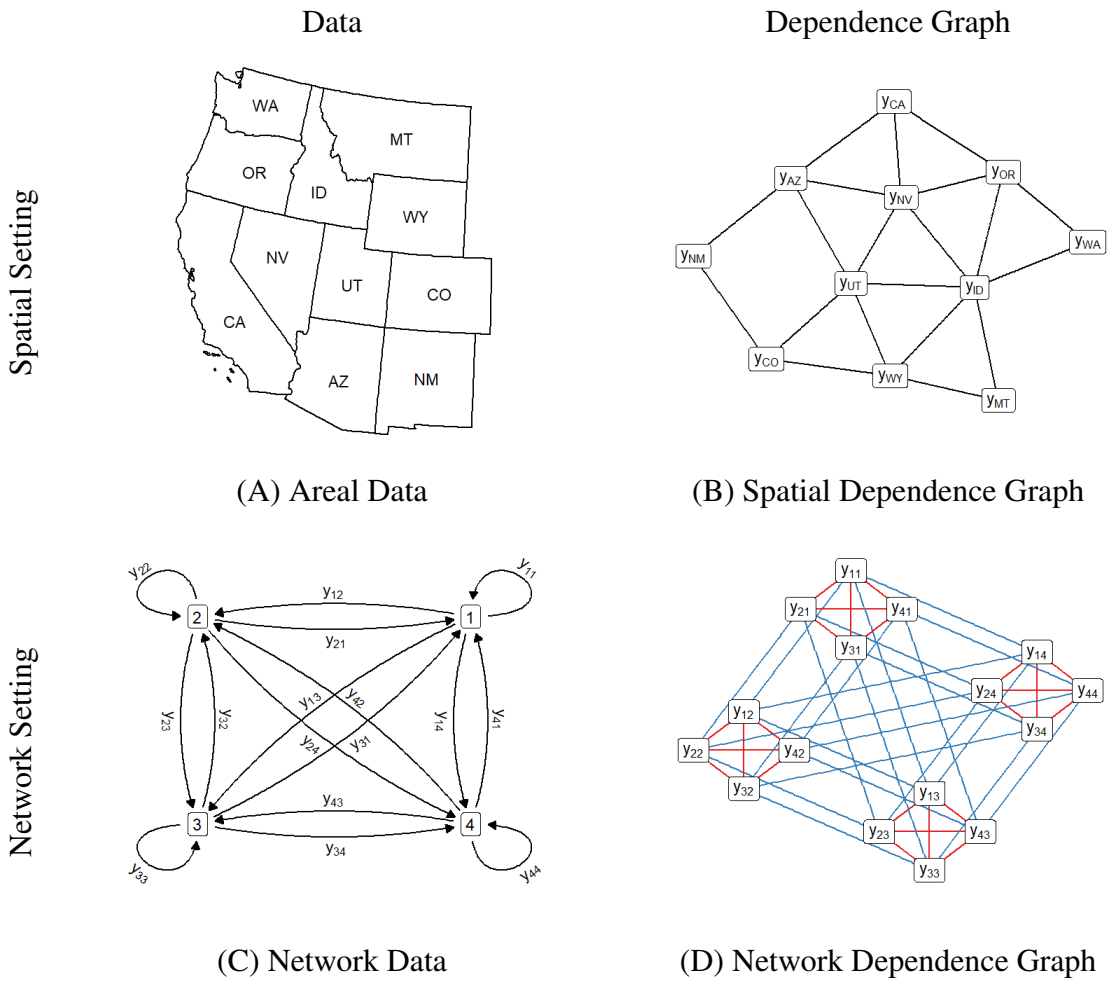


Figure 2.1: Comparing spatial dependence to network dependence. (A): A map of 11 western states. (B): The dependence graph corresponding to observations on each state in (A), where observations on states sharing a border are dependent. (C): A directed network with 4 nodes numbered 1 through 4, and observations measured on each directed edge. (D): The network dependence graph corresponding to observations in (C) . Each edge in the network is now a node in the dependence graph. There are two types of dependence - observations with a common sender (blue) and a common receiver (red). Both blue and red edges form distinct fully-connected clusters of observations.

Spatial models in which regions can be neighboring in two distinct ways have been studied in Reich et al. (2007) for the case of periodontal health measurements. Measurements were considered to be neighboring other measurements on the same tooth, or measurements on an adjacent tooth, but these kinds of neighbor relations were considered to create different kinds of dependence between measurements. The dependence in the periodontal health measurements was modeled using the random effect prior

$$p(\boldsymbol{\eta}) \propto c(\tau_1, \tau_2) \exp\{-0.5\boldsymbol{\eta}^\top(\tau_1\boldsymbol{Q}_1 + \tau_2\boldsymbol{Q}_2)\boldsymbol{\eta}\}, \tag{2.12}$$

where $\boldsymbol{Q}_1$ and $\boldsymbol{Q}_2$ are Laplacians for each neighbor relation. However, the distribution in (2.12) cannot represent the same dependence as the sender and receiver random effects in (2.5) because the distribution of $\boldsymbol{\eta}$ is full rank, while the distribution of the vectorized $(a_i + b_j)$ is not full rank.

In this chapter we introduce Restricted Network Regression as a method that mitigates network confounding. We approach network regression in a Bayesian framework, estimating parameters using their posterior distributions given the data. We characterize the posterior mean and variance of regression parameters in Restricted Network Regression with continuous data and show through simulation that Restricted Network Regression mitigates network confounding for continuous and binary data. Specifically, we demonstrate that a Restricted Network Regression model results in smaller bias and posterior credible intervals that are more appropriately calibrated to capture the generative parameter values than corresponding network regression models without random effects or with non-restricted random effects.

The remainder of this chapter is organized as follows. Section 2.2 defines network confounding and requirements for methods to "alleviate" and "mitigate" network confounding. Section 2.3 introduces Restricted Network Regression, characterizes the collinearity of effects within the Restricted Network Regression model, and provides theorems about the posterior distribution of the regression parameters in the continuous Restricted Network Regression model. Section 2.4 describes the results of a simulation study involving both continuous and binary network regression. Section 2.5 is a case study of Eurovision Song Contest voting data showing the changes

in inference that occur when using a Restricted Network Regression approach, relative to models without random effects and non-restricted random effect models. Finally, Section 2.6 closes with a discussion.

2.2 Network Confounding

Network confounding is the collinearity between fixed effects and network-structured random effects in a network regression model. This collinearity creates difficulty in estimating regression parameters by introducing bias in estimates and increasing their posterior variance. In this section we explore network confounding in more detail and define conditions for a method to alleviate or mitigate network confounding.

Consider the network regression model,

$$y_{ij} = g(z_{ij}), \;\; 1 \leq i, j \leq n, \tag{2.13}$$

$$\boldsymbol{z} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{A}\boldsymbol{a} + \boldsymbol{B}\boldsymbol{b} + \boldsymbol{\varepsilon}, \tag{2.14}$$

$$a_i \overset{\text{i.i.d.}}{\sim} g_a(\boldsymbol{\theta_a}), \tag{2.15}$$

$$b_j \overset{\text{i.i.d.}}{\sim} g_b(\boldsymbol{\theta_b}), \tag{2.16}$$

$$\boldsymbol{\varepsilon} \sim N(\boldsymbol{0}, \sigma_e^2 \boldsymbol{I}), \tag{2.17}$$

where $\boldsymbol{z}$ is an n^2-vector of latent continuous responses, $\boldsymbol{X}$ is an $n^2 \times p$ fixed matrix of covariates, $\boldsymbol{\beta}$ is a p-vector of regression parameters, and $\boldsymbol{a}$ and $\boldsymbol{b}$ are n-vectors composed of sender and receiver random effects, respectively, for each node in the network. The matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are $n^2 \times n$ matrices of zeros and ones which broadcast the elements of $\boldsymbol{a}$ and $\boldsymbol{b}$ into the appropriate rows of $\boldsymbol{z}$ depending on the sender and receiver of each dyad. To match the vectorized form of $\boldsymbol{z}$, we write $\boldsymbol{y}$ as an n^2-vector of the observed network data.

Now consider partitioning $\boldsymbol{X} = [\boldsymbol{1}\ \boldsymbol{X}_s\ \boldsymbol{X}_r\ \boldsymbol{X}_d]$ into an intercept, an $n^2 \times p_s$ matrix of sender covariates $\boldsymbol{X}_s$, an $n^2 \times p_r$ matrix of receiver covariates $\boldsymbol{X}_r$, and an $n^2 \times p_d$ matrix of dyadic covariates $\boldsymbol{X}_d$ as done in Hoff (2021), where p_s, p_r, and p_d are the number of sender, receiver, and

dyadic covariates, respectively, and $p = 1+p_s+p_r+p_d$. Similarly, partition $\boldsymbol{\beta} = (\beta_0\ \boldsymbol{\beta}_s^\top\ \boldsymbol{\beta}_r^\top\ \boldsymbol{\beta}_d^\top)^\top$. The column space of the matrix $\boldsymbol{A}$ contains all n^2-vectors that have repeated values for dyads with a common sender. Similarly, the column space of the matrix $\boldsymbol{B}$ contains all n^2-vectors that have repeated values for dyads with a common receiver. Since the columns of $\boldsymbol{X}_s$ are sender covariates, every column of $\boldsymbol{X}_s$ is in the column space of $\boldsymbol{A}$. Therefore, we can write $\boldsymbol{X}_s = \boldsymbol{A}\boldsymbol{X}_s'$, where $\boldsymbol{X}_s'$ is an $n \times p_s$ matrix created by collapsing equivalent rows of $\boldsymbol{X}_s$. Since the columns of $\boldsymbol{X}_r$ are receiver covariates, every column of $\boldsymbol{X}_r$ is in the column space of $\boldsymbol{B}$, allowing us to write $\boldsymbol{X}_r = \boldsymbol{B}\boldsymbol{X}_r'$. The model in (2.14) can be written

$$\boldsymbol{z} = [\mathbf{1}\ \boldsymbol{X}_d](\beta_0\ \boldsymbol{\beta}_d^\top)^\top + \boldsymbol{X}_s\boldsymbol{\beta}_s + \boldsymbol{X}_r\boldsymbol{\beta}_r + \boldsymbol{A}\boldsymbol{a} + \boldsymbol{B}\boldsymbol{b} + \boldsymbol{\varepsilon}, \tag{2.18}$$

$$= [\mathbf{1}\ \boldsymbol{X}_d](\beta_0\ \boldsymbol{\beta}_d^\top)^\top + \boldsymbol{A}(\boldsymbol{X}_s'\boldsymbol{\beta}_s + \boldsymbol{a}) + \boldsymbol{B}(\boldsymbol{X}_r'\boldsymbol{\beta}_r + \boldsymbol{b}) + \boldsymbol{\varepsilon}. \tag{2.19}$$

From this, we can see that $\boldsymbol{\beta}_s$ and $\boldsymbol{a}$ are confounded in the sense that they occupy the same linear space in the response, i.e., $\mathcal{C}(\boldsymbol{X}_s) \subset \mathcal{C}(\boldsymbol{A})$, where $\mathcal{C}$ is the column space operator. Similarly, $\boldsymbol{\beta}_r$ and $\boldsymbol{b}$ are also confounded. By restricting the random effects to be orthogonal to the fixed effects, Restricted Network Regression (introduced formally in the next section) fixes this collinearity by removing the intersection of the column spaces of $\boldsymbol{X}$, $\boldsymbol{A}$, and $\boldsymbol{B}$.

We now distinguish between two types of methods: those that *alleviate* network confounding and those that *mitigate* network confounding. We adapt a definition from the spatial statistics literature to define what it means for a method to alleviate network confounding:

Definition 2.2.1. A network regression method modeling network data, $\boldsymbol{y}$, with unconditional regression parameters, $\boldsymbol{\delta}$, which results in posterior mean $\mathrm{E}[\boldsymbol{\delta}|\boldsymbol{y}]$ and marginal posterior variances $\mathrm{Var}(\delta_\ell|\boldsymbol{y}), \ell = 1, \ldots, p$ alleviates network confounding if the following conditions are met:

1. $\mathrm{E}[\boldsymbol{\delta}|\boldsymbol{y}] = \mathrm{E}[\boldsymbol{\delta}_{NN}|\boldsymbol{y}]$
2. $\mathrm{Var}(\delta_{NN,\ell}|\boldsymbol{y}) \leq \mathrm{Var}(\delta_\ell|\boldsymbol{y}) \leq \mathrm{Var}(\beta_{Network,\ell}|\boldsymbol{y})$ for $\ell = 1, \ldots, p$

Here $\delta_{NN,\ell}$ are the unconditional regression coefficients of the corresponding network model without network-structured random effects and $\beta_{Network,\ell}$ are the conditional regression coefficients from a model with non-restricted network-structured random effects.

Definition 2.2.1 is adapted from the definition of spatial confounding in Khan and Calder (2022). This definition of alleviating network confounding reflects the intuition that models with network confounding will result in excess uncertainty on regression parameter estimates, reflected in large posterior variances of those parameters and that a method that alleviates network confounding should have lower posterior variances than one exhibiting network confounding. At the same time, a model that alleviates network confounding should model the unconditional effects of the fixed covariates, and so is expected to have the same posterior means as the model without network-structured random effects.

Alleviation of network confounding has useful interpretation in terms of parameter uncertainty, however, it does not relate directly to the accuracy of estimates made using models with network confounding. Also, Definition 2.2.1 provides no way to compare two models if neither alleviates network confounding. For these reasons, we introduce an alternative notion of network confounding mitigation. We expect a method that mitigates network confounding relative to another approach to produce better estimates and uncertainty quantification of the unconditional regression effects $\boldsymbol{\delta}$. We give a more precise definition of this mitigation:

Definition 2.2.2. A network regression method modeling network data, $\boldsymbol{y}$, with unconditional regression parameters, $\boldsymbol{\delta}$, which results in posterior mean $\mathrm{E}[\boldsymbol{\delta}|\boldsymbol{y}]$, denoted $\boldsymbol{m}$, and marginal posterior $100c\%$ credible intervals $I_{c,\ell}$, $0 < c < 1$, for components δ_ℓ, mitigates network confounding relative to another method which produces $\boldsymbol{m}'$ and $I'_{c,\ell}$, if for true unconditional regression effects $\boldsymbol{\delta}^*$,

1. $|\mathrm{E}\left[m_\ell - \delta^*_\ell\right]| \leq |\mathrm{E}\left[m'_\ell - \delta^*_\ell\right]|$ for $\ell = 1, \ldots, p$,

2. $|\mathrm{P}\left(\delta^*_\ell \in I_{c,\ell}\right) - c| \leq \left|\mathrm{P}\left(\delta^*_\ell \in I'_{c,\ell}\right) - c\right|$ for $\ell = 1, \ldots, p$,

where the expectation in item 1 and probability in item 2 are taken over $\boldsymbol{y}$.

Definition 2.2.2 combines two useful model evaluations, bias and credible interval coverage, into a comparison which can be used to evaluate the relative improvement of one model over another in the presence of network confounding. Even if a model does not meet the requirements of Definition 2.2.1, it can be compared to other models using Definition 2.2.2 to assess its mitigation of network confounding.

2.3 Restricted Network Regression

For the network regression model given in (2.14)-(2.17), we propose the following Restricted Network Regression model:

$$\boldsymbol{z} = \boldsymbol{X}\boldsymbol{\delta} + (\boldsymbol{I} - \boldsymbol{P}_X)(\boldsymbol{A}\boldsymbol{a} + \boldsymbol{B}\boldsymbol{b}) + \boldsymbol{\varepsilon}, \tag{2.20}$$

$$a_i \overset{\text{i.i.d.}}{\sim} g_a(\boldsymbol{\theta_a}), \tag{2.21}$$

$$b_j \overset{\text{i.i.d.}}{\sim} g_b(\boldsymbol{\theta_b}), \tag{2.22}$$

$$\boldsymbol{\varepsilon} \sim \mathrm{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_{n^2}), \tag{2.23}$$

where $\boldsymbol{z}$, $\boldsymbol{X}$, $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{P}_X$ are as described earlier. This model is distinguished from the model in (2.14)-(2.17) by the application of the projection matrix $\boldsymbol{I} - \boldsymbol{P}_X$, projecting the random effects orthogonal to the column space of $\boldsymbol{X}$, and the interpretation of the regression effects, $\boldsymbol{\delta}$, as unconditional on the values of the random effects. With $\boldsymbol{\delta}$ related to the unconditional regression parameters, $\boldsymbol{\beta}$, by $\boldsymbol{\delta} = \boldsymbol{\beta} + (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top (\boldsymbol{A}\boldsymbol{a} + \boldsymbol{B}\boldsymbol{b})$, the regression equation (2.20) is equivalent to (2.14).

2.3.1 Properties of Continuous Restricted Network Regression Posterior Distributions

In this section, we provide relationships between the posterior means and variances of Restricted Network Regression and a model without random effects. With these relationships, we show that Restricted Network Regression with a continuous response does not satisfy the condi-

tions of Definition 2.2.1, and therefore does not alleviate network confounding. We also relate these theoretical results to recent results in the spatial statistics literature by Khan and Calder (2022), which showed a similar result in spatial regression.

To understand the behavior of the Restricted Network Regression model with respect to network confounding, we give an expression for the posterior mean and variance of $\boldsymbol{\delta}$ in the Restricted Network Regression model.

Theorem 2.3.1. *In a continuous Restricted Network Regression model with two additive normally distributed node-level random effects,*

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\delta} + (\boldsymbol{I} - \boldsymbol{P}_X)\boldsymbol{A}\boldsymbol{a} + (\boldsymbol{I} - \boldsymbol{P}_X)\boldsymbol{B}\boldsymbol{b} + \boldsymbol{\varepsilon}, \tag{2.24}$$

$$p(\boldsymbol{\delta}) \propto 1, \tag{2.25}$$

$$\boldsymbol{a} \sim N(\boldsymbol{0}, \sigma_a^2\boldsymbol{I}),\ \boldsymbol{b} \sim N(\boldsymbol{0}, \sigma_b^2\boldsymbol{I}), \boldsymbol{\varepsilon} \sim N(\boldsymbol{0}, \sigma_\varepsilon^2\boldsymbol{I}), \tag{2.26}$$

the posterior distribution of $\boldsymbol{\delta}$ will have mean and variance

$$\mathrm{E}[\boldsymbol{\delta}|\boldsymbol{y}] = (\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top\boldsymbol{y}, \tag{2.27}$$

$$\mathrm{Var}(\boldsymbol{\delta}|\boldsymbol{y}) = (\boldsymbol{X}^\top\boldsymbol{X})^{-1}\mathrm{E}[\sigma_\varepsilon^2|\boldsymbol{y}]. \tag{2.28}$$

The proof of this theorem is provided in Appendix A.1. The prior distribution of σ_ε^2 is not specified for this theorem, but will partially determine $\mathrm{E}[\sigma_\varepsilon^2|\boldsymbol{y}]$. The posterior mean in (2.27) is equal to the posterior mean from a model without network-structured random effects, as required by Definition 2.2.1. Therefore Restricted Network Regression alleviates network confounding if and only if the inequality on the posterior variances in Definition 2.2.1 is true. We show this property of the posterior variances using a more general model with two restricted random effects.

Theorem 2.3.2. *Consider the restricted regression model with two random effects,*

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\delta} + \boldsymbol{W}_1\boldsymbol{\eta}_1 + \boldsymbol{W}_2\boldsymbol{\eta_2} + \boldsymbol{\epsilon}, \tag{2.29}$$

$$p(\boldsymbol{\delta}) \propto 1, \tag{2.30}$$

$$p(\boldsymbol{\eta}_1|\tau_1) \sim \tau_1^{\text{rank}(\boldsymbol{F}_1)/2} \exp\left\{-\frac{\tau_1}{2}\boldsymbol{\eta}_1^\top \boldsymbol{F}_1\boldsymbol{\eta}_1\right\}, \tag{2.31}$$

$$p(\boldsymbol{\eta}_2|\tau_2) \sim \tau_2^{\text{rank}(\boldsymbol{F}_2)/2} \exp\left\{-\frac{\tau_2}{2}\boldsymbol{\eta}_2^\top \boldsymbol{F}_2\boldsymbol{\eta}_2\right\}, \tag{2.32}$$

$$\boldsymbol{\epsilon} \sim N(\boldsymbol{0}, \boldsymbol{I}/\tau_\epsilon). \tag{2.33}$$

If $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ have orthonormal columns such that $\mathcal{C}(\boldsymbol{X})$, $\mathcal{C}(\boldsymbol{W}_1)$ and $\mathcal{C}(\boldsymbol{W}_2)$ are pairwise orthogonal, $\boldsymbol{F}_1$ and $\boldsymbol{F}_2$ are positive definite symmetric matrices, $\tau_j \sim$ gamma(a_j, b_j) for $j = 1, 2$, $\tau_\epsilon \sim gamma(a_\epsilon, b_\epsilon)$ and

$$\frac{\text{E}[r_1|\boldsymbol{y}]/b_1 + \text{E}[r_2|\boldsymbol{y}]/b_2}{\text{E}[\tau_1]/b_1 + \text{E}[\tau_2]/b_2} \leq E[\sigma^2_{\epsilon,NN}|\boldsymbol{y}], \tag{2.34}$$

where $r_j = \tau_j/\tau_\epsilon$ for $j = 1, 2$, then $\text{Var}(\delta_\ell|\boldsymbol{y}) \leq \text{Var}(\delta_{NN,\ell}|\boldsymbol{y})$ for $\ell = 1, \ldots, p$.

The proof of this theorem is provided in Appendix A.1. This theorem shows that under the specified conditions, a model with two random effects restricted to be orthogonal to the covariates yields posterior variances that do not meet the conditions in Definition 2.2.1. Applying Theorem 2.3.2 to Restricted Network Regression with $\boldsymbol{W}_1 = (\boldsymbol{I} - \boldsymbol{P}_X)\boldsymbol{A}$, $\boldsymbol{W}_2 = (\boldsymbol{I} - \boldsymbol{P}_X)\boldsymbol{B}$, and $\boldsymbol{F}_1 = \boldsymbol{F}_2 = \boldsymbol{I}_n$ shows that Restricted Network Regression with a continuous response does not meet the conditions in Definition 2.2.1.

Khan and Calder (2022) prove similar theorems for restricted spatial regression models of the form,

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\delta} + \boldsymbol{W}\boldsymbol{\eta} + \boldsymbol{\varepsilon}, \tag{2.35}$$

$$p(\boldsymbol{\eta}|\tau_s) \propto \tau_s^{\text{rank}(\boldsymbol{F})/2} \exp\left\{-\frac{\tau_s}{2}\boldsymbol{\eta}^\top \boldsymbol{F}\boldsymbol{\eta}\right\} \tag{2.36}$$

$$\boldsymbol{\varepsilon} \sim N(\boldsymbol{0}, \boldsymbol{I}/\tau_\varepsilon), \tag{2.37}$$

observing that that the model form in (2.35)-(2.37) encompasses the ICAR model, the non-spatial model, and restricted spatial regression models from Reich et al. (2006), Hughes and Haran (2013), and Prates et al. (2019). The model in (2.35)-(2.37) is "restricted" if $\mathcal{C}(\boldsymbol{X})$ and $\mathcal{C}(\boldsymbol{W})$ are orthogonal. In fact, this form also encompasses continuous network models that include one additive sender or one additive receiver random effect, but not both, motivating the need for Theorem 2.3.2.

2.4 Simulation Study

In this section we investigate properties of Restricted Network Regression through simulation. We confirm the theoretical results from Section 2.3.1 using continuous network data and show that neither continuous nor binary Restricted Network Regression alleviate network confounding. However, we show that both mitigate network confounding relative to non-restricted network regression and network regression with no random effects. All of the following simulations involve data with varying levels of excess nodal variation, using models with both sender/receiver covariates and sender/receiver random effects.

In addition to evaluating Restricted Network Regression, we chose to evaluate choices of prior distribution on σ_a^2 and σ_b^2. A common choice is the inverse-gamma distribution, e.g., in the `amen` package (Hoff et al., 2020). However, the half-Cauchy distribution, $\sigma_a \sim \text{Cauchy}^+(0, 1)$, is a less informative distribution recommended by Gelman (2006) for random effect variances in hierarchical models. We compare five models:

(NoRE) Network model with no random effects,

(NR.ig) Network model with additive random effects and inverse-gamma priors,

(NR.hc) Network model with additive random effects and half-Cauchy priors,

(RNR.ig) Network model with restricted additive random effects and inverse-gamma priors,

(RNR.hc) Network model with restricted additive random effects and half-Cauchy priors.

We show that Restricted Network Regression mitigates network confounding by providing estimates of $\boldsymbol{\delta}$ with lower bias and properly calibrated credible intervals.

2.4.1 Simulation 1: Continuous Network Data

Continuous network data were simulated from the network model in (2.14)-(2.17), using the identity function, $g(z_{ij}) = z_{ij}$. The design matrix $\boldsymbol{X}$ contained an intercept, one sender covariate, one receiver covariate, and one dyadic covariate whose values were drawn independently from a standard normal distribution. Unobserved excess nodal variation (simulated values of a_i and b_j, denoted $\boldsymbol{a}^*$ and $\boldsymbol{b}^*$) was then simulated from one of seven possible scenarios, which varied in the magnitude of the nodal variation and the degree of collinearity between the nodal variation and observed covariates. We control and quantify this latter degree of collinearity using the canonical correlation (ccor) between $\boldsymbol{A}\boldsymbol{a}^* + \boldsymbol{B}\boldsymbol{b}^*$ and $\boldsymbol{X}$.

(G1) No excess variation: $\boldsymbol{a}^* = \boldsymbol{b}^* = \boldsymbol{0}$,

(G2) Small magnitude, no correlation:

$\boldsymbol{a}^* \sim \text{Normal}(0, 0.25)$, $\boldsymbol{b}^* \sim \text{Normal}(0, 0.25)$, $\text{ccor}(\boldsymbol{A}\boldsymbol{a}^* + \boldsymbol{B}\boldsymbol{b}^*, \boldsymbol{X}) = 0$,

(G3) Small magnitude, slight correlation:

$\boldsymbol{a}^* \sim \text{Normal}(0, 0.25)$, $\boldsymbol{b}^* \sim \text{Normal}(0, 0.25)$, $\text{ccor}(\boldsymbol{A}\boldsymbol{a}^* + \boldsymbol{B}\boldsymbol{b}^*, \boldsymbol{X}) = 0.1$,

(G4) Small magnitude, strong correlation:

$\boldsymbol{a}^* \sim \text{Normal}(0, 0.25)$, $\boldsymbol{b}^* \sim \text{Normal}(0, 0.25)$, $\text{ccor}(\boldsymbol{A}\boldsymbol{a}^* + \boldsymbol{B}\boldsymbol{b}^*, \boldsymbol{X}) = 0.9$,

(G5) Large magnitude, no correlation:

$\boldsymbol{a}^* \sim \text{Normal}(0, 1)$, $\boldsymbol{b}^* \sim \text{Normal}(0, 1)$, $\text{ccor}(\boldsymbol{A}\boldsymbol{a}^* + \boldsymbol{B}\boldsymbol{b}^*, \boldsymbol{X}) = 0$,

(G6) Large magnitude, slight correlation:

$\boldsymbol{a}^* \sim \text{Normal}(0, 1)$, $\boldsymbol{b}^* \sim \text{Normal}(0, 1)$, $\text{ccor}(\boldsymbol{A}\boldsymbol{a}^* + \boldsymbol{B}\boldsymbol{b}^*, \boldsymbol{X}) = 0.1$,

(G7) Large magnitude, strong correlation:

$\boldsymbol{a}^* \sim \text{Normal}(0, 1)$, $\boldsymbol{b}^* \sim \text{Normal}(0, 1)$, $\text{ccor}(\boldsymbol{A}\boldsymbol{a}^* + \boldsymbol{B}\boldsymbol{b}^*, \boldsymbol{X}) = 0.9$.

In all scenarios, the components of $\boldsymbol{a}^*$ and $\boldsymbol{b}^*$ were first generated i.i.d., then the vectors were projected to have the desired canonical correlation according to the algorithm in Appendix A.2. Scenarios G2, G3, and G4 were chosen to represent nodal variation smaller than the effect of one covariate, while scenarios G5, G6, and G7 were chosen to represent variation comparable to the effect of one covariate. The slight correlation scenarios (G3 and G6) represent situations with subjectively low correlation between nodal variation and covariates, while the strong correlation scenarios (G4 and G7) represent situations with subjectively high correlation between nodal variation and covariates. For each of the seven scenarios, 100 values of $\boldsymbol{X}$, $\boldsymbol{a}^*$, and $\boldsymbol{b}^*$ were generated. Then for each set of covariates and random effects, 200 values of the random error $\boldsymbol{\epsilon}$ were generated each from a standard normal distribution ($\sigma^2_\epsilon = 1$), resulting in 20,000 simulated data sets for each scenario.

We compare the posterior means and posterior variances of the unconditional regression parameters in NoRE and RNR.ig. Figure 2.2 shows these values for the receiver covariate, δ_r. We see that the posterior means are equal for both models on all simulated data sets, and the posterior variance in the Restricted Network Regression model (RNR.ig) is less than or equal to the posterior variance in the model with no random effects (NoRE). Similar results were observed for the sender covariate. The equality of posterior means and this observed inequality of posterior variances validate the result of Theorem 2.3.2 empirically, and show that continuous Restricted Network Regression does not alleviate network confounding according to Definition 2.2.1.

To investigate the ability of Restricted Network Regression to mitigate network confounding relative to a model with no random effects and models with non-restricted random effects, we compare bias and coverage of posterior credible intervals for across all models (NoRE through RNR.hc). For each of the 200 trials with each of the 100 values of $\boldsymbol{X}$, $\boldsymbol{a}^*$, and $\boldsymbol{b}^*$, we recorded the difference between the posterior means of $\boldsymbol{\delta}$ and the value of $\boldsymbol{\delta}$ used to generate the data. We also recorded whether the 90% credible for $\boldsymbol{\delta}$ captures $\boldsymbol{\delta}^*$. Finally we calculated the average bias across the 200 values of $\boldsymbol{y}$ generated for each of the 100 values of $\boldsymbol{X}$, and the proportion of the 200 trials for which $\boldsymbol{\delta}^*$ was captured by the credible interval.

(A)

(B)

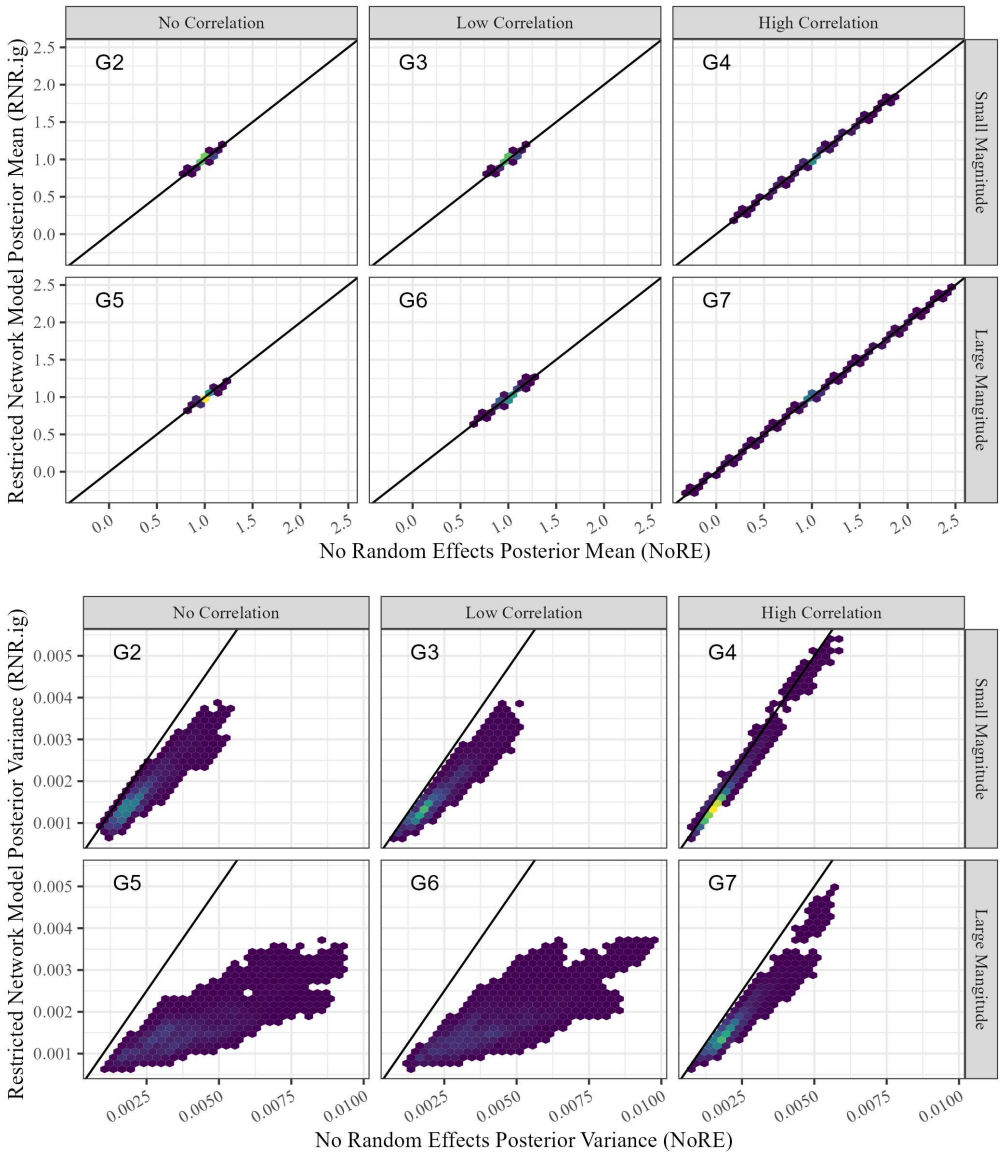


Figure 2.2: Comparison of posterior means (A) and variances (B) for a receiver covariate using a network regression model with no random effects (NoRE) and a Restricted Network Regression model (RNR.ig) with a continuous response. Each panel is one of scenarios G2 through G7. Each panel shows a heatmap of the 100×200 simulated data sets for each scenario, and the $y = x$ line is drawn on each panel, which represents an equal value from both models. The area below the line contains smaller values in the Restricted Network Regression model, and the area above the line contains larger values in the Restricted Network Regression model.

(A)

(B)

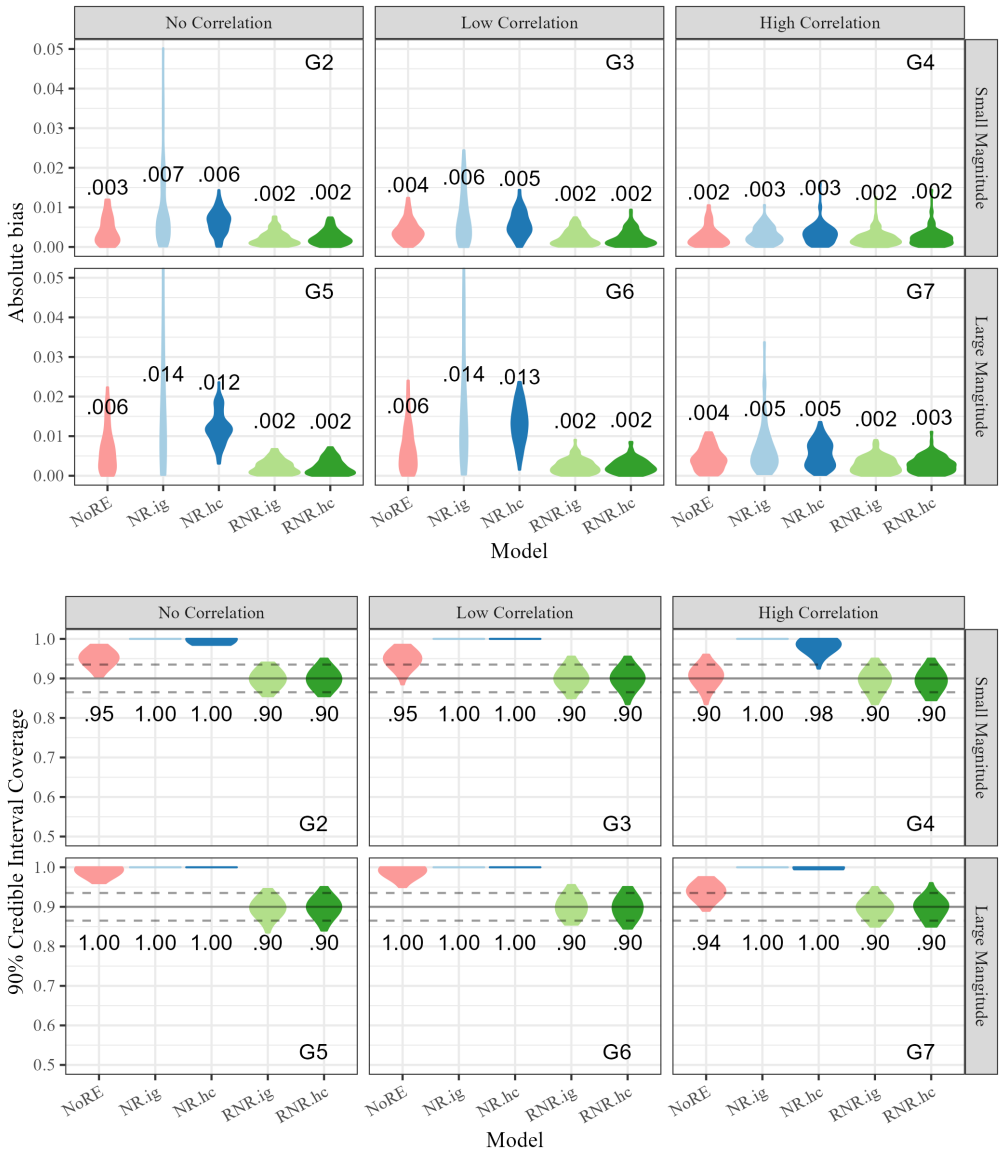


Figure 2.3: Absolute bias (A) and credible interval coverage (B) for all models when estimating δ_r with continuous data. Median bias and coverage values are printed for each model. Each panel is one of scenarios G2 through G7. Violin plots show the distribution of the estimated absolute bias and coverage values of the 100 simulated values of X, $\boldsymbol{a}^*$, and $\boldsymbol{b}^*$. Solid horizontal lines indicate the nominal coverage (90%) and dashed horizontal lines at 86.5% and 93.5% indicate bounds within which the average coverage of a 90% credible interval should fall over 200 trials.

Figure 2.3 shows the distribution of the absolute bias and credible interval coverage for δ_r using each model in each scenario, G2 through G7. Bias appears lower for the restricted models (RNR.ig and RNR.hc) than for other models in scenarios G2, G3, G5, G6, and G7, and approximately equal in G4. The most noticeable difference between the Restricted Network Regression models and others is in credible interval coverage, where models RNR.ig and RNR.hc appear properly calibrated and NoRE, NR.ig, and NR.hc have coverage that is too high. Together, these results suggest the continuous Restricted Network Regression mitigates network confounding relative to both non-restricted network regression and network regression with no random effects.

2.4.2 Simulation 2: Binary Network Data

Data for the binary network models were simulated in the same way as for the continuous network model (Section 2.4.1), using scenarios G1 through G7, but with the indicator function $g(z_{ij}) = I(z_{ij} > 0)$ to convert latent continuous responses to binary responses. All models were fit to each set of simulated data. We similarly assessed the results of these simulations by comparing posterior means and posterior variances of the unconditional regression parameters in models NoRE and RNR.ig.

Figure 2.4 shows the comparison between posterior means and variances of δ_r for models NoRE and RNR.ig with binary data. This comparison is notably different from the comparison for continuous data in Figure 2.2. First, the posterior means produced by each model are not equal. Second, the posterior variance of the regression coefficient in the Restricted Network Regression model is now greater than in the network model with no random effects. This demonstrates empirically that the implications of Theorem 2.3.1 and Theorem 2.3.2 do not apply to models with non-Gaussian responses due to the inequality of posterior means. However, this inequality of means also demonstrates that probit Restricted Network Regression also does not alleviate network confounding.

Figure 2.5 shows the absolute bias and coverage estimates for δ_r in scenarios G2 through G7 using all models. The bias for model NoRE is the highest in general, with all other models having

(A)

(B)

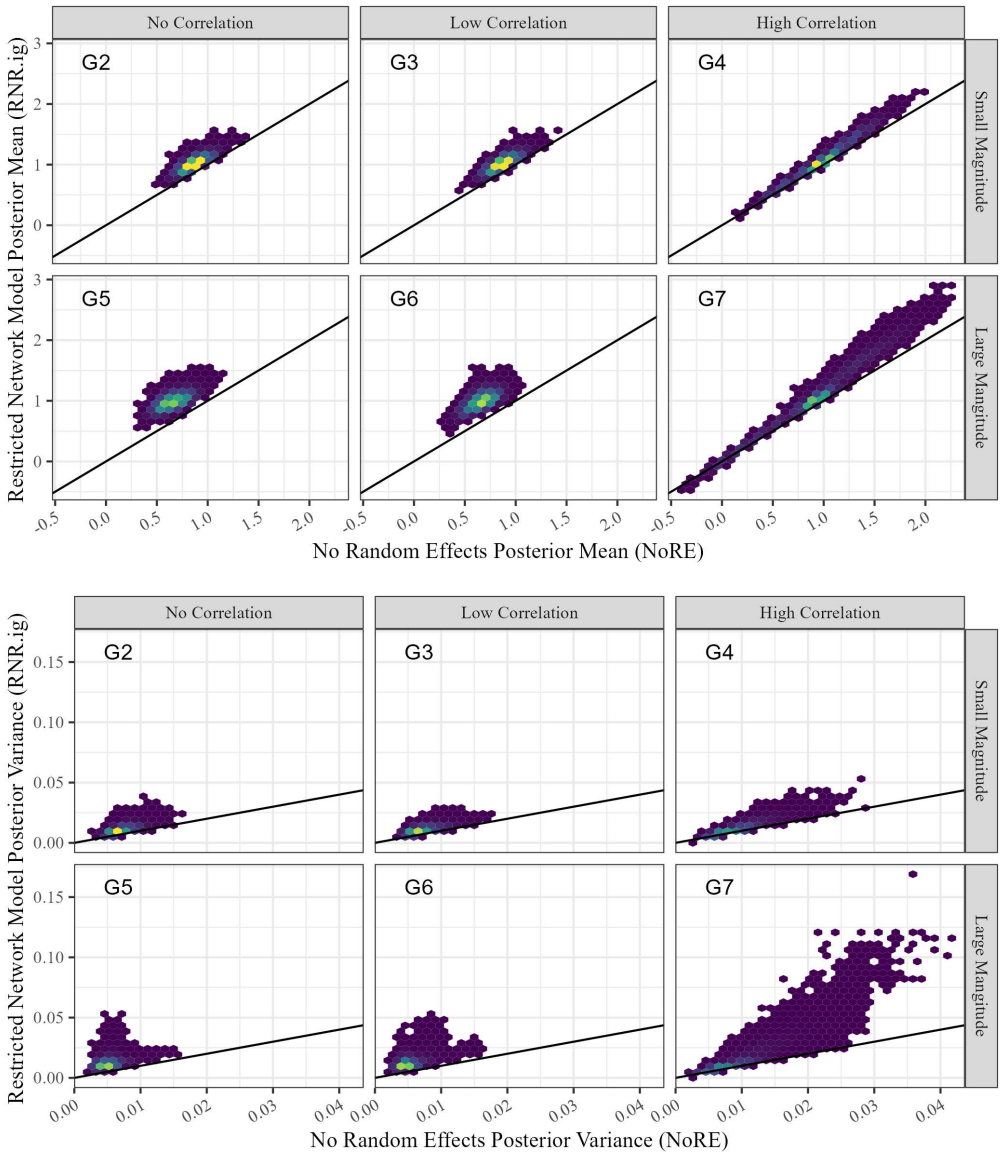


Figure 2.4: Comparison of posterior means (A) and variances (B) for a receiver covariate using a network regression model with no random effects (NoRE) and a Restricted Network Regression model (RNR.ig) with a binary response. Each panel is one of scenarios G2 through G7. Each panel shows a heatmap of the 100×200 simulated data sets for each scenario, and the $y = x$ line is drawn on each panel, which represents an equal value from both models. The area below the line contains smaller values in the Restricted Network Regression model, and the area above the line contains larger values in the Restricted Network Regression model.

(A)

(B)

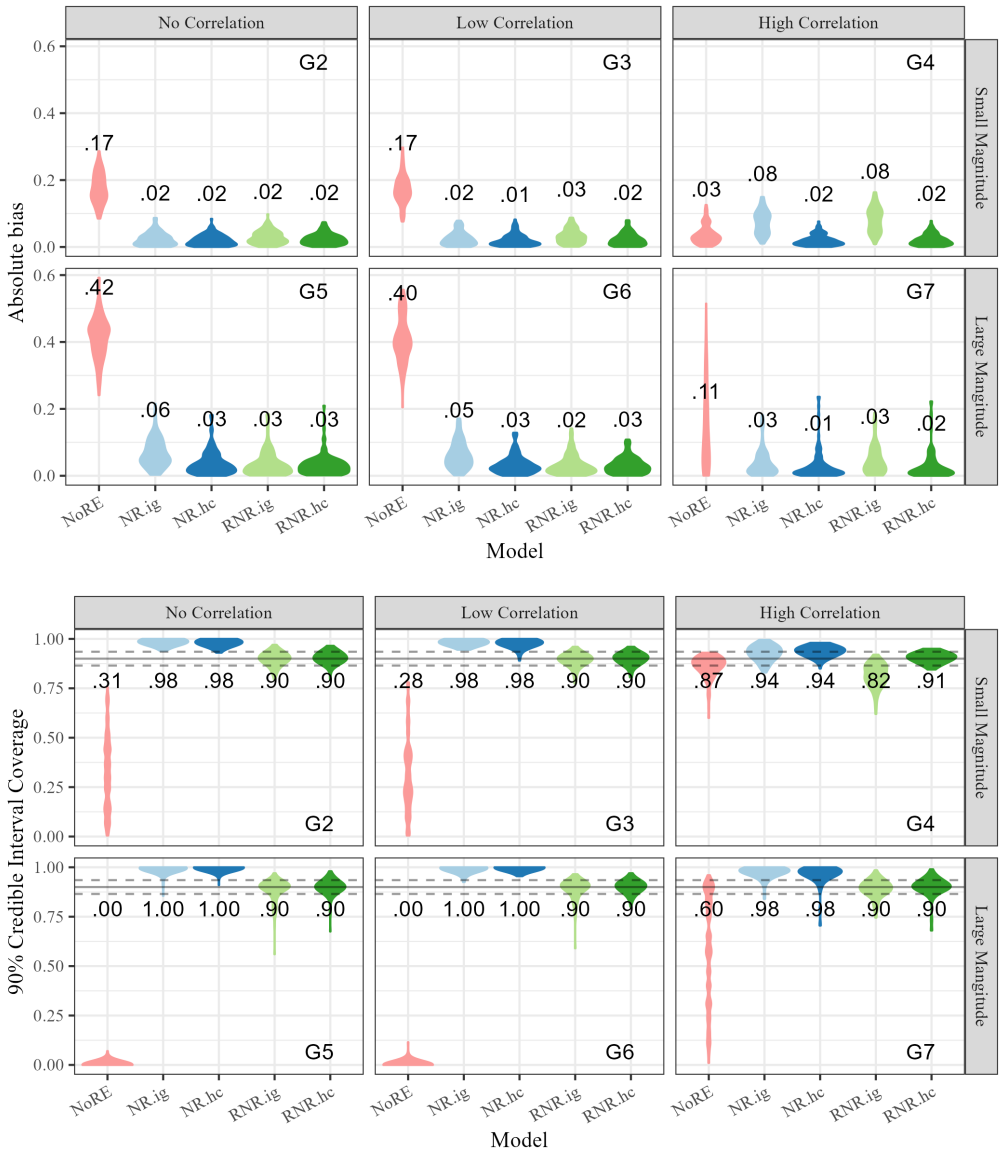


Figure 2.5: Absolute bias (A) and credible interval coverage (B) for all models when estimating δ_r with binary data. Median absolute bias and coverage values are printed for each model. Each panel is one of scenarios (G2) - (G7). Violin plots show the distribution of the estimated absolute bias and coverage values of the 100 simulated values of $\boldsymbol{X}$, $\boldsymbol{a}^*$, and $\boldsymbol{b}^*$. Solid horizontal lines indicate the nominal coverage (90%) and dashed horizontal lines at 86.5% and 93.5% indicate bounds within which the average coverage of a 90% credible interval should fall over 200 trials.

approximately equal absolute bias except in scenario G4. In most scenarios, the coverage of model NoRE is significantly lower than 90%, while the coverage of the Restricted Network Regression model with both inverse-gamma (RNR.ig) and half-Cauchy (RNR.hc) priors is much closer to 90%. Again, the non-restricted models (NR.ig and NR.hc) have coverage that is higher than 90% in all scenarios. The Restricted Network Regression model with half-Cauchy random effect priors (RNR.hc) has coverage within the expected range in all scenarios. A notable exception is scenario G4, in which the models with inverse-gamma priors on σ_a and σ_b (NR.ig and RNR.ig), have higher bias than their half-Cauchy counterparts. The coverage for model RNR.ig is also noticeably lower than RNR.hc. Here, the prior selection affects the model's ability to mitigate network confounding. While model RNR.hc mitigates network confounding relative to NR.hc and RNR.ig mitigates network confounding relative to NR.ig, RNR.ig does not mitigate network confounding relative to NR.hc due to higher bias in this scenario.

2.5 Eurovision Voting Network Analysis

The Eurovision Song Contest is an annual competition in which European countries compete by submitting the best song by an artist from their country. The contest culminates in a final round, where the remaining 26 competitors perform their songs for a TV audience. All participating countries then vote for their top ten songs through judges and/or phone-in voting. Points are awarded according to votes (12 points for first, 10 points for second, then 8 through 1 points for third through tenth) and the total determines the winner.

The contest is extremely popular, drawing 182 million viewers in 2019 (Eurovision, 2019). This popularity has meant that the contest has been of interest for study, especially the study of voting patterns (e.g., Ginsburgh and Noury, 2008; Spierdijk and Vellekoop, 2009). Because competing countries are a subset of voting countries, vote data can naturally be represented as a directed graph with the countries as nodes and an edge from country i to country j representing a top-10 vote by country i for country j. Edges may be labeled with ranked votes if desired.

Eurovision votes have been analyzed as a network by, for example, Yair (1995), Fenn et al. (2006), and D'Angelo et al. (2019).

Countries may have other measurable qualities that are related to how well they score in the contest results. For example, countries with larger populations have more musicians from which to select a contestant. A country's wealth may be associated with the reach of its cultural exports, leading to more votes received from its trading partners. The Eurovision Song Contest is also the subject of bets predicting its winner. Betting markets reflect the collective knowledge of their participants, which in this case includes knowledge of the specific songs entered by each country, and it is reasonable to think they may be predictive of the outcome. For example, Spann and Skiera (2009) finds betting odds to be predictive of match results in the German premier soccer league. Analyzing the relationship between population, wealth, and betting odds, and the Eurovision contest voting outcome fits a network regression framework naturally.

We restrict our analysis to the year 2015 and the 27 countries with entries in the final round of that year (Australia, as a new contestant, was given an automatic berth to the final round). The response data consist of a vote network represented as a 27×27 matrix (Figure 2.6). The receiver covariate data consist of three dimension 27 vectors of song or country attributes: the log median odds from 16 popular European betting sites for each song to win the contest, the log 2015 population of each competing country, and the log 2015 GDP per capita of each competing country. We log-transform the receiver covariates before using them as covariates in a regression model because they are either right-skewed (GDP, population) or because we believe there to be a logarithmic relationship between the predictor and the response (betting odds). We also include a dyadic covariate for country contiguity which was found to be explanatory in D'Angelo et al. (2019). Country contiguity is an undirected network represented as a symmetric 27×27 binary matrix where a 1 indicates that two countries share a border and a 0 indicates otherwise. Visualizations of these covariates are available in Appendix A.3. Votes were represented as ranks (1-10 with 10 the highest). This data is freely available and easily compiled by hand (Eurovision, 2015; Eurovisionworld, 2015; Conte et al., 2022; United Nations, 2015; World Bank, 2022). Any

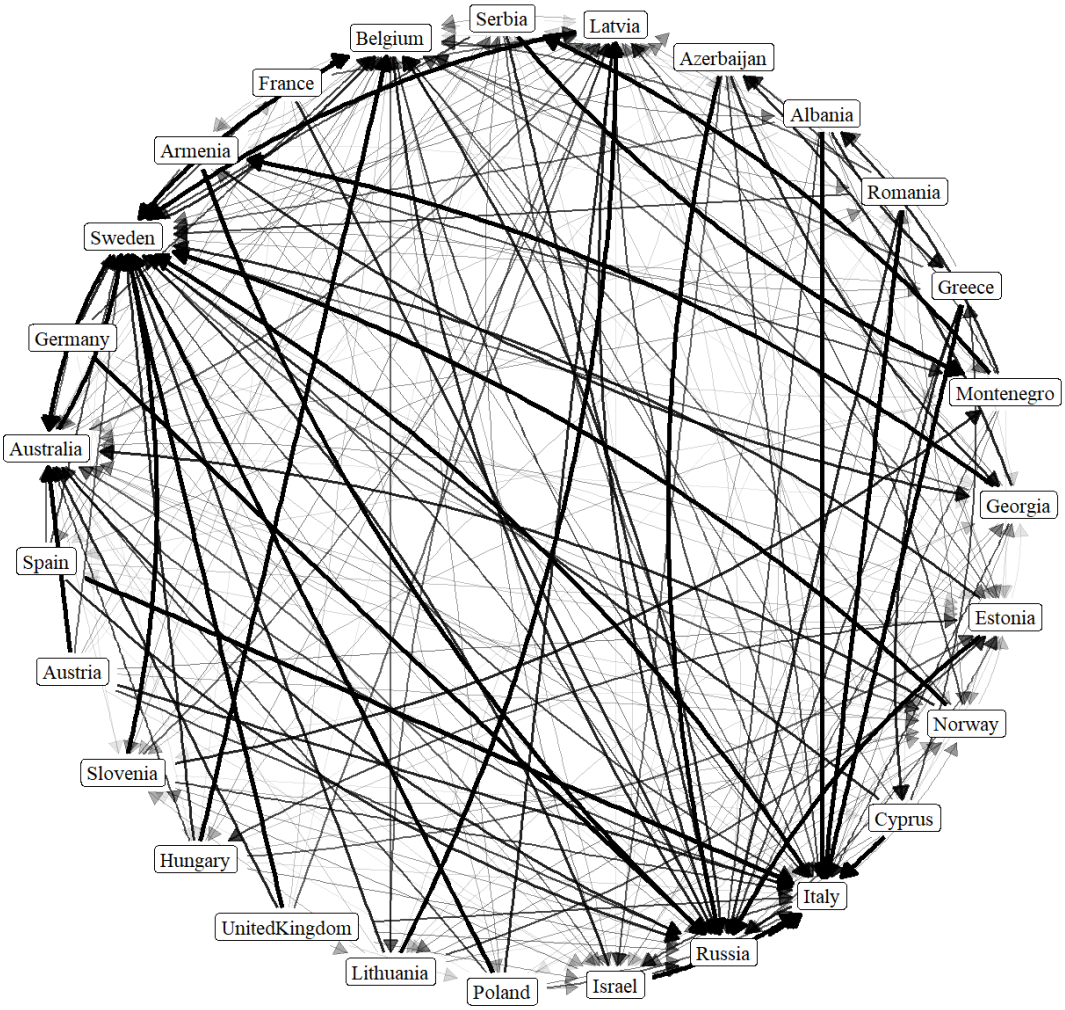


Figure 2.6: Illustration of vote data from the 2015 Eurovision Song Contest final round as a network. Edges point from voter to song, with darker lines indicating higher rankings. The contest's eventual winner, Sweden, receives a majority of the high-ranked votes.

pairs i, j where country i did not vote for country j in its top 10 were coded as zero. Our focus on a single year is due to the fact that both betting odds and column random effects are song-specific, and therefore year-specific. Therefore additional years in this analysis cannot be treated like replicates in the style of D'Angelo et al. (2019). We examine the effects of the covariates on Eurovision voting by performing a network regression analysis without random effects, with receiver random effects, and with restricted receiver random effects. We investigate the effect of Restricted Network Regressions on regression parameter estimates and interpretation compared to either alternative model.

2.5.1 Network Model with No Random Effects

The base model with no network-structured random effects has the form,

$$\boldsymbol{y} = g(\boldsymbol{z}), \tag{2.38}$$

$$z_{ij} = \delta_{CC} x_{ij}^1 + \delta_{Odds} x_j^2 + \delta_{Pop} x_j^3 + \delta_{GDP} x_j^4 + \varepsilon_{ij}, \tag{2.39}$$

$$\varepsilon_{ij} \overset{\text{i.i.d.}}{\sim} N(0, 1). \tag{2.40}$$

We used the relative rank likelihood (RRL; Pettitt, 1982; Hoff et al., 2013), implemented with a function $g(\cdot)$ which maps continuous values z_{ij} to the observed ranks y_{ij} in the following way: for any voting country i and two entered songs j and j', $y_{ij} > y_{ij'}$ implies $z_{ij} > z_{ij'}$. This imposes no relationship between the responses for different voting countries, so we cannot infer row effects (Hoff et al., 2013).

We analyze and interpret the regression parameter estimates through posterior means and 90% credible intervals for $\boldsymbol{\delta}$ (Figure 2.7). As these are estimates of $\boldsymbol{\delta}$ and there are no random effects in the model, we interpret them as the unconditional effect of the fixed effects on the response. All fixed covariates–country contiguity, log betting odds, log GDP per capita, and log population–appear to have an effect on the voting outcomes. Country contiguity has a large positive effect, agreeing with D'Angelo et al. (2019) that countries are more likely to vote for their neighbors'

Figure 2.7: Comparison of posterior means and 90% credible intervals for β (Non-Restricted Network Model) or δ (No Random Effects and Restricted Network Model). As betting odds, population, and GDP are receiver covariates confounded with the random effects, inference on these regression parameters changes the most between the three models. The posterior distribution for country contiguity, a dyadic covariate, is less affected by the choice of model.

songs. Log betting odds have a large negative effect, which shows that betting markets are predictive of the Eurovision outcome as larger odds are associated with lower predicted probability of winning. Log GDP per capita has a small positive effect, indicating that wealthier countries are more likely to receive votes than less wealthy countries. Log population has a small negative effect. All other things being equal, more populous countries are less likely to receive votes than less populous countries.

2.5.2 Non-Restricted Network Model

We fit a network model with a receiver random effect (b_j) to account for song heterogeneity not explained by the betting odds, population, or GDP. This network model has the form,

$$\boldsymbol{y} = g(\boldsymbol{z}), \tag{2.41}$$

$$z_{ij} = \beta_{CC}x_{ij}^1 + \beta_{Odds}x_j^2 + \beta_{Pop}x_j^3 + \beta_{GDP}x_j^4 + b_j + \varepsilon_{ij}, \tag{2.42}$$

$$b_j \overset{\text{i.i.d.}}{\sim} N(0, \sigma_b^2), \tag{2.43}$$

$$\varepsilon_{ij} \overset{\text{i.i.d.}}{\sim} N(0, 1). \tag{2.44}$$

Leaving the random effects non-restricted is appropriate depending on the intended interpretation of the terms in the model. For example, restricted regression is not recommended in the case of "Scheffé-style" random effects: random effects whose values are considered as draws from a population which is of interest, even though the values of the effects themselves are not (Hodges and Reich, 2010). If we consider the population of countries to be all those eligible to participate in the contest, or all those who competed in the initial rounds, then the selection of countries in the final round is only a subset of the population. If the primary interest is studying the population of all eligible countries rather than the propensity of individual countries to receive votes, the receiver random effects could be considered "Scheffé-style" and restricted regression may not be an appropriate choice.

Because the model contains receiver random effects, the regression effects represent the effect of the covariates on the response conditioned on $\boldsymbol{b}$. The posterior means and credible intervals of the regression parameters in this model are noticeably different than in the network model without random effects (Figure 2.7). In this case, the country contiguity retains its large positive effect and log betting odds retains its large negative effect. We notice that the width of the 90% posterior credible intervals for β_{Odds}, β_{GDP}, and β_{Pop} are wider than the credible intervals for δ_{Odds}, δ_{GDP}, and δ_{Pop} from the model in (2.39), indicating greater uncertainty about the conditional effect of these covariates than the unconditional effect. The estimate of β_{Pop} is also smaller in

Table 2.1: Comparison of covariate effect estimates from Restricted Network Regression to other models. For each covariate and each comparison model, the ratio of posterior means and posterior credible interval widths are shown. Numbers smaller than 1 indicate smaller posterior means or narrower credible intervals in the Restricted Network Regression model.

	Comparison Model			
	No Random Effects		Non-Restricted Network Model	
Covariate	Mean Ratio	Width Ratio	Mean Ratio	Width Ratio
Log Betting Odds	1.004	1.045	1.008	0.539
Log Population	0.389	1.247	1.000	0.598
Log GDP per Capita	1.656	1.173	1.004	0.566
Country Contiguity	1.091	1.031	1.032	0.997

magnitude and has a credible interval which includes zero. The changes in credible interval width and posterior mean illustrate the impact of network confounding.

2.5.3 Restricted Network Model

We fit a network model with a receiver random effect ($\boldsymbol{b}$) to account for song heterogeneity not explained by the column covariates, projected to be orthogonal to the fixed effects of betting odds, population, or GDP. Specifically, we set

$$\boldsymbol{y} = g(\boldsymbol{z}), \tag{2.45}$$

$$\boldsymbol{z} = X\boldsymbol{\delta} + (\boldsymbol{I} - \boldsymbol{P}_X)\boldsymbol{B}\boldsymbol{b} + \boldsymbol{\varepsilon}. \tag{2.46}$$

Since the association between $\boldsymbol{y}$ and $\boldsymbol{X}$ is of primary interest, we would like estimates of $\boldsymbol{\delta}$ instead of $\boldsymbol{\beta}$. If we do not want to infer about the population of countries which did not compete in this year's final round, then the random effects in the model constitute the entire population of interest. Since in this analysis we are using data only from the final round of the 2015 contest, restricted regression would be appropriate. In the Restricted Network Regression model, the regression effects once again represent the unconditional effect of the covariates on the response as the collinearity with the random effect has been removed.

Table 2.1 compares the magnitude of the regression parameter estimates and their credible interval widths from the Restricted Network Regression model to those in the other models. Compared to the model with no random effects, the receiver effects for log population and log GDP per capita exhibit noticeably different posterior means and larger posterior credible intervals, while the effect for log betting odds shows approximately equal posterior mean and only slightly larger credible interval (Figure 2.7). These results mirror what was observed with the binary data in the earlier simulation study. Based on that study, we expect the estimates from Restricted Network Regression to more accurately capture the unconditional effect of the covariates on the response. Compared to the non-restricted network model, all regression parameters have approximately equal posterior means and the receiver covariate effects have smaller posterior credible intervals. Restricted Network Regression allows the excess network-structured variation in $\boldsymbol{y}$ to be accounted for via the random effects b_j, while avoiding the network confounding and inflated standard errors in the non-restricted network model.

2.6 Discussion

In this chapter, we introduced Restricted Network Regression for models with additive network random effects and established its connection to restricted spatial regression. We characterized the network confounding of the network regression model with additive random effects and node-level covariates, which Restricted Network Regression addresses by forcing the column spaces of the fixed and random effects to be mutually orthogonal. We provided conditions for network regression models to alleviate and mitigate network confounding and proved that Restricted Network Regression does not alleviate network confounding with theoretical results and through simulation. However, we showed through simulation that Restricted Network Regression does mitigate network confounding relative to network regression with no random effects and non-restricted network regression with continuous and binary response data. Restricted Network Regression produces less bias and properly calibrated credible intervals for regression parameters relative to network regression without random effects and non-restricted network regression.

We also explored through simulation the effect of a half-Cauchy prior on the variance components of the network random effects. We found that this prior resulted in comparable bias and credible interval coverage for the unconditional regression effects to the conjugate inverse-gamma prior in probit Restricted Network Regression, with better bias and credible interval coverage in some scenarios.

Finally, we applied Restricted Network Regression to a dataset of Eurovision Song Contest voting. We interpreted the model estimates of Restricted Network Regression alongside those from a model without network-structured random effects and those from a non-restricted network regression model. For the three receiver covariates in the model, the choice of model affected both their posterior mean point estimates and their credible interval estimates. The change in credible interval width is noticeable for all three receiver covariates. Uncertainty, as indicated by the width of posterior credible intervals, increases after adding random effects to the model, but decreases again after restricting them. The widths of the credible intervals for the receiver covariates with restricted random effects is larger than without random effects but smaller than with non-restricted random effects.

Future work in this area includes developing Restricted Network Regression for other forms of network random effects such as multiplicative effects (Hoff, 2021) or latent space distance effects (Hoff et al., 2002). It is less clear which covariates may be confounded with such effects, and whether restricting these random effects can have the same benefits as with additive effects and non-Gaussian data. Theoretical results for binary or other non-Gaussian data that describe the posterior distribution are also needed to make stronger conclusions about Restricted Network Regression on non-Gaussian data. Restricted Network Regression can also be expanded to include bipartite network data or longitudinal network data (e.g., Marrs et al., 2020).

Chapter 3

Fast Bayesian Record Linkage for Streaming Data Contexts

3.1 Introduction

Record linkage is the task of resolving duplicates in two or more overlapping sets of records, or files, from multiple noisy data sources, often without the benefit of having a unique identifier. For example, in a longitudinal survey setting it is possible to have multiple responses from the same person with misspellings or other data errors. This type of error is shown in Table 3.1, where records 1 and 5 represent responses from the same person that were stored with a misspelling in the surname. This presents a problem for those that wish to use this data to make inferences. With the current accessibility and continuity of data, record linkage has become crucial for many areas of application including healthcare (Fleming et al., 2012; Hof et al., 2017), official statistics (Winkler, 2006; Kaplan et al., 2023; Wortman, 2019), and fraud detection and national security (Vatsalan et al., 2017).

Although probabilistic approaches for record linkage have become more common in recent years, principled approaches that are computationally tractable and scalable for large data sets are limited (Binette and Steorts, 2022). Moreover, existing approaches are not suited for streaming data settings, where inference is desired continuously. In the streaming context, data files are expected to arrive sequentially in time with no predetermined number of files. A limited portion of the machine learning literature has targeted the area of near real-time record linkage from a data-driven perspective (Christen et al., 2009; Ioannou et al., 2010; Dey et al., 2011; Altwaijry et al., 2017; Karapiperis et al., 2018).

In this work, we propose new methodology to perform record linkage with streaming data in an efficient and statistically principled fashion under a Bayesian framework. A model-based ap-

Table 3.1: An example of noisy data in need of deduplication. Rows 1 and 5 refer to the same entity but differ due to an error in 'Given Name'.

Given Name	Surname	Age	Occupation
maddisom	ryan	f	3
marleikh	hoffman	d	4
samara	pater5on	d	5
lili	wheatlry	f	7
maddison	ryan	f	3

proach, such as the one we propose, provides interpretable parameters and a way to encode prior knowledge about the data generation process. Bayesian inference also provides natural uncertainty quantification, allowing uncertainty from record linkage to propagate to downstream analysis (Kaplan et al., 2023). This work presents the first model-based approach to perform record linkage in streaming data contexts.

A significant portion of the probabilistic record linkage literature has focused on linking two data files (Fellegi and Sunter, 1969; Tancredi and Liseo, 2011; Gutman et al., 2013; Sadinle, 2017). Recently, Bayesian approaches for multi-file record linkage have become popular (Sadinle and Fienberg, 2013; Sadinle, 2014; Steorts et al., 2016; Betancourt et al., 2016; Aleshin-Guendel and Sadinle, 2023). In particular, Aleshin-Guendel and Sadinle (2023) extend the Bayesian Fellegi-Sunter model of Sadinle (2017) through the use of a partition prior. However, the existing literature is limited to non-streaming settings where the number of files is fixed and known in advance, and record linkage is performed offline in a single procedure. Recent advances have made record linkage possible for big offline data settings, either by jointly performing blocking and entity resolution (Marchant et al., 2021) or by quickly computing point estimates and approximating the posterior distribution (McVeigh et al., 2019). Nonetheless, these approaches are not suited to efficiently assimilate new data. To address this gap in the literature from a fully model-driven perspective, we focus on developing a Bayesian model for multi-file record linkage that enables online data scenarios. Our approach uses recursive Bayesian techniques to produce samples from the full posterior that efficiently update existing draws from the previous posterior. To date, such recursive Bayesian

updates have not been used for linkage in a streaming setting. Our proposed model is constructed under the Fellegi-Sunter paradigm, which entails pairwise comparisons of records (Fellegi and Sunter, 1969; Sadinle and Fienberg, 2013). We explore diffuse and informative prior distributions and provide two streaming samplers.

The remainder of this chapter proceeds as follows. Section 3.2 defines the Bayesian record linkage model for streaming data and defines the problem context, notation, assumptions, and constraints for the model. Section 3.3 introduces two streaming samplers which can be used to perform updates of parameter estimates upon the arrival of a new file. Section 3.4 evaluates these methods on both the quality of samples they produce as well as their speed on simulated data sets. Section 3.5 provides the result of performing streaming record linkage on real-world survey panel data. Section 3.6 contains discussion of further advantages and disadvantages of each streaming update method.

3.2 Bayesian Record Linkage Model for Streaming Data

We will begin this section with a description of the streaming data context, definition of notation, and enumeration of assumptions. We then define the likelihood and prior specification for the multi-file record linkage model.

3.2.1 Streaming Record Linkage Notation

We consider k files $X_1, \ldots, X_k$ that are collected temporally, so that file X_m is available at time T_m, with $T_1 < T_2 < \cdots < T_k$. See Figure B.1 in Appendix B.1 for a diagram depicting this context. Each file X_m contains $n_m \geq 1$ records $X_m = \{\boldsymbol{x}_{mi}\}_{i=1}^{n_m}$, with each n_m potentially distinct. Each record is comprised of p_m fields, and it is assumed that there is a common set of F fields numbered $f = 1, \ldots, F$ across the k files which can be numeric, text, or categorical. Records representing an individual (or entity) can be noisily duplicated across files. Each individual or entity is recorded at most once in each file, corresponding to an assumption that there are no duplicates within a file. This setting has a growing complexity— with k files, all records in

$k(k-1)/2$ pairs of files must be compared and linked. The goal of the record linkage problem is identifying which records in files $X_1, \ldots, X_k$ refer to the same entities. This context is considered "streaming" because data is continuously generated with no predetermined stopping point, and our goal is to update the inference pipeline as new information becomes available.

Our record linkage model for the streaming data context extends the ideas of Fellegi and Sunter (1969) and Sadinle (2017). Within this paradigm, the comparisons are assumed to come from one of two distributions, $\mathcal{M}$ for coreferent pairs and $\mathcal{U}$ for non-coreferent pairs. Two records are coreferent if they refer to the same entity. The Fellegi and Sunter (1969) framework was extended to the Bayesian paradigm for two-file record linkage in Sadinle (2017). In this work, we further extend the model for a general k-file scenario. In contrast to the Aleshin-Guendel and Sadinle (2023) model which also extends the Sadinle (2017) model for the multi-file case, we parameterize the record matching as vectors linking to the most recent previous occurrence of an individual and place an informative prior on these vectors to avoid overlinking between files. This parameterization is the mechanism by which streaming updates are possible.

We denote comparison between two records, $\boldsymbol{x}_{m_1 i}$ in file X_{m_1} and $\boldsymbol{x}_{m_2 j}$ in file X_{m_2}, as a function, $\gamma(\boldsymbol{x}_{m_1 i}, \boldsymbol{x}_{m_2 j})$, which compares the values in each field, f, dependent on field type. Each comparison results in discrete levels $0, \ldots, L_f$ with 0 representing exact equality and subsequent levels representing increased difference. For example, categorical values can be compared in a binary fashion, numerical fields can be compared by binned absolute difference, and text fields can be compared by binned Levenshtein distance (Christen, 2012). We define $P = \sum_{f=1}^{F}(L_f + 1)$, as the total number of levels of disagreement of all fields. The comparison $\gamma(\boldsymbol{x}_{m_1 i}, \boldsymbol{x}_{m_2 j})$ takes the form of a P-vector of binary indicators containing F ones and $P-F$ zeros which indicates the level of disagreement between $\boldsymbol{x}_{m_1 i}$ and $\boldsymbol{x}_{m_2 j}$ in each field. Exactly one 1 must appear in the first $L_1 + 1$ elements of $\gamma(\boldsymbol{x}_{m_1 i}, \boldsymbol{x}_{m_2 j})$, one 1 in the next $L_2 + 1$ elements, and so on. The comparison vectors are collected into matrices $\boldsymbol{\Gamma}^{(1)}, \ldots, \boldsymbol{\Gamma}^{(k-1)}$ where $\boldsymbol{\Gamma}^{(m-1)}$ contains all comparisons between the records in file X_m and previous files. The comparison matrix $\boldsymbol{\Gamma}^{(m-1)}$ thus has $n_m \cdot (n_1 + \cdots + n_{m-1})$ rows and P columns. Define $\boldsymbol{\Gamma}^{(1:m)}$ as $\{\boldsymbol{\Gamma}^{(1)}, \ldots, \boldsymbol{\Gamma}^{(m)}\}$ for $m \in 1, \ldots, k-1$.

Records can be represented as a k-partite graph, with nodes representing records in each file and a link between two records indicating that they are coreferent. This graph can be segmented according to the order of files. First, a bipartite graph between X_1 and X_2; then a tripartite graph between X_1, X_2, and X_3, where records in X_3 link to records in X_1 and X_2; until finally a k-partite graph between $X_1, \ldots, X_k$ where records in X_k link to records in $X_1, \ldots, X_{k-1}$. These graphs can be represented with $k-1$ matching vectors, with one vector per file $X_2, \ldots, X_k$. Each vector, denoted $\boldsymbol{Z}^{(m-1)}$, has length n_m with the value in index j, denoted $Z_j^{(m-1)}$, corresponding to the record $\boldsymbol{x}_{mj}$ as follows,

$$Z_j^{(m-1)} = \begin{cases} \sum_{\ell=1}^{t-1} n_\ell + i & \text{for } t < m, \text{ if } \boldsymbol{x}_{ti} \in X_t \text{ and } \boldsymbol{x}_{mj} \text{ are coreferent,} \\ \sum_{\ell=1}^{m-1} n_\ell + j & \text{otherwise.} \end{cases}$$

Let $\boldsymbol{Z}^{(m-1)} = \left(Z_j^{(m-1)}\right)_{j=1}^{n_m}$ and $\boldsymbol{Z}^{(1:m)} = \left\{\boldsymbol{Z}^{(1)}, \ldots, \boldsymbol{Z}^{(m)}\right\}$ for $m \in 1, \ldots, k-1$. These vectors identify which records are coreferent and are therefore the main parameters of interest in the record linkage problem.

We also define parameters $\boldsymbol{m}$ and $\boldsymbol{u}$, which specify the distributions $\mathcal{M}$ and $\mathcal{U}$ respectively. Both $\boldsymbol{m}$ and $\boldsymbol{u}$ are P-vectors which can be separated into the sub-vectors $\boldsymbol{m} = \begin{bmatrix} \boldsymbol{m}_1 & \ldots & \boldsymbol{m}_F \end{bmatrix}$ and $\boldsymbol{u} = \begin{bmatrix} \boldsymbol{u}_1 & \ldots & \boldsymbol{u}_F \end{bmatrix}$, where $\boldsymbol{m}_f$ and $\boldsymbol{u}_f$ have length $L_f + 1$. Then $\mathcal{M}(\boldsymbol{m}) = \prod_{f=1}^{F} \text{Multinomial}(1; \boldsymbol{m}_f)$ and $\mathcal{U}(\boldsymbol{u}) = \prod_{f=1}^{F} \text{Multinomial}(1; \boldsymbol{u}_f)$ are the distributions for matches and non-matches, respectively.

3.2.2 Preserving the Duplicate-Free File Assumption

Preserving the assumption of duplicate-free files with a large number of files is a challenge because the combination of several links throughout the parameters $\boldsymbol{Z}^{(1:(k-1))}$ may imply that two records in the same file are coreferent. For example if $Z_1^{(1)} = 1$, $Z_1^{(2)} = 1$, and $Z_2^{(2)} = n_1 + 1$, then the records $\boldsymbol{x}_{31}$ and $\boldsymbol{x}_{32}$ are implied to be coreferent even though they are not directly linked to the same record. We address this by placing constraints on the values of these parameters such that no two records may link directly to the same record in a previous file. Because each record

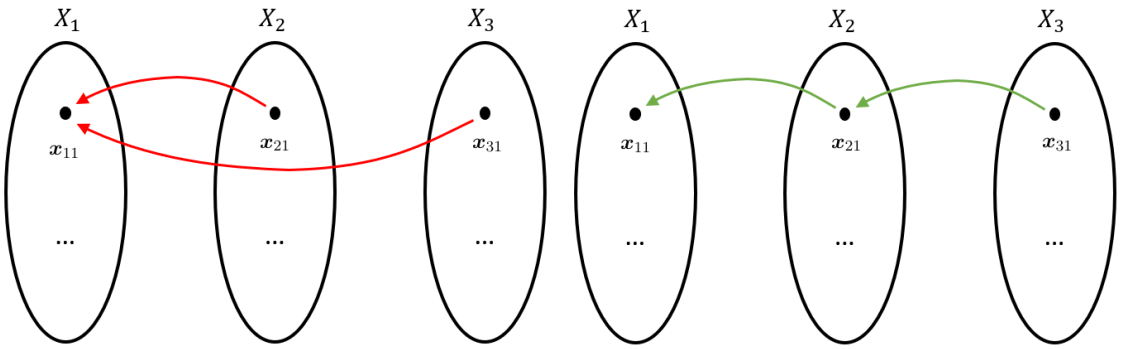


Figure 3.1: Examples of both prohibited (left) and allowed (right) links between records in three files. On the left $Z_1^{(1)} = 1$ and $Z_1^{(2)} = 1$, while on the right $Z_1^{(1)} = 1$ and $Z_1^{(2)} = n_1 + 1$. The left configuration is prohibited because the record $\boldsymbol{x}_{11}$ receives a link from both $\boldsymbol{x}_{21}$ and $\boldsymbol{x}_{31}$. Both configurations define the same cluster containing these three records.

can send at most one link to a previous record and receive at most one link from a later record, we guarantee that no two records in the same file are transitively linked. Figure 3.1 depicts a three-file example of prohibited and allowed values of $\boldsymbol{Z}^{(1)}$ and $\boldsymbol{Z}^{(2)}$. Both values are logically equivalent, but without this constraint the prohibited configuration could allow for one record in file X_4 to link to record $\boldsymbol{x}_{31}$ while another links to record $\boldsymbol{x}_{21}$, becoming coreferent and violating the assumption.

The bipartite matching, $\boldsymbol{Z}^{(1)}$, is constrained in a manner consistent with Sadinle (2017). Namely, that there can be no two $Z_i^{(1)} = Z_{i'}^{(1)}$ where $i \neq i'$. The tripartite matching must be similarly restricted to enforce our link validity constraint. Specifically, for some $1 \leq i \leq n_3$ and $1 \leq j \leq n_1$, $Z_i^{(2)}$ cannot equal j if $Z_k^{(1)} = j$ for any $k \leq n_2$. That is, record i cannot be linked to a record j in X_1 which already has a match in X_2. To enforce transitivity of the coreference relationship, comparisons with files $X_m, m \geq 3$ will be constrained.

Definition 3.2.1. Link Validity Constraint. Let $\mathcal{C}_k$ be the set of all matching vectors $\boldsymbol{Z}^{(1:(k-1))}$ such that every record $\boldsymbol{x}_{m_1 i}$ receives at most one link from a record $\boldsymbol{x}_{m_2 j}$ where $m_2 > m_1$. That is, there is at most one value in any $\boldsymbol{Z}^{(m_2-1)}$ with $m_2 > m_1$ that equals $\sum_{\ell=1}^{m_1-1} n_\ell + i$. Matching vectors $\boldsymbol{Z}^{(1:(k-1))}$ are valid if and only if $\boldsymbol{Z}^{(1:(k-1))} \in \mathcal{C}_k$.

This constraint aids in the identifiability of the parameters $\boldsymbol{Z}^{(1:(k-1))}$. Under these constraints each logical cluster of at most one record from each file has one unique valid representation,

namely a chain of links from the latest-appearing record to the earliest-appearing record, linking records in order of appearance. The chain nature aids in computation— it becomes possible to list all members of a cluster by starting at one of its members and traversing the chain forwards and backwards without needing to branch or double back.

3.2.3 Likelihood

In this section and Section 3.2.4, we define the likelihood and priors that contribute to the streaming record linkage model posterior. The full posterior distribution is presented in Appendix B.2.1. Consistent with the formulation in Sadinle (2017), the likelihood for the two-file case is defined as

$$P(\boldsymbol{\Gamma}^{(1)}|\boldsymbol{Z}^{(1)},\boldsymbol{m},\boldsymbol{u}) = \prod_{i=1}^{n_1}\prod_{j=1}^{n_2} P(\gamma_{ij}|\boldsymbol{Z}^{(1)},\boldsymbol{m},\boldsymbol{u}) = \prod_{i=1}^{n_1}\prod_{j=1}^{n_2}\prod_{f=1}^{F}\prod_{\ell=0}^{L_f}\left[m_{f\ell}^{\mathbb{I}(Z_j^{(1)}=i)}\, u_{f\ell}^{\mathbb{I}(Z_j^{(1)}\neq i)}\right]^{\gamma_{ij}^{f\ell}},$$

where $\gamma_{ij} := \gamma(\boldsymbol{x}_{1i}, \boldsymbol{x}_{2j})$, $\gamma_{ij}^{f\ell}$ is the component corresponding to level ℓ of field f, and $\mathbb{I}(\cdot)$ is the indicator function taking a value of 1 if its argument holds and 0 otherwise. For every pair of records, one from each file, $\boldsymbol{m}$ contributes to the distribution if the records are linked by $\boldsymbol{Z}^{(1)}$ and $\boldsymbol{u}$ contributes otherwise. We extend this to the k-file case by defining the match set, $M := M(\boldsymbol{Z}^{(1:(k-1))}) = \{(\boldsymbol{x}_{m_1 i}, \boldsymbol{x}_{m_2 j}) : \boldsymbol{x}_{m_1 i} \text{ and } \boldsymbol{x}_{m_2 j} \text{ are linked}\}$, to contain all pairs of records that are linked either directly or transitively through a combination of multiple vectors $\boldsymbol{Z}^{(1:(k-1))}$. Testing whether $(\boldsymbol{x}_{m_1 i}, \boldsymbol{x}_{m_2 j}) \in M$ for $m_1 < m_2$ is done by the process of *link tracing*. This is the process by which we determine the links implied by transitivity in the match vectors. To perform link tracing, we start at $\boldsymbol{x}_{m_2 j}$ and follow the values in $\boldsymbol{Z}^{(1:(k-1))}$ to travel down the chain of links, starting with $Z_j^{(m_2-1)}$. If $\boldsymbol{x}_{m_1 j}$ is ever reached, then $(\boldsymbol{x}_{m_1 i}, \boldsymbol{x}_{m_2 j}) \in M$, while if a dead end is reached first, then $(\boldsymbol{x}_{m_1 i}, \boldsymbol{x}_{m_2 j}) \notin M$.

The full data model in the k-file case is then

$$P(\boldsymbol{\Gamma}^{(1:(k-1))}|\boldsymbol{m},\boldsymbol{u},\boldsymbol{Z}^{(1:(k-1))}) \\ = \prod_{m_1<m_2}^{k}\prod_{i=1}^{n_{m_1}}\prod_{j=1}^{n_{m_2}}\prod_{f=1}^{F}\prod_{\ell=0}^{L_f}\left[m_{f\ell}^{\mathbb{I}((\boldsymbol{x}_{m_1 i},\boldsymbol{x}_{m_2 j})\in M)}u_{f\ell}^{\mathbb{I}((\boldsymbol{x}_{m_1 i},\boldsymbol{x}_{m_2 j})\notin M)}\right]^{\gamma^{f\ell}(\boldsymbol{x}_{m_1 i},\boldsymbol{x}_{m_2 j})}. \tag{3.1}$$

The likelihood of the k-file Bayesian record linkage model encodes the assumption that all comparisons, $\boldsymbol{\Gamma}$, are conditionally independent given the parameters $\boldsymbol{m}$, $\boldsymbol{u}$, and $\boldsymbol{Z}^{(1:(k-1))}$. The same $\boldsymbol{m}$ and $\boldsymbol{u}$ probabilities appear in the distribution of comparisons between each pair of files, corresponding to an assumption of equal propensity for error in each file. Alternatively separate probabilities, $\boldsymbol{m}_{t_1t_2}$ and $\boldsymbol{u}_{t_1t_2}$, can be specified for the comparisons between files X_{t_1} and X_{t_2}, as in Aleshin-Guendel and Sadinle (2023). However, every new file, X_k, will require $2(k-1)$ new parameters, $\boldsymbol{m}_{t_1k}$ and $\boldsymbol{u}_{t_1k}$ for all $t_1 < k$, which may affect the model's performance in a streaming setting. The support of the data distribution is dependent on the vectors $\boldsymbol{Z}^{(1:(k-1))}$, specifically, the matching vectors must satisfy the link validity constraint given in Definition 3.2.1. We explicitly write this constraint as an indicator function in the likelihood:

$$L(\boldsymbol{m},\boldsymbol{u},\boldsymbol{Z}^{(1:(k-1))}) = \mathbb{I}(\boldsymbol{Z}^{(1:(k-1))}\in\mathcal{C}_k)\cdot P(\boldsymbol{\Gamma}^{(1:(k-1))}|\boldsymbol{m},\boldsymbol{u},\boldsymbol{Z}^{(1:(k-1))}). \tag{3.2}$$

We discuss the benefit to prior selection in Section 3.2.4.

3.2.4 Prior Specification

Priors for $\boldsymbol{m}$ and $\boldsymbol{u}$

The parameters $\boldsymbol{m}$ and $\boldsymbol{u}$ are probabilities of a multinomial distribution, so we specify conjugate Dirichlet priors. Specifically, we let $\boldsymbol{m}_f \sim \text{Dirichlet}(\boldsymbol{a}_f)$ and $\boldsymbol{u}_f \sim \text{Dirichlet}(\boldsymbol{b}_f)$, for $f = 1,\ldots,F$, where $\boldsymbol{a}_f$ and $\boldsymbol{b}_f$ are vectors with the same dimension, L_f+1, as $\boldsymbol{m}_f$ and $\boldsymbol{u}_f$. For a diffuse prior we can set $\boldsymbol{a} = \boldsymbol{b} = \mathbf{1}$. Also it can be useful to encode prior knowledge about the propensity for duplicates to have errors in the prior for $\boldsymbol{m}$. For example, if we know that an error

in field f of a duplicated record has probability p of occurring, we can let

$$\boldsymbol{a}_f = s \cdot \begin{bmatrix} 1-p & p/L_f & \dots & p/L_f \end{bmatrix}, \tag{3.3}$$

with s determining the strength of the prior knowledge. We empirically investigate the effect of this informative prior specification on $\boldsymbol{m}$ in simulated data scenarios in Section 3.4.

Priors for $\boldsymbol{Z}^{(k-1)}$

We construct the prior for streaming matching vectors using the same hierarchy as specified in Sadinle (2017). For the parameter $\boldsymbol{Z}^{(k-1)}$ after the arrival of the k^{th} file, we let

$$\mathbb{I}\left(Z_j^{(k-1)} \leq \sum_{m=1}^{k-1} n_m\right) \Bigg| \pi \sim \text{Bernoulli}(\pi)$$

$$\boldsymbol{Z}^{(k-1)} \Bigg| \left\{ \mathbb{I}\left(Z_j^{(k-1)} \leq \sum_{m=1}^{k-1} n_m\right) \right\}_{j=1}^{n_k} \sim \text{Uniform}\left(\{\text{all } k\text{-partite matchings}\}\right)$$

Allowing $\pi \sim \text{Beta}(\alpha_\pi, \beta_\pi)$ results in the marginal streaming prior

$$P(\boldsymbol{Z}^{(k-1)} | \alpha_\pi, \beta_\pi) = \frac{(N - n_{k\cdot}(\boldsymbol{Z}^{(k-1)}))!}{N!} \cdot \frac{\text{B}(n_{k\cdot}(\boldsymbol{Z}^{(k-1)}) + \alpha_\pi, n_k - n_{k\cdot}(\boldsymbol{Z}^{(k-1)}) + \beta_\pi)}{\text{B}(\alpha_\pi, \beta_\pi)}, \tag{3.4}$$

where $N = \sum_{m=1}^{k-1} n_m$ and $n_{k\cdot}(\boldsymbol{Z}^{(k-1)}) = \sum_{j=1}^{n_k} I(Z_j^{(k-1)} \leq N)$

This streaming prior enforces the condition that no two records within the same file can link to the same record in a previous file. However, the more general link validity constraint in Definition 3.2.1 is not enforced in the prior. Not enforcing the general constraint allows the priors for all $\boldsymbol{Z}^{(m)}$ to be independent. We define a constrained version of the prior in Equation 3.4 in Section 3.2.4 and demonstrate its inferiority due to over-linking.

Another Prior for $\boldsymbol{Z}^{(k-1)}$ that Enforces Link Validity but Overlinks

Define the set $C := C(\boldsymbol{Z}^{(1:(k-2))})$ to be the *candidate set*, the set of records with no links from later files. Let $I_C := I_C(\boldsymbol{Z}^{(1:(k-2))})$ be the corresponding indices of those records. In other

words, C is the set of records which may be linked directly to a record in file X_k via $\boldsymbol{Z}^{(k-1)}$ without violating the link validity constraint in Definition 3.2.1 and I_C are the corresponding values allowed for components of $\boldsymbol{Z}^{(k-1)}$. We can create an alternative prior for $\boldsymbol{Z}^{(k-1)}$ that will directly enforce the link validity constraint,

$$\begin{aligned} \mathbb{I}\left(Z_j^{(k-1)} \leq \sum_{m=1}^{k-1} n_m\right) \Bigg| \pi &\sim \text{Bernoulli}(\pi) \\ \boldsymbol{Z}^{(k-1)} \Bigg| \left\{\mathbb{I}\left(Z_j^{(k-1)} \leq \sum_{m=1}^{k-1} n_m\right)\right\}_{j=1}^{n_k} &\sim \text{Uniform}\left(\left\{\begin{array}{l}\text{all } k\text{-partite matchings satis-} \\ \text{fying Definition 3.2.1}\end{array}\right\}\right) \end{aligned} \tag{3.5}$$

This slight change in line 3.5 results in the (conditional) prior for $\boldsymbol{Z}^{(k-1)}$,

$$\begin{aligned} &P(\boldsymbol{Z}^{(k-1)} | \boldsymbol{Z}^{(1:(k-2))}, \alpha_\pi, \beta_\pi) \\ &\quad = \frac{(|C| - n_{k\cdot}(\boldsymbol{Z}^{(k-1)}))!}{|C|!} \cdot \frac{\mathrm{B}(n_{k\cdot}(\boldsymbol{Z}^{(k-1)}) + \alpha_\pi, n_k - n_{k\cdot}(\boldsymbol{Z}^{(k-1)}) + \beta_\pi)}{\mathrm{B}(\alpha_\pi, \beta_\pi)}. \end{aligned} \tag{3.6}$$

This prior results in increased over-linking as the number of files increases.

Theorem 3.2.1. *Consider a k-file record linkage problem with an initial state $\boldsymbol{Z}^{(1:(k-1))}$ and an alternate state $(\boldsymbol{Z}^{*(1:(k-2))}, \boldsymbol{Z}^{(k-1)})$ such that $\boldsymbol{Z}^{*(1:(k-2))}$ are identical to $\boldsymbol{Z}^{(1:(k-2))}$ except for the addition of one link. That is, there exists an $\ell < k$, $j \leq n_\ell$ and $i \leq n_1 + \cdots + n_{\ell-1}$ such that $Z_j^{*(\ell-1)} = i$ and $Z_j^{(\ell-1)} = n_1 + \cdots + n_{\ell-1} + j$. Let*

$$R = \frac{P(\boldsymbol{Z}^{*(1:(k-2))}, \boldsymbol{Z}^{(k-1)})}{P(\boldsymbol{Z}^{(1:(k-2))}, \boldsymbol{Z}^{(k-1)})} \div \frac{P(\boldsymbol{Z}^{*(1:(k-2))})}{P(\boldsymbol{Z}^{(1:(k-2))})}.$$

When the prior in Equation 3.6 is specified for $\boldsymbol{Z}^{(2:(k-1))}$, $R \geq 1$ with equality only when there are no links in $\boldsymbol{Z}^{(k-1)}$. When the prior in Equation 3.4 is specified for $\boldsymbol{Z}^{(2:(k-1))}$, $R = 1$.

Proof. See Appendix B.3.1. □

The ratio R represents the relative prior probability of an additional link in $\boldsymbol{Z}^{(1:(k-2))}$ after file k arrives compared to before file k arrives, with any given state of $\boldsymbol{Z}^{(k-1)}$. Therefore with the

prior in Equation 3.6, the relative prior probability to add one link in $\boldsymbol{Z}^{(1:(k-2))}$ is higher after the arrival of file X_k than before. For this reason, we use the prior in Equation 3.4 and enforce the link validity constraint in Definition 3.2.1 via the likelihood.

3.3 Streaming Sampling

The key to Bayesian streaming record linkage is an efficient means of updating the posterior distribution of existing parameters after the arrival of a new file, X_k. In this section, we introduce two sampling approaches we have adapted to address this problem, Prior-Proposal-Recursive-Bayes (PPRB) and Sequential MCMC (SMCMC).

3.3.1 Prior-Proposal-Recursive Bayes (PPRB)

Prior-Proposal-Recursive Bayes is a recursive Bayesian sampling technique in which existing posterior samples from a previous stage are used as independent Metropolis proposals to sample from a later stage posterior distribution, conditioned on new data (Hooten et al., 2021). We consider a model with parameters $\boldsymbol{\theta}$ and data $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$:

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \end{bmatrix} \sim p(\boldsymbol{y}|\boldsymbol{\theta}) = p(\boldsymbol{y}_1|\boldsymbol{\theta})p(\boldsymbol{y}_2|\boldsymbol{\theta}, \boldsymbol{y}_1), \;\; \boldsymbol{\theta} \sim p(\boldsymbol{\theta})$$

We assume $\boldsymbol{y}_1$ arrives before $\boldsymbol{y}_2$ and posterior samples $\boldsymbol{\theta}_{(1)} \dots \boldsymbol{\theta}_{(S)}$ are obtained from $p(\boldsymbol{\theta}|\boldsymbol{y}_1)$. After $\boldsymbol{y}_2$ arrives, these samples are resampled as independent Metropolis proposals for the updated posterior distribution $p(\boldsymbol{\theta}|\boldsymbol{y}_1, \boldsymbol{y}_2)$. The acceptance ratio α for the proposal $\boldsymbol{\theta}'$ and current value $\boldsymbol{\theta}$ simplifies to

$$\alpha = \min\left(\frac{p(\boldsymbol{y}_2|\boldsymbol{\theta}', \boldsymbol{y}_1)}{p(\boldsymbol{y}_2|\boldsymbol{\theta}, \boldsymbol{y}_1)}, 1\right).$$

This ratio depends only on the full conditional distribution of the new data, $\boldsymbol{y}_2$, and so can be calculated quickly. If $\boldsymbol{y}_2$ and $\boldsymbol{y}_1$ are conditionally independent given $\boldsymbol{\theta}$, then the old data $\boldsymbol{y}_1$ does not need to be stored in order to calculate α or perform PPRB.

To apply PPRB to the Bayesian record linkage model when a file X_k arrives, we have $\boldsymbol{y}_2 = \boldsymbol{\Gamma}^{(k-1)}$, $\boldsymbol{y}_1 = \boldsymbol{\Gamma}^{(1:(k-2))}$, and $\boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{m} & \boldsymbol{u} & \boldsymbol{Z}^{(1:(k-2))} \end{bmatrix}$. Since all comparisons are assumed conditionally independent given the parameters $\boldsymbol{m}$, $\boldsymbol{u}$, and $\boldsymbol{Z}^{(1:(k-2))}$, the past calculated comparisons $\boldsymbol{\Gamma}^{(1:(k-2))}$ would not be needed to calculate α or perform PPRB. However, the streaming record linkage model requires additional parameters, $\boldsymbol{Z}^{(k-1)}$, for the distribution of the new data, $\boldsymbol{\Gamma}^{(k-1)}$, so a straight forward application of PPRB is not possible. Hooten et al. (2021) propose drawing values of the new parameter from its predictive distribution and appending those values to the existing samples prior to PPRB, which retains the simplified form of the acceptance ratio, α. In the streaming record linkage problem, the predictive distribution of $\boldsymbol{Z}^{(k-1)}$ reduces to its prior: $p(\boldsymbol{Z}^{(k-1)}|\boldsymbol{m}, \boldsymbol{u}, \boldsymbol{Z}^{(1:(k-2))}, \boldsymbol{\Gamma}^{(1:(k-2))}) = p(\boldsymbol{Z}^{(k-1)})$. However, because the space of possible values of $\boldsymbol{Z}^{(k-1)}$ is on the order of $(\sum_{\ell=1}^{k-1} n_\ell)^{n_k}$ and the proposed prior is diffuse, these values are rarely good proposals for the updated posterior distribution, leading to low acceptance rates and slow mixing.

For this reason, we propose PPRB-within-Gibbs, a Gibbs sampler in which one of the steps is an independent Metropolis proposal from prior stage posterior samples.

Definition 3.3.1. PPRB-within-Gibbs algorithm. Consider a general model with partitioned data $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$, and parameters $\boldsymbol{\theta}_1$, $\boldsymbol{\theta}_2$, and $\boldsymbol{\theta}_3$:

$$\begin{aligned} \boldsymbol{y}_1|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 &\sim p(\boldsymbol{y}_1|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) \\ \boldsymbol{y}_2|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3 &\sim p(\boldsymbol{y}_2|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3) \\ \boldsymbol{\theta}_1 \sim p(\boldsymbol{\theta}_1),\ \boldsymbol{\theta}_2 &\sim p(\boldsymbol{\theta}_2),\ \boldsymbol{\theta}_3 \sim p(\boldsymbol{\theta}_3) \end{aligned}$$

The parameters $\boldsymbol{\theta}_1$, $\boldsymbol{\theta}_2$, and $\boldsymbol{\theta}_3$ have independent priors, $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ are conditionally independent given the parameters, and the first wave of data, $\boldsymbol{y}_1$, is not dependent on $\boldsymbol{\theta}_3$. Let there be existing posterior samples, $\{\boldsymbol{\theta}_1^s\}_{s=1}^S$ from the distribution $p(\boldsymbol{\theta}_1|\boldsymbol{y}_1)$. Then for the desired number of posterior samples,

1. Update the parameter $\boldsymbol{\theta}_2$ from the full conditional distribution $[\boldsymbol{\theta}_2|\boldsymbol{\theta}_1, \boldsymbol{\theta}_3, \boldsymbol{y}_1, \boldsymbol{y}_2]$,

2. (PPRB step) Propose a new value $\boldsymbol{\theta}_1^*$ by drawing from the existing posterior samples $\{\boldsymbol{\theta}_1^s\}_{s=1}^S$ with replacement. Accept or reject the proposal using the Metropolis-Hastings ratio

$$\alpha = \min\left(\frac{p(\boldsymbol{y}_2|\boldsymbol{\theta}_1^*, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3)}{p(\boldsymbol{y}_2|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3)} \frac{p(\boldsymbol{\theta}_2|\boldsymbol{\theta}_1^*, \boldsymbol{y}_1)}{p(\boldsymbol{\theta}_2|\boldsymbol{\theta}_1, \boldsymbol{y}_1)}, 1\right),$$

3. Update the parameter $\boldsymbol{\theta}_3$ from the full conditional distribution $[\boldsymbol{\theta}_3|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{y}_1, \boldsymbol{y}_2]$,

recording the values of $\boldsymbol{\theta}_1$, $\boldsymbol{\theta}_2$, and $\boldsymbol{\theta}_3$ at the end of each iteration.

Theorem 3.3.1. *The PPRB-within-Gibbs sampler (Definition 3.3.1) produces an ergodic Markov chain with the model's posterior distribution as its target distribution if the posterior distribution satisfies the following positivity condition,*

$$p(\boldsymbol{\theta}_1|\boldsymbol{y}_1, \boldsymbol{y}_2) > 0,\ p(\boldsymbol{\theta}_2|\boldsymbol{y}_1, \boldsymbol{y}_2) > 0,\ p(\boldsymbol{\theta}_3|\boldsymbol{y}_1, \boldsymbol{y}_2) > 0 \implies p(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3|\boldsymbol{y}_1, \boldsymbol{y}_2) > 0.$$

Proof. See Appendix B.3.2. □

S is the number of samples drawn from the previous posterior distribution, $p(\boldsymbol{\theta}_1|\boldsymbol{y}_1)$ and generally cannot be increased. As the pool of samples, $\{\boldsymbol{\theta}_1^s\}_{s=1}^S$, approximates the distribution $p(\boldsymbol{\theta}_1|\boldsymbol{y}_1)$ for the purpose of proposals, a larger S will lead to better proposals. However, we see in Section 3.4.4 that the pool of samples available to PPRB or PPRB-within-Gibbs will degrade over time after repeated applications in a streaming setting. A large S can extend the utility of the pool but will not keep it from degrading. We briefly mention future work that could address this degradation in Section 3.6.

PPRB-within-Gibbs is applicable to the streaming record linkage model via the relationships $\boldsymbol{\theta}_1 = \boldsymbol{Z}^{(1:(k-2))}$, $\boldsymbol{\theta}_2 = [\boldsymbol{m}, \boldsymbol{u}]$, $\boldsymbol{\theta}_3 = \boldsymbol{Z}^{(k-1)}$, $\boldsymbol{y}_1 = \boldsymbol{\Gamma}^{(1:(k-2))}$, $\boldsymbol{y}_2 = \boldsymbol{\Gamma}^{(k-1)}$, which satisfies all the preconditions of the algorithm. The algorithm steps for the streaming record linkage model as defined in Section 3.2 are listed in Appendix B.3.2. The acceptance ratio, α, is now the product of two ratios. The first ratio is of the data distribution of new data, as in original PPRB, evaluated both at the proposed $\boldsymbol{Z}_*^{(1:(k-1))}$ and the current $\boldsymbol{Z}^{(1:(k-1))}$. The second ratio is of the full conditional

density of $\boldsymbol{m}$ and $\boldsymbol{u}$, but only conditioned on the pre-arrived data and pre-existing parameters. As such, these values can be pre-calculated for every existing posterior sample from the previous stage posterior.

This approach retains the appealing speed and low storage requirements of PPRB by utilizing existing posterior samples, while also avoiding an identified challenge of the original method proposed by Hooten et al. (2021) by drawing from the full conditional distribution of $\boldsymbol{Z}^{(k-1)}$ rather than its prior. However, as resampling filtering methods, PPRB and PPRB-within-Gibbs can never sample states of any $\boldsymbol{Z}^{(m)}$ not present in the first pool of posterior samples of that parameter. As a result, the pool of samples for any $\boldsymbol{Z}^{(m)}$ will converge to a degenerate distribution as $k \to \infty$ (Lunn et al., 2013). We see evidence in Section 3.4.4 and discuss potential ways to address this in Section 3.6.

3.3.2 Sequential MCMC (SMCMC)

Sequential MCMC is a sampling algorithm based on parallel sequential approximation (Yang and Dunson, 2013). Starting from an existing ensemble of posterior samples from $P(\boldsymbol{m}, \boldsymbol{u}, \boldsymbol{Z}^{(1:(k-2))}|\boldsymbol{\Gamma}^{(1:(k-2))})$, SMCMC uses two kernels:

1. The Jumping Kernel — a probability distribution $J(\boldsymbol{Z}^{(k-1)}|\cdot)$ which is responsible for initializing a value of $\boldsymbol{Z}^{(k-1)}$ for each sample, potentially conditioning on old or new data.
2. The Transition Kernel — any MCMC kernel, T, that targets the updated posterior distribution, $P(\boldsymbol{m}, \boldsymbol{u}, \boldsymbol{Z}^{(1:(k-1))}|\boldsymbol{\Gamma}^{(1:(k-1))})$.

These kernels are applied in parallel initialized at each existing sample, first using the jumping kernel to initialize $\boldsymbol{Z}^{(k-1)}$ and then repeatedly applying the transition kernel T until desired convergence is achieved. Final states of each parallel chain are taken as the new ensemble. SMCMC is a massively parallel MCMC algorithm that is expected to have fast convergence if the posterior based on new data and the posterior based on current data are similar in shape. Both jumping and transition kernels may depend on previously arrived data as well as new data. For Bayesian

multi-file record linkage, we choose the transition kernel T as a Gibbs-style kernel which updates all parameters in sequence, and the jumping kernel J to be the full conditional update of $\boldsymbol{Z}^{(k-1)}$.

SMCMC differs from PPRB-within-Gibbs in that it operates on an independent ensemble of samples. If the initial size of the ensemble is S, SMCMC produces S *independent* samples from the updated posterior distribution by nature of the parallel algorithm. Therefore the ensemble can remain relatively small, and only a small number of posterior draws need to be saved after the arrival of each file. The ensemble is never filtered, so converging to a degenerate distribution is not a concern for SMCMC. The transition kernel within SMCMC updates all parameters and maintains the same speed as MCMC for the updated posterior using the full data. The speed benefits of SMCMC then come from the ability to use as many as S parallel chains with well-chosen initial values. By contrast, PPRB-within-Gibbs's speed benefits come from simplifying the parameter update step. Unlike PPRB-within-Gibbs, SMCMC requires the full data be stored in perpetuity because with every new file the transition kernel will update all parameters.

3.3.3 Proposals for Matching Vector Updates

Both streaming samplers, PPRB-within-Gibbs and SMCMC, depend on full conditional updates of matching vectors. Step 3 of PPRB-within-Gibbs and the jumping kernel from SMCMC are both full conditional updates of the most recent vector $\boldsymbol{Z}^{(k-1)}$, and the transition kernel of SMCMC must update all matching vectors. The choice of update is crucial for both speed and convergence of the sampler.

A straight-forward method for performing updates of $\boldsymbol{Z}^{(k-1)}$ is to update each component $Z_j^{(k-1)}$ in turn for $j = 1, \ldots, n_k$. This method is used by Sadinle (2017) to update the matching vector in the two-file Bayesian record linkage model. The support for each component $Z_j^{(k-1)}$ is enumerable as $\{1, \ldots, \sum_{\ell=1}^{k-1} n_\ell, \sum_{\ell=1}^{k-1} n_\ell + j\}$. To draw from the full conditional distribution of each $Z_j^{(k-1)}$, the product of the likelihood and priors is evaluated for each potential value, normalized, and used as probabilities to sample the new value. The full transition kernel using these component-wise proposals for matching vectors is defined in Definition B.3.1 in Appendix B.3.2.

Zanella (2020) describes a class of locally balanced pointwise informed proposals distributions to improve sampling in high-dimensional discrete spaces. For a sample space $\mathcal{X}$ with a target distribution given by $\pi(x)$, these proposals have the form,

$$Q_g(x, y) = \frac{g\left(\frac{\pi(y)}{\pi(x)}\right) K(x, y)}{Z_g(x)},$$

where the proposed move is from a point x to a point y. $K(x, y)$ is a symmetric uninformed local proposal distribution, $g : \mathbb{R}^+ \to \mathbb{R}^+$ is a function and $Z_g(x)$ is the normalizing constant. The goal of these proposals is to improve the uninformed proposal K by biasing towards points with higher probability through the multiplicative term $g(\pi(y)/\pi(x))$. The uninformed kernel K is arbitrary, and Q_g is called *locally balanced* if and only if $g(t) = tg(1/t)$. A consequence of this property of g along with a symmetric local proposal K is that the Metropolis-Hastings acceptance ratio for locally balanced proposals simplifies to the ratio of normalizing constants, $\min(Z_g(x)/Z_g(y), 1)$.

To apply locally balanced proposals to the Bayesian multi-file record linkage model, we choose $g(t) = t/(1 + t)$ and K to be the kernel defined by making a single randomly chosen add, delete, swap, or double-swap move. The kernel K can optionally be blocked, where first a subset of records in file X_k and an equally sized subset of records in files $X_1, \ldots, X_{k-1}$ are randomly selected and then, only moves which affect links between these subsets are considered. Blocking limits the scope of possible moves for each update, which in turn decreases the time required per update. However, blocking also increases the chance of proposing a move to a lower probability state which is more likely to be rejected, requiring more updates to sample effectively. We use a block size in Section 3.4 which is fast while still producing many accepted proposals. The full transition kernel using these locally balanced proposals for matching vectors is defined in Definition B.3.2 in Appendix B.3.2.

The component-wise full conditional updates can take larger steps than the locally balanced proposals because each value in $\boldsymbol{Z}^{(k-1)}$ has the potential to be updated. In contrast, the locally balanced proposals can at most update two components of $\boldsymbol{Z}^{(k-1)}$ with a double-swap operation.

The component-wise full conditional updates, however, are more computationally intensive as the likelihood needs to be calculated at more potential states and there is no option for blocking. We use locally balanced proposals to update $\boldsymbol{Z}^{(k-1)}$ in PPRB-within-Gibbs. In Section 3.4 both locally balanced and component-wise proposals are used within SMCMC and their speed and sampling performance are compared.

3.4 Simulation Study

To assess both the performance of the model and speed of the streaming update, we evaluate our Bayesian multi-file record linkage model and both streaming samplers on simulated data. We choose to focus on the four file case, since the arrival of the fourth file is the earliest point at which two sequential streaming updates can have been used, demonstrating the potential for use in streaming settings.

3.4.1 Data Simulation

Data were simulated using the GeCo software package (Tran et al., 2013) which creates realistic simulated data about individuals. Each record was given 10 fields: first name, last name, occupation, and age, plus 6 categorical fields with values drawn uniformly from 12 possible categories. For each of four levels of overlap (10%, 30%, 50%, and 90%), four files of 200 records each were created. Duplicate records were allowed in consecutive and non-consecutive datasets. In each duplicated record in files X_2, X_3 and X_4, a maximum of either 2, 4, or 6 errors were inserted. Errors were inserted into text fields of first name and last name by simulating typos, common misspellings, and OCR errors using the GeCo package (Tran et al., 2013). Errors were inserted into the remaining categorical fields by replacing their value with a category selected randomly uniform from all possible categories. Each field could have errors, with text fields more likely than categorical fields. A total of 12 datasets were created, one at each combination of error and overlap. This simulation is intended to mimic a longitudinal survey in which we have demographic information and the answers to 6 identifying categorical questions with varying levels of noise and

overlap. Comparison vectors were created by comparing each field between pairs of records. Text fields were compared using binned normalized Levenshtein distance with 4 levels: exact equality, $(0, 0.25]$, $(0.25, 0.5]$, and $(0.5, 1]$. Categorical fields were compared in a binary fashion. All computation in this section and in Section 3.5 was performed using the RMACC Summit Supercomputer (Anderson et al., 2017). We utilized the accompanying package `bstrl` (Taylor et al., 2022) on `R` version 3.5.0 on Intel Haswell CPUs with 24 cores and 4.84 GB of memory per CPU.

3.4.2 Link Accuracy

We assess the accuracy of our multi-file record linkage model by evaluating samples from the posterior distribution obtained using a non-streaming Gibbs sampler. The streaming samplers should target the same posterior distribution as the Gibbs sampler, thus we present a comparison on model performance alone. We compare the streaming samplers on runtime in Section 3.4.3. We use three strengths of prior on the parameter $\boldsymbol{m}$. For the diffuse prior (Flat), we set $\boldsymbol{a} = \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix}$. Then for weakly informed (Weak) and strongly informed (Strong) priors, we use Equation 3.3 to determine $\boldsymbol{a}$. We use $s = 12$ for the weakly informed prior and $s = 120$ for the strongly informed prior. In both the weakly and strongly informed priors, $p = 1/2$ for string fields and $p = 1/8$ for categorical fields. These values of p reflect a prior probability of error of $1/2$ in string fields and $1/8$ in categorical fields, and an average of 2 errors per record. For comparison, we evaluate the multi-file Bayesian linkage model of Aleshin-Guendel and Sadinle (2023) as implemented in the `multilink` package (Multilink), the empirically motivated Bayesian entity resolution model of Steorts (2015) as implemented in the `blink` package (Blink), and a semi-supervised Fellegi-Sunter model with support vector machine used to classify links as implemented in the `RecordLinkage` package (SVM) with 1% of the record pairs used as training data. Multilink is similar to our proposed model in that it is a Bayesian multi-file Fellegi-Sunter extension. However, it differs from ours in that it is based on a partitioning prior and does not enable streaming data. We have included both the recommended separate likelihoods, which models comparisons differently for each pair of files, and a single likelihood version (Single Likelihood), which is more analogous

to the model presented in Section 3.2.3. Blink and SVM are both deduplication models, and so may link records within the same file. Where possible, we have chosen default or recommended values for tuning and hyperparameters in these comparison models. Further details about the comparison models can be found in Appendix B.4.1.

We compare the accuracy of the resulting links by examining the posterior distribution of the F_1-score, $F_1 = 2(\text{recall}^{-1}+\text{precision}^{-1})^{-1}$ (Blair, 1979). Recall is the proportion of true coreferent record pairs that are correctly identified, and precision is the proportion of identified coreferent pairs that are true duplicates. Table 3.2 shows these posterior distributions as means and standard deviations of posterior samples drawn from each model, after discarding burn-in. We also evaluate the models through the posterior distribution of the number of estimated distinct entities across all files in Figure 3.2. Because the SVM may result in non-transitive links, we consider only the accuracy of the link labels for this method rather than number of estimated entities. The model presented in this chapter performs as well or better than the comparison models using both metrics. Additional error levels are included in the supplemental material.

Overall the link accuracy of our model is comparable to the comparison models. In all but one case (90% overlap and 6 errors) our proposed model has the highest F_1-score, and in that case our model's F_1-score is close to the best-performing comparison model. As expected, performance is generally worse for all models in scenarios with fewer duplicates and more errors in the duplicates. We would hesitate to generalize these comparison results to other scenarios, particularly because two comparison models (Blink, SVM) allow for duplicates within files which are not present in this simulated data. Additionally, the SVM method relies on having training data, which is not always available and expensive to produce, while the proposed model is fully unsupervised. With higher amounts of error and low overlap, the strength of the prior on $\boldsymbol{m}$ can be used to compensate for a lack of clean identifying information. We see in these cases, that the Strong Prior model outperforms the Weak and Flat Prior models, even though the strong prior is slightly misspecified for higher error cases. Similar prior information may be provided for the other Bayesian comparison models (Blink, Multilink), which may also improve their performance in these more difficult cases.

Table 3.2: Posterior means and standard deviations of F_1-score for simulated datasets. Within rows, each model is listed: the model presented in this chapter (Streaming) and three comparison models. Larger values represent more accurate links in the posterior distribution. The support vector machine, a non-bayesian method, is represented only by the F_1-score of its resulting point estimate.

Model	10% overlap	30% overlap	50% overlap	90% overlap
Errors: 2				
Streaming (Flat Prior)	**0.992 (0.0054)**	**1.000 (0.0009)**	0.991 (0.0018)	0.990 (0.0000)
Streaming (Weak Prior)	**0.992 (0.0056)**	**1.000 (0.0009)**	**0.999 (0.0015)**	**1.000 (0.0000)**
Streaming (Strong Prior)	0.978 (0.0102)	0.999 (0.0022)	0.994 (0.0020)	**1.000 (0.0000)**
Multilink	0.985 (0.0089)	0.996 (0.0041)	0.985 (0.0019)	0.944 (0.0000)
Multilink (Single Likelihood)	0.991 (0.0047)	0.999 (0.0016)	0.994 (0.0015)	0.992 (0.0000)
Blink	0.578 (0.0165)	0.974 (0.0021)	0.993 (0.0005)	0.996 (0.0004)
SVM (1% training)	0.962	**1.000**	0.986	0.999
Errors: 4				
Streaming (Flat Prior)	0.979 (0.0123)	0.957 (0.0067)	0.974 (0.0036)	0.997 (0.0001)
Streaming (Weak Prior)	**0.981 (0.0107)**	0.971 (0.0072)	**0.986 (0.0034)**	**0.998 (0.0001)**
Streaming (Strong Prior)	0.978 (0.0101)	**0.976 (0.0052)**	**0.986 (0.0036)**	**0.998 (0.0001)**
Multilink	0.161 (0.0038)	0.640 (0.0402)	0.982 (0.0048)	0.978 (0.0015)
Multilink (Single Likelihood)	0.913 (0.0283)	0.960 (0.0092)	0.983 (0.0035)	0.997 (0.0004)
Blink	0.504 (0.0117)	0.887 (0.0065)	0.962 (0.0043)	0.994 (0.0011)
SVM (1% training)	0.933	0.827	0.919	0.947
Errors: 6				
Streaming (Flat Prior)	0.227 (0.0073)	0.797 (0.0200)	0.952 (0.0071)	0.993 (0.0016)
Streaming (Weak Prior)	0.808 (0.0592)	0.910 (0.0157)	**0.954 (0.0065)**	0.977 (0.0011)
Streaming (Strong Prior)	**0.896 (0.0180)**	**0.929 (0.0103)**	0.952 (0.0054)	0.983 (0.0012)
Multilink	0.064 (0.0013)	0.482 (0.0118)	0.822 (0.0263)	0.985 (0.0017)
Multilink (Single Likelihood)	0.064 (0.0021)	0.393 (0.0151)	0.913 (0.0147)	**0.997 (0.0012)**
Blink	0.456 (0.0127)	0.803 (0.0092)	0.910 (0.0058)	0.986 (0.0022)
SVM (1% training)	0.674	0.668	0.707	0.675

Each Bayesian model was run using 3 different random seeds and all exhibited some multimodality in higher overlap cases where links are more constrained, particularly those with duplicate-free file constraints (Streaming, Multilink).

3.4.3 Speed

Our streaming samplers from Section 3.3 more efficiently produce samples from the model's posterior distribution. We demonstrate this improved efficiency by recording the amount of time required by each sampler to produce an effective sample size of 1000. For each of the 16 simulated data sets, five samplers were used to sample from the posterior distribution of $\boldsymbol{m}, \boldsymbol{u}, \boldsymbol{Z}^{(1)}, \boldsymbol{Z}^{(2)}, \boldsymbol{Z}^{(3)} | \boldsymbol{\Gamma}^{(1)}, \boldsymbol{\Gamma}^{(2)}, \boldsymbol{\Gamma}^{(3)}$. We compared PPRB-within-Gibbs using locally balanced

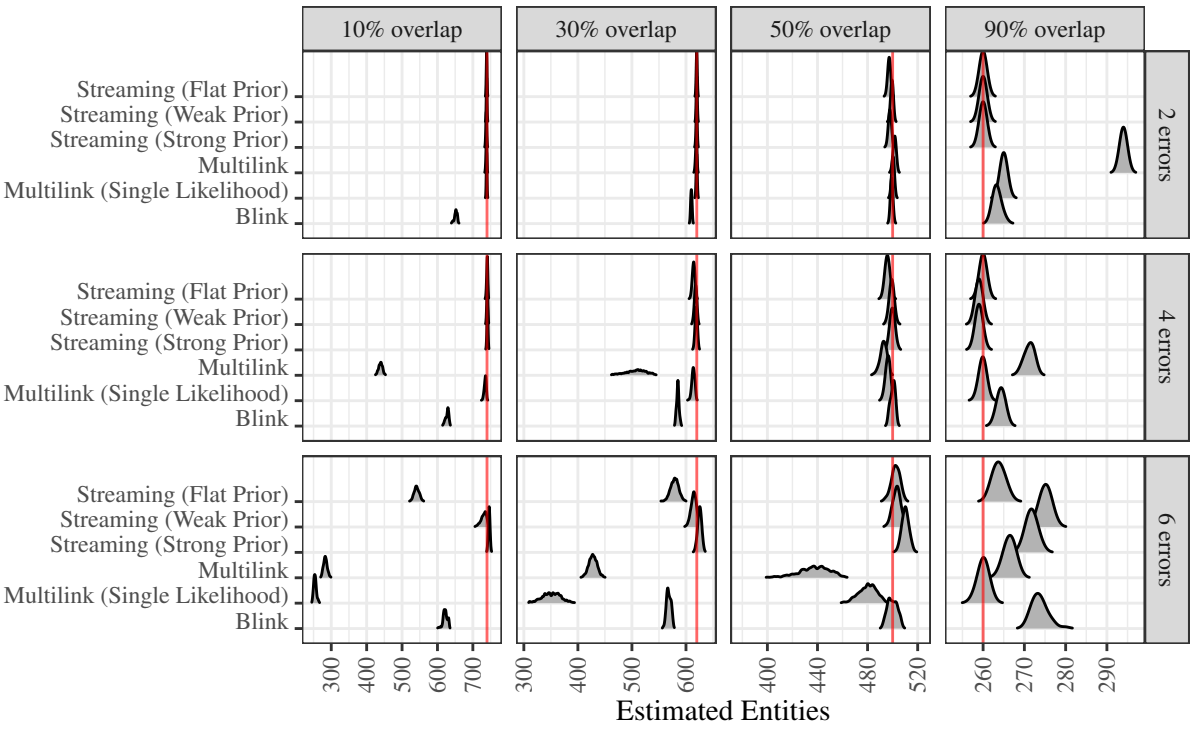


Figure 3.2: Posterior distribution of the number of estimated entities for simulated datasets. A vertical line indicates the true number of distinct entities in each dataset. Compared models are on the y-axis: the model presented in this chapter (Streaming) and three comparison models.

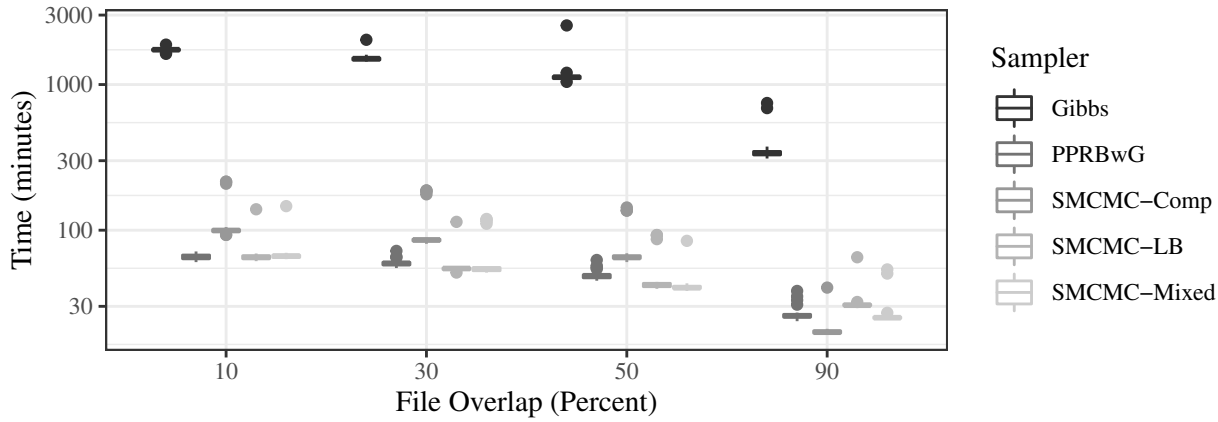


Figure 3.3: Time required for each sampler to produce an effective sample size of 1000. The effective sample size is measured on the continuous parameters, $\boldsymbol{m}$ and $\boldsymbol{u}$. Lower values indicate more efficient sampling. The SMCMC sampling time is estimated assuming 1000 available cores so that each ensemble member can be updated in parallel.

$\boldsymbol{Z}^{(3)}$ updates (PPRBwG), SMCMC with locally balanced proposals for both jumping and transition kernels (SMCMC-LB), SMCMC with component-wise full conditional draws for both jumping and transition kernels (SMCMC-Comp), SMCMC with component-wise full conditional draws for the jumping kernel and locally balanced proposals for the transition kernel (SMCMC-Mixed), and a non-streaming Gibbs sampler fit to the full data using the sampler in Definition B.3.1 in Appendix B.3.2 (Gibbs). All streaming samplers used the `BRL` package (Sadinle, 2017) to sample from the bipartite record linkage posterior distribution, $\boldsymbol{m}, \boldsymbol{u}, \boldsymbol{Z}^{(1)} | \boldsymbol{\Gamma}^{(1)}$. More details about these simulations are in Appendix B.4.2.

We choose effective sample size to capture both the number of samples produced in a given time and their quality. To summarize the effective sample size of each run, we calculate the effective sample size of each component of the continuous parameters $\boldsymbol{m}$ and $\boldsymbol{u}$, and find the median across all values. Since SMCMC produces independent samples, the effective sample size of any parameter is equal to the size of the SMCMC ensemble. The three SMCMC methods are assumed to be run fully parallel, where the samples produced are not limited by time but by available computational resources. The streaming samplers take an order of magnitude less time to obtain 1000 effective samples than the non-streaming sampler (Figure 3.3). With fewer cores available the time advantage for SMCMC will not be as stark, however there is still a benefit with as few as 36 cores.

As the number of records in each file, $n_1, \ldots, n_k$, grows, the time required by each component-wise $\boldsymbol{Z}^{(k-1)}$ full conditional update will grow quadratically because it iterates through every combination of a record in file X_k and a record in all previous files, $n_k \cdot \sum_{\ell=1}^{k-1} n_\ell$ total pairs of records. This will affect the time of any sampler using component-wise full conditional updates. The time for locally balanced proposals, if blocked, does not grow with the number of records per file. However the smaller the block size becomes relative to the file size, the less effective blocked locally balanced proposals will be at exploring the parameter space. As the number of files, k, grows, the time required by each component-wise $\boldsymbol{Z}^{(k-1)}$ full conditional update will grow linearly since the number of records in file k does not increase, only the total number of records in previous files. The time for locally balanced proposals, if blocked, does not grow with the number of files.

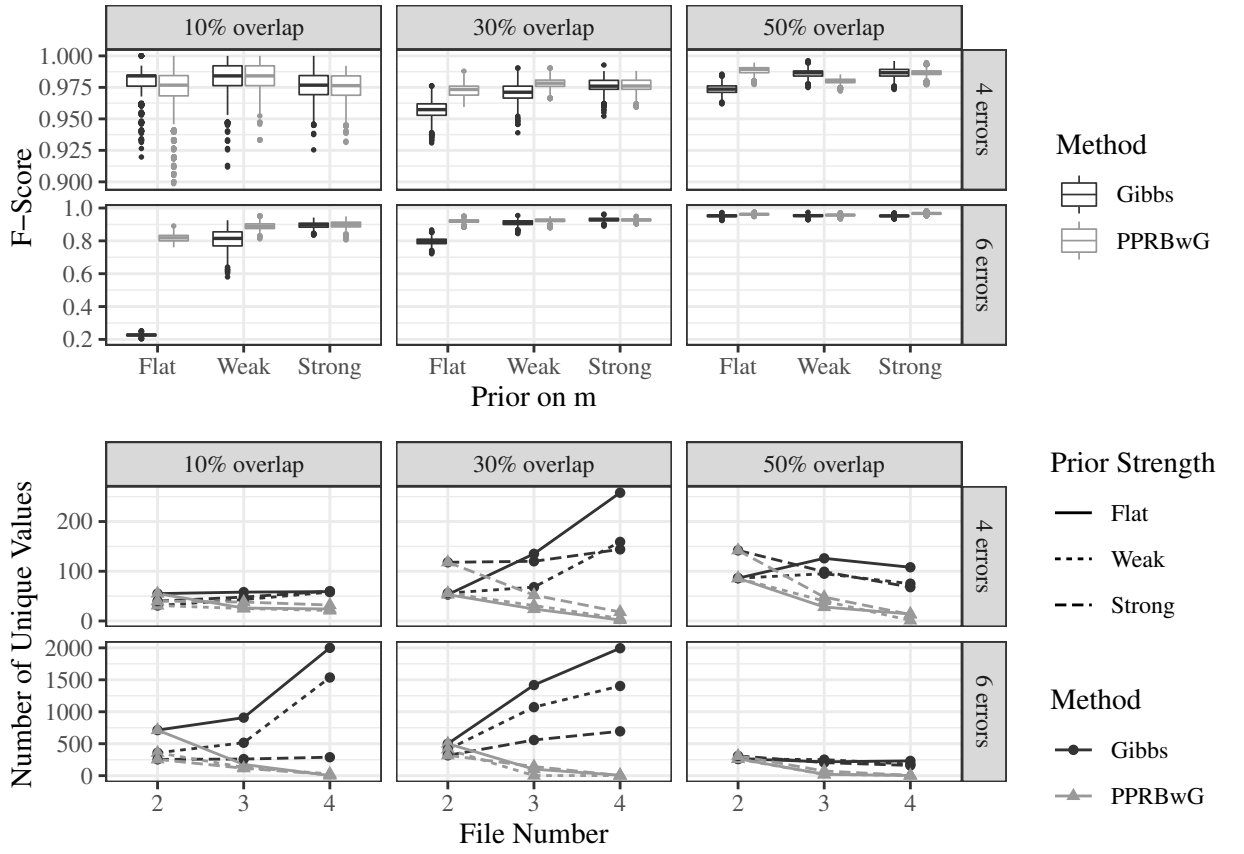


Figure 3.4: Demonstrations of PPRB-within-Gibbs sample degradation. The data scenarios in which the difference in distinct values is most visible are shown. TOP: Posterior F1-score for PPRB-within-Gibbs and non-streaming samplers. On the x-axis are different strengths of prior distribution on the parameter $\boldsymbol{m}$. In some datasets, PPRB-within-Gibbs appears to produce different posterior distributions than the non-streaming sampler. BOTTOM: Number of distinct values of $\boldsymbol{Z}^{(1)}$ produced by PPRB-within-Gibbs and Gibbs, out of 2000 iterations. Lines connect points with the same prior information for $\boldsymbol{m}$.

A growing number of files will also increase the time required by SMCMC as more full conditional updates will be required per iteration of the transition kernel. The time required for the transition kernel will grow at most quadratically with increasing k because a linear series of new full conditional updates are required which are themselves require at most linearly increasing time with k. As k increases, the amount of time required for PPRB-within-Gibbs is not affected unless using component-wise full conditional updates for $\boldsymbol{Z}^{(k-1)}$ or also increasing the locally balanced proposal block-size.

3.4.4 PPRB Degeneracy

As is true of all filtering methods, PPRB and PPRB-within-Gibbs have the undesirable property that the pool of samples for any $\boldsymbol{Z}^{(m)}$ will converge to a degenerate distribution as $k \to \infty$. We see an example of this phenomenon in Figure 3.4, particularly for overlaps of 50% or less and 4 or more errors, where the posterior distribution of F_1-score from PPRB-within-Gibbs differs from the other samplers. For 10% overlap, 4 errors, and a flat prior on $\boldsymbol{m}$, we even see a very large difference between PPRB-within-Gibbs and the non-streaming sampler. To investigate further, we compare the samples produced from PPRB-within-Gibbs to those produced from a non-streaming (Gibbs) sampler. In 4-file record linkage, PPRB-within-Gibbs produces noticeably fewer unique values of $\boldsymbol{Z}^{(1)}$ than Gibbs for the same number of posterior samples. This indicates that degradation is occurring due to the filtering of the initial pool of samples from two sequential PPRB-within-Gibbs updates. As more files are added and the pool of $\boldsymbol{Z}^{(1)}$ samples is further filtered, this contrast will become more apparent, eventually leading to a single value of $\boldsymbol{Z}^{(1)}$ being sampled.

3.5 Real Data Application

We now apply streaming record linkage to a sample of records from a longitudinal survey with a known true identity for each record. The Social Diagnosis Survey (SDS) of quality of life in Poland (Czapinski and Panek, 2015) is a biennial survey of households that was first conducted in the year 2000. Individuals may be recorded multiple times in separate years but there is no duplication of individuals within a year. Four files of data were selected from the full dataset from the years 2007 through 2013. The four files have varying sizes, with $n_1 = 151$, $n_2 = 464$, $n_3 = 688$, and $n_4 = 677$, for a total of 1980 records. The files were created by randomly sampling, without replacement, 910 individuals from all individuals appearing in at least one of the included years. Of the 910 individuals, 306 appear in just one file, 240 appear in two files, 262 appear in three files, and 102 appear in all four files.

Linkage was performed using six fields: gender, province, educational attainment, and year, month, and day of birth. All fields are categorical and were compared using binary comparisons.

Table 3.3: Posterior means and standard deviations of F_1-score and estimated number of entities, and total sampling time, for the four-file Poland SDS data set using five samplers. There are 910 true entities in the four files. Sampling time is given in cumulative hours required to produce posterior samples of the parameters conditioned first on three files, then on four files using each sampling method. The SMCMC sampling time is estimated assuming 1000 available cores so that each ensemble member can be updated in parallel.

Sampler	F1-Score	Estimated Entities	Sampling Time
Gibbs	0.985 (9e-04)	915 (1.6)	121.1
PPRBwG	0.992 (0.0010)	915 (2.0)	10.9
SMCMC-Comp	0.992 (0.0012)	916 (1.9)	3.5
SMCMC-LB	0.99 (0.0022)	916 (1.9)	6.9
SMCMC-Mixed	0.992 (0.0010)	916 (1.8)	3.5

We chose hyperparameters to produce flat priors in $\boldsymbol{m}$, $\boldsymbol{u}$, and $\boldsymbol{Z}^{(\ell)}$ for $\ell = 1, 2, 3$. We compared five samplers: a non-streaming Gibbs sampler (Gibbs), sequentially applied PPRB-within-Gibbs updates with locally balanced proposals (PPRBwG), and sequentially applied SMCMC updates with component-wise proposals (SMCMC-Comp), locally balanced proposals (SMCMC-LB) or a mix using component-wise jumping kernel proposals and locally balanced transition kernel proposals (SMCMC-Mixed). More details of the MCMC runs can be found in Appendix B.4.3.

The streaming record linkage models were able to recover the true coreferent records with high accuracy. Table 3.3 shows the posterior F_1-score distribution for each of the 5 samplers, the posterior distribution of the estimated number of entities resulting from the linkage, and the time to generate the posterior samples. All samplers performed equally well at recovering the true coreferent record sets with a posterior mean F_1-score between 0.985 and 0.992. Streaming samplers were significantly faster than the non-streaming Gibbs sampler, with times given for the cumulative time required to produce both three-file and four-file inference using each sampling method. This is representative of the streaming data setting where inference is required after each new file arrives. The streaming samplers show between 11 times and 35 times speedup when compared to the non-streaming Gibbs sampler, where SMCMC time estimates are based on the assumption that enough cores are available for each ensemble to be run simultaneously in parallel.

3.6 Discussion

In this chapter we have introduced a model for multi-file Bayesian record linkage based on the Fellegi-Sunter paradigm that is appropriate for streaming data contexts. We have shown this model to work as well as comparison models on realistic simulated data at varying amounts of duplication and error. With this work, we have proposed the first model-based streaming record linkage procedures that update inference on existing parameters and estimate new parameters as new data arrives. Our model provides interpretable parameters for estimating not only links between records, but the probability of different levels of error between fields of coreferent records. These streaming samplers allow for near-identical inference to the model fit using the full data. Having two distinct streaming options for this model allows for the selection of one based on the needs of the user, and we have detailed the trade-offs that one might consider. We have demonstrated that these streaming samplers can provide significant computational gains when compared to a Gibbs sampler using both simulated and real-world data.

Our simulation study shows a noticeable effect of the strength of the prior on $\boldsymbol{m}$ on the accuracy of the resulting posterior samples. In Section 3.2.4 we describe a way to use the prior on $\boldsymbol{m}$ to incorporate prior knowledge about the probability of errors in duplicated fields, and in Section 3.4.2 we suggest how this can be used to compensate for a lack of clean identifying fields in each record. The priors on $\boldsymbol{Z}^{(1:(k-1))}$ can also be tuned through the values of α_π and β_π, but practitioners are unlikely to have prior knowledge about the level of overlap between files.

The scalability of this model to files with very large numbers of records could be limited in two ways. First, the dimension of the model's parameter space grows directly with the number of records included. A larger parameter space requires both larger storage for posterior samples and slower computation of the transition kernel. A very large file also poses difficulties for the computation of comparisons. With the arrival of a new file, a comparison vector needs to be computed comparing each record in the new file to each record in previous files. These challenges with large files could be mitigated by blocking to prohibit links across large time differences or breaking large files into several smaller files.

Future work in streaming record linkage includes relaxing the assumption of no duplicates within files to develop an entity resolution model that can identify duplicates both within and between files in a streaming context. Further streaming sampling methods may be explored by combining techniques of PPRB-within-Gibbs and SMCMC into a streaming sampler with more of the strengths of both methods: the ease of computation and low data storage demands of PPRB with the non-degenerate sampling of SMCMC.

Chapter 4

Generative Filtering for Recursive Bayesian Inference with Streaming Data

4.1 Introduction

Modern data collection has given rise to the streaming setting, where data arrives continuously or in frequent batches. In a typical analysis, when a pre-determined amount of data is collected, estimates of model parameters can be produced in one offline procedure. However, in the streaming data setting, estimates of model parameters may be desired after each arrival of new data. This setting poses a computational challenge as the data model becomes more complex with each new arrival and producing model estimates is increasingly more time-consuming.

In Bayesian statistics, parameters are assumed to be random variables with probability distributions (priors), and the data are assumed to have come from distributions conditioned on the parameter values. The parameters are estimated using the distribution of the parameters conditioned on the values of the data (posterior). Because these posterior distributions are frequently intractable, they are typically approximated using samples produced by Markov chain Monte Carlo (MCMC, Gelfand and Smith, 1990). MCMC can be computationally intensive, especially if a model is complex or has strong dependencies among the parameters. Sampling techniques such as Hamiltonian Monte Carlo (Duane et al., 1987) or the No-U-Turn Sampler (Hoffman and Gelman, 2014) more efficiently sample from the posterior distribution. However even with efficient sampling, offline MCMC begins from scratch each time new data arrives. This frequent restarting in the streaming setting ignores the previous parameter estimates, which may be helpful to more efficiently produce inference.

Approximate methods such as Variational Bayes (Blei et al., 2017) or Approximate Bayesian Computation (Tavaré et al., 1997) provide Bayesian inference by approximating the posterior dis-

tribution analytically instead of sampling from the exact posterior distribution. These methods improve efficiency for intractable posterior distributions, and while the posterior means may be captured by the approximation, the entire shape of the posterior distribution is not. A practitioner may not be willing to make this trade-off between efficiency and inference. Additionally, these methods require the selection of a family of approximate posteriors (in the case of Variational Bayes) or summary statistics of data (in the case of Approximate Bayesian Computation) for accurate approximation of the posterior, which may not be obvious.

Recursive Bayesian updates have been proposed to leverage the streaming setting, in which the posterior distribution of one analysis is used as the prior for a subsequent analysis (Särkkä, 2013). In the streaming setting, the posterior after the arrival of data up to time t is used as the prior when the next data arrives at time $t + 1$. When models are designed with conjugate prior distributions, recursive Bayesian updates can be performed analytically. Otherwise, samples approximating the previous posterior distribution can be used to inform the next analysis. Methods that resample existing samples from time t in such a way to approximate the posterior distribution at time $t + 1$ include sequential importance sampling (e.g., Hendry and Richard, 1992; Liu and Chen, 1995), the Bootstrap filter (Gordon et al., 1993), and Monte Carlo filtering (Kitagawa, 1996). Because these methods refine existing samples, we refer to them as "filtering" methods. Filtering methods are desirable for their speed but suffer from eventual degeneracy as samples are filtered. Sequential Monte Carlo (SMC, Gordon et al., 1993) methods apply importance sampling to a particular class of state space models which may not be applicable to a given problem. We focus on two MCMC strategies for recursive Bayesian updates: Sequential MCMC (SMCMC, Yang and Dunson, 2013) and Prior-Proposal-Recursive Bayes (PPRB, Hooten et al., 2021). SMCMC uses each existing sample as an initial value for an independent Markov chain targeting the updated posterior. PPRB is a multistage MCMC method that filters existing samples through independent Metropolis-Hastings (MH) steps based on new data.

In this chapter, we propose Generative Filtering, a new method for recursive Bayesian inference in a streaming data setting that retains the speed benefits of filtering while avoiding the associated

sample degradation. In Section 4.2 we derive theoretical bounds for the approximation error introduced through successive applications of PPRB. In Section 4.3 we define Generative Filtering and describe Generative Filtering's relationship to the existing methods of SMCMC and PPRB. We also provide theoretical upper bounds on the convergence of Generative Filtering to the target posterior and derive conditions to reduce data storage requirements. In Section 4.4 we apply Generative Filtering to two numerical examples and evidence the inferential and computational benefits. In Section 4.5 we apply Generative Filtering to a publicly available species survey to estimate the population intensity of sea lion pups in Alaska. We next provide necessary background information and notation that ground our development of Generative Filtering.

4.1.1 Sequential Markov Chain Monte Carlo

Sequential MCMC (SMCMC) is an algorithm to sample from a sequence of probability distributions, corresponding to posterior distributions at different times in streaming data contexts (Yang and Dunson, 2013). Consider a model,

$$\boldsymbol{y}_1 \sim p(\boldsymbol{y}_1|\boldsymbol{\theta}), \quad \boldsymbol{y}_2 \sim p(\boldsymbol{y}_2|\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{y}_1), \quad \boldsymbol{\theta} \sim p(\boldsymbol{\theta}), \quad \boldsymbol{\phi} \sim p(\boldsymbol{\phi}|\boldsymbol{\theta}), \tag{4.1}$$

where $\boldsymbol{y}_1$ is available first and $\boldsymbol{y}_2$ is available later. The parameters, $\boldsymbol{\theta}$, encode the distribution of all data, while the parameters, $\boldsymbol{\phi}$, encode only the distribution of $\boldsymbol{y}_2$. SMCMC operates on an ensemble of samples, $\{\boldsymbol{\theta}_s\}_{s=1}^S$, from the posterior distribution, $p(\boldsymbol{\theta}|\boldsymbol{y}_1)$. First, a jumping kernel is applied to each sample $\boldsymbol{\theta}_s$ in parallel to append a value $\boldsymbol{\phi}_s$. Then, in parallel with each pair $(\boldsymbol{\theta}_s, \boldsymbol{\phi}_s)$ as an initial value, a transition kernel is applied m_t times. The transition kernel targets the posterior distribution $p(\boldsymbol{\theta}, \boldsymbol{\phi}|\boldsymbol{y}_1, \boldsymbol{y}_2)$. After m_t iterations, the final value of each of the S parallel chains is saved and comprise a sample to approximate the posterior distribution $p(\boldsymbol{\theta}|\boldsymbol{y}_1, \boldsymbol{y}_2)$.

4.1.2 Prior-Proposal-Recursive Bayes

Prior-Proposal-Recursive Bayes (PPRB) is a method for performing recursive Bayesian updates using existing samples from the previous posterior distribution (Hooten et al., 2021). Con-

sider the general model in Eq. (4.1), but simplified so that there is no parameter ϕ and both data distributions are characterized by $\boldsymbol{\theta}$. In PPRB the posterior samples, $\{\boldsymbol{\theta}_s\}_{s=1}^S$, from $p(\boldsymbol{\theta}|\boldsymbol{y}_1)$ are used as independent MH proposals in an MCMC to sample from the target $p(\boldsymbol{\theta}|\boldsymbol{y}_1, \boldsymbol{y}_2)$, where the MH acceptance ratio for a proposal $\boldsymbol{\theta}^*$ and a current value $\boldsymbol{\theta}$ simplifies to

$$\alpha = \min\left(\frac{p(\boldsymbol{y_2}|\boldsymbol{\theta}^*, \boldsymbol{y}_1)}{p(\boldsymbol{y_2}|\boldsymbol{\theta}, \boldsymbol{y}_1)}, 1\right).$$

One limitation of this method is its inability to account for a changing parameter space in a Bayesian update, represented in Equation (4.1) by the parameter $\boldsymbol{\phi}$. Hooten et al. (2021) propose first drawing $\boldsymbol{\phi}_s$ from the predictive distribution $p(\boldsymbol{\phi}|\boldsymbol{\theta}_s, \boldsymbol{y}_1)$, to produce augmented samples $\{(\boldsymbol{\theta}_s, \boldsymbol{\phi}_s)\}_{s=1}^S$ which are from $p(\boldsymbol{\theta}, \boldsymbol{\phi}|\boldsymbol{y}_1)$. Then with these samples as proposals, proceed with PPRB as usual. In Chapter 3, we note that this sample augmentation approach is problematic in situations where the predictive distribution $p(\boldsymbol{\phi}|\boldsymbol{\theta}_s, \boldsymbol{y}_1)$ is diffuse, because values drawn from $p(\boldsymbol{\phi}|\boldsymbol{\theta}_s, \boldsymbol{y}_1)$ are poor proposals for the full conditional distribution $p(\boldsymbol{\phi}|\boldsymbol{\theta}_s, \boldsymbol{y}_1, \boldsymbol{y}_2)$, leading to low acceptance rates of the PPRB independent MH proposals. This problem arises, for example, if the prior $p(\boldsymbol{\phi}|\boldsymbol{\theta}) = p(\boldsymbol{\phi})$ has no dependence on $\boldsymbol{\theta}$. The authors propose PPRB-within-Gibbs as an adaptation to PPRB.

Definition 4.1.1. PPRB-within-Gibbs. Consider the streaming model defined in Eq. (4.1). Let there be existing posterior samples, $\{\boldsymbol{\theta}^s\}_{s=1}^S$ from the distribution $p(\boldsymbol{\theta}|\boldsymbol{y}_1)$. Then for the desired number of posterior samples,

1. (PPRB step) Propose a new value $\boldsymbol{\theta}^*$ by drawing from the existing posterior samples $\{\boldsymbol{\theta}^s\}_{s=1}^S$ with replacement. Accept or reject the proposal using the MH ratio

$$\alpha = \min\left(\frac{p(\boldsymbol{y}_2|\boldsymbol{y}_1, \boldsymbol{\theta}^*, \boldsymbol{\phi})}{p(\boldsymbol{y}_2|\boldsymbol{y}_1, \boldsymbol{\theta}, \boldsymbol{\phi})}\frac{p(\boldsymbol{\phi}|\boldsymbol{\theta}^*)}{p(\boldsymbol{\phi}|\boldsymbol{\theta})}, 1\right), \tag{4.2}$$

2. Update the parameter $\boldsymbol{\phi}$ from the full conditional distribution $p(\boldsymbol{\phi}|\boldsymbol{\theta}, \boldsymbol{y}_1, \boldsymbol{y}_2)$.

The PPRB step acceptance ratio (Eq. 4.2) is now the product of two ratios: the ratio of the distribution of the new data, and the ratio of the conditional prior of the new parameters, $\boldsymbol{\phi}$. If the prior of $\boldsymbol{\phi}$ is not dependent on $\boldsymbol{\theta}$, i.e., $p(\boldsymbol{\phi}|\boldsymbol{\theta}) = p(\boldsymbol{\phi})$, then this second ratio cancels. Further, if the data $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ are conditionally independent, the first ratio does not depend on $\boldsymbol{y}_1$. In Chapter 3 we proposed a three-step variant with a separate update for components within $\boldsymbol{\theta}$ with conjugate full conditional updates. In this Chapter, unless otherwise noted, we refer to the two-step version in Definition 4.1.1 as PPRB-within-Gibbs.

4.2 Filtering Degradation

Degradation refers to the tendency for MCMC samples resulting from filtering methods to contain many repeated values due to rejected proposals. As a result, when using existing samples resampled with replacement as proposals for the next update (as suggested in Hooten et al., 2021), the number of unique values decreases for continuous distributions. Due to degradation, the performance of filtering methods suffer for the streaming data setting, where a potentially large number of Bayesian updates must be performed while resampling the same pool of samples. As the number of unique values decreases within a set number, S, of posterior samples produced after each update, the ability of those S samples to approximate the posterior distribution degrades. We next give an intuitive explanation for the causes of filtering degradation, provide theoretical bounds for the error introduced, and demonstrate filtering degradation via simulation in Section 4.4, all for the PPRB algorithm.

Each application of PPRB to perform a streaming update introduces error in the approximation to the updated posterior. To see this, consider a simple model with parameters $\boldsymbol{\theta}$ and data partitioned in time, $\boldsymbol{y}_t$ for $t = 1, 2, \ldots$. Samples $\{\boldsymbol{\theta}_s^{(1)}\}_{s=1}^S$ are drawn from $p(\boldsymbol{\theta}|\boldsymbol{y}_1)$. At time $t = 2$, $\boldsymbol{y}_2$ are available and the posterior $p(\boldsymbol{\theta}|\boldsymbol{y}_1, \boldsymbol{y}_2)$ must be approximated. By resampling $\{\boldsymbol{\theta}_s^{(1)}\}_{s=1}^S$ with replacement as PPRB proposals, the true proposal distribution is the empirical posterior distribution of these samples, which we call $F_S^{(1)}(\cdot)$. During the first PPRB update at $t = 2$, the target

distribution is thus

$$p(\boldsymbol{y}_2|\boldsymbol{\theta}, \boldsymbol{y}_1)F_S^{(1)}(\boldsymbol{\theta}) \approx p(\boldsymbol{y}_2|\boldsymbol{\theta}, \boldsymbol{y}_1)p(\boldsymbol{\theta}|\boldsymbol{y}_1) \propto p(\boldsymbol{\theta}|\boldsymbol{y}_1, \boldsymbol{y}_2),$$

an approximation of the desired posterior at $t = 2$. S samples, $\{\boldsymbol{\theta}_s^{(2)}\}_{s=1}^S$, are drawn from this approximate posterior, which yields another empirical posterior distribution $F_S^{(2)}(\cdot)$. During the second PPRB update at $t = 3$, resampling $\boldsymbol{\theta}_s^{(2)}$ with replacement as proposals targets

$$\begin{aligned} p(\boldsymbol{y}_3|\boldsymbol{\theta}, \boldsymbol{y}_1, \boldsymbol{y}_2)F_S^{(2)}(\boldsymbol{\theta}) &\approx p(\boldsymbol{y}_3|\boldsymbol{\theta}, \boldsymbol{y}_1, \boldsymbol{y}_2)p(\boldsymbol{y}_2|\boldsymbol{\theta}, \boldsymbol{y}_1)F_S^{(1)}(\boldsymbol{\theta}) \\ &\approx p(\boldsymbol{y}_3|\boldsymbol{\theta}, \boldsymbol{y}_1, \boldsymbol{y}_2)p(\boldsymbol{y}_2|\boldsymbol{\theta}, \boldsymbol{y}_1)p(\boldsymbol{\theta}|\boldsymbol{y}_1) \propto p(\boldsymbol{\theta}|\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{y}_3), \end{aligned}$$

an approximation of an approximation of the desired posterior at $t = 3$.

Intuitively, as this process repeats in subsequent updates, an accumulation of approximation error occurs. Whatever acceptable level of approximation for using PPRB once is surpassed in using PPRB a second time or beyond.

4.2.1 Bounds on PPRB Approximation Error

We derive upper and lower bounds on the PPRB approximation error in a streaming setting where PPRB is applied sequentially by resampling the samples produced at a previous stage as proposals in the next stage. We start by defining notation,

$$\pi_t = p(\boldsymbol{\theta}|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_t), \text{ true posterior at time } t, \tag{4.3}$$

$$A_t \propto p(\boldsymbol{y}_t|\boldsymbol{\theta}, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{t-1})F_S^{(t-1)}(\boldsymbol{\theta}), \text{ approximate PPRB posterior at time } t, \tag{4.4}$$

$$F_S^{(t)} = \text{empirical distribution of } S \text{ samples drawn from } A_t. \tag{4.5}$$

The quantity of interest is then $\|A_t - \pi_t\|$, where $\|\cdot\|$ is a norm on probability distributions. The following upper and lower bounds exist for this quantity.

Theorem 4.2.1. *Let π_t, A_t, and $F_S^{(t)}$ be defined as in Eq. (4.3)-(4.5) and let $\|\cdot\|$ be a norm on probability measures that has a triangle inequality. Then,*

$$\|A_t - \pi_t\| \leq \underbrace{\|A_{t-1} - \pi_{t-1}\|}_{(1)} + \underbrace{\left\|F_S^{(t-1)} - A_{t-1}\right\|}_{(2)} + \underbrace{\|\pi_t - \pi_{t-1}\|}_{(3)} + \underbrace{\left\|A_t - F_S^{(t-1)}\right\|}_{(4)} \tag{4.6}$$

$$\|A_t - \pi_t\| \geq \left| \underbrace{\|A_{t-1} - \pi_{t-1}\|}_{(1)} - \underbrace{\left\|F_S^{(t-1)} - A_{t-1}\right\|}_{(2)} \right| - \underbrace{\|\pi_t - \pi_{t-1}\|}_{(3)} - \underbrace{\left\|A_t - F_S^{(t-1)}\right\|}_{(4)} \tag{4.7}$$

Proof. Appendix C.1 □

The form of these inequalities has the advantage that all terms on the right hand sides of Eq. (4.6) and Eq. (4.7) are easily interpreted in the context of streaming Bayesian updates:

(1) $\|A_{t-1} - \pi_{t-1}\|$ is the existing approximation error at time $t-1$.

(2) $\left\|F_S^{(t-1)} - A_{t-1}\right\|$ is the error introduced by producing a finite number, S, of MCMC samples from the approximate posterior.

(3) $\|\pi_t - \pi_{t-1}\|$ is the difference in the true posterior after receiving $\boldsymbol{y}_t$.

(4) $\left\|A_t - F_S^{(t-1)}\right\|$ is the difference in approximate distribution, from the prior $F_S^{(t-1)}$ to the posterior A_t after receiving $\boldsymbol{y}_t$.

Together, Equations (4.6) and (4.7) give upper and lower bounds for the approximation error at time t, $\|A_t - \pi_t\|$, which are especially useful for large t. As $t \to \infty$, the proportion of all data $(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_t)$ contained in $\boldsymbol{y}_t$ becomes smaller, so we have the intuition that the Bayesian updates will be smaller, in the sense that the difference between the prior and posterior goes to zero. Thus, $\|\pi_t - \pi_{t-1}\| \to 0$ and $\left\|A_t - F_S^{(t-1)}\right\| \to 0$ are reasonable to assume as $t \to \infty$. Then $\|A_t - \pi_t\|$ is approximately bounded by

$$\|A_t - \pi_t\| \lessapprox \|A_{t-1} - \pi_{t-1}\| + \left\|F_S^{(t-1)} - A_{t-1}\right\| \tag{4.8}$$

$$\|A_t - \pi_t\| \gtrapprox \left| \|A_{t-1} - \pi_{t-1}\| - \left\|F_S^{(t-1)} - A_{t-1}\right\| \right|. \tag{4.9}$$

For large t, the inequalities in Equations (4.8) and (4.9) show that the PPRB error at t is approximately bounded by a triangle inequality with $\|A_{t-1} - \pi_{t-1}\|$, the approximation error at $t-1$, and $\left\|F_S^{(t-1)} - A_{t-1}\right\|$, the finite sample error at $t-1$. If there exists some $\epsilon > 0$ such that $\left\|F_S^{(t)} - A_t\right\| > \epsilon$ for all t, then the approximation error cannot decay to zero as $t \to \infty$. For large t, if $\|A_t - \pi_t\| < \epsilon/2$ then $\|A_{t+1} - \pi_{t+1}\| > \epsilon/2$. Similar results apply to PPRB-within-Gibbs, see Appendix C.2.

4.3 Generative Filtering

We introduce Generative Filtering, a novel MCMC sampler for recursive Bayesian updates which avoids filtering degeneracy and achieves faster convergence than SMCMC. Consider data, $\boldsymbol{y}$, partitioned into sequential batches, $\boldsymbol{y}_t$ for $t = 1, 2, \ldots$, and data model

$$\begin{aligned}
\boldsymbol{y}_1 &\sim p(\boldsymbol{y}_1|\boldsymbol{\theta}_1), \\
\boldsymbol{y}_t &\sim p(\boldsymbol{y}_t|\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_t, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{t-1}), \text{ for } t \geq 2, \\
\boldsymbol{\theta_1} &\sim p(\boldsymbol{\theta}_1), \\
\boldsymbol{\theta_t} &\sim p(\boldsymbol{\theta}_t|\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{t-1}), \text{ for } t \geq 2.
\end{aligned}$$

That is, each batch of data has a distribution which can depend on some set of parameters and previously arrived batches of data. The parameters, $\boldsymbol{\theta}$, are divided into batches, $\boldsymbol{\theta}_t$ for $t = 1, 2, \ldots$, such that the batch $\boldsymbol{y}_t$ of data depends only on the parameters $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_t$. Each batch of parameters has a prior that may depend on parameters in previous batches. For convenience, define $\boldsymbol{y}_{1:t} := (\boldsymbol{y}_1 \ \ldots \ \boldsymbol{y}_t)$ and $\boldsymbol{\theta}_{1:t} := (\boldsymbol{\theta}_1 \ \ldots \ \boldsymbol{\theta}_t)$. The Generative Filtering algorithm begins at time t, after the arrival of $\boldsymbol{y}_t$.

Definition 4.3.1. Generative Filtering.

Let there be samples $\{\boldsymbol{\theta}_{1:(t-1),s}\}_{s=1}^S$ from the posterior distribution $p(\boldsymbol{\theta}_{1:(t-1)}|\boldsymbol{y}_{1:(t-1)})$. Let T_t be a transition kernel targeting the updated posterior distribution, $p(\boldsymbol{\theta}_{1:t}|\boldsymbol{y}_{1:t})$. The Generative Filtering update consists of two steps:

1. Perform a filtering step to produce S samples, $\{\boldsymbol{\theta}^*_{1:t,s}\}_{s=1}^S$.
2. Using each $\{\boldsymbol{\theta}^*_{1:t,s}\}_{s=1}^S$ as an initial value, apply the transition kernel T_t in parallel m_t times in S parallel chains, saving the final value of each chain.

The resulting samples, $\{\boldsymbol{\theta}_{1:t,s}\}_{s=1}^S$ approximate the updated posterior distribution, $p(\boldsymbol{\theta}_{1:t}|\boldsymbol{y}_{1:t})$. The filtering method in step 1 is a method which resamples the ensemble $\{\boldsymbol{\theta}_{1:(t-1),s}\}_{s=1}^S$ to produce samples approximating the distribution, $p(\boldsymbol{\theta}_{1:t}|\boldsymbol{y}_{1:t})$. When the filtering method consists of a PPRB-within-Gibbs update, the Generative Filtering algorithm can be seen as an extension of both PPRB-within-Gibbs and SMCMC. It extends PPRB-within-Gibbs by applying some number of transition kernel steps in parallel to each resulting sample and extends SMCMC by replacing the jumping kernel with a PPRB-within-Gibbs update, which is able to update all parameters in the ensemble instead of just new parameters. Generative Filtering seeks to quickly converge to and sample from its target distribution through the use of existing posterior samples from an earlier posterior distribution, while avoiding the problem of filtering methods that degrade to degenerate distributions.

4.3.1 Convergence Results

We derive bounds for the convergence of Generative Filtering to its target posterior distribution and find sufficient conditions under which Generative Filtering has a stronger convergence guarantee than SMCMC in fewer iterations of the transition kernel.

Theorem 4.3.1. *Let $P_t^S(\boldsymbol{\theta}_{1:(t-1)}, \cdot)$ represent the kernel resulting from S applications of a filtering method at time t, which is a probability density for $\boldsymbol{\theta}_{1:t} := (\boldsymbol{\theta}_{1:(t-1)}, \boldsymbol{\theta}_t)$. Let $\pi_t = p(\boldsymbol{\theta}_{1:t}|\boldsymbol{y}_{1:t})$ be the target posterior at time t. Assuming the following conditions:*

1. *(Universal ergodicity) There exist $\rho_t \in (0, 1)$, such that for all $t > 0$ and $x \in \mathcal{X}$,*

$$||T_t(x, \cdot) - \pi_t||_1 \leq 2\rho_t.$$

2. *(Filtering consistency) For a sequence of $\lambda_t \to 0$ and a bounded sequence of positive integers S_t, the following holds:*

$$\sup_{\boldsymbol{\theta}_{1:(t-1)}} ||\pi_t - P_t^{S_t}(\boldsymbol{\theta}_{1:(t-1)}, \cdot)||_1 \leq 2\lambda_t.$$

Let $\epsilon_t = \rho_t^{m_t}$ and let $Q_t = T_t^{m_t} \circ P_t^{S_t}$ be a Generative Filtering update at time t. Then for any initial distribution π_0,

$$||Q_t \circ \cdots \circ Q_1 \circ \pi_0 - \pi_t||_1 \leq \sum_{v=1}^{t} \left\{ \prod_{u=v+1}^{t} \epsilon_u(1-\lambda_u) \right\} \epsilon_v \lambda_v \leq \sum_{v=1}^{t} \left\{ \prod_{u=v}^{t} \epsilon_u \right\} \lambda_v.$$

Remark. When PPRB-within-Gibbs is used as the filtering method within Generative Filtering, the Filtering consistency condition contains two key requirements related to PPRB-within-Gibbs and the posterior distributions, $\pi_1, \ldots, \pi_t$. First, the PPRB-within-Gibbs transition kernel must produce an irreducible Markov chain. A sufficient condition for irreducibility is the positivity critereon for PPRB-within-Gibbs from Chapter 3. Second, in order for the sequence of S_t to be bounded, the difference between subsequent posteriors must not be too large as t increases, or the PPRB-within-Gibbs transition kernel must have fast enough convergence to overcome the differences.

Proof. Appendix C.1 □

Comparison of upper bounds

We next compare the upper bounds for convergence of SMCMC and Generative Filtering, providing conditions for when Generative Filtering's upper bound is at least as small as that of SMCMC.

Theorem 4.3.2. *Assume the following conditions hold:*

1. *(Universal ergodicity) There exists $\epsilon \in (0, 1)$, such that for all $t > 0$ and $x \in \mathcal{X}$,*

$$||T_t(x, \cdot) - \pi_t||_1 \leq 2\rho_t.$$

2. *(Stationary convergence) The stationary distribution π_t of T_t satisfies*

$$\alpha_t = \frac{1}{2}||\pi_t - \pi_{t-1}||_1 \to 0,$$

where π_t is the marginal posterior of $\theta_{1:(t-1)}$ at time t in α_t.

3. *(Filtering consistency) For a sequence of $\lambda_t^{(F)} \to 0$ and a bounded sequence of positive integers S_t, the following holds:*

$$\sup_{\theta_{1:(t-1)}} ||\pi_t - P_t^{S_t}(\theta_{1:(t-1)}, \cdot)||_1 \leq 2\lambda_t^{(F)}.$$

4. *(Jumping consistency) For a sequence of $\lambda_t^{(J)} \to 0$, the following holds:*

$$\sup_{\theta_{1:(t-1)}} ||\pi_t(\cdot|\theta_{1:(t-1)}) - J_t(\theta_{1:(t-1)}, \cdot)||_1 \leq 2\lambda_t^{(J)}.$$

Let $\epsilon_t = \rho_t^{m_t}$. Define

$$\gamma_t^{(F)} = \sum_{v=1}^{t} \left\{ \prod_{u=v+1}^{t} \epsilon_u (1 - \lambda_u^{(F)}) \right\} \epsilon_v \lambda_v^{(F)}$$

and

$$\gamma_t^{(J)} = \sum_{v=1}^{t} \left\{ \prod_{u=v}^{t} \epsilon_u \right\} (\lambda_v^{(J)} + \alpha_v)$$

to be the bounds from Theorem 4.3.1 and Theorem 3.9 of Yang and Dunson (2013), respectively. If, for all $u \leq t$, $\lambda_u^{(F)} \leq \alpha_u + \lambda_u^{(J)}$, then $\gamma_t^{(F)} \leq \gamma_t^{(J)}$.

Proof. Appendix C.1 □

Theorem 4.3.2 gives a sufficient, but somewhat restrictive, condition on the convergence of relative pieces such that Generative Filtering's convergence is bounded more tightly than SMCMC. It reveals an interesting relationship between the use of filtering or the jumping kernel. For a fixed $t \geq 1$, no matter how good the jumping kernel is, e.g., $\lambda_t^{(J)} = 0$, there is a fixed $\alpha_t > 0$ contributing

to the SMCMC upper bound. When PPRB-within-Gibbs is used as the filtering method, there exists an S_t such that $\lambda_t^{(F)} \leq \alpha_t \leq \alpha_t + \lambda_t^{(J)}$.

Theorem 4.3.3. *With the conditions and definitions of Theorem 4.3.2, assume* $\gamma_{t-1}^{(F)} = \gamma_{t-1}^{(J)}$ *and define* $\gamma := \gamma_{t-1}^{(F)} = \gamma_{t-1}^{(J)}$. *If* $\gamma < 1$ *and* $\lambda_t^{(F)} \leq \frac{\alpha_t + \lambda_t^{(J)}}{1-\gamma}$, *then* $\gamma_t^{(F)} \leq \gamma_t^{(J)}$. *If* $\gamma \geq 1$ *then* $\gamma_t^{(F)} \leq \gamma_t^{(J)}$ *always.*

Proof. Appendix C.1 □

Theorem 4.3.3 shows a sufficient condition for a claim that is slightly weaker than that in Theorem 4.3.2, but as a result, the condition is less strict. Theorem 4.3.3 is relevant to the relative gain in the bound of Generative Filtering over that of SMCMC, when both are starting from the same position. As a result we only need conditions on time t, not all times $u \leq t$. In the $\gamma < 1$ case, this condition is less strict than the equivalent condition in Theorem 4.3.2, because of the denominator $0 < 1 - \gamma \leq 1$. For larger γ, the bound on the filtering step can be significantly worse than the bound on the jumping kernel and still result in a lower upper bound for the total Generative Filtering process. In the $\gamma \geq 1$ case, when starting from the same condition, the upper bound for Generative Filtering at the next time point will *always* be lower than that of SMCMC.

4.3.2 Transition kernel and m_t, the required iterations

In the above Section 4.3.1, we assume that Generative Filtering and SMCMC use the same transition kernel for the same number of iterations, m_t. Here we will continue to use the same transition kernel(s), T_t, however, we will investigate under what conditions Generative Filtering can use fewer transition kernel iterations to achieve the same convergence.

Lemmas 3.2 and 3.1 of Yang and Dunson (2013) together show that if a transition kernel T_t satisfies $\sup_x ||T_t(x, \cdot) - \pi_t||_1 \leq 2\rho_t$, then for a distribution p_0 we have $||T_t^{m_t} \circ p_0 - \pi_t||_1 \leq \rho_t^{m_t} ||p_0 - \pi_t||_1$. We wish to compare the case when $p_0 = J_t \circ \pi_{t-1}$ and when $p_0 = P_t^{S_t} \circ \pi_{t-1}$. Thus, filtering results in a distribution closer to π_t than the jumping kernel, the result of Generative Filtering using the same transition kernel with the same number of steps as SMCMC achieves a smaller upper bound on its distance from π_t than SMCMC.

Now, consider the case when the transition kernel is applied a different number of times in Generative Filtering ($m_t^{(F)}$) than in SMCMC ($m_t^{(S)}$). If

$$m_t^{(F)} \geq m_t^{(S)} - \frac{\log\left(||J_t \circ \pi_{t-1} - \pi_t||_1 / ||P_t^{S_t} \circ \pi_{t-1} - \pi_t||_1\right)}{\log(1/\rho_t)}, \tag{4.10}$$

then $\rho_t^{m_t^{(F)}} ||P_t^{S_t} \circ \pi_{t-1} - \pi_t||_1 \leq \rho_t^{m_t^{(S)}} ||J_t \circ \pi_{t-1} - \pi_t||_1$. Since $\rho_t < 1$, $\log(1/\rho_t) > 0$ and if $||P_t^{S_t} \circ \pi_{t-1} - \pi_t||_1 < ||J_t \circ \pi_{t-1} - \pi_t||_1$, then

$$\frac{\log\left(||J_t \circ \pi_{t-1} - \pi_t||_1 / ||P_t^{S_t} \circ \pi_{t-1} - \pi_t||_1\right)}{\log(1/\rho_t)} > 0.$$

Generative Filtering can achieve a lower convergence bound than SMCMC with either more transition kernel steps, or some smaller number of steps determined by Eq. (4.10).

In order for fewer transition kernel steps to be required, we need

$$\frac{\log\left(||J_t \circ \pi_{t-1} - \pi_t||_1 / ||P_t^{S_t} \circ \pi_{t-1} - \pi_t||_1\right)}{\log(1/\rho_t)} \geq k,$$

where the integer $k \geq 1$ is the number of steps difference, which is true if and only if

$$\frac{||P_t^{S_t} \circ \pi_{t-1} - \pi_t||_1}{||J_t \circ \pi_{t-1} - \pi_t||_1} \leq \rho_t^k. \tag{4.11}$$

This reveals a relationship between the gains of PPRB-within-Gibbs over the jumping kernel, $||P_t^{S_t} \circ \pi_{t-1} - \pi_t||_1 / ||J_t \circ \pi_{t-1} - \pi_t||_1$, and the mixing of the transition kernel, ρ_t. If the transition kernel is weakly mixing, $\rho_t \approx 1$, PPRB-within-Gibbs can result in a larger reduction in the number of required transition kernel steps over the SMCMC jumping kernel than in the case when the transition kernel is strongly mixing, $\rho_t \ll 1$.

While Eq. (4.11) provides a sufficient condition for Generative Filtering to require fewer transition kernel steps for an equivalent convergence bound to SMCMC, a lower number of transition kernel steps does not necessarily mean Generative Filtering takes less time to complete. This

is because while SMCMC's jumping kernel can be applied to each ensemble sample in parallel, filtering methods such as PPRB-within-Gibbs are inherently sequential. Thus when many cores are available for computation, the filtering step of Generative Filtering is a potential bottleneck. The improvement in convergence provided by PPRB-within-Gibbs may or may not overcome the time saved by parallelizing the jumping kernel in SMCMC. In Section 4.4 and Section 4.5, we demonstrate both cases.

4.3.3 Streaming Data Storage Considerations

In this section, we provide conditions under which a transition kernel at time t can be applied without requiring storage of the full data through time t. In general, we want a sufficient statistic, $U_t(\boldsymbol{y}_t)$, to exist for each batch of data, $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots$, so that the likelihood can be evaluated without the full data, and transition kernels can be applied while needing to store only the statistics U_t for $t = 1, 2, \ldots$. The data themselves are always a trivial sufficient statistic, however when $\dim U_t \ll \dim \boldsymbol{y}_t$, storage of the sufficient statistics allows for significant reductions in data storage requirements and potentially faster computation of the transition kernel. We describe conditions that are sufficient to allow reduced data storage when applying the transition kernel.

Theorem 4.3.4. *Assume:*

1. *The data $\boldsymbol{y}_{t_1}$ and $\boldsymbol{y}_{t_2}$, for all $t_1 < t_2$, are conditionally independent given $\boldsymbol{\theta}_{1:t_2}$.*
2. *Each distribution $p(\boldsymbol{y}_t|\boldsymbol{\theta}_{1:t})$ has a sufficient statistic $U_t(\boldsymbol{y}_t)$ where $\dim U_t \ll \dim \boldsymbol{y}_t$.*

Then any transition kernel can be computed while storing only the sufficient statistics, U_t, instead of the data, $\boldsymbol{y}_t$, for all t.

Proof. Appendix C.1 □

The sufficient condition of low dimensional sufficient statistics for a data distribution is general and fairly broad. However, it is difficult to know in practice when such sufficient statistics exist. We provide a more specific sufficient condition on the data distributions for when data storage needs can be reduced.

Theorem 4.3.5. *Assume:*

1. *The data $\boldsymbol{y}_{t_1}$ and $\boldsymbol{y}_{t_2}$, for all $t_1 < t_2$, are conditionally independent given $\boldsymbol{\theta}_{1:t_2}$.*
2. *Each $\boldsymbol{y}_t$ is a sample of n_t i.i.d. observations $\boldsymbol{y}_{t,i}$ for $i = 1, \ldots, n_t$.*
3. *Each observation $\boldsymbol{y}_{t,i}$ comes from an exponential family distribution.*

Then storage of the full data can be avoided through the use of sufficient statistics.

Proof. Appendix C.1 □

Note that in both Theorem 4.3.4 and Theorem 4.3.5, there is no requirement that the sufficient statistics $U_t(\cdot)$ be the same function at each time t, nor that the data distribution of $\boldsymbol{y}_t$ be the same for each time t. These conditions are thus flexible, applying to a wide range of streaming data settings including time-varying data sources, data collection techniques, or measurement devices.

4.4 Simulation Studies

In this section, we apply Generative Filtering to two sets of simulated data, a Gaussian hidden Markov model and a streaming record linkage model, and compare to other streaming samplers for Bayesian updates with respect to speed and sampling accuracy.

4.4.1 Gaussian Hidden Markov Model

We analyze the Gaussian hidden Markov model,

$$\begin{aligned} y_{t,i} &\sim N(\theta_t, \sigma^2), \text{ for } t = 1, 2, \ldots \text{ and } i = 1, \ldots, n \\ \theta_1 &\sim N(0, \phi^2) \\ \theta_t &\sim N(\theta_{t-1}, \phi^2), \text{ for } t > 1, \end{aligned}$$

where n, σ^2, and ϕ^2 are fixed. This model is a simple representation of a streaming data problem where a new parameter, θ_t, is required to parameterize the distribution of data $\boldsymbol{y}_t$. As the data arrive

sequentially, $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots$, updated estimates of the parameters, $\boldsymbol{\theta}_{1:t}$, are desired through the posterior distribution of $p(\boldsymbol{\theta}_{1:t}|\boldsymbol{y}_{1:t})$.

Data were generated at each combination of $n = 1, 5, 10, 50$ and $\sigma^2 = 0.25, 0.5, 1, 2, 4$, with $\phi^2 = 1$. For practical reasons of generating the data, we generate data for $t = 1, \ldots, T$ with the endpoint $T = 100$. The values of n and σ^2 create varying levels of signal from the data arriving at each time, $\boldsymbol{y}_t$. For each of the 20 combinations of n and σ^2, 20 sets of data were produced with differing random seeds by first drawing $\boldsymbol{\theta}_{1:T}$ from the prior, then generating $\boldsymbol{y}_{1:T}$ conditioned on $\boldsymbol{\theta}_{1:T}$.

For each simulated dataset, and for each value of $t = 1, \ldots, T$, we estimated the posterior distribution of $p(\boldsymbol{\theta}_{1:t}|\boldsymbol{y}_{1:t})$ first using three samplers: non-streaming Gibbs, PPRB-within-Gibbs, and Generative Filtering. Each component, θ_ℓ, in $\boldsymbol{\theta}_{1:t}$ has a conjugate Gaussian full conditional distribution, making a Gibbs sampler a natural choice for a non-streaming MCMC. The Gibbs sampler produced samples directly from the posterior distribution, $p(\boldsymbol{\theta}_{1:t}|\boldsymbol{y}_{1:t})$, for $t = 2, \ldots, T$. Each Gibbs update was run for 1100 iterations, discarding the first 100 as burn-in with 10 independent chains.

The PPRB-within-Gibbs and Generative Filtering updates were sequentially applied in a streaming fashion using samples from the previous time point as the initial ensemble. The first streaming updates at time $t = 2$ used samples drawn independently from the true Gaussian posterior at $t = 1$, $p(\theta_1|y_1)$, as their initial ensemble. For the update of the new parameter, θ_t, in PPRB-within-Gibbs, we used its full conditional Gaussian distribution. Each PPRB-within-Gibbs sampler was run for 1100 iterations, discarding the first 100 as burn-in. The PPRB-within-Gibbs updates that served as the filtering step in Generative Filtering used the same configuration. The transition kernel used in Generative Filtering updates all parameters $\boldsymbol{\theta}_{1:t}$ simultaneously using a Metropolis random walk with multivariate Gaussian proposals. As the true posterior distribution is known, we choose a proposal distribution based on the adaptive Metropolis proposal of Gelman et al. (1997) and Haario et al. (2001), i.e., $N(\mathbf{0}, 2.4^2\boldsymbol{\Sigma}/t)$, where $\boldsymbol{\Sigma}$ is the true posterior variance. We use a random walk Metropolis transition kernel to simulate a slowly-converging transition ker-

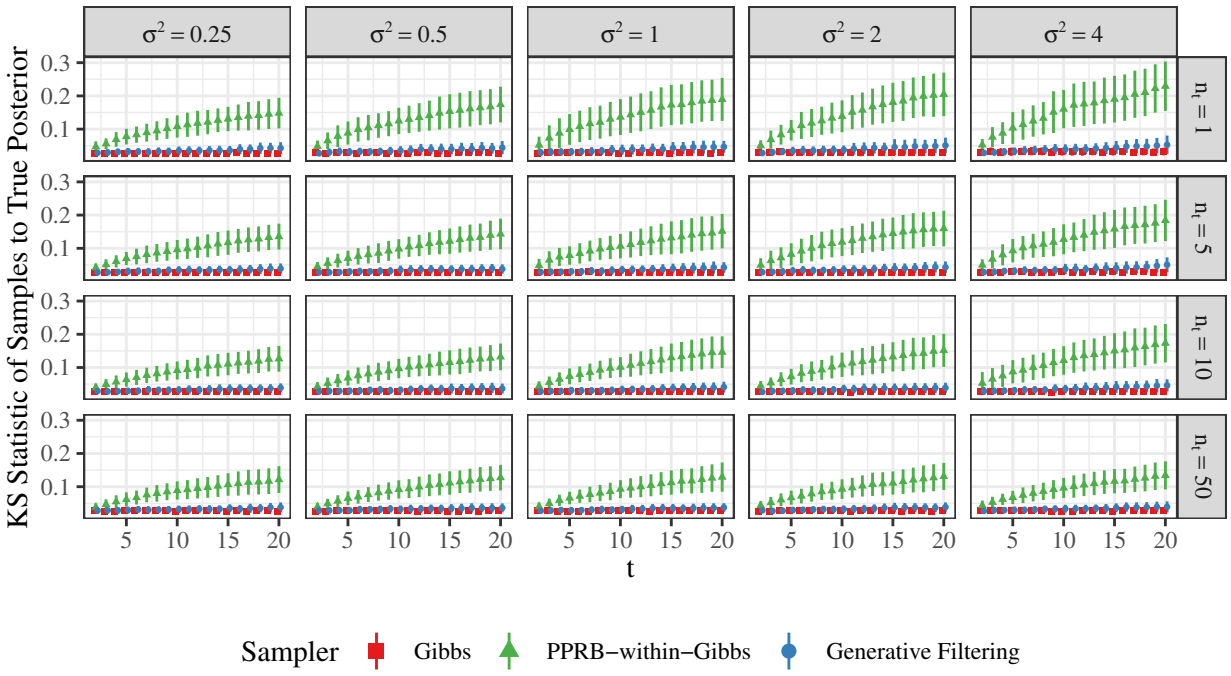


Figure 4.1: KS statistic (mean plus or minus standard deviation) for samples of θ_1 after repeated application of non-streaming Gibbs, PPRB-within-Gibbs, and Generative Filtering, for $t = 2, \ldots, 20$. Higher KS statistics indicate more MCMC error in the samples. After repeated applications, PPRB-within-Gibbs produces worse approximations to the posterior distribution while Generative Filtering does not.

nel, specifically, one which converges more slowly as the number of parameters increases. This slow convergence can be an issue in many real-world MCMC applications, and would be a scenario in which a user would wish to improve convergence times. The transition kernel in Generative Filtering was applied $m_t = 5$ times for each update. The full streaming update process from time $t = 2$ to $t = T$ was run 10 times on each dataset, using 10 different random seeds.

To explore the phenomenon of filtering degradation in this simulated data, we compare the samples produced by Gibbs, PPRB-within-Gibbs, and Generative Filtering to the true posterior distribution of $p(\boldsymbol{\theta}_{1:t}|\boldsymbol{y}_{1:t})$. This model can be written as a Gaussian linear regression model with a multivariate normal prior on the regression parameters and fixed variance, so the posterior distribution can be calculated in closed form for data through any time t. We used the max absolute difference between the true CDF and empirical CDF, i.e., the Kolmogorov-Smirnov test statistic, of each parameter, $\theta_1, \ldots, \theta_t$, as a measure of the MCMC error in the samples for that parame-

ter. Figure 4.1 shows these measured errors for θ_1 at each time up to $t = 20$. As expected, the non-streaming Gibbs sampler is able to maintain a constant level of MCMC error at each time $t = 2, \ldots, 20$ because it is producing new samples from the posterior at each time. However, the MCMC error of PPRB-within-Gibbs increases with each update as the number of unique values of θ_1 in its samples decreases. Generative Filtering avoids the degradation observed in PPRB-within-Gibbs, having MCMC error comparable to Gibbs.

To investigate the relative convergence of Generative Filtering and SMCMC, we also compared the required number of transition kernel steps for SMCMC and Generative Filtering to estimate the parameters $\boldsymbol{\theta}_{1:t}$ using the simulated data. At each time, t, the jumping kernel for SMCMC and the update for θ_t in the PPRB-within-Gibbs step of Generative Filtering were chosen to be the conjugate Gaussian full conditional distribution for θ_t. The Metropolis random walk transition kernel was used in both SMCMC and Generative Filtering, and the PPRB-within-Gibbs filtering step of Generative filtering was as described earlier. At each time t we began both Generative Filtering and SMCMC using the same initial ensemble of 1000 samples from $p(\boldsymbol{\theta}_{1:(t-1)}|\boldsymbol{y}_{1:(t-1)})$ produced by the non-streaming Gibbs sampler. Thus, we are evaluating the difference in minimum transition steps to converge for each method while controlling the quality of the initial ensemble.

Each sampler was stopped when the KS-statistics comparing the marginal empirical posterior distributions of θ_t and θ_{t-1} in the current ensemble to their respective true posterior distributions were both below 0.055, the critical value of the KS distribution. The cumulative number of transition kernel steps required to converge for each sampler up to each time is shown in Figure 4.2. By $t = 20$, Generative Filtering has required many fewer transition kernel steps than SMCMC in all datasets due to its use of PPRB-within-Gibbs instead of the jumping kernel. However, while the jumping kernel in SMCMC is parallelizable, PPRB-within-Gibbs is inherently sequential. This presents a tradeoff between the two methods in environments with parallel computation available. We compare the cumulative runtime required for each method to converge in Appendix C.3. In this numerical experiment, for SMCMC to be faster than Generative Filtering on average in all settings would require 45 cores.

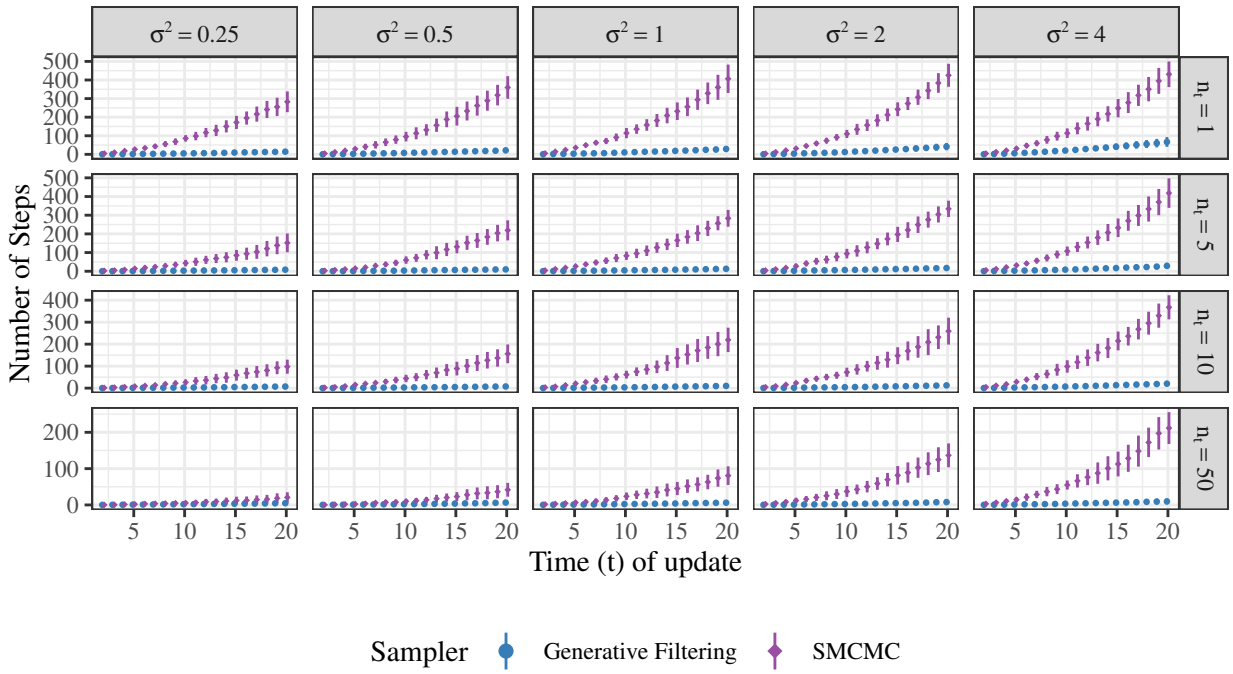


Figure 4.2: Cumulative number (mean plus or minus standard deviation) of transition kernel steps to converge to the posterior distribution.

4.4.2 Streaming Record Linkage

Record linkage is the task of consolidating records which refer to overlapping sets of entities. Often this task must be performed without the presence of a unique identifier. Noisy duplicated records present a problem for those who wish to use the data to make inferences. Bayesian record linkage estimates duplicate entities via a posterior distribution of linkages and provides natural uncertainty quantification (e.g., Tancredi and Liseo, 2011; Steorts et al., 2016; Sadinle, 2017; Zanella, 2020; Aleshin-Guendel and Sadinle, 2023).

In streaming record linkage, sets of records (files) arrive sequentially in time with no predetermined number and estimates of links are updated after the arrival of each. Recent advancements in record linkage allow for near real-time data-driven record linkage (e.g., Christen et al., 2009; Ioannou et al., 2010; Dey et al., 2011; Altwaijry et al., 2017; Karapiperis et al., 2018) and model-driven Bayesian record linkage in the streaming data setting can be performed using recursive Bayesian updates with SMCMC or PPRB-within-Gibbs (Chapter 3). In Chapter 3 we note PPRB-

within-Gibbs is efficient, but degrades, and SMCMC requires a parallel computing environment but maintains accuracy. We expand on the simulation study performed in that paper by comparing to Generative Filtering. We demonstrate that Generative Filtering does not suffer from the same degradation as PPRB-within-Gibbs on the streaming record linkage problem.

For this study, we use the data files as described in Chapter 3. Data were simulated using the GeCo software package (Tran et al., 2013) to create realistic demographic information and insert realistic errors into randomly selected fields. Each record had 10 fields: given name, surname, age, occupation, and 6 categorical fields with 12 possible levels. Each record had errors inserted to mimic common typos, misspellings, or OCR errors. Files were generated with records that contained up to 4, 6, or 8 errors. The files were generated with varying overlap, having 10%, 30%, 50% or 90% of their records coreferent with records with previous files. For each of the 12 combinations of error and overlap, a set of 4 files was generated to arrive sequentially, simulating streaming data.

We use the streaming record linkage model in Chapter 3, with diffuse, weak, and strong priors for the parameter governing the distribution of disagreement levels for matched records. We begin the streaming updates using posterior samples from a two file record linkage using the `BRL` package (Sadinle, 2017). Then, we perform sequential PPRB-within-Gibbs updates and sequential Generative Filtering updates with the third, fourth, and fifth file of each dataset. The Generative Filtering updates consisted of an ensemble of 200 samples. Each filtering step used PPRB-within-Gibbs for 2000 iterations, discarding the first 1000 and thinning to 200 before applying the transition kernel. The Generative Filtering transition kernel used locally balanced proposals for link updates, and was run for 200 iterations. The number of transition kernel iterations was conservatively chosen by examining traceplots for convergence. The PPRB-within-Gibbs updates use the three step version originally proposed in Chapter 3 and implemented in the `bstrl` R package (Taylor et al., 2022). The PPRB-within-Gibbs sampler was run using locally balanced link updates for 5000 iterations, discarding the first 1000 as burn-in. For comparison, we also fit the model using a non-streaming

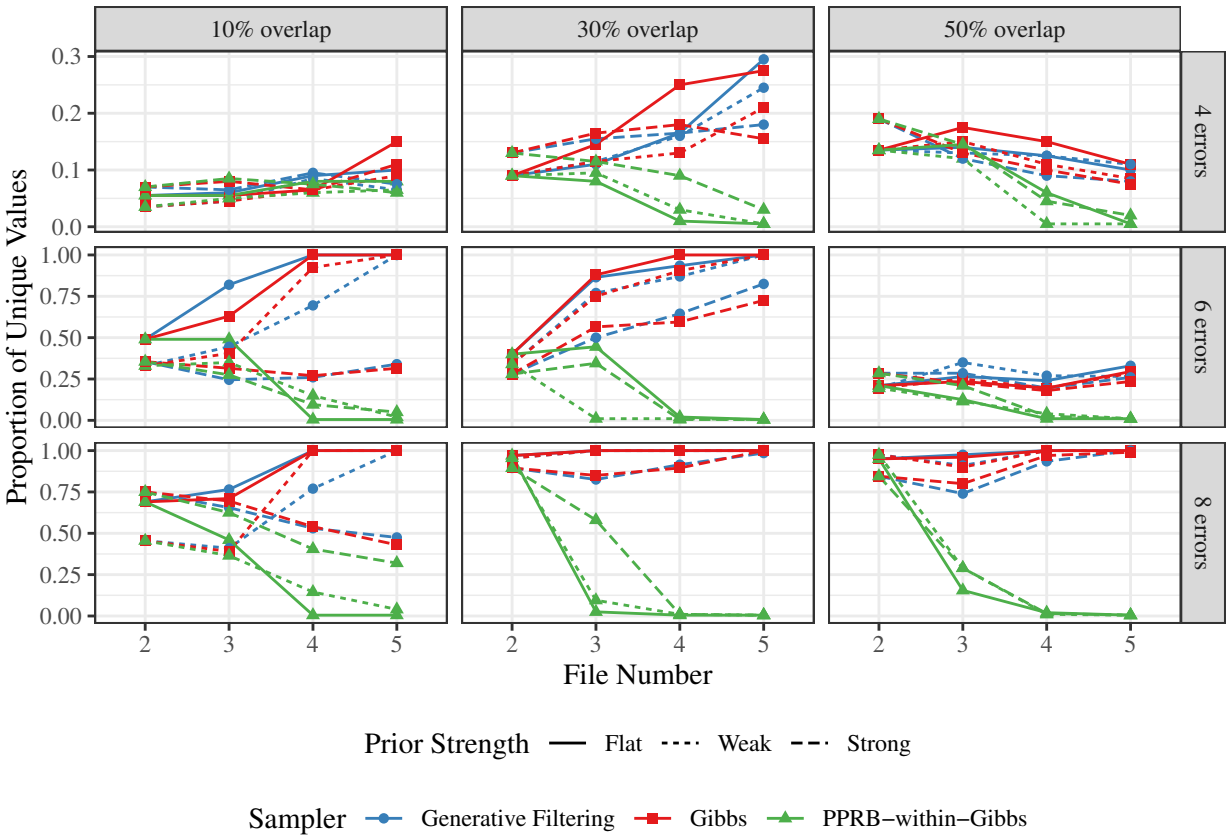


Figure 4.3: Proportion of unique values within samples of $Z^{(1)}$ produced by Gibbs, PPRB-within-Gibbs, and Generative Filtering. While the proportion of unique values from the PPRB-within-Gibbs samples decreases with each successive update, the proportion of unique values from Generative Filtering remains similar to that of samples produced by Gibbs.

Gibbs sampler. The Gibbs sampler was run using component-wise link updates for 2500 iterations, discarding the first 500 as burn-in.

We examine the unique values within the produced samples for $\boldsymbol{Z}^{(1)}$, the parameter that encodes links from file 2 to file 1. Before comparing, we thinned samples produced by Gibbs and PPRB-within-Gibbs from 2000 and 4000, respectively, to 200 to match the ensemble size of Generative Filtering. The number of unique values produced by PPRB-within-Gibbs decreases with each Bayesian update relative to Gibbs, while the number of unique values produced by Generative Filtering is comparable to Gibbs (Figure 4.3). This indicates that Generative Filtering is able to avoid the degradation problems of PPRB-within-Gibbs.

4.5 Application

We investigate the effectiveness of Generative Filtering on data of Steller sea lion pup counts. Steller sea lions are an endangered species[1] whose population have been reduced by a shrinking habitat due to climate change and human activity (Hooten et al., 2021). The National Marine Mammal Laboratory of the National Oceanographic and Atmospheric Administration performs aerial surveys of the number of pups born each year across several sites in Alaska. The data are available in the `R` package `agTrend` (Johnson, 2017) and contain 713 observations between the years 1973 and 2016, at 72 sites. Not every site is measured at every year, with sites having between 1 and 22 observations.

We assume that counts of sea lion pups are produced from latent population intensities for each site-year combination, which are related through a log-scale autoregressive process. We model these data using the following hierarchical model,

$$\begin{aligned}
y_{s,t} &\sim \text{Pois}(\lambda_{s,t}) \\
\log(\lambda_{s,1}) &\sim \text{N}(\mu_1, \sigma_1^2) \\
\log(\lambda_{s,t}) &\sim \text{N}(\phi_s + \log(\lambda_{s,t-1}), \sigma_s^2) \\
\phi_s &\sim \text{N}(0, \sigma_\phi^2) \\
\sigma_s^2 &\sim \text{Inverse-gamma}(\alpha, \beta),
\end{aligned}$$

where $s = 1, 2, 3, 4$ for each of our four studied sites, and $t = 1978, \ldots, 2016$. The parameters $\lambda_{s,t}$ represent the latent population intensities, and the parameters ϕ_s and σ_s^2 define the relationship between population intensity parameters over time. The parameters of interest are population intensities $\lambda_{s,t}$ and population intensity trends ϕ_s. Negative values of ϕ_s indicate the population intensities are decreasing at site s while positive values of ϕ_s indicate the population intensites are increasing at site s.

[1] United States Endangered Species Act of 1973

These data were analyzed by Hooten et al. (2021) who used PPRB to update a temporal model of two measured sites (Marmot and Sugarloaf) a single time from year 2013 to 2015. We extend this analysis in two ways. First, we include two additional sites (Seal Rocks and Atkins) from the same survey. Second, we perform a series of 16 streaming updates from the year 2000 to 2016. We assume a first-stage analysis has been performed for all data through year 2000 which will form the basis for the series of streaming updates performed. Each update incorporates the data from an additional year. We use the same hyperparameters as in Hooten et al. (2021), specifically, $\mu_1 = 8.7$, $\sigma_1^2 = 1.69$, $\sigma_\phi^2 = 1$, $\alpha = 1$, and $\beta = 20$.

We perform the series of updates using four methods: a non-streaming Gibbs sampler for each year fitting the full model (Gibbs), sequential PPRB-within-Gibbs updates (PPRB-within-Gibbs), sequential SMCMC updates (SMCMC), and sequential Generative Filtering. The full sampler details can be found in Appendix C.4. Similarly to the simulation in Section 4.4.1, we perform transition kernel steps in SMCMC and Generative Filtering until a desired level of convergence to the posterior is reached. Unlike in Section 4.4.1, however, the true posterior is unknown in this model. Thus we use the Gibbs samples as a reference, and stop SMCMC and Generative filtering when the KS statistic comparing the current state of the ensemble to the Gibbs samples is below a desired threshold. This approach is not feasible in scenarios where streaming is required and we discuss practical options for choosing m_t in Section 4.6.

Figure 4.4 (left) shows the posterior means and credible intervals for $\log(\lambda_{s,t})$. In three of four sites, population intensity estimates declined, then began to rise after 2000. We are also able to recover estimates of $\log(\lambda_{s,t})$ in years where there is missing data for the sites, though the credible intervals are wide for these parameters. Figure 4.4 (right) shows the changing posterior mean and credible intervals for each parameter ϕ_s, with each Bayesian update. Corresponding to the trends observed in the estimates of $\log(\lambda_{s,t})$, the posterior means and credible intervals of ϕ_s become less negative with each new year of data. The inference for both $\log(\lambda_{s,t})$ and ϕ_s using Generative Filtering is nearly identical to non-streaming Gibbs, but with cumulatively 32% less time required with only one core.

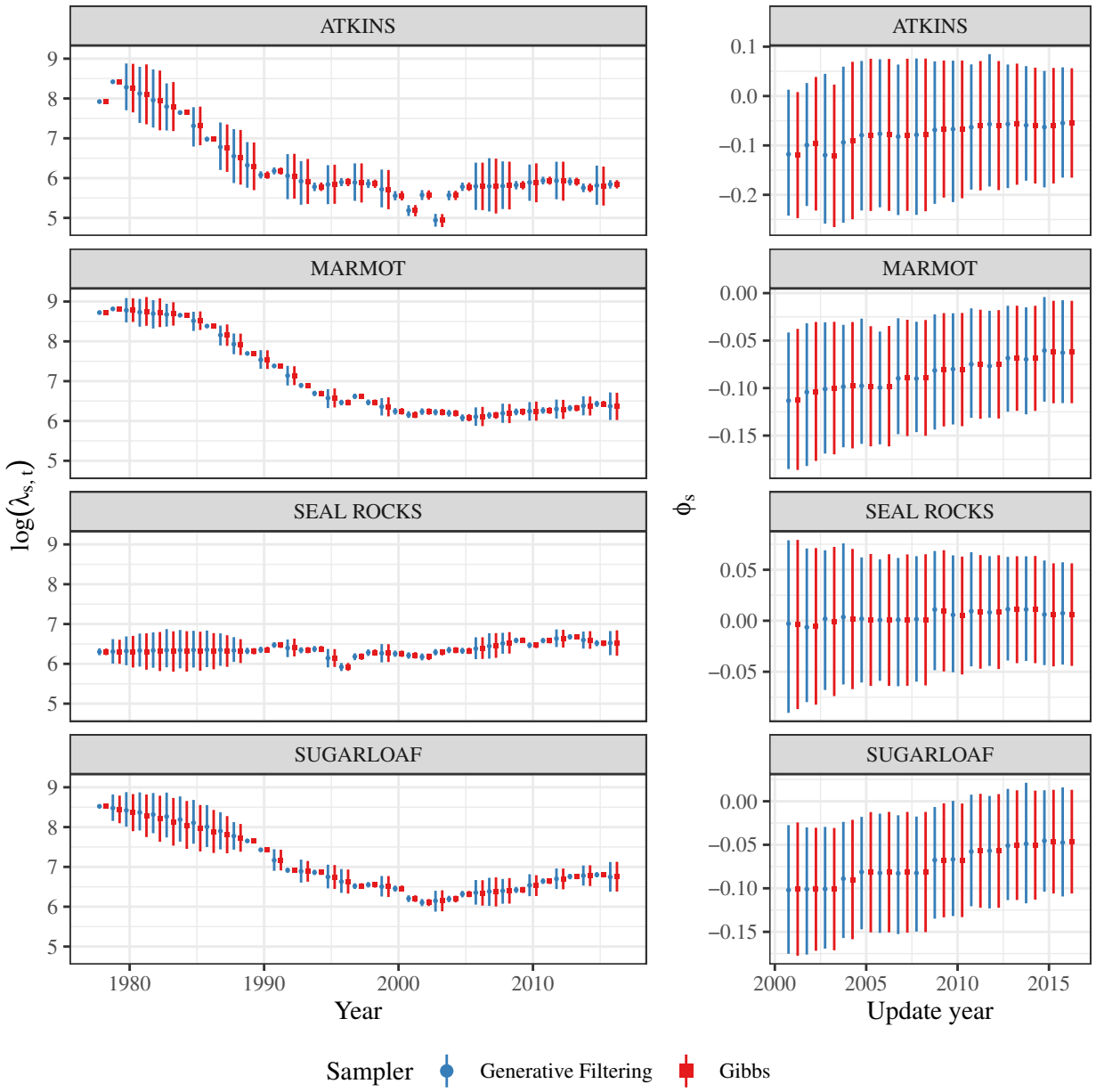


Figure 4.4: LEFT: Posterior means and credible intervals for $\log(\lambda_{s,t})$ parameters after all updates. Credible intervals are wider where there was missing data. In three of four sites, population intensity estimates declined, then began to rise after 2000. RIGHT: Posterior means and credible intervals for ϕ_s following each streaming update. Means shift positive and credible intervals narrow with the arrival of new data. On both the left and right, means and credible intervals agree between Gibbs and Generative Filtering.

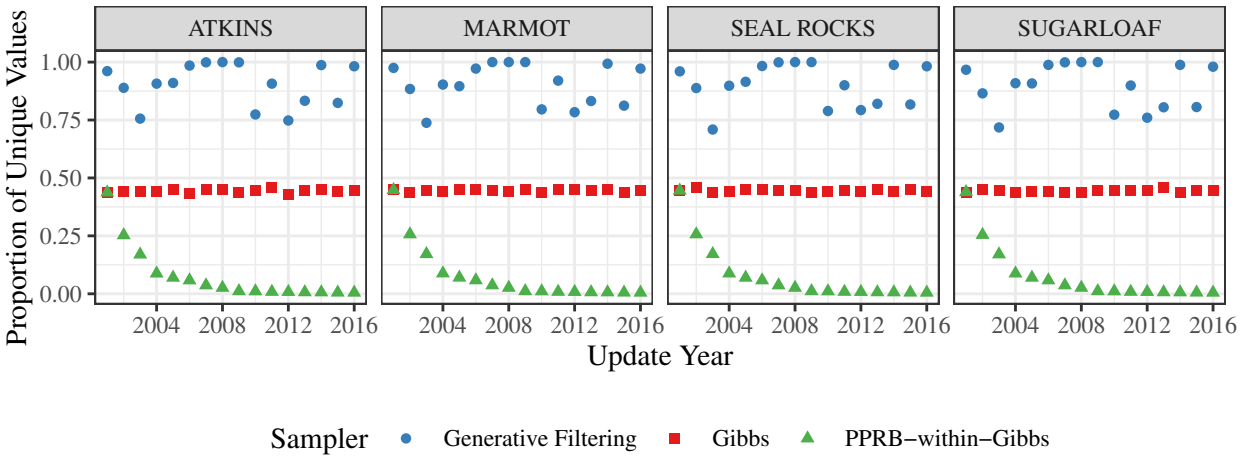


Figure 4.5: Number of unique values of $\log(\lambda_{s,t})$, for $t = 2001$, present in samples as a proportion of the number of samples produced with each sampler. The unique values produced by PPRB-within-Gibbs decays with each successive update. The Gibbs sampler produces a consistent number of unique values, approximately 44%, due to the tuning of its component updates. Generative Filtering produces consistently the highest proportion of unique values among its samples.

We compare the samplers' ability to obtain these estimates through three metrics: the number of distinct values present in repeatedly updated parameters (Figure 4.5), the number of transition kernel steps required by each of SMCMC and Generative Filtering (Figure 4.6, top), and the total time required to perform all updates for SMCMC, Generative Filtering, and Gibbs with varying numbers of cores available (Figure 4.6, bottom). First, Generative filtering outperforms PPRB-within-Gibbs in terms of the proportion of unique values within its produced samples (Figure 4.5). Generative Filtering also has a higher proportion of unique values than Gibbs, due to the Gibbs sampler's component updates tuned to 44% acceptance rate. We also see that Generative Filtering outperforms SMCMC in both number of transition kernel steps required, and both SMCMC and Gibbs in runtime (Figure 4.6). In particular, two cases in the runtime comparison are especially interesting. For one available core, i.e., sequential execution, Generative Filtering is faster even than Gibbs. This indicates that even when no parallel execution is possible, Generative Filtering takes advantage of previously produced samples and is faster than refitting the model from scratch. For 1000 available cores, we simulate the situation where essentially unlimited parallel computing resources are available and each thread in either SMCMC or Generative Filtering is able to be run in

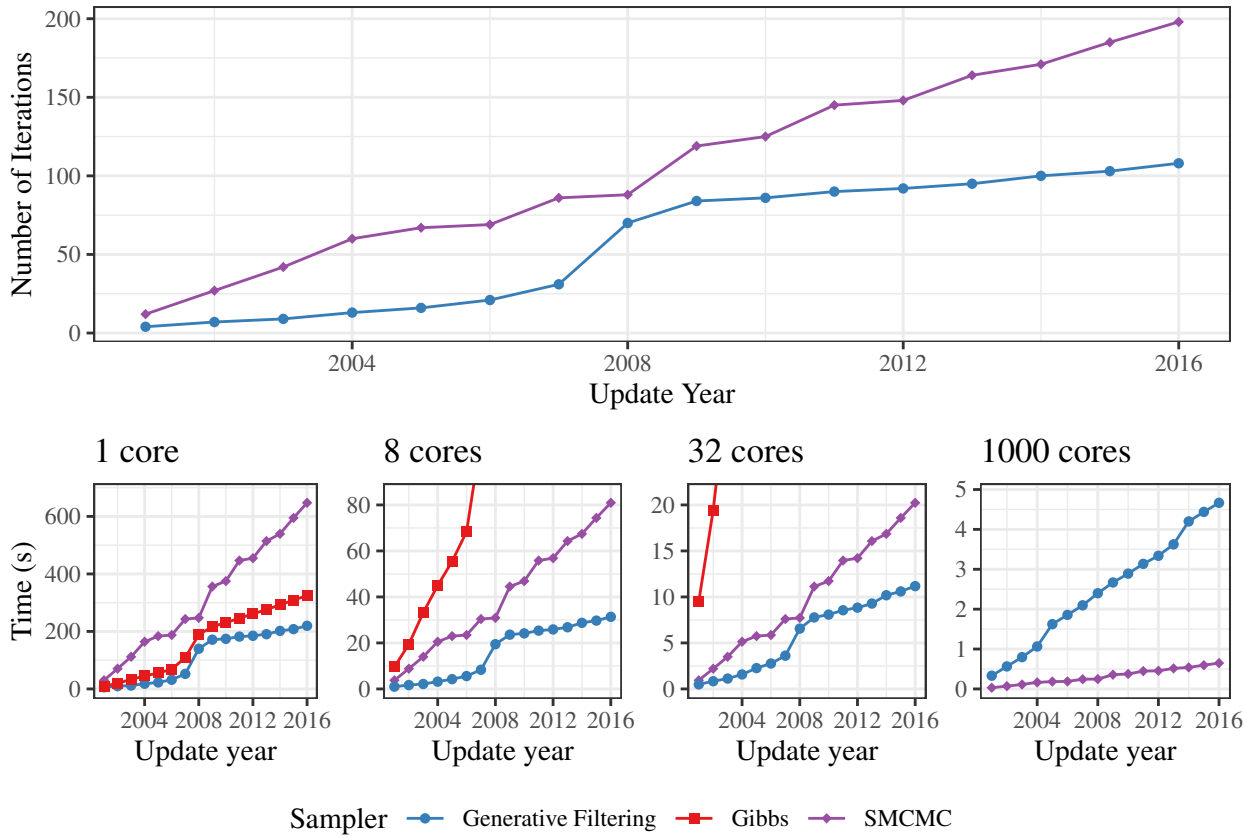


Figure 4.6: TOP: Total number of transition kernel steps required for each method to converge to the posterior distribution. Generative Filtering requires fewer transition kernel steps due to its initial PPRB-within-Gibbs step. BOTTOM: Cumulative time to converge to the posterior distribution in each update, for varying numbers of available parallel cores. Left to right: 1 core, 8 cores, 32 cores, 1000 cores. These represent, respectively, no available parallel resources, a typical laptop or workstation, a single node in a high performance computing environment, and unlimited parallel resources.

parallel. In this case, SMCMC overtakes Generative Filtering in speed because its jumping kernel is able to be parallelized while PPRB-within-Gibbs is not. However, this is an extreme setting. We also note that both methods outperform Gibbs by wide margins. Together, these timing results suggest that Generative Filtering is an attractive method for streaming Bayesian updates where time is a consideration, but few or moderate computational resources are available.

4.6 Discussion

We define Generative Filtering as an efficient way to perform recursive Bayesian updates in a streaming data context when moderate parallel computing resources are available. We show that Generative Filtering resolves the problems of the two methods it extends, avoiding the degradation of filtering methods such as PPRB and PPRB-within-Gibbs and converging faster with fewer resources than SMCMC. We characterize the degradation of PPRB and PPRB-within-Gibbs samples after repeated application, and provide novel bounds on the error introduced by this degradation. We additionally provide sufficient conditions for reducing the data storage needs of Generative Filtering.

We conduct two simulation studies with streaming data models to demonstrate the effectiveness of Generative Filtering. We find that repeated application of PPRB-within-Gibbs in a streaming setting leads to a measurable accumulation of MCMC error and show that Generative Filtering avoids this error to reach the same level of convergence as SMCMC using less time with moderate computational resources.

We use Generative Filtering to analyze Steller sea lion pup counts discovering trends in count intensity at four different sites and observing changes in parameter estimates with the arrival of new data. We find that Generative Filtering required fewer transition kernel steps than SMCMC and less time with moderate computation resources (32 cores or fewer). We also find that Generative Filtering requires less time to produce the equivalent of 1000 independent samples than Gibbs when only one core is available.

Throughout this chapter, when comparing the time to convergence for streaming samplers, we have used an objective standard to determine when a given algorithm converged to the posterior distribution. We compared samples to either the true posterior distribution (if known), or to reference samples produced by a non-streaming alternative sampler. This worked for our purposes of demonstrating faster convergence for Generative Filtering than SMCMC. However, this approach is not relevant in practice and determining the number of transition kernel iterations to perform remains an open question. Yang and Dunson (2013) propose a correlation-based stopping, however, this approach is not appropriate for many cases, e.g., discrete parameters without a natural meaning or measure of correlation. Future work for samplers of this type will include a more general stopping criterion.

Chapter 5

Conclusion

In this dissertation we presented three methods for the analysis of complex data. In Chapter 2 we introduced restricted regression for network data to mitigate network confounding. We find that for binary data, Restricted Network Regression results in better estimation of the unconditional regression effects than network regression without random effects. In Chapter 3, we developed a Bayesian record linkage model for the streaming data setting and present two streaming samplers for performing Bayes updates with the arrival of each file. Motivated by resolving the trade-offs between these two streaming samplers, in Chapter 4 we develop Generative Filtering, a new sampler for performing recursive Bayes updates. We show that Generative Filtering retains the speed of PPRB while avoiding filtering degradation, and the parallel computation of SMCMC while converging faster with constrained computational resources. We conclude by discussing potential future directions of research related to the topics in this dissertation.

5.1 Restricted Network Regression Future Work

In Chapter 2, we define network confounding and introduce Restricted Network Regression to mitigate network confounding. For a general network model with a random effect,

$$y_{ij} = \boldsymbol{x}_{d,ij}^\top \boldsymbol{\beta}_d + \boldsymbol{x}_{s,i}^\top \boldsymbol{\beta}_s + \boldsymbol{x}_{r,j}^\top \boldsymbol{\beta}_r + \eta_{ij} + \varepsilon_{ij},$$

we specifically introduce network regression for additive node-structured random effects of the form, $\eta_{ij} = a_i + b_j$. Other kinds of network-structured random effects exist for which network confounding and solutions can be studied, for example, multiplicative random effects of the form $\eta_{ij} = \boldsymbol{u}_i^\top \boldsymbol{v}_j$ for vectors $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$ (Hoff, 2021) or Euclidean latent space random effects of the form $\eta_{ij} = -\|\boldsymbol{w}_i - \boldsymbol{w}_j\|_2$ for vectors $\boldsymbol{w}_i$ (Hoff et al., 2002). These random effects both place each network node in a latent space where two nodes' relative positions affect the relation between

them. While confounding with additive random effects has a clear relationship to collinearity of fixed and random effects, neither multiplicative nor latent space random effects form a linear space. However, there are network covariates that have the same structure as these random effects. Common group membership effects (e.g., Hoff et al., 2013) can be written in the form $x_{ij} = x_i x_j$ where x_i are binary values indicating whether each individual is in a particular group, which has a multiplicative structure. Where nodes represent entities with physical locations a covariate may be the phyiscal distances between entities, which has a Euclidean distance structure. An interesting area of future work would be to investigate the estimation of such covariates in the presence of similarly structured random effects.

5.2 Streaming Record Linkage Future Work

Several extensions to the streaming record linkage model in Chapter 3 are possible to further leverage the temporal nature of the arriving files. First, as the number of files increases, the number of potential links for any record in the most recent file, X_k becomes larger. This increases the complexity of estimating the links in $\boldsymbol{Z}^{(k-1)}$. If we have a priori knowledge that entities are unlikely to be absent from files for long periods of time before reappearing, we can limit this complexity by blocking, i.e., limiting links to only a subset of other records. We can choose a value Δt as the maximum allowed difference in time for directly linked records and modify the link validity criterion (Definition 3.2.1) to be the following.

Definition 5.2.1. Link Validity Constraint with blocking. Let $\Delta t \geq 1$. Let $\mathcal{C}_k$ be the set of all matching vectors $\boldsymbol{Z}^{(1:(k-1))}$ such that every record $\boldsymbol{x}_{m_1 i}$ receives at most one link from a record $\boldsymbol{x}_{m_2 j}$ where $m_1 < m_2 \leq m_1 + \Delta t$, and receives no links from any record $\boldsymbol{x}_{m_2 j}$ where $m_2 > m_1 + \Delta t$. That is, there is at most one value in any $\boldsymbol{Z}^{(m_2-1)}$ with $m_1 < m_2 \leq m_1 + \Delta t$ that equals $\sum_{\ell=1}^{m_1-1} n_\ell + i$, and there are no values in any $\boldsymbol{Z}^{(m_2-1)}$ with $m_2 > m_1 + \Delta t$ that equal $\sum_{\ell=1}^{m_1-1} n_\ell + i$. Matching vectors $\boldsymbol{Z}^{(1:(k-1))}$ are valid if and only if $\boldsymbol{Z}^{(1:(k-1))} \in \mathcal{C}_k$.

If all files are approximately equal in size or are bounded in size, then this modified link validity constraint limits the number of potential values of $\boldsymbol{Z}^{(k-1)}$ for large k. This does not prevent far-

reaching links in general, however. If an entity regularly appears in files at most Δt apart in time, then by link tracing, the most recent records can still be linked to the earliest records under this new constraint.

Second, we can incorporate changing temporal likelihoods to represent drift in semi-identifying fields over time. Aleshin-Guendel and Sadinle (2023) allows for multiple likelihoods in multifile record linkage by using parameters $\boldsymbol{m}_{ij}$ and $\boldsymbol{u}_{ij}$ specifically for comparisons between records in files X_i and X_j. However, for k files in an offline batch procedure, this approach requires $k(k-1)/2$ separate likelihoods, which increases the complexity of the model and prevents information borrowing between pairs of files. With temporally-structured files, we can use a restricted version of multiple likelihoods based on time difference. Let there be parameters $\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots$ and $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots$, then comparisons between two files X_i and X_j are parameterized by $\boldsymbol{m}_{|i-j|}$ and $\boldsymbol{u}_{|i-j|}$. This parameterization lets the distribution of matches and nonmatches vary in time, reflecting a belief that the semi-identifying fields used to compare records may have values that drift over time. Unlike other multiple likelihood approaches however, with k files only $k-1$ likelihoods are required, and information can be borrowed between comparisons.

5.3 Generative Filtering Future Work

Throughout the simulations and application in Chapter 4, we have used ad-hoc stopping criteria to determine the number of transition kernel steps to use that were dependent on the specific simulation study setting and would be impossible or impractical in applications. In the case of the simulation study in Section 4.4.1 we compared the current state of the ensemble to the known posterior distribution, which would not be possible in all cases in which MCMC would be used. In the application in section 4.5 we compared the current state of the ensemble to samples produced from a non-streaming reference Gibbs sampler, which would not be practical as it would defeat the purpose of using a streaming sampler. Determining the number of transition kernel steps, m_t, is critical for the application of Generative Filtering as too few iterations would result in large MCMC error and too many iterations defeats the computational benefits of the method. Therefore

a fruitful area of future work would be in developing an automatic stopping criterion for Generative Filtering, which could determine the appropriate number of transition kernel steps without external reference.

Yang and Dunson (2013) propose a correlation-based stopping criterion for SMCMC using the function

$$f_t(k) = \max_{\ell=1\ldots,p} \text{corr}(\theta_\ell^{(k)}, \theta_\ell^{(0)}),$$

where $\theta_\ell^{(k)}$ is the random variable representing the ℓ^{th} component of the parameters, $\boldsymbol{\theta}$, after k applications of the transition kernel at time t. This function can be estimated as $\hat{f}_t(k)$ using the sample correlation of the S samples in the ensemble at each point k. The value of m_t is chosen online as the first k for which this maximal correlation is below a threshold, $\hat{f}_t(k) < \epsilon$. For well-chosen ϵ, this guarantees the desired convergence. This stopping criterion, however, depends on the sample correlation being interpretable for each component of the parameters. For continuous parameters on $\mathbb{R}$, this will generally be the case. However, for discrete-valued parameters, the sample correlation may not be meaningful. For example, consider the link parameters, $\boldsymbol{Z}^{(k-1)}$, from Chapter 3, where recall, $Z_j^{(k-1)} = i$ means record j in file X_k is linked to record i. The use of the standard sample correlation here is problematic here as due to the exchangeability of the records within files, values of $Z_j^{(k-1)} = i$ and $i + 1$ are no more similar than values of $Z_j^{(k-1)} = i$ and $i + 100$, while these would contribute very differently to the sample correlation.

A simple approach to resolving this difficulty would be to transform discrete parameters into a summary statistic for which correlation is meaningful. For the record linkage example, we can consider a function $n(\boldsymbol{Z}^{(1)}, \ldots, \boldsymbol{Z}^{(k-1)})$ which returns the number of distinct entities represented by the links and use the correlation of this function. This allows for a straight-forward application of the correlation criterion but the summary function does not capture all information in the parameter and may result in convergence that is worse than expected. It may be possible to adapt this correlation-based criterion using more flexible measures of correlation. If the discrete parameters in a model exist in a metric space, the Fréchet correlation (Fout and Fosdick, 2023) may be used as a substitute. We may also move away from correlations to diagnostics for multi-chain

MCMC convergence, e.g., $\hat{R}$ (Gelman and Rubin, 1992), though it may be more difficult to use these diagnostics to guarantee rates of convergence in the overall procedure.

Bibliography

Ahmad, S., Lavin, A., Purdy, S., and Agha, Z. (2017). Unsupervised real-time anomaly detection for streaming data. *Neurocomputing*, 262:134–147. Online Real-Time Learning Strategies for Data Streams.

Albert, J. H. and Chib, S. (1993). Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88(422):669–679.

Aleshin-Guendel, S. and Sadinle, M. (2023). Multifile Partitioning for Record Linkage and Duplicate Detection. *Journal of the American Statistical Association*, 118(543):1786–1795. PMID: 37771512.

Aleskerov, F., Meshcheryakova, N., Rezyapova, A., and Shvydun, S. (2017). Network analysis of international migration. In Kalyagin, V. A., Nikolaev, A. I., Pardalos, P. M., and Prokopyev, O. A., editors, *Models, Algorithms, and Technologies for Network Analysis*, pages 177–185, Cham. Springer International Publishing.

Altwaijry, H., Kalashnikov, D. V., and Mehrotra, S. (2017). QDA: A Query-Driven Approach to Entity Resolution. *IEEE Transactions on Knowledge and Data Engineering*, 29(2):402–417.

Anderson, J., Burns, P. J., Milroy, D., Ruprecht, P., Hauser, T., and Siegel, H. J. (2017). Deploying RMACC Summit: An HPC Resource for the Rocky Mountain Region. In *Proceedings of the Practice and Experience in Advanced Research Computing 2017 on Sustainability, Success and Impact*, PEARC17, New York, NY, USA. Association for Computing Machinery.

Becker, R., Eick, S., and Wilks, A. (1995). Visualizing network data. *IEEE Transactions on Visualization and Computer Graphics*, 1(1):16–28.

Besag, J., York, J., and Mollié, A. (1991). Bayesian image restoration, with two applications in spatial statistics. *Annals of the Institute of Statistical Mathematics*, 43(1):1–20.

Betancourt, B., Zanella, G., Miller, J. W., Wallach, H., Zaidi, A., and Steorts, R. C. (2016). Flexible Models for Microclustering with Application to Entity Resolution. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Bifet, A. and Frank, E. (2010). Sentiment knowledge discovery in twitter streaming data. In Pfahringer, B., Holmes, G., and Hoffmann, A., editors, *Discovery Science*, pages 1–15, Berlin, Heidelberg. Springer Berlin Heidelberg.

Binette, O. and Steorts, R. C. (2022). (Almost) all of entity resolution. *Science Advances*, 8(12):eabi8021.

Blair, D. C. (1979). Information Retrieval, 2nd ed. C.J. Van Rijsbergen. London: Butterworths; 1979. *Journal of the American Society for Information Science*, 30(6):374–375.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877.

Campbell, B. W., Marrs, F. W., Böhmelt, T., Fosdick, B. K., and Cranmer, S. J. (2019). Latent Influence Networks in Global Environmental Politics. *PLOS ONE*, 14(3):1–17.

Cantner, U. and Graf, H. (2006). The Network of Innovators in Jena: An Application of Social Network Analysis. *Research Policy*, 35(4):463–480.

Christen, P. (2012). *Data matching concepts and techniques for record linkage, entity resolution, and duplicate detection*. Data-centric systems and applications. Springer, Berlin.

Christen, P., Gayler, R., and Hawking, D. (2009). Similarity-Aware Indexing for Real-Time Entity Resolution. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, page 1565–1568, New York, NY, USA. Association for Computing Machinery.

Clayton, D. G., Bernardinelli, L., and Montomoli, C. (1993). Spatial Correlation in Ecological Analysis. *International Journal of Epidemiology*, 22(6):1193–1202.

Conte, M., Cotterlaz, P., and Mayer, T. (2022). The CEPII Gravity Database. Working Papers 2022-05, CEPII.

Czapinski, J. and Panek, T. (2015). Social diagnosis - objective and subjective quality of life in Poland. http://www.diagnoza.com/index-en.html. Accessed: 2022-06-09.

Dey, D., Mookerjee, V., and Liu, D. (2011). Efficient Techniques for Online Record Linkage. *IEEE Transactions on Knowledge and Data Engineering*, 23(3):373–387.

Duane, S., Kennedy, A., Pendleton, B. J., and Roweth, D. (1987). Hybrid Monte Carlo. *Physics Letters B*, 195(2):216–222.

D'Angelo, S., Murphy, T. B., and Alfò, M. (2019). Latent space modelling of multidimensional networks with application to the exchange of votes in Eurovision Song Contest. *The Annals of Applied Statistics*, 13(2):900 – 930.

Eurovision (2015). Eurovision Song Contest 2015 Grand Final. https://web.archive.org/web/20150924042946/http://www.eurovision.tv/page/history/by-year/contest?event=2083#Scoreboard. Accessed 2023-09-15 via Internet Archive.

Eurovision (2019). 182 million viewers tuned in to the 2019 Eurovision Song Contest. https://eurovision.tv/story/182-million-viewers-2019-eurovision-song-contest. Accessed 2023-09-14.

Eurovisionworld (2015). Odds Eurovision Song Contest 2015. https://eurovisionworld.com/odds/eurovision-2015. Accessed 2023-09-15.

Fellegi, I. P. and Sunter, A. B. (1969). A Theory for Record Linkage. *Journal of the American Statistical Association*, 64(328):1183–1210.

Fenn, D., Suleman, O., Efstathiou, J., and Johnson, N. F. (2006). How does Europe Make Its Mind Up? Connections, cliques, and compatibility between countries in the Eurovision Song Contest. *Physica A: Statistical Mechanics and its Applications*, 360(2):576–598.

Fleming, M., Kirby, B., and Penny, K. I. (2012). Record linkage in Scotland and its applications to health research. *Journal of Clinical Nursing*, 21(19pt20):2711–2721.

Fout, A. and Fosdick, B. K. (2023). Fréchet Covariance and MANOVA Tests for Random Objects in Multiple Metric Spaces. https://arxiv.org/abs/2306.12066.

Gelfand, A. E. and Smith, A. F. M. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85(410):398–409.

Gelman, A. (2006). Prior distributions for variance parameters in hierarchical models (comment on article by Browne and Draper). *Bayesian Analysis*, 1(3):515 – 534.

Gelman, A., Gilks, W. R., and Roberts, G. O. (1997). Weak convergence and optimal scaling of random walk Metropolis algorithms. *The Annals of Applied Probability*, 7(1):110 – 120.

Gelman, A. and Rubin, D. B. (1992). Inference from Iterative Simulation Using Multiple Sequences. *Statistical Science*, 7(4):457 – 472.

Ginsburgh, V. and Noury, A. G. (2008). The Eurovision Song Contest. Is Voting Political or Cultural? *European Journal of Political Economy*, 24(1):41–52.

Gordon, N., Salmond, D., and Smith, A. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F (Radar and Signal Processing)*, 140:107–113(6).

Gutman, R., Afendulis, C. C., and Zaslavsky, A. M. (2013). A Bayesian Procedure for File Linking to Analyze End-of-Life Medical Costs. *Journal of the American Statistical Association*, 108(501):34–47. PMID: 23645944.

Gwon, Y., Mo, M., Chen, M.-H., Chi, Z., Li, J., Xia, A. H., and Ibrahim, J. G. (2020). Network Meta-Regression for Ordinal Outcomes: Applications in Comparing Crohn's Disease Treatments. *Statistics in Medicine*, 39(13):1846–1870.

Haario, H., Saksman, E., and Tamminen, J. (2001). An Adaptive Metropolis Algorithm. *Bernoulli*, 7(2):223–242.

Hanks, E. M., Schliep, E. M., Hooten, M. B., and Hoeting, J. A. (2015). Restricted spatial regression in practice: geostatistical models, confounding, and robustness under model misspecification. *Environmetrics*, 26(4):243–254.

Hendry, D. F. and Richard, J.-F. (1992). Likelihood evaluation for dynamic latent variables models. In Amman, H. M., Belsley, D. A., and Pau, L. F., editors, *Computational Economics and Econometrics*, pages 3–17. Springer Netherlands, Dordrecht.

Hodges, J. S. and Reich, B. J. (2010). Adding Spatially-Correlated Errors Can Mess Up the Fixed Effect You Love. *The American Statistician*, 64(4):325–334.

Hof, M. H., Ravelli, A. C., and Zwinderman, A. H. (2017). A Probabilistic Record Linkage Model for Survival Data. *Journal of the American Statistical Association*, 112(520):1504–1515.

Hoff, P. (2021). Additive and Multiplicative Effects Network Models. *Statistical Science*, 36(1):34 – 50.

Hoff, P., Fosdick, B., and Volfovsky, A. (2020). *amen: Additive and Multiplicative Effects Models for Networks and Relational Data*. R package version 1.4.4.

Hoff, P., Fosdick, B., Volfovsky, A., and Stovel, K. (2013). Likelihoods for fixed rank nomination networks. *Network Science*, 1(3):253–277.

Hoff, P. D. (2005). Bilinear mixed-effects models for dyadic data. *Journal of the American Statistical Association*, 100(469):286–295.

Hoff, P. D., Raftery, A. E., and Handcock, M. S. (2002). Latent Space Approaches to Social Network Analysis. *Journal of the American Statistical Association*, 97(460):1090–1098.

Hoffman, M. D. and Gelman, A. (2014). The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(47):1593–1623.

Holland, P. W., Laskey, K. B., and Leinhardt, S. (1983). Stochastic Blockmodels: First Steps. *Social Networks*, 5(2):109–137.

Hooten, M. B., Johnson, D. S., and Brost, B. M. (2021). Making Recursive Bayesian Inference Accessible. *The American Statistician*, 75(2):185–194.

Hughes, J. and Haran, M. (2013). Dimension Reduction and Alleviation of Confounding for Spatial Generalized Linear Mixed Models. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 75(1):139–159.

Ioannou, E., Nejdl, W., Niederée, C., and Velegrakis, Y. (2010). On-the-Fly Entity-Aware Query Processing in the Presence of Linkage. *Proc. VLDB Endow.*, 3(1–2):429–438.

Johnson, D. S. (2017). *agTrend: Estimate Linear Trends for Aggregated Abundance Data*. R package version 0.17.7.

Kaplan, A., Betancourt, B., and Steorts, R. C. (2023). A Practical Approach to Proper Inference with Linked Data. *The American Statistician*, 118(543):1786–1795. PMID: 37771512.

Karapiperis, D., Gkoulalas-Divanis, A., and Verykios, V. S. (2018). Summarization Algorithms for Record Linkage. In Böhlen, M. H., Pichler, R., May, N., Rahm, E., Wu, S., and Hose, K., editors, *Proceedings of the 21st International Conference on Extending Database Technology, EDBT 2018, Vienna, Austria, March 26-29, 2018*, pages 73–84. OpenProceedings.org.

Khan, K. and Calder, C. A. (2022). Restricted Spatial Regression Methods: Implications for Inference. *Journal of the American Statistical Association*, 117(537):482–494.

Kitagawa, G. (1996). Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models. *Journal of Computational and Graphical Statistics*, 5(1):1–25.

Lee, J. W. and Sohn, S. Y. (2022). Evaluating borrowers' default risk with a spatial probit model reflecting the distance in their relational network. *PLOS ONE*, 16(12):1–11.

Li, H., Chen, M.-H., Ibrahim, J. G., Kim, S., Shah, A. K., Lin, J., and Tershakovec, A. M. (2018). Bayesian Inference for Network Meta-Regression Using Multivariate Random Effects With Applications to Cholesterol Lowering Drugs. *Biostatistics*, 20(3):499–516.

Li, H. and Loken, E. (2002). A unified theory of statistical analysis and inference for variance component models for dyadic data. *Statistica Sinica*, 12(2):519–535.

Liu, J. S. and Chen, R. (1995). Blind deconvolution via sequential imputations. *Journal of the American Statistical Association*, 90(430):567–576.

Lunn, D., Barrett, J., Sweeting, M., and Thompson, S. (2013). Fully Bayesian hierarchical modelling in two stages, with application to meta-analysis. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 62(4):551–572.

Marchant, N. G., Kaplan, A., Elazar, D. N., Rubinstein, B. I. P., and Steorts, R. C. (2021). d-blink: Distributed End-to-End Bayesian Entity Resolution. *Journal of Computational and Graphical Statistics*, 30(2):406–421.

Marrs, F. W., Campbell, B. W., Fosdick, B. K., Cranmer, S. J., and Böhmelt, T. (2020). Inferring Influence Networks from Longitudinal Bipartite Relational Data. *Journal of Computational and Graphical Statistics*, 29(3):419–431.

Marrs, F. W., Fosdick, B. K., and Mccormick, T. H. (2022). Regression of exchangeable relational arrays. *Biometrika*, 110(1):265–272.

McVeigh, B. S., Spahn, B. T., and Murray, J. S. (2019). Scaling Bayesian Probabilistic Record Linkage with Post-Hoc Blocking: An Application to the California Great Registers. https://arxiv.org/abs/1905.05337.

Moreira-Matias, L., Gama, J., Ferreira, M., Mendes-Moreira, J., and Damas, L. (2013). Predicting taxi–passenger demand using streaming data. *IEEE Transactions on Intelligent Transportation Systems*, 14(3):1393–1402.

Pettitt, A. N. (1982). Inference for the Linear Model Using a Likelihood Based on Ranks. *Journal of the Royal Statistical Society. Series B (Methodological)*, 44(2):234–243.

Prates, M. O., Assunção, R. M., and Rodrigues, E. C. (2019). Alleviating Spatial Confounding for Areal Data Problems by Displacing the Geographical Centroids. *Bayesian Analysis*, 14(2):623 – 647.

Reich, B. J., Hodges, J. S., and Carlin, B. P. (2007). Spatial Analyses of Periodontal Data Using Conditionally Autoregressive Priors Having Two Classes of Neighbor Relations. *Journal of the American Statistical Association*, 102(477):44–55.

Reich, B. J., Hodges, J. S., and Zadnik, V. (2006). Effects of Residual Smoothing on the Posterior of the Fixed Effects in Disease-Mapping Models. *Biometrics*, 62(4):1197–1206.

Robert, C. and Casella, G. (2013). *Monte Carlo statistical methods*. Springer texts in statistics. Springer.

Robins, G., Pattison, P., Kalish, Y., and Lusher, D. (2007). An Introduction to Exponential Random Graph (p*) Models for Social Networks. *Social Networks*, 29(2):173–191. Special Section: Advances in Exponential Random Graph (p*) Models.

Sadinle, M. (2014). Detecting duplicates in a homicide registry using a Bayesian partitioning approach. *The Annals of Applied Statistics*, 8(4):2404 – 2434.

Sadinle, M. (2017). Bayesian Estimation of Bipartite Matchings for Record Linkage. *Journal of the American Statistical Association*, 112(518):600–612.

Sadinle, M. and Fienberg, S. E. (2013). A Generalized Fellegi–Sunter Framework for Multiple Record Linkage With Application to Homicide Record Systems. *Journal of the American Statistical Association*, 108(502):385–397.

Sengupta, S. and Chen, Y. (2018). A Block Model for Node Popularity in Networks With Community Structure. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(2):365–386.

Spann, M. and Skiera, B. (2009). Sports forecasting: A comparison of the forecast accuracy of prediction markets, betting odds and tipsters. *Journal of Forecasting*, 28(1):55–72.

Spierdijk, L. and Vellekoop, M. (2009). The structure of bias in peer voting systems: Lessons from the Eurovision Song Contest. *Empirical Economics*, 36(2):403–425.

Steorts, R. C. (2015). Entity Resolution with Empirically Motivated Priors. *Bayesian Analysis*, 10(4):849 – 875.

Steorts, R. C., Hall, R., and Fienberg, S. E. (2016). A Bayesian Approach to Graphical Record Linkage and Deduplication. *Journal of the American Statistical Association*, 111(516):1660–1672.

Särkkä, S. (2013). *Bayesian Filtering and Smoothing*. Institute of Mathematical Statistics Textbooks. Cambridge University Press.

Tancredi, A. and Liseo, B. (2011). A hierarchical Bayesian approach to record linkage and population size problems. *The Annals of Applied Statistics*, 5(2B):1553 – 1585.

Tavaré, S., Balding, D. J., Griffiths, R. C., and Donnelly, P. (1997). Inferring Coalescence Times From DNA Sequence Data. *Genetics*, 145(2):505–518.

Taylor, I., Kaplan, A., and Betancourt, B. (2022). *bstrl: Bayesian Streaming Record Linkage*. R package version 1.0.2.

Tran, K.-N., Vatsalan, D., and Christen, P. (2013). GeCo: An Online Personal Data Generator and Corruptor. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management*, CIKM '13, page 2473–2476, New York, NY, USA. Association for Computing Machinery.

United Nations (2015). The World Population Prospects: 2015 Revision. https://www.un.org/en/development/desa/publications/world-population-prospects-2015-revision.html.

Vatsalan, D., Sehili, Z., Christen, P., and Rahm, E. (2017). Privacy-Preserving Record Linkage for Big Data: Current Approaches and Research Challenges. In Zomaya, A. Y. and Sakr, S., editors, *Handbook of Big Data Technologies*, pages 851–895. Springer International Publishing, Cham.

Wang, Y. J. and Wong, G. Y. (1987). Stochastic Blockmodels for Directed Graphs. *Journal of the American Statistical Association*, 82(397):8–19.

Warner, R. M., Kenny, D. A., and Stoto, M. (1979). A New Round Robin Analysis of Variance for Social Interaction Data. *Journal of Personality and Social Psychology*, 37(10):1742–1757.

Winkler, W. E. (2006). Overview of Record Linkage and Current Research Directions. Technical report, U.S. Bureau of the Census Statistical Research Division.

World Bank (2022). GDP per capita (constant 2015 US$). https://data.worldbank.org/indicator/NY.GDP.PCAP.KD. Accessed 2023-09-15.

Wortman, J. P. H. (2019). *Record Linkage Methods with Applications to Causal Inference and Election Voting Data*. PhD thesis, Duke University.

Yair, G. (1995). 'Unite Unite Europe' The Political and Cultural Structures of Europe as Reflected in the Eurovision Song Contest. *Social Networks*, 17(2):147–161.

Yang, Y. and Dunson, D. B. (2013). Sequential Markov Chain Monte Carlo. https://arxiv.org/abs/1308.3861.

Zachary, W. W. (1977). An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33(4):452–473.

Zanella, G. (2020). Informed Proposals for Local MCMC in Discrete Spaces. *Journal of the American Statistical Association*, 115(530):852–865.

Zhou, Y., Reid, E., Qin, J., Chen, H., and Lai, G. (2005). US domestic extremist groups on the Web: Link and content analysis. *IEEE Intelligent Systems*, 20(5):44–51.

Zimmerman, D. L. and Hoef, J. M. V. (2022). On deconfounding spatial confounding in linear models. *The American Statistician*, 76(2):159–167.

Appendix A

Supplement to *Restricted Regression in Networks*

A.1 Theoretical Results

A.1.1 Proof of Theorem 2.3.1

We use the laws of total expectation and variance. Starting with the full conditional distribution of $\boldsymbol{\delta}$,

$$\boldsymbol{\delta}|\boldsymbol{a}, \boldsymbol{b}, \sigma_a^2, \sigma_b^2, \sigma_\varepsilon^2, \boldsymbol{y} \sim N\left((\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top \boldsymbol{y}, \sigma_\varepsilon^2 (\boldsymbol{X}^\top \boldsymbol{X})^{-1}\right),$$

we have

$$\mathrm{E}[\boldsymbol{\delta}|\boldsymbol{y}] = \mathrm{E}[\mathrm{E}[\boldsymbol{\delta}|\boldsymbol{a}, \boldsymbol{b}, \sigma_a^2, \sigma_b^2, \sigma_\varepsilon^2, \boldsymbol{y}]|\boldsymbol{y}] = (\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top \boldsymbol{y} \tag{A.1}$$

$$\mathrm{Var}(\boldsymbol{\delta}|\boldsymbol{y}) = \mathrm{E}[\mathrm{Var}(\boldsymbol{\delta}|\boldsymbol{a}, \boldsymbol{b}, \sigma_a^2, \sigma_b^2, \sigma_\varepsilon^2, \boldsymbol{y})|\boldsymbol{y}] + \mathrm{Var}(\mathrm{E}[\boldsymbol{\delta}|\boldsymbol{a}, \boldsymbol{b}, \sigma_a^2, \sigma_b^2, \sigma_\varepsilon^2, \boldsymbol{y}]|\boldsymbol{y}) \tag{A.2}$$

$$= \mathrm{E}[\sigma_\varepsilon^2 (\boldsymbol{X}^\top \boldsymbol{X})^{-1}|\boldsymbol{y}] \tag{A.3}$$

$$= (\boldsymbol{X}^\top \boldsymbol{X})^{-1}\mathrm{E}[\sigma_\varepsilon^2|\boldsymbol{y}]. \tag{A.4}$$

The above identities depend on having a proper joint posterior distribution $f(\boldsymbol{\delta}, \boldsymbol{a}, \boldsymbol{b}, \sigma_a^2, \sigma_b^2, \sigma_\varepsilon^2|\boldsymbol{y})$ and a finite posterior expectation $\mathrm{E}[\sigma_\varepsilon^2|\boldsymbol{y}] < \infty$ which are both true assuming proper priors for σ_a^2, σ_b^2 and σ_ε^2 with a finite prior mean $\mathrm{E}[\sigma_\varepsilon^2] < \infty$.

□

A.1.2 Proof of Theorem 2.3.2

We first simplify the model by combining the random effects as follows:

$$\boldsymbol{\eta}_* := \begin{bmatrix} \boldsymbol{\eta}_1 \\ \boldsymbol{\eta}_2 \end{bmatrix} \tag{A.5}$$

$$\boldsymbol{W}_* := \begin{bmatrix} \boldsymbol{W}_1 & \boldsymbol{W}_2 \end{bmatrix} \tag{A.6}$$

$$\boldsymbol{F}_* := \boldsymbol{F}_*(r_1, r_2) = \begin{bmatrix} r_1\boldsymbol{F}_1 & 0 \\ 0 & r_2\boldsymbol{F}_2 \end{bmatrix} \tag{A.7}$$

Then we can rewrite the model as

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\delta} + \boldsymbol{W}_*\boldsymbol{\eta}_* + \boldsymbol{\epsilon} \tag{A.8}$$

$$p(\boldsymbol{\eta}_*|\tau_\epsilon, r_1, r_2) = p(\boldsymbol{\eta}_1|\tau_\epsilon, r_1)p(\boldsymbol{\eta}_2|\tau_\epsilon, r_2) \tag{A.9}$$

$$\propto \tau_\epsilon^{\mathrm{rank}(\boldsymbol{F}_*)/2} r_1^{\mathrm{rank}(\boldsymbol{F}_1)/2} r_2^{\mathrm{rank}(\boldsymbol{F}_2)/2} \exp\left\{-\frac{\tau_\epsilon}{2}\boldsymbol{\eta}_*^\top \boldsymbol{F}_*\boldsymbol{\eta}_*\right\}. \tag{A.10}$$

Note that $\mathrm{rank}(\boldsymbol{F}_*) = \mathrm{rank}(\boldsymbol{F}_1) + \mathrm{rank}(\boldsymbol{F}_2)$, and $\boldsymbol{W}_*$ has orthonormal columns because $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ have orthonormal columns and $\mathcal{C}(\boldsymbol{W}_1) \perp \mathcal{C}(\boldsymbol{W}_2)$.

We know that $\mathrm{Var}(\boldsymbol{\delta}) = (\boldsymbol{X}^\top\boldsymbol{X})^{-1}\mathrm{E}[\sigma_\epsilon^2|\boldsymbol{y}]$ where $\sigma_\epsilon^2 = 1/\tau_\epsilon$, so it suffices to show that $\mathrm{E}[\sigma_\epsilon^2|\boldsymbol{y}] \leq \mathrm{E}[\sigma_{\epsilon,NN}^2|\boldsymbol{y}]$. We find the posterior and conditional posterior distributions,

$$\sigma_{\epsilon,NN}^2|\boldsymbol{y} \sim \text{inverse-gamma}\left(a_\epsilon + (n-p)/2, \frac{1}{b_\epsilon} + \frac{1}{2}\boldsymbol{y}^\top\boldsymbol{P}_{X^\perp}\boldsymbol{y}\right) \tag{A.11}$$

$$\sigma_\epsilon^2|\boldsymbol{y}, r_1, r_2 \sim \text{inverse-gamma}\left(\begin{array}{l} a_\epsilon + a_1 + a_2 + (n-p)/2, \\ \frac{1}{b_\epsilon} + \frac{r_1}{b_1} + \frac{r_2}{b_2} + \frac{1}{2}\boldsymbol{y}^\top(\boldsymbol{P}_{X^\perp} - \boldsymbol{W}_*(\boldsymbol{I} + \boldsymbol{F}_*)^{-1}\boldsymbol{W}_*^\top)\boldsymbol{y} \end{array}\right) \tag{A.12}$$

Then,

$$\mathrm{E}\left[\sigma_\epsilon^2|\boldsymbol{y}, r_1, r_2\right] = \frac{\frac{1}{b_\epsilon} + \frac{r_1}{b_1} + \frac{r_2}{b_2} + \frac{1}{2}\boldsymbol{y}^\top(\boldsymbol{P}_{X^\perp} - \boldsymbol{W}_*(\boldsymbol{I} + \boldsymbol{F}_*)^{-1}\boldsymbol{W}_*^\top)\boldsymbol{y}}{a_\epsilon + a_1 + a_2 + (n-p)/2 - 1}.$$

Using total expectation,

$$\mathrm{E}\left[\sigma_\epsilon^2|\boldsymbol{y}\right] = \frac{\frac{1}{b_\epsilon} + \frac{\mathrm{E}[r_1|\boldsymbol{y}]}{b_1} + \frac{\mathrm{E}[r_2|\boldsymbol{y}]}{b_2} + \frac{1}{2}\boldsymbol{y}^\top \boldsymbol{P}_{X^\perp}\boldsymbol{y} - \mathrm{E}\left[\frac{1}{2}\boldsymbol{y}^\top \boldsymbol{W}_*(\boldsymbol{I}+\boldsymbol{F}_*)^{-1}\boldsymbol{W}_*^\top \boldsymbol{y}|\boldsymbol{y}\right]}{a_\epsilon + a_1 + a_2 + (n-p)/2 - 1}.$$

Now define

$$M(x) := \frac{\frac{1}{b_\epsilon} + \mathrm{E}[r_1|\boldsymbol{y}]x + \frac{b_1\mathrm{E}[r_2|\boldsymbol{y}]}{b_2}x + \frac{1}{2}\boldsymbol{y}^\top \boldsymbol{P}_{X^\perp}\boldsymbol{y}}{a_\epsilon + a_1b_1x + a_2b_1x + (n-p)/2 - 1},$$

and similary note that $M(0) = E[\sigma^2_{\epsilon,NN}|\boldsymbol{y}]$ and $M(1/b_1) \leq E[\sigma^2_\epsilon]$. We want to find conditions such that M is decreasing in x on $[0, 1/b_1]$.

$$\begin{aligned}
\frac{\partial M}{\partial x} &= \frac{(a_\epsilon + a_1b_1x + a_2b_1x + (n-p)/2 - 1)(\mathrm{E}[r_1|\boldsymbol{y}] + \frac{b_1\mathrm{E}[r_2|\boldsymbol{y}]}{b_2})}{(a_\epsilon + a_1b_1x + a_2b_1x + (n-p)/2 - 1)^2} \\
&\quad - \frac{(\frac{1}{b_\epsilon} + \mathrm{E}[r_1|\boldsymbol{y}]x + \frac{b_1\mathrm{E}[r_2|\boldsymbol{y}]}{b_2}x + \frac{1}{2}\boldsymbol{y}^\top \boldsymbol{P}_{X^\perp}\boldsymbol{y})(a_1b_1 + a_2b_1)}{(a_\epsilon + a_1b_1x + a_2b_1x + (n-p)/2 - 1)^2} \qquad \text{(A.13)} \\
&= \frac{(a_\epsilon + (n-p)/2 - 1)(\mathrm{E}[r_1|\boldsymbol{y}] + \frac{b_1\mathrm{E}[r_2|\boldsymbol{y}]}{b_2}) - (\frac{1}{b_\epsilon} + \frac{1}{2}\boldsymbol{y}^\top \boldsymbol{P}_{X^\perp}\boldsymbol{y})(a_1b_1 + a_2b_1)}{(a_\epsilon + a_1b_1x + a_2b_1x + (n-p)/2 - 1)^2} \qquad \text{(A.14)} \\
&= \frac{\left\{(\mathrm{E}[r_1|\boldsymbol{y}] + \frac{b_1\mathrm{E}[r_2|\boldsymbol{y}]}{b_2}) - E[\sigma^2_{\epsilon,NN}|\boldsymbol{y}](a_1b_1 + a_2b_1)\right\}(a_\epsilon + (n-p)/2 - 1)}{(a_\epsilon + a_1b_1x + a_2b_1x + (n-p)/2 - 1)^2}. \qquad \text{(A.15)}
\end{aligned}$$

So we have $\frac{\partial M}{\partial x} \leq 0$ if $(\mathrm{E}[r_1|\boldsymbol{y}] + \frac{b_1\mathrm{E}[r_2|\boldsymbol{y}]}{b_2})/(a_1b_1 + a_2b_1) \leq E[\sigma^2_{\epsilon,NN}|\boldsymbol{y}]$, or equivalently if

$$\frac{\mathrm{E}[r_1|\boldsymbol{y}]/b_1 + \mathrm{E}[r_2|\boldsymbol{y}]/b_2}{\mathrm{E}[\tau_1]/b_1 + \mathrm{E}[\tau_2]/b_2} \leq E[\sigma^2_{\epsilon,NN}|\boldsymbol{y}].$$

□

A.1.3 Restricted Network Regression With a Single Random Effect

The need for Theorem 2.3.2 is motivated by the fact that the model form in (2.35)-(2.37) encompasses continuous restricted network models that include one additive sender or one additive receiver random effect, but not both. Taking the sender random effect as an example, we obtain the Restricted Network Regression model through the relations $\boldsymbol{\eta} = \boldsymbol{a}$, $\boldsymbol{W} = (\boldsymbol{I} - \boldsymbol{P}_X)\boldsymbol{A}$, $\boldsymbol{F} = \boldsymbol{I}_n$, and $\tau_s = \sigma_a^{-2}$ (Table A.1). From the results in Khan and Calder (2022), any restricted regression

Table A.1: Table showing the mapping between the general form of spatial models studied in Khan and Calder (2022), (2.35)-(2.37), and additive network models with a single sender random effect.

Spatial	Restricted Network	Description
$\boldsymbol{\eta}$	$\boldsymbol{a}$	Random effect
$\boldsymbol{W}$	$(\boldsymbol{I} - \boldsymbol{P}_X)\boldsymbol{A}$	Random effect design matrix
$\boldsymbol{F}$	$\boldsymbol{I}$	Random effect precision matrix
τ_s	σ_a^{-2}	Random effect precision

model of the form in (2.35)-(2.37) with a continuous response will also not alleviate spatial (or network) confounding according to Definition 2.2.1.

A.2 Simulation of excess variation at specified canonical correlations

The simulation studies in Section 2.4 depend on the ability to generate a vector of unobserved variation at a specified magnitude and a specified canonical correlation with a set of covariates. This is achieved through rescaling the parallel and orthogonal components of an initial vector through the following `R` function:

```
gencancor <- function(X, Y, rho) {
  PX <- X %*% MASS::ginv(t(X) %*% X) %*% t(X)
  Ypar <- c(PX %*% Y)
  Yperp <- Y - Ypar
  Ynew <- (rho * sqrt(sum(Yperp^2)) * Ypar +
      sqrt(1-rho^2) * sqrt(sum(Ypar^2)) * Yperp)
  return(Ynew * sqrt(sum(Y^2)) / sqrt(sum(Ynew^2)))
}
```

This function works by taking an initial, i.i.d. normal random vector Y and decomposing it into components parallel to and orthogonal to X. Those components are then scaled and recombined

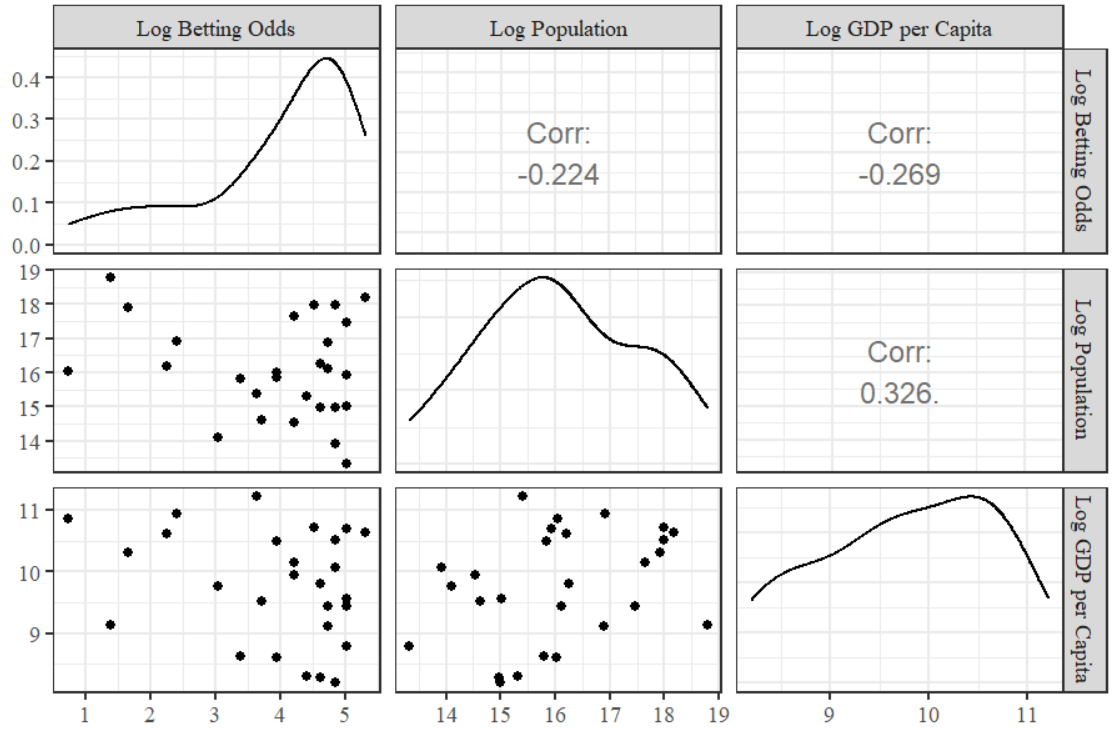


Figure A.1: Pairs plot of the 3 receiver covariates in the Eurovision data analysis. There is low correlation between each pair of covariates. LogMedianOdds and LogGDP appear left skewed, LogPopulation is approximately normally-distributed.

into a vector Y', such that Y' has the desired canonical correlation ρ with X. The vector Y' is then rescaled to have the same magnitude as the original Y and returned.

A.3 Eurovision Data

Figures A.1 and A.2 visualize the covariates in the Eurovision data analysis (Section 2.5). Log-transforming the receiver covariates (Figure A.1) results in no right skew. The receiver covariates are also not highly correlated with each other, avoiding multicollinearity in the model. As the Eurovision analysis found that country contiguity had a positive effect on voting, countries with large numbers of neighbors (e.g., Russia, Hungary) are more likely to receive votes, while countries with fewer neighbors (e.g., Australia, Cyprus, Israel, and the United Kingdom) are less likely to receive votes.

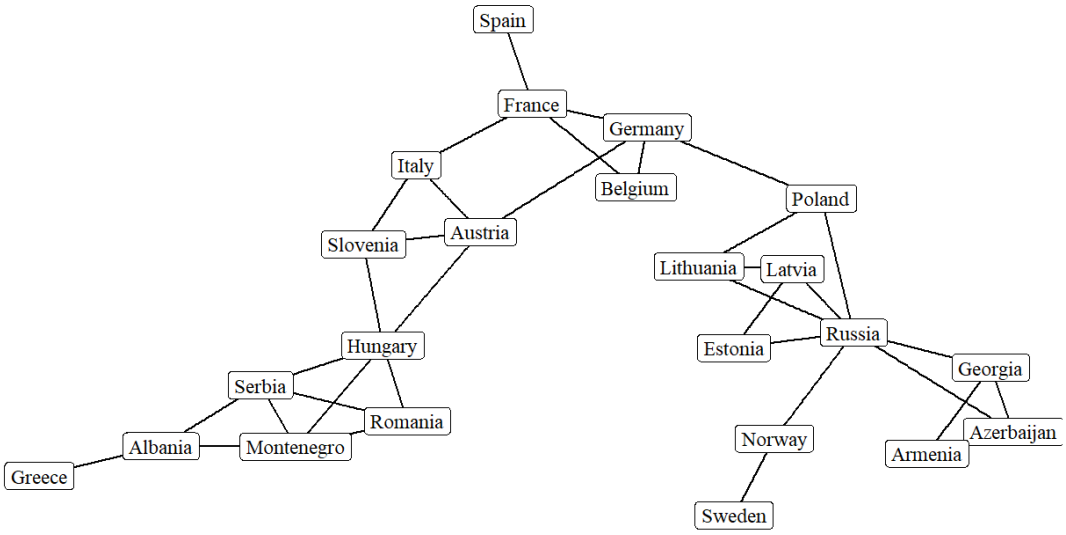


Figure A.2: Eurovision country contiguity covariate illustrated as a network. Countries sharing a border are connected by an edge. Australia, Cyprus, Israel, and the United Kingdom border no other competing countries and are not shown.

Appendix B

Supplement to *Fast Bayesian Record Linkage for Streaming Data Contexts*

B.1 Supplemental Figures and Tables

Figure B.1 depicts the streaming record linkage problem up to time T_k.

Table B.1 and Figure B.2 show F1-scores and entity errors for additional error levels from the simulation in Section 3.4.

B.2 Posterior and Full Conditional Distributions

B.2.1 Posterior Distribution

Here we specify a function that is proportional to the full streaming record linkage posterior density.

$$P(\boldsymbol{m}, \boldsymbol{u}, \boldsymbol{Z}^{(1)}, \ldots, \boldsymbol{Z}^{(k-1)} | \boldsymbol{\Gamma}^{(1)}, \ldots, \boldsymbol{\Gamma}^{(k-1)}) \tag{B.1}$$

$$\propto P(\boldsymbol{m}) P(\boldsymbol{u}) P(\boldsymbol{Z}^{(1)}) \cdots P(\boldsymbol{Z}^{(k-1)}) P(\boldsymbol{\Gamma}^{(1)}, \ldots, \boldsymbol{\Gamma}^{(k-1)} | \boldsymbol{m}, \boldsymbol{u}, \boldsymbol{Z}^{(1)}, \ldots, \boldsymbol{Z}^{(k-1)}) \tag{B.2}$$

$$\begin{aligned} &\propto \prod_{f=1}^{F} \prod_{\ell=0}^{L_f} m_{f\ell}^{a_{f\ell}} u_{f\ell}^{b_{f\ell}} \\ &\quad \times \prod_{t=2}^{k} \left[\frac{(N_{t-1} - n_{t\cdot}(\boldsymbol{Z}^{(t-1)}))!}{N_{t-1}!} \cdot \frac{\mathrm{B}(n_{t\cdot}(\boldsymbol{Z}^{(t-1)}) + \alpha_\pi, n_t - n_{t\cdot}(\boldsymbol{Z}^{(t-1)}) + \beta_\pi)}{\mathrm{B}(\alpha_\pi, \beta_\pi)} \right] \\ &\quad \times \prod_{t_1 < t_2}^{k} \prod_{i=1}^{n_{t_1}} \prod_{j=1}^{n_{t_2}} \prod_{f=1}^{F} \prod_{\ell=0}^{L_f} \left[m_{f\ell}^{\mathbb{I}((\boldsymbol{x}_{t_1 i}, \boldsymbol{x}_{t_2 j}) \in M)} u_{f\ell}^{\mathbb{I}((\boldsymbol{x}_{t_1 i}, \boldsymbol{x}_{t_2 j}) \notin M)} \right]^{\gamma^{f\ell}(\boldsymbol{x}_{t_1 i}, \boldsymbol{x}_{t_2 j})}, \end{aligned} \tag{B.3}$$

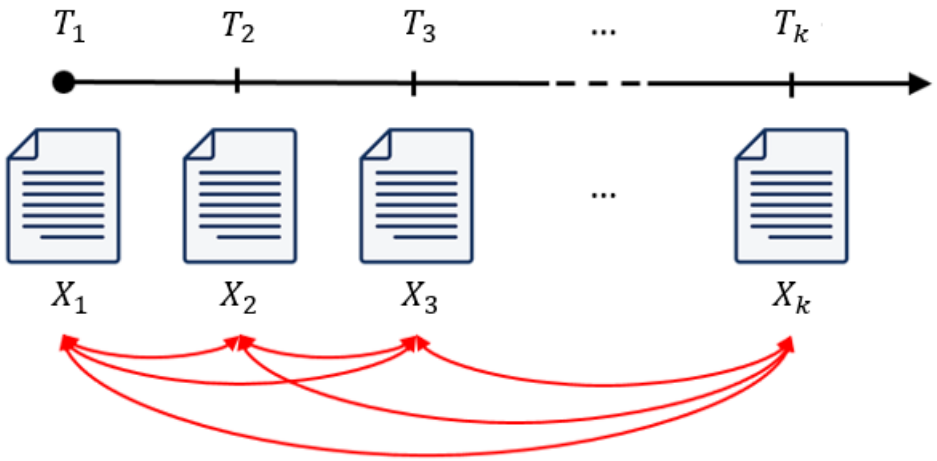


Figure B.1: A depiction of the streaming record linkage problem up to time T_k. Files 1 through k arrive sequentially and are duplicate-free. The red arrows illustrate the growing complexity of the linkage problem on multiple files: with k files, records in $k(k-1)/2$ pairs of files must be compared and linked.

Table B.1: Posterior means and standard deviations of F_1-score for simulated datasets. Within rows, each model is listed: the model presented in this chapter (Streaming) and three comparison models. Larger values represent more accurate links in the posterior distribution. The support vector machine, a non-bayesian method, is represented only by the F_1-score of its resulting point estimate.

Model	10% overlap	30% overlap	50% overlap	90% overlap
Errors: 1				
Streaming (Flat Prior)	0.999 (0.0026)	0.988 (0.0003)	**0.999 (0.0010)**	0.998 (0.0000)
Streaming (Weak Prior)	0.998 (0.0038)	**1.000 (0.0001)**	**0.999 (0.0012)**	**1.000 (0.0000)**
Streaming (Strong Prior)	0.988 (0.0067)	0.995 (0.0016)	0.992 (0.0011)	**1.000 (0.0000)**
Multilink	0.987 (0.0088)	0.995 (0.0021)	0.982 (0.0010)	0.915 (0.0000)
Multilink (Single Likelihood)	0.999 (0.0035)	0.995 (0.0004)	0.991 (0.0011)	0.946 (0.0000)
Blink	0.869 (0.0136)	0.988 (0.0007)	**0.999 (0.0007)**	**1.000 (0.0000)**
SVM (1% training)	**1.000**	0.998	0.997	**1.000**
Errors: 3				
Streaming (Flat Prior)	0.955 (0.0195)	0.990 (0.0058)	0.987 (0.0021)	0.995 (0.0001)
Streaming (Weak Prior)	0.970 (0.0193)	0.990 (0.0057)	**0.996 (0.0021)**	**1.000 (0.0001)**
Streaming (Strong Prior)	**0.978 (0.0159)**	0.983 (0.0055)	**0.996 (0.0022)**	**1.000 (0.0001)**
Multilink	0.095 (0.0055)	0.981 (0.0059)	0.983 (0.0027)	0.954 (0.0000)
Multilink (Single Likelihood)	0.940 (0.0210)	**0.991 (0.0052)**	0.985 (0.0023)	**1.000 (0.0000)**
Blink	0.543 (0.0176)	0.944 (0.0031)	0.988 (0.0023)	0.999 (0.0002)
SVM (1% training)	0.933	0.958	0.984	0.974
Errors: 8				
Streaming (Flat Prior)	0.231 (0.0077)	0.414 (0.0093)	0.822 (0.0163)	0.950 (0.0031)
Streaming (Weak Prior)	0.240 (0.0085)	0.415 (0.0103)	0.843 (0.0157)	0.911 (0.0026)
Streaming (Strong Prior)	**0.710 (0.0277)**	**0.817 (0.0128)**	**0.898 (0.0075)**	0.908 (0.0030)
Multilink	0.204 (0.0084)	0.372 (0.0084)	0.647 (0.0097)	**0.977 (0.0032)**
Multilink (Single Likelihood)	0.136 (0.0057)	0.369 (0.0070)	0.647 (0.0097)	0.972 (0.0022)
Blink	0.340 (0.0216)	0.663 (0.0104)	0.836 (0.0140)	0.918 (0.0080)
SVM (1% training)	0.586	0.482	0.556	0.542

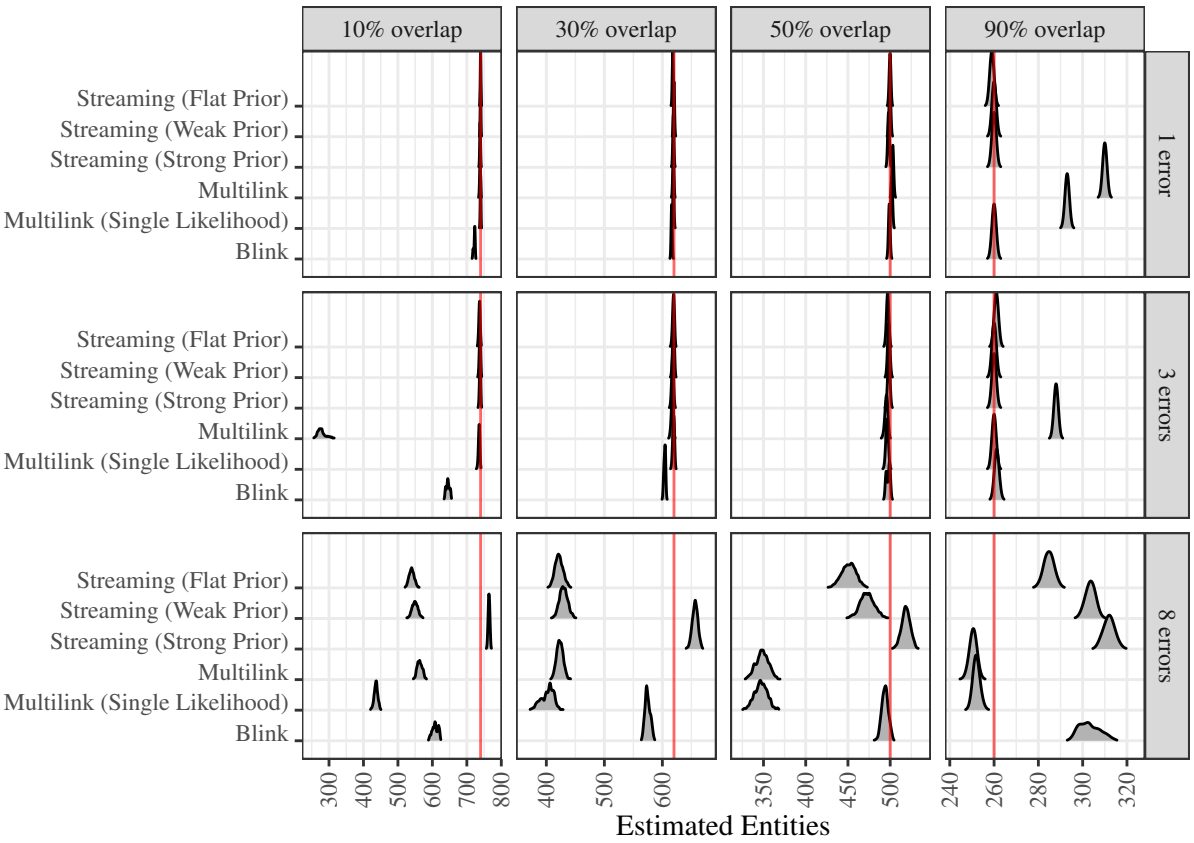


Figure B.2: Posterior distribution of the number of estimated entities for simulated datasets. A vertical line indicates the true number of distinct entities in each dataset. Distributions to the right or left of the vertical line indicate underlinking or overlinking, respectively, in the posterior. Compared models are on the y-axis: the model presented in this chapter (Streaming) and three comparison models.

where

$$N_{t-1} = n_1 + \cdots + n_{t-1}$$
$$n_{t\cdot}(\boldsymbol{Z}^{(t-1)}) = \sum_{j=1}^{n_t} \mathbb{I}(Z_j^{(t-1)} \leq N_{t-1})$$
$$M := M(\boldsymbol{Z}^{(1)}, \ldots, \boldsymbol{Z}^{(k-1)}) = \{(\boldsymbol{x}_{t_1 i}, \boldsymbol{x}_{t_2 j}) : \boldsymbol{x}_{t_1 i} \text{ and } \boldsymbol{x}_{t_2 j} \text{ are linked}\}.$$

B.2.2 Full conditional for $\boldsymbol{m}$ and $\boldsymbol{u}$

We provide the full conditional distribution for $\boldsymbol{m}$ starting from the posterior in Equation B.3.

$$P(\boldsymbol{m}|\boldsymbol{u}, \boldsymbol{Z}^{(1)}, \ldots, \boldsymbol{Z}^{(k-1)}, \boldsymbol{\Gamma}^{(1)}, \ldots, \boldsymbol{\Gamma}^{(k-1)}) \tag{B.4}$$
$$\propto P(\boldsymbol{m}, \boldsymbol{u}, \boldsymbol{Z}^{(1)}, \ldots, \boldsymbol{Z}^{(k-1)}|\boldsymbol{\Gamma}^{(1)}, \ldots, \boldsymbol{\Gamma}^{(k-1)}) \tag{B.5}$$
$$\propto \prod_{f=1}^{F} \prod_{\ell=0}^{L_f} m_{f\ell}^{a_{f\ell} + \sum_{t_1<t_2}^{k} \sum_{i=1}^{n_{t_1}} \sum_{j=1}^{n_{t_2}} \mathbb{I}((\boldsymbol{x}_{t_1 i}, \boldsymbol{x}_{t_2 j}) \in M) \cdot \gamma^{f\ell}(\boldsymbol{x}_{t_1 i}, \boldsymbol{x}_{t_2 j})}. \tag{B.6}$$

We recognize the inside products in Equation B.6 as the kernel of a Dirichlet distribution, and so each vector $\boldsymbol{m}_f$ for $f = 1, \ldots, F$ has a conjugate Dirichlet full conditional distribution. Similarly, we can derive

$$P(\boldsymbol{u}|\boldsymbol{m}, \boldsymbol{Z}^{(1)}, \ldots, \boldsymbol{Z}^{(k-1)}, \boldsymbol{\Gamma}^{(1)}, \ldots, \boldsymbol{\Gamma}^{(k-1)})$$
$$\propto \prod_{f=1}^{F} \prod_{\ell=0}^{L_f} u_{f\ell}^{b_{f\ell} + \sum_{t_1<t_2}^{k} \sum_{i=1}^{n_{t_1}} \sum_{j=1}^{n_{t_2}} \mathbb{I}((\boldsymbol{x}_{t_1 i}, \boldsymbol{x}_{t_2 j}) \notin M) \cdot \gamma^{f\ell}(\boldsymbol{x}_{t_1 i}, \boldsymbol{x}_{t_2 j})}, \tag{B.7}$$

and so each vector $\boldsymbol{u}_f$ for $f = 1, \ldots, F$ also has a conjugate Dirichlet full conditional distribution.

B.2.3 Full conditional for $\boldsymbol{Z}^{(t-1)}$

Let T be a file number, $2 \leq T \leq k$. We derive the full conditional distribution for $\boldsymbol{Z}^{(T-1)}$, the matching vector introduced with file X_T, starting from the posterior in Equation B.3.

$$P(\boldsymbol{Z}^{(T-1)}|\boldsymbol{m}, \boldsymbol{u}, \boldsymbol{Z}^{(1)}, \ldots, \boldsymbol{Z}^{(T-2)}, \boldsymbol{Z}^{(T)}, \ldots, \boldsymbol{Z}^{(k-1)}, \boldsymbol{\Gamma}^{(1)}, \ldots, \boldsymbol{\Gamma}^{(k-1)}) \tag{B.8}$$

$$\propto P(\boldsymbol{m}, \boldsymbol{u}, \boldsymbol{Z}^{(1)}, \ldots, \boldsymbol{Z}^{(k-1)}|\boldsymbol{\Gamma}^{(1)}, \ldots, \boldsymbol{\Gamma}^{(k-1)}) \tag{B.9}$$

$$\propto \left[\frac{(N_{T-1} - n_{T\cdot}(\boldsymbol{Z}^{(T-1)}))!}{N_{T-1}!} \cdot \frac{\mathrm{B}(n_{T\cdot}(\boldsymbol{Z}^{(T-1)}) + \alpha_\pi, n_T - n_{T\cdot}(\boldsymbol{Z}^{(T-1)}) + \beta_\pi)}{\mathrm{B}(\alpha_\pi, \beta_\pi)}\right]$$
$$\times \prod_{t_2=T}^{k} \prod_{t_1=1}^{t_2-1} \prod_{i=1}^{n_{t_1}} \prod_{j=1}^{n_{t_2}} \prod_{f=1}^{F} \prod_{\ell=0}^{L_f} \left[m_{f\ell}^{\mathbb{I}((\boldsymbol{x}_{t_1 i}, \boldsymbol{x}_{t_2 j}) \in M)} u_{f\ell}^{\mathbb{I}((\boldsymbol{x}_{t_1 i}, \boldsymbol{x}_{t_2 j}) \notin M)}\right]^{\gamma^{f\ell}(\boldsymbol{x}_{t_1 i}, \boldsymbol{x}_{t_2 j})}, \tag{B.10}$$

where

$$N_{t-1} = n_1 + \cdots + n_{t-1}$$
$$n_{t\cdot}(\boldsymbol{Z}^{(t-1)}) = \sum_{j=1}^{n_t} \mathbb{I}(Z_j^{(t-1)} \leq N_{t-1})$$
$$M := M(\boldsymbol{Z}^{(1)}, \ldots, \boldsymbol{Z}^{(k-1)})$$
$$= \{(\boldsymbol{x}_{t_1 i}, \boldsymbol{x}_{t_2 j}) : \boldsymbol{x}_{t_1 i} \text{ and } \boldsymbol{x}_{t_2 j} \text{ are linked}\}.$$

Pairs of records, $\boldsymbol{x}_{t_1 i}$ and $\boldsymbol{x}_{t_2 j}$, where $t_1, t_2 < T$ do not depend on $\boldsymbol{Z}^{(T-1)}$ to be linked because of the constraints outlined in Section 3.2.2.

B.3 Supplemental Definitions and Theorems

B.3.1 Matching Vector Prior Theorem

Theorem B.3.1. *Consider a k-file record linkage problem with an initial state $(\boldsymbol{Z}^{(1)}, \ldots, \boldsymbol{Z}^{(k-1)})$ and an alternate state $(\boldsymbol{Z}^{*(1)}, \ldots, \boldsymbol{Z}^{*(k-2)}, \boldsymbol{Z}^{(k-1)})$ such that $\boldsymbol{Z}^{*(1)}, \ldots, \boldsymbol{Z}^{*(k-2)}$ are identical to $\boldsymbol{Z}^{(1)}, \ldots, \boldsymbol{Z}^{(k-2)}$ except for the addition of one link, that is there exists an $\ell < k$, $j \leq n_\ell$ and $i \leq n_1 + \cdots + n_{\ell-1}$ such that $Z_j^{*(\ell-1)} = i$ and $Z_j^{(\ell-1)} = n_1 + \cdots + n_{\ell-1} + j$.*

Let

$$R = \frac{P(\boldsymbol{Z}^{*(1)}, \ldots, \boldsymbol{Z}^{*(k-2)}, \boldsymbol{Z}^{(k-1)})}{P(\boldsymbol{Z}^{(1)}, \ldots, \boldsymbol{Z}^{(k-2)}, \boldsymbol{Z}^{(k-1)})} \div \frac{P(\boldsymbol{Z}^{*(1)}, \ldots, \boldsymbol{Z}^{*(k-2)})}{P(\boldsymbol{Z}^{(1)}, \ldots, \boldsymbol{Z}^{(k-2)})}.$$

When the prior in Equation 3.6 is specified for $\boldsymbol{Z}^{(2)}, \ldots, \boldsymbol{Z}^{(k-1)}$, $R \geq 1$ *with equality only when there are no links in* $\boldsymbol{Z}^{(k-1)}$. *When the prior in Equation 3.4 is specified for* $\boldsymbol{Z}^{(2)}, \ldots, \boldsymbol{Z}^{(k-1)}$, $R = 1$.

Proof. First, simplifying R gives

$$R = \frac{P(\boldsymbol{Z}^{(k-1)} | \boldsymbol{Z}^{*(1)}, \ldots, \boldsymbol{Z}^{*(k-2)})}{P(\boldsymbol{Z}^{(k-1)} | \boldsymbol{Z}^{(1)}, \ldots, \boldsymbol{Z}^{(k-2)})}.$$

In the case of the prior in Equation 3.4,

$$P(\boldsymbol{Z}^{(k-1)} | \boldsymbol{Z}^{*(1)}, \ldots, \boldsymbol{Z}^{*(k-2)}) = P(\boldsymbol{Z}^{(k-1)} | \boldsymbol{Z}^{(1)}, \ldots, \boldsymbol{Z}^{(k-2)}) = P(\boldsymbol{Z}^{(k-1)}),$$

so $R = 1$.

In the case of the prior in Equation 3.6,

$$\begin{aligned} R &= \frac{(|C(\boldsymbol{Z}^{*(1)}, \ldots, \boldsymbol{Z}^{*(k-2)})| - n_{k\cdot}(\boldsymbol{Z}^{(k-1)}))!}{(|C(\boldsymbol{Z}^{(1)}, \ldots, \boldsymbol{Z}^{(k-2)})| - n_{k\cdot}(\boldsymbol{Z}^{(k-1)}))!} \cdot \frac{|C(\boldsymbol{Z}^{(1)}, \ldots, \boldsymbol{Z}^{(k-2)})|!}{|C(\boldsymbol{Z}^{*(1)}, \ldots, \boldsymbol{Z}^{*(k-2)})|!} \\ &= \frac{|C(\boldsymbol{Z}^{(1)}, \ldots, \boldsymbol{Z}^{(k-2)})|}{|C(\boldsymbol{Z}^{(1)}, \ldots, \boldsymbol{Z}^{(k-2)})| - n_{k\cdot}(\boldsymbol{Z}^{(k-1)})}, \end{aligned}$$

since the extra link removes exactly one candidate. So $R \geq 1$ with equality only when $n_{k\cdot}(\boldsymbol{Z}^{(k-1)}) = 0$, or there are no links in $\boldsymbol{Z}^{(k-1)}$. □

B.3.2 Sampler Definitions and Theorems

Definition B.3.1. Component-wise sampler. Define the component-wise sampler for sampling from the streaming record linkage model as follows:

1. For $f = 1, \ldots, F$

    a. Update the vector $\boldsymbol{m}_f$ from its conjugate full conditional Dirichlet distribution.

b. Update the vector $\boldsymbol{u}_f$ from its conjugate full conditional Dirichlet distribution.

2. For each vector $\boldsymbol{Z}^{(\ell)}$, $\ell = 1, \ldots, k-1$

   a. For each index $j = 1, \ldots, n_{\ell+1}$, update the component $Z_j^{(\ell)}$ from its full conditional distribution over all possible values, $1, \ldots, (n_1 + \cdots + n_\ell), (n_1 + \cdots + n_\ell + j)$.

3. Repeat steps 1 and 2 for $s = 1, \ldots, S$ times.

Definition B.3.2. Locally balanced sampler. Define the locally balanced sampler for sampling from the streaming record linkage model as follows:

1. For $f = 1, \ldots, F$

   a. Update the vector $\boldsymbol{m}_f$ from its conjugate full conditional Dirichlet distribution.

   b. Update the vector $\boldsymbol{u}_f$ from its conjugate full conditional Dirichlet distribution.

2. For each vector $\boldsymbol{Z}^{(\ell)}$, $\ell = 1, \ldots, k-1$

   a. Propose a new value of $\boldsymbol{Z}^{(\ell)}$ using locally balanced proposals (Zanella, 2020). Each potential proposal takes a step through either the addition of a link, the removal of a link, swapping one end of a link, or exchanging ends of two links (double-swap). Proposal probabilities are weighted with barker weights, $g(t) = t/(1+t)$.

   b. Accept or reject the proposal using the standard Metropolis-Hastings acceptance ratio for asymmetric proposals.

3. Repeat steps 1 and 2 for $s = 1, \ldots, S$ times.

Theorem B.3.2. *The component-wise sampler (Definition B.3.1) produces an ergodic Markov chain with the streaming record linkage model posterior distribution as its target distribution.*

Proof. The sampler in Definition B.3.1 is a Gibbs algorithm which samples directly from the full conditional distributions of the parameters in sequence. Therefore if we prove that the resulting Markov chain is irreducible, then it is ergodic and samples from the posterior distribution. From an

initial state with non-zero probability, $(\boldsymbol{m}, \boldsymbol{u}, \boldsymbol{Z}^{(1)}, \ldots, \boldsymbol{Z}^{(k-1)})$, a new state with non-zero probability, $(\boldsymbol{m}_*, \boldsymbol{u}_*, \boldsymbol{Z}_*^{(1)}, \ldots, \boldsymbol{Z}_*^{(k-1)})$, may always be reached through a sequence of non-zero probability steps. For the matching vectors, first remove all existing links from $(\boldsymbol{Z}^{(1)}, \ldots, \boldsymbol{Z}^{(k-1)})$ one component at a time until the completely unlinked state is reached. In the next iteration, add all links in $(\boldsymbol{Z}_*^{(1)}, \ldots, \boldsymbol{Z}_*^{(k-1)})$ one component at a time. All components of $\boldsymbol{m}$ and $\boldsymbol{u}$ are strictly positive, so states have zero posterior probability if and only if the state is invalid (Definition 3.2.1) and the indicator in the likelihood equals zero. As states are invalid due to conflicting links, removing links can never turn a valid state to invalid. Since $(\boldsymbol{Z}_*^{(1)}, \ldots, \boldsymbol{Z}_*^{(k-1)})$ is valid and has nonzero posterior probability, constructing it one link at a time will never result in an invalid state. □

Theorem B.3.3. *The locally balanced sampler (Definition B.3.2) produces an ergodic Markov chain with the streaming record linkage model posterior distribution as its target distribution.*

Proof. The sampler in Definition B.3.2 is a Metropolis-Hastings within Gibbs algorithm. Therefore it is sufficient to show that the resulting chain is irreducible. Similarly to the proof of Theorem B.3.2, we show there is a non-zero probability path between a starting state, $(\boldsymbol{m}, \boldsymbol{u}, \boldsymbol{Z}^{(1)}, \ldots, \boldsymbol{Z}^{(k-1)})$, and an ending state, $(\boldsymbol{m}_*, \boldsymbol{u}_*, \boldsymbol{Z}_*^{(1)}, \ldots, \boldsymbol{Z}_*^{(k-1)})$, via the completely unlinked state. In each iteration, the locally balanced proposals may remove a single link or add a single link to each vector $\boldsymbol{Z}^{(1)}, \ldots, \boldsymbol{Z}^{(k-1)}$. As in the proof of Theorem B.3.2, each of these steps are to states with positive probability. Since the locally balanced proposals are weighted by the target density, they can be proposed with positive probability. □

Theorem B.3.4. *The PPRB-within-Gibbs sampler (Definition 3.3.1) produces an ergodic Markov chain with the model's posterior distribution as its target distribution if the target distribution satisfies the positivity condition,*

$$p(\boldsymbol{\theta}_1|\boldsymbol{y}_1, \boldsymbol{y}_2) > 0,\ p(\boldsymbol{\theta}_2|\boldsymbol{y}_1, \boldsymbol{y}_2) > 0,\ p(\boldsymbol{\theta}_3|\boldsymbol{y}_1, \boldsymbol{y}_2) > 0 \implies p(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3|\boldsymbol{y}_1, \boldsymbol{y}_2) > 0.$$

Proof. First, we show that the Metropolis-Hastings acceptance ratio, α, in step 2 is appropriate for the target distribution. Since the proposals come from the distribution, $p(\boldsymbol{\theta}_1^*|\boldsymbol{y}_1)$, the acceptance

ratio would be

$$\begin{aligned}
\alpha &= \frac{p(\boldsymbol{\theta}_1^*|\boldsymbol{\theta}_2,\boldsymbol{\theta}_3,\boldsymbol{y}_1,\boldsymbol{y}_2)}{p(\boldsymbol{\theta}_1|\boldsymbol{\theta}_2,\boldsymbol{\theta}_3,\boldsymbol{y}_1,\boldsymbol{y}_2)}\frac{p(\boldsymbol{\theta}_1|\boldsymbol{y}_1)}{p(\boldsymbol{\theta}_1^*|\boldsymbol{y}_1)} \\
&= \frac{p(\boldsymbol{\theta}_1^*,\boldsymbol{\theta}_2,\boldsymbol{\theta}_3|\boldsymbol{y}_1,\boldsymbol{y}_2)}{p(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2,\boldsymbol{\theta}_3|\boldsymbol{y}_1,\boldsymbol{y}_2)}\frac{p(\boldsymbol{\theta}_1|\boldsymbol{y}_1)}{p(\boldsymbol{\theta}_1^*|\boldsymbol{y}_1)} \\
&= \frac{p(\boldsymbol{y}_1|\boldsymbol{\theta}_1^*,\boldsymbol{\theta}_2)p(\boldsymbol{y}_2|\boldsymbol{\theta}_1^*,\boldsymbol{\theta}_2,\boldsymbol{\theta}_3)p(\boldsymbol{\theta}_1^*)p(\boldsymbol{\theta}_2)p(\boldsymbol{\theta}_3)}{p(\boldsymbol{y}_1|\boldsymbol{\theta}_1,\boldsymbol{\theta}_2)p(\boldsymbol{y}_2|\boldsymbol{\theta}_1,\boldsymbol{\theta}_2,\boldsymbol{\theta}_3)p(\boldsymbol{\theta}_1)p(\boldsymbol{\theta}_2)p(\boldsymbol{\theta}_3)}\frac{p(\boldsymbol{\theta}_1|\boldsymbol{y}_1)}{p(\boldsymbol{\theta}_1^*|\boldsymbol{y}_1)} \\
&= \frac{p(\boldsymbol{y}_2|\boldsymbol{\theta}_1^*,\boldsymbol{\theta}_2,\boldsymbol{\theta}_3)}{p(\boldsymbol{y}_2|\boldsymbol{\theta}_1,\boldsymbol{\theta}_2,\boldsymbol{\theta}_3)}\frac{p(\boldsymbol{\theta}_1^*,\boldsymbol{\theta}_2|\boldsymbol{y}_1)}{p(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2|\boldsymbol{y}_1)}\frac{p(\boldsymbol{\theta}_1|\boldsymbol{y}_1)}{p(\boldsymbol{\theta}_1^*|\boldsymbol{y}_1)} \\
&= \frac{p(\boldsymbol{y}_2|\boldsymbol{\theta}_1^*,\boldsymbol{\theta}_2,\boldsymbol{\theta}_3)}{p(\boldsymbol{y}_2|\boldsymbol{\theta}_1,\boldsymbol{\theta}_2,\boldsymbol{\theta}_3)}\frac{p(\boldsymbol{\theta}_2|\boldsymbol{\theta}_1^*,\boldsymbol{y}_1)}{p(\boldsymbol{\theta}_2|\boldsymbol{\theta}_1,\boldsymbol{y}_1)}.
\end{aligned}$$

Second, we have that $p(\boldsymbol{\theta}_1|\boldsymbol{y}_1) = 0 \implies p(\boldsymbol{\theta}_1|\boldsymbol{\theta}_2,\boldsymbol{\theta}_3,\boldsymbol{y}_1,\boldsymbol{y}_2) = 0$ since the latter distribution is conditioned on a superset of random variables as the former. Therefore the distribution $p(\boldsymbol{\theta}_1|\boldsymbol{y}_1)$ works as an independent Metropolis-Hastings proposal distribution for the target $p(\boldsymbol{\theta}_1|\boldsymbol{\theta}_2,\boldsymbol{\theta}_3,\boldsymbol{y}_1,\boldsymbol{y}_2)$.

Finally, the positivity condition implies that a Gibbs sampler is irreducible, and so the algorithm produces an ergodic Markov chain. (Robert and Casella, 2013) □

PPRB-within-Gibbs sampler for StreamingRL model

We perform the three steps of each iteration as

1. For $f = 1, \ldots, F$
    a. Update the vector $\boldsymbol{m}_f$ from its conjugate full conditional Dirichlet distribution (see Appendix B.2.2).
    b. Update the vector $\boldsymbol{u}_f$ from its conjugate full conditional Dirichlet distribution (see Appendix B.2.2).
2. (PPRB step) Propose a new value $(\boldsymbol{Z}_*^{(1)}, \ldots, \boldsymbol{Z}_*^{(k-2)})$ by drawing from the existing posterior samples (with replacement). Accept or reject the proposal using the Metropolis-Hastings

ratio,

$$\alpha = \min \left(\begin{array}{l} \frac{p(\boldsymbol{\Gamma}^{(k-1)} | \boldsymbol{Z}_*^{(1)}, \ldots, \boldsymbol{Z}_*^{(k-2)}, \boldsymbol{m}, \boldsymbol{u}, \boldsymbol{Z}^{(k-1)})}{p(\boldsymbol{\Gamma}^{(k-1)} | \boldsymbol{Z}^{(1)}, \ldots, \boldsymbol{Z}^{(k-2)}, \boldsymbol{m}, \boldsymbol{u}, \boldsymbol{Z}^{(k-1)})} \\ \times \frac{p(\boldsymbol{m}, \boldsymbol{u} | \boldsymbol{Z}_*^{(1)}, \ldots, \boldsymbol{Z}_*^{(k-2)}, \boldsymbol{\Gamma}^{(1)}, \ldots, \boldsymbol{\Gamma}^{(k-2)})}{p(\boldsymbol{m}, \boldsymbol{u} | \boldsymbol{Z}^{(1)}, \ldots, \boldsymbol{Z}^{(k-2)}, \boldsymbol{\Gamma}^{(1)}, \ldots, \boldsymbol{\Gamma}^{(k-2)})} \end{array} , 1 \right)$$

3. Update the value of $\boldsymbol{Z}^{(k-1)}$ using a Metropolis-Hastings proposal targeting its full conditional distribution. We examine two such possible proposals in Section 3.3.3.

B.4 Simulation and Sampling Details

This appendix contains details for MCMC runs and simulation studies whose results are presented in the main body of the chapter.

B.4.1 Link Accuracy Comparison

Proposed Model: Streaming Record Linkage

The sampler was run for 2500 iterations, discarding the first 500. We set $\alpha_\pi = \beta_\pi = 1$ as an uninformative prior for $\boldsymbol{Z}^{(1)}$, $\boldsymbol{Z}^{(2)}$, and $\boldsymbol{Z}^{(3)}$. Flat Dirichlet priors were chosen for $\boldsymbol{u}$, and three choices of prior strength were used for $\boldsymbol{m}$ (Flat, Weak, Strong). Component-wise proposals were used for $\boldsymbol{Z}^{(1)}$, $\boldsymbol{Z}^{(2)}$, and $\boldsymbol{Z}^{(3)}$ to avoid needing excessive burn-in. We found that a Gibbs sampler with locally balanced proposals required too many iterations to converge to the target posterior distribution to be computationally feasible.

Multilink (Aleshin-Guendel and Sadinle, 2023)

We use flat Dirichlet priors for the $\boldsymbol{m}$ and $\boldsymbol{u}$ parameters, $\boldsymbol{\alpha} = \mathbf{1}$ for the Dirichlet-multinomial overlap table prior on the partitions and a uniform prior on the number of clusters. For each of the simulated datasets, we produce 1000 posterior samples after a 500 iteration burn-in from an initial state of no linked pairs.

Blink (Steorts, 2015)

For string fields, we choose a steepness parameter $c = 1$ and the generalized Levenshtein distance of the `R` function `adist`. For categorical fields, we choose beta parameters $a = 5$ and $b = 20$ to encode prior knowledge of between 1 and 4 errors per record, or a distortion probability

of between 0.1 and 0.4. For each simulated dataset, we produce 1000 posterior samples after a 5000 iteration burn-in.

Support Vector Machine

Training pairs were chosen as evenly as possible between coreferent and non-coreferent pairs, which sometimes resulted in all coreferent pairs being included in the training set.

B.4.2 Speed Comparison

The Gibbs sampler was run using component-wise full conditional updates for $\boldsymbol{Z}^{(1)}$, $\boldsymbol{Z}^{(2)}$ and $\boldsymbol{Z}^{(3)}$ for 2500 iterations, discarding the first 500 for burn-in. Each PPRB update was run for 5000 iterations, discarding the first 1000 for burn-in. The SMCMC updates used ensembles of size 200 and were computed with 12 parallel processes. SMCMC-Comp used 5 jumping kernel iterations and 50 transition kernel iterations, SMCMC-LB used 50 jumping kernel iterations and and 200 transition kernel iterations, and SMCMC-Mixed used 5 jumping kernel iterations and 200 transition kernel iterations. All locally balanced proposals used a block size of 75 records.

B.4.3 Social Diagnosis Survey Analysis

The Gibbs sampler was run for 2500 iterations, discarding the first 500 for burn-in. Each PPRB update was run for 5000 iterations, discarding the first 1000 for burn-in. The SMCMC updates used ensembles of size 200 and were computed with 12 parallel processes. SMCMC-Comp used 5 jumping kernel iterations and 50 transition kernel iterations, SMCMC-LB used 500 jumping kernel iterations and and 200 transition kernel iterations, and SMCMC-Mixed used 5 jumping kernel iterations and 200 transition kernel iterations. All locally balanced proposals used a block size of 150 records.

Appendix C

Supplement to *Generative Filtering for Recursive Bayesian Inference with Streaming Data*

C.1 Theorems and Proofs

C.1.1 PPRB-within-Gibbs

Theorem C.1.1. *The PPRB-within-Gibbs sampler (Definition 4.1.1) produces an ergodic Markov chain with the model's posterior distribution as its target distribution if the posterior distribution satisfies the following positivity condition,*

$$p(\boldsymbol{\theta}|\boldsymbol{y}_1, \boldsymbol{y}_2) > 0,\ p(\boldsymbol{\phi}|\boldsymbol{y}_1, \boldsymbol{y}_2) > 0 \implies p(\boldsymbol{\theta}, \boldsymbol{\phi}|\boldsymbol{y}_1, \boldsymbol{y}_2) > 0.$$

Proof. First, we show that the acceptance ratio, α, in step 1 is appropriate for the target full conditional distribution. Since the proposals, $\boldsymbol{\theta}^*$ are produced from the distribution $p(\boldsymbol{\theta}^*|\boldsymbol{y}_1)$ and the target distribution is the full conditional, $p(\boldsymbol{\theta}|\boldsymbol{\phi}, \boldsymbol{y}_1, \boldsymbol{y}_2)$, the MH acceptance ratio would be

$$\begin{aligned}
\alpha &= \frac{p(\boldsymbol{\theta}^*|\boldsymbol{\phi}, \boldsymbol{y}_1, \boldsymbol{y}_2)}{p(\boldsymbol{\theta}|\boldsymbol{\phi}, \boldsymbol{y}_1, \boldsymbol{y}_2)} \cdot \frac{p(\boldsymbol{\theta}|\boldsymbol{y}_1)}{p(\boldsymbol{\theta}^*|\boldsymbol{y}_1)} \\
&= \frac{p(\boldsymbol{\theta}^*, \boldsymbol{\phi}|\boldsymbol{y}_1, \boldsymbol{y}_2)}{p(\boldsymbol{\theta}, \boldsymbol{\phi}|\boldsymbol{y}_1, \boldsymbol{y}_2)} \cdot \frac{p(\boldsymbol{\theta}|\boldsymbol{y}_1)}{p(\boldsymbol{\theta}^*|\boldsymbol{y}_1)} \\
&= \frac{p(\boldsymbol{y}_2|\boldsymbol{\phi}, \boldsymbol{\theta}^*, \boldsymbol{y}_1)p(\boldsymbol{y}_1|\boldsymbol{\theta}^*)p(\boldsymbol{\phi}|\boldsymbol{\theta}^*)p(\boldsymbol{\theta}^*)}{p(\boldsymbol{y}_2|\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{y}_1)p(\boldsymbol{y}_1|\boldsymbol{\theta})p(\boldsymbol{\phi}|\boldsymbol{\theta})p(\boldsymbol{\theta})} \cdot \frac{p(\boldsymbol{\theta}|\boldsymbol{y}_1)}{p(\boldsymbol{\theta}^*|\boldsymbol{y}_1)} \\
&= \frac{p(\boldsymbol{y}_2|\boldsymbol{\phi}, \boldsymbol{\theta}^*, \boldsymbol{y}_1)p(\boldsymbol{\phi}|\boldsymbol{\theta}^*) \cdot p(\boldsymbol{\theta}^*|\boldsymbol{y}_1)}{p(\boldsymbol{y}_2|\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{y}_1)p(\boldsymbol{\phi}|\boldsymbol{\theta}) \cdot p(\boldsymbol{\theta}|\boldsymbol{y}_1)} \cdot \frac{p(\boldsymbol{\theta}|\boldsymbol{y}_1)}{p(\boldsymbol{\theta}^*|\boldsymbol{y}_1)} \\
&= \frac{p(\boldsymbol{y}_2|\boldsymbol{y}_1, \boldsymbol{\theta}^*, \boldsymbol{\phi})}{p(\boldsymbol{y}_2|\boldsymbol{y}_1, \boldsymbol{\theta}, \boldsymbol{\phi})} \cdot \frac{p(\boldsymbol{\phi}|\boldsymbol{\theta}^*)}{p(\boldsymbol{\phi}|\boldsymbol{\theta})}.
\end{aligned}$$

Second, the distribution $p(\boldsymbol{\theta}|\boldsymbol{y}_1)$ works as independent MH proposals for the distribution $p(\boldsymbol{\theta}|\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{\phi})$ because $p(\boldsymbol{\theta}|\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{\phi}) \propto p(\boldsymbol{\theta}|\boldsymbol{y}_1)p(\boldsymbol{y}_2|\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{y}_1)p(\boldsymbol{\phi}|\boldsymbol{\theta})$, so $p(\boldsymbol{\theta}|\boldsymbol{y}_1) = 0$ implies $p(\boldsymbol{\theta}|\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{\phi}) = 0$.

Finally, the positivity condition implies that a Gibbs sampler is irreducible (Robert and Casella, 2013), so the PPRB-within-Gibbs algorithm produces an ergodic Markov chain. □

Theorem C.1.2. ***(Theorem 4.2.1 restated)*** *Let π_t, A_t, and $F_S^{(t)}$ be defined as in Eq. (4.3)-(4.5) and let $\|\cdot\|$ be a norm on probability measures that has a triangle inequality. Then,*

$$\|A_t - \pi_t\| \leq \underbrace{\|A_{t-1} - \pi_{t-1}\|}_{(1)} + \underbrace{\left\|F_S^{(t-1)} - A_{t-1}\right\|}_{(2)} + \underbrace{\|\pi_t - \pi_{t-1}\|}_{(3)} + \underbrace{\left\|A_t - F_S^{(t-1)}\right\|}_{(4)}$$

$$\|A_t - \pi_t\| \geq \left| \underbrace{\|A_{t-1} - \pi_{t-1}\|}_{(1)} - \underbrace{\left\|F_S^{(t-1)} - A_{t-1}\right\|}_{(2)} \right| - \underbrace{\|\pi_t - \pi_{t-1}\|}_{(3)} - \underbrace{\left\|A_t - F_S^{(t-1)}\right\|}_{(4)}$$

Proof. We derive the upper bound by the triangle inequality as

$$\|A_t - \pi_t\| \leq \left\|A_t - F_S^{(t-1)}\right\| + \left\|F_S^{(t-1)} - A_{t-1}\right\| + \|A_{t-1} - \pi_{t-1}\| + \|\pi_t - \pi_{t-1}\| \tag{C.1}$$

$$= \|A_{t-1} - \pi_{t-1}\| + \left\|F_S^{(t-1)} - A_{t-1}\right\| + \|\pi_t - \pi_{t-1}\| + \left\|A_t - F_S^{(t-1)}\right\|. \tag{C.2}$$

We use reverse triangle inequalities to derive the lower bound:

$$\|A_t - \pi_t\| \geq \Big| \|A_t - A_{t-1}\| - \|A_{t-1} - \pi_t\| \Big| \tag{C.3}$$

$$= \max \left\{ \begin{array}{l} \|A_t - A_{t-1}\| - \|A_{t-1} - \pi_t\|, \\ \|A_{t-1} - \pi_t\| - \|A_t - A_{t-1}\| \end{array} \right\} \tag{C.4}$$

$$\geq \max \left\{ \begin{array}{l} \|A_t - A_{t-1}\| - (\|A_{t-1} - \pi_{t-1}\| + \|\pi_{t-1} - \pi_t\|), \\ \|A_{t-1} - \pi_t\| - \left(\left\|A_t - F_S^{(t-1)}\right\| + \left\|F_S^{(t-1)} - A_{t-1}\right\| \right) \end{array} \right\} \tag{C.5}$$

$$\geq \max \left\{ \begin{array}{l} \left| \left\|A_t - F_S^{(t-1)}\right\| - \left\|F_S^{(t-1)} - A_{t-1}\right\| \right| - \|A_{t-1} - \pi_{t-1}\| - \|\pi_{t-1} - \pi_t\|, \\ \|A_{t-1} - \pi_{t-1}\| - \|\pi_{t-1} - \pi_t\| \Big| - \left\|A_t - F_S^{(t-1)}\right\| - \left\|F_S^{(t-1)} - A_{t-1}\right\| \end{array} \right\} \tag{C.6}$$

$$= \max \left\{ \begin{array}{l} \left\|A_t - F_S^{(t-1)}\right\| - \left\|F_S^{(t-1)} - A_{t-1}\right\| - \|A_{t-1} - \pi_{t-1}\| - \|\pi_{t-1} - \pi_t\|, \\ \left\|F_S^{(t-1)} - A_{t-1}\right\| - \left\|A_t - F_S^{(t-1)}\right\| - \|A_{t-1} - \pi_{t-1}\| - \|\pi_{t-1} - \pi_t\|, \\ \|A_{t-1} - \pi_{t-1}\| - \|\pi_{t-1} - \pi_t\| - \left\|A_t - F_S^{(t-1)}\right\| - \left\|F_S^{(t-1)} - A_{t-1}\right\|, \\ \|\pi_{t-1} - \pi_t\| - \|A_{t-1} - \pi_{t-1}\| - \left\|A_t - F_S^{(t-1)}\right\| - \left\|F_S^{(t-1)} - A_{t-1}\right\| \end{array} \right\} \tag{C.7}$$

$$\geq \max \left\{ \begin{array}{l} \left\|F_S^{(t-1)} - A_{t-1}\right\| - \|A_{t-1} - \pi_{t-1}\| - \|\pi_{t-1} - \pi_t\| - \left\|A_t - F_S^{(t-1)}\right\|, \\ \|A_{t-1} - \pi_{t-1}\| - \left\|F_S^{(t-1)} - A_{t-1}\right\| - \|\pi_{t-1} - \pi_t\| - \left\|A_t - F_S^{(t-1)}\right\| \end{array} \right\} \tag{C.8}$$

$$= \left| \|A_{t-1} - \pi_{t-1}\| - \left\|F_S^{(t-1)} - A_{t-1}\right\| \right| - \|\pi_{t-1} - \pi_t\| - \left\|A_t - F_S^{(t-1)}\right\| \tag{C.9}$$

Line (C.4) is true by expanding $|x| = \max\{x, -x\}$. Line (C.5) follows by using a triangle inequality on the negative terms. Line (C.6) follows by using a reverse triangle inequality on the positive terms. Line (C.7) comes from expanding the abolute value as before. Line (C.8) is true because $\max A \geq \max B$ if $B \subset A$. Finally, line (C.9) recombines the maximum into an absolute value. □

C.1.2 Generative Filtering

Lemma C.1.1. *Let $P_t^S(\boldsymbol{\theta}_{1:(t-1)}, \cdot)$ represent the kernel resulting from S applications of PPRB-within-Gibbs at time t, which is a probability density for $\boldsymbol{\theta}_{1:t} := (\boldsymbol{\theta}_{1:(t-1)}, \boldsymbol{\theta}_t)$. Then for any*

probability density $p(\cdot)$ for $\boldsymbol{\theta}_{1:(t-1)}$, the following holds:

$$||\pi_t - P_t^S \circ p||_1 \leq \sup_{\boldsymbol{\theta}_{1:(t-1)}} ||\pi_t - P_t^S(\boldsymbol{\theta}_{1:(t-1)}, \cdot)||_1.$$

Proof.

$$||\pi_t - P_t^S \circ p||_1 = \int \left| \pi_t(\boldsymbol{\theta}_{1:t}) - \int p(\boldsymbol{\theta}'_{1:(t-1)}) P_t^S(\boldsymbol{\theta}'_{1:(t-1)}, \boldsymbol{\theta}_{1:t}) \mathrm{d}\boldsymbol{\theta}'_{1:(t-1)} \right| \mathrm{d}\boldsymbol{\theta}_{1:t} \tag{C.10}$$

$$= \int \left| \int \left(p(\boldsymbol{\theta}'_{1:(t-1)}) \pi_t(\boldsymbol{\theta}_{1:t}) - p(\boldsymbol{\theta}'_{1:(t-1)}) P_t^S(\boldsymbol{\theta}'_{1:(t-1)}, \boldsymbol{\theta}_{1:t}) \right) \mathrm{d}\boldsymbol{\theta}'_{1:(t-1)} \right| \mathrm{d}\boldsymbol{\theta}_{1:t} \tag{C.11}$$

$$\leq \int \int \left| p(\boldsymbol{\theta}'_{1:(t-1)}) \pi_t(\boldsymbol{\theta}_{1:t}) - p(\boldsymbol{\theta}'_{1:(t-1)}) P_t^S(\boldsymbol{\theta}'_{1:(t-1)}, \boldsymbol{\theta}_{1:t}) \right| \mathrm{d}\boldsymbol{\theta}'_{1:(t-1)} \mathrm{d}\boldsymbol{\theta}_{1:t} \tag{C.12}$$

$$= \int \int p(\boldsymbol{\theta}'_{1:(t-1)}) \left| \pi_t(\boldsymbol{\theta}_{1:t}) - P_t^S(\boldsymbol{\theta}'_{1:(t-1)}, \boldsymbol{\theta}_{1:t}) \right| \mathrm{d}\boldsymbol{\theta}'_{1:(t-1)} \mathrm{d}\boldsymbol{\theta}_{1:t} \tag{C.13}$$

$$\leq \int \int p(\boldsymbol{\theta}'_{1:(t-1)}) \sup_{\boldsymbol{\theta}'_{1:(t-1)}} \left| \pi_t(\boldsymbol{\theta}_{1:t}) - P_t^S(\boldsymbol{\theta}'_{1:(t-1)}, \boldsymbol{\theta}_{1:t}) \right| \mathrm{d}\boldsymbol{\theta}'_{1:(t-1)} \mathrm{d}\boldsymbol{\theta}_{1:t} \tag{C.14}$$

$$= \int \sup_{\boldsymbol{\theta}'_{1:(t-1)}} \left| \pi_t(\boldsymbol{\theta}_{1:t}) - P_t^S(\boldsymbol{\theta}'_{1:(t-1)}, \boldsymbol{\theta}_{1:t}) \right| \mathrm{d}\boldsymbol{\theta}_{1:t} \tag{C.15}$$

$$= \sup_{\boldsymbol{\theta}'_{1:(t-1)}} ||\pi_t - P_t^S(\boldsymbol{\theta}'_{1:(t-1)}, \cdot)||_1 \tag{C.16}$$

□

Theorem C.1.3. ***(Theorem 4.3.1 restated)*** *Let $P_t^S(\boldsymbol{\theta}_{1:(t-1)}, \cdot)$ represent the kernel resulting from S applications of a filtering method at time t, which is a probability density for $\boldsymbol{\theta}_{1:t} := (\boldsymbol{\theta}_{1:(t-1)}, \boldsymbol{\theta}_t)$. Let $\pi_t = p(\boldsymbol{\theta}_{1:t}|\boldsymbol{y}_{1:t})$ be the target posterior at time t. Assuming the following conditions:*

1. *(Universal ergodicity) There exist $\rho_t \in (0, 1)$, such that for all $t > 0$ and $x \in \mathcal{X}$,*

$$||T_t(x, \cdot) - \pi_t||_1 \leq 2\rho_t.$$

2. *(Filtering consistency) For a sequence of $\lambda_t \to 0$ and a bounded sequence of positive integers S_t, the following holds:*

$$\sup_{\boldsymbol{\theta}_{1:(t-1)}} ||\pi_t - P_t^{S_t}(\boldsymbol{\theta}_{1:(t-1)}, \cdot)||_1 \leq 2\lambda_t.$$

Let $\epsilon_t = \rho_t^{m_t}$ and let $Q_t = T_t^{m_t} \circ P_t^{S_t}$ be a Generative Filtering update at time t. Then for any initial distribution π_0,

$$||Q_t \circ \cdots \circ Q_1 \circ \pi_0 - \pi_t||_1 \leq \sum_{v=1}^{t} \left\{ \prod_{u=v+1}^{t} \epsilon_u(1-\lambda_u) \right\} \epsilon_v \lambda_v \leq \sum_{v=1}^{t} \left\{ \prod_{u=v}^{t} \epsilon_u \right\} \lambda_v.$$

Proof. The proof follows closely the proof of Theorem 3.9 in Yang and Dunson (2013).

We will construct two time inhomogeneous Markov chains $\{X_{t,r} : r = 1, \ldots, m_t, t \geq 0\}$ and $\{X'_{t,r} : r = 1, \ldots, m_t, t \geq 0\}$. The chains proceed in the double index first in r then t, i.e., $(t, r) = (0, 1), \ldots, (0, m_0), (1, 1), \ldots, (1, m_1), \ldots$. The two chains are constructed as follows:

1. $X_{0,1} \sim \pi_0$, $X'_{0,1} \sim \pi_0$.
2. For $t \geq 1$

   a. For $r = 1$. Let $X_{t-1,m_{t-1}} = x$, $X'_{t-1,m_{t-1}} = x'$. Draw $X_{t,1} = x^* \sim P_t^{S_t}(x, \cdot)$. With probability $\min\left\{1, \frac{\pi_t(x^*)}{P_t^{S_t} \circ \pi_{t-1}(x^*)}\right\}$, set $X'_{t,1} = x^*$; with probability $1 - \min\left\{1, \frac{\pi_t(x^*)}{P_t^{S_t} \circ \pi_{t-1}(x^*)}\right\}$, draw

$$X'_{t,1} \sim \frac{\pi_t(\cdot) - \min\left\{\pi_t(\cdot), P_t^{S_t} \circ \pi_{t-1}(\cdot)\right\}}{\tilde{\alpha}_t}, \tag{C.17}$$

   where $\tilde{\alpha}_t = \frac{1}{2}||\pi_t - P_t^{S_t} \circ \pi_{t-1}||_1$.

   b. For $1 < r \leq m_t$. Let $X_{t,r-1} = x$ and $X'_{t,r-1} = x'$.

      i. If $x = x'$, choose $X_{t,r} = X'_{t,r} \sim T_t(x, \cdot)$;

ii. else, first choose $X'_{t,r} = y \sim T_t(x', \cdot)$, then with probability $\min\left\{1, \frac{T_t(x,y)}{\pi_t(y)}\right\}$, set $X_{t,s} = y$, with probability $1 - \min\left\{1, \frac{T_t(x,y)}{\pi_t(y)}\right\}$, draw

$$X_{t,r} \sim \frac{T_t(x, \cdot) - \min\{T_t(x, \cdot), \pi_t(\cdot)\}}{\delta_t(x)}, \tag{C.18}$$

where $\delta_t(x) = \frac{1}{2}||T_t(x, \cdot) - \pi_t||_1$.

First, for $t \geq 1$ and $1 < r \leq m_t$, both chains have the same transition kernel, T_t, which targets π_t. This is apparent for $\{X'\}_{t,s}$, while for $\{X\}_{t,s}$, we can see that its transition kernel is a mixture of π_t and the distribution given by (C.18) which equals T_t. For $t \geq 1$ and $r = 1$, the distribution of $X'_{t,1}$ is π_t because its distribution is a mixture of $P_t^{S_t} \circ \pi_{t-1}$ and the distribution given by (C.17), which equals π_t. Therefore for any (t, r), the marginal distribution of $X'_{t,r}$ is π_t.

For any (t, r), the marginal distribution of $X_{t,r}$ is $T_t^r \circ P_t^{S_t} \circ Q_{t-1} \circ \cdots \circ Q_1 \circ \pi_0$. Therefore,

$$||Q_t \circ \cdots \circ Q_1 \circ \pi_0 - \pi_t||_1 \leq P(X_{t,m_t} \neq X'_{t,m_t}). \tag{C.19}$$

Conditional on $X_{t-1,m_{t-1}} = X'_{t-1,m_{t-1}}$, the distribution of $X_{t-1,m_{t-1}}$ is π_{t-1}. So $P(X_{t,1} \neq X'_{t,1} | X_{t-1,m_{t-1}} = X'_{t-1,m_{t-1}}) = \tilde{\alpha}_t$, which by Lemma C.1.1, $\tilde{\alpha}_t \leq \lambda_t$.

Then,

$$P(X_{t,m_t} \neq X'_{t,m_t}) = P(X_{t-1,m_{t-1}} \neq X'_{t-1,m_{t-1}}, X_{t,m_t} \neq X'_{t,m_t}) \tag{C.20}$$

$$+ P(X_{t-1,m_{t-1}} = X'_{t-1,m_{t-1}}, X_{t,m_t} \neq X'_{t,m_t}) \tag{C.21}$$

$$= [P(X_{t,m_t} \neq X'_{t,m_t} | X_{t-1,m_{t-1}} \neq X'_{t-1,m_{t-1}})$$

$$\cdot P(X_{t-1,m_{t-1}} \neq X'_{t-1,m_{t-1}})] \tag{C.22}$$

$$+ [P(X_{t,m_t} \neq X'_{t,m_t} | X_{t-1,m_{t-1}} = X'_{t-1,m_{t-1}})$$

$$\cdot (1 - P(X_{t-1,m_{t-1}} \neq X'_{t-1,m_{t-1}}))] \tag{C.23}$$

$$\leq \rho_t^{m_t} \cdot P(X_{t-1,m_{t-1}} \neq X'_{t-1,m_{t-1}}) \tag{C.24}$$

$$+ \tilde{\alpha}_t \rho_t^{m_t} \cdot (1 - P(X_{t-1,m_{t-1}} \neq X'_{t-1,m_{t-1}})) \tag{C.25}$$

$$\leq \rho_t^{m_t} \cdot P(X_{t-1,m_{t-1}} \neq X'_{t-1,m_{t-1}}) \tag{C.26}$$

$$+ \lambda_t \rho_t^{m_t} \cdot (1 - P(X_{t-1,m_{t-1}} \neq X'_{t-1,m_{t-1}})) \tag{C.27}$$

$$= \lambda_t \rho_t^{m_t} + (1 - \lambda_t) \rho_t^{m_t} \cdot P(X_{t-1,m_{t-1}} \neq X'_{t-1,m_{t-1}}) \tag{C.28}$$

Line (C.24) follows from line (C.22) and line (C.25) follows from line (C.23) because ρ_t is the probability of $X_{t,r}$ and $X'_{t,r}$ remaining unequal given that the chains are unequal at step $r-1$, and $P(X_{t,1} \neq X'_{t,1} | X_{t-1,m_{t-1}} = X'_{t-1,m_{t-1}}) = \tilde{\alpha}_t$ and $P(X_{t,1} \neq X'_{t,1} | X_{t-1,m_{t-1}} \neq X'_{t-1,m_{t-1}}) \leq 1$. Line (C.28) follows from Lemma C.1.1.

There is now a recursive relation ship between t and $t-1$. We can repeat this for all $t >= 1$, and using $P(X_{0,m_0} \neq X'_{0,m_0}) \leq 1$ and $\epsilon_t = \rho_t^{m_t}$, we arrive at the result. □

Theorem C.1.4. ***(Theorem 4.3.2 restated)*** *Assume the following conditions hold:*

1. *(Universal ergodicity) There exists $\epsilon \in (0,1)$, such that for all $t > 0$ and $x \in \mathcal{X}$,*

$$||T_t(x, \cdot) - \pi_t||_1 \leq 2\rho_t.$$

2. *(Stationary convergence) The stationary distribution π_t of T_t satisfies*

$$\alpha_t = \frac{1}{2}||\pi_t - \pi_{t-1}||_1 \to 0,$$

where π_t is the marginal posterior of $\theta_{1:(t-1)}$ at time t in α_t.

3. *(Filtering consistency) For a sequence of $\lambda_t^{(F)} \to 0$ and a bounded sequence of positive integers S_t, the following holds:*

$$\sup_{\theta_{1:(t-1)}} ||\pi_t - P_t^{S_t}(\theta_{1:(t-1)}, \cdot)||_1 \leq 2\lambda_t^{(F)}.$$

4. *(Jumping consistency) For a sequence of $\lambda_t^{(J)} \to 0$, the following holds:*

$$\sup_{\theta_{1:(t-1)}} ||\pi_t(\cdot|\theta_{1:(t-1)}) - J_t(\theta_{1:(t-1)}, \cdot)||_1 \leq 2\lambda_t^{(J)}.$$

Let $\epsilon_t = \rho_t^{m_t}$. Define

$$\gamma_t^{(F)} = \sum_{v=1}^{t} \left\{ \prod_{u=v+1}^{t} \epsilon_u (1 - \lambda_u^{(F)}) \right\} \epsilon_v \lambda_v^{(F)}$$

and

$$\gamma_t^{(J)} = \sum_{v=1}^{t} \left\{ \prod_{u=v}^{t} \epsilon_u \right\} (\lambda_v^{(J)} + \alpha_v)$$

to be the bounds from Theorem 4.3.1 and Theorem 3.9 of Yang and Dunson (2013), respectively. If, for all $u \leq t$, $\lambda_u^{(F)} \leq \alpha_u + \lambda_u^{(J)}$, then $\gamma_t^{(F)} \leq \gamma_t^{(J)}$.

Proof. Define

$$\gamma_t^{(F)} = \sum_{v=1}^{t} \left\{ \prod_{u=v+1}^{t} \epsilon_u (1 - \lambda_u^{(F)}) \right\} \epsilon_v \lambda_s^{(F)}$$

and

$$\gamma_t^{(J)} = \sum_{v=1}^{t} \left\{ \prod_{u=v}^{t} \epsilon_u \right\} (\lambda_v^{(J)} + \alpha_v).$$

Assume that for all $u \leq t$, $\lambda_u^{(F)} \leq \alpha_u + \lambda_u^{(J)}$. Then we have,

$$\begin{aligned}
\gamma_t^{(F)} &= \sum_{v=1}^{t} \left\{ \prod_{u=v+1}^{t} \epsilon_u (1 - \lambda_u^{(F)}) \right\} \epsilon_v \lambda_v^{(F)} \\
&\leq \sum_{v=1}^{t} \left\{ \prod_{u=v+1}^{t} \epsilon_u \right\} \epsilon_v \lambda_v^{(F)} \\
&= \sum_{v=1}^{t} \left\{ \prod_{u=v}^{t} \epsilon_u \right\} \lambda_v^{(F)} \\
&\leq \sum_{v=1}^{t} \left\{ \prod_{u=v}^{t} \epsilon_u \right\} (\lambda_v^{(J)} + \alpha_v) \\
&= \gamma_t^{(J)}.
\end{aligned}$$

Then $\gamma_t^{(F)} \leq \gamma_t^{(J)}$. □

Theorem C.1.5. ***(Theorem 4.3.3 restated)*** *With the conditions and definitions of Theorem 4.3.2, assume $\gamma_{t-1}^{(F)} = \gamma_{t-1}^{(J)}$ and define $\gamma := \gamma_{t-1}^{(F)} = \gamma_{t-1}^{(J)}$. If $\gamma < 1$ and $\lambda_t^{(F)} \leq \frac{\alpha_t + \lambda_t^{(J)}}{1-\gamma}$, then $\gamma_t^{(F)} \leq \gamma_t^{(J)}$. If $\gamma \geq 1$ then $\gamma_t^{(F)} \leq \gamma_t^{(J)}$ always.*

Proof. We have the following recursive relationships for $\gamma_t^{(F)}$ and $\gamma_t^{(J)}$,

$$\gamma_t^{(J)} = \epsilon_t \gamma_{t-1}^{(J)} + \epsilon_t (\alpha_t + \lambda_t^{(J)}) \tag{C.29}$$

$$\gamma_t^{(F)} = \epsilon_t (1 - \lambda^{(F)}) \gamma_{t-1}^{(F)} + \epsilon_t \lambda_t^{(F)} \tag{C.30}$$

Then for $\gamma < 1$,

$$\gamma_t^{(F)} = \epsilon_t(1-\lambda^{(F)})\gamma_{t-1}^{(F)} + \epsilon_t\lambda_t^{(F)} \tag{C.31}$$

$$= \epsilon_t(1-\lambda^{(F)})\gamma + \epsilon_t\lambda_t^{(F)} \tag{C.32}$$

$$= \epsilon_t\lambda^{(F)}(1-\gamma) + \epsilon_t\gamma \tag{C.33}$$

$$\leq \epsilon_t\frac{\alpha_t + \lambda_t^{(J)}}{1-\gamma}(1-\gamma) + \epsilon_t\gamma \tag{C.34}$$

$$= \epsilon_t(\alpha_t + \lambda_t^{(J)}) + \epsilon_t\gamma \tag{C.35}$$

$$= \gamma_t^{(J)}. \tag{C.36}$$

For $\gamma \geq 1$,

$$\gamma_t^{(F)} = \epsilon_t(1-\lambda^{(F)})\gamma + \epsilon_t\lambda_t^{(F)} \tag{C.37}$$

$$= \epsilon_t\lambda^{(F)}(1-\gamma) + \epsilon_t\gamma \tag{C.38}$$

$$\leq \epsilon_t\gamma \tag{C.39}$$

$$\leq \epsilon_t\gamma + \epsilon_t(\alpha_t + \lambda_t^{(J)}) \tag{C.40}$$

$$= \gamma_t^{(J)} \tag{C.41}$$

□

Theorem C.1.6. ***(Theorem 4.3.4 restated)*** *Assume:*

1. *The data $\boldsymbol{y}_{t_1}$ and $\boldsymbol{y}_{t_2}$, for all $t_1 < t_2$, are conditionally independent given $\boldsymbol{\theta}_{1:t_2}$.*
2. *Each distribution $p(\boldsymbol{y}_t|\boldsymbol{\theta}_{1:t})$ has a sufficient statistic $U_t(\boldsymbol{y}_t)$ where $\dim U_t \ll \dim \boldsymbol{y}_t$.*

Then any transition kernel can be computed while storing only the sufficient statistics, U_t, instead of the data, $\boldsymbol{y}_t$, for all t.

Proof. Each data distribution can be factored by the Fisher-Neyman factorization theorem as

$$p(\boldsymbol{y}_t|\boldsymbol{\theta}_{1:t}) = h_t(\boldsymbol{y}_t)g_t(U_t(\boldsymbol{y}_t);\boldsymbol{\theta}_{1:t}). \tag{C.42}$$

Then the posterior at time T can be evaluated, up to a constant, using only these functions:

$$p(\boldsymbol{\theta}_{1:T}|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_T) \propto p(\boldsymbol{\theta}_{1:T}) \prod_{t=1}^{T} p(\boldsymbol{y}_t|\boldsymbol{\theta}_{1:t}) \tag{C.43}$$

$$\propto p(\boldsymbol{\theta}_{1:T}) \prod_{t=1}^{T} g_t(U_t(\boldsymbol{y}_t);\boldsymbol{\theta}_{1:t}), \tag{C.44}$$

which only requires the sufficient statistics be stored after the arrival of each batch of data. □

Theorem C.1.7. ***(Theorem 4.3.5 restated)*** *Assume:*

1. *The data $\boldsymbol{y}_{t_1}$ and $\boldsymbol{y}_{t_2}$, for all $t_1 < t_2$, are conditionally independent given $\boldsymbol{\theta}_{1:t_2}$.*
2. *Each $\boldsymbol{y}_t$ is a sample of n_t i.i.d. observations $\boldsymbol{y}_{t,i}$ for $i = 1,\ldots,n_t$.*
3. *Each observation $\boldsymbol{y}_{t,i}$ comes from an exponential family distribution.*

Then storage of the full data can be avoided through the use of sufficient statistics.

Proof. We have

$$p(\boldsymbol{y}_{t,i}|\boldsymbol{\theta}_{1:t}) = h(\boldsymbol{y}_{t,i})g(\boldsymbol{\theta}_{1:t}) \exp\left\{\eta'(\boldsymbol{\theta}_{1:t}) \cdot T(\boldsymbol{y}_{t,i})\right\}, \tag{C.45}$$

where h and g are scalar-valued functions, and η and T are (possibly) vector-valued functions of the same dimension. Then

$$U(\boldsymbol{y}_t) := \sum_{i=1}^{n_t} T(\boldsymbol{y}_{t,i}) \tag{C.46}$$

is a sufficient statistic for the distribution $p(\boldsymbol{y}_t|\boldsymbol{\theta}_{1:t}) = \prod_{i=1}^{n_t} p(\boldsymbol{y}_{t,i}|\boldsymbol{\theta}_{1:t})$. Further, $\dim U(\boldsymbol{y}_t) \approx \dim \boldsymbol{\theta}_{1:t}$, with $\dim U(\boldsymbol{y}_t) \leq \dim \boldsymbol{\theta}_{1:t}$ unless the distribution is curved.

Then by Theorem 4.3.4, any transition kernel can be computed while only storing the sufficient statistics, U_t. □

C.2 PPRB-within-Gibbs approximation error

Section 4.2.1 deals only with the case when PPRB is used with a parameter space for $\boldsymbol{\theta}$ which is not expanding. In this section, we extend these results to the case in which the parameter space expands with new data.

Lemma C.2.1. *Let $f_1(x)$ and $f_2(x)$ be densities on the same measure space. Let $f_x(y) := f(y|x)$ be a the probability distribution of y conditioned on x, and define the joint distributions $f_i(x,y) = f(y|x)f_i(x)$. Then $\inf_x \|f_x(y)\|_p \cdot \|f_1(x) - f_2(x)\|_p \leq \|f_1(x,y) - f_2(x,y)\|_p \leq \sup_x \|f_x(y)\|_p \cdot \|f_1(x) - f_2(x)\|_p$.*

Proof.

$$\|f_1(x,y) - f_2(x,y)\|_p = \left(\int \int |f(y|x)f_1(x) - f(y|x)f_2(x)|^p \, \mathrm{d}y\mathrm{d}x \right)^{\frac{1}{p}} \tag{C.47}$$

$$= \left(\int \int f_x(y)^p \, |f_1(x) - f_2(x)|^p \, \mathrm{d}y\mathrm{d}x \right)^{\frac{1}{p}} \tag{C.48}$$

$$= \left(\int \|f_x(y)\|_p^p \, |f_1(x) - f_2(x)|^p \, \mathrm{d}x \right)^{\frac{1}{p}} \tag{C.49}$$

$$\leq \left(\int \sup_x \|f_x(y)\|_p^p \, |f_1(x) - f_2(x)|^p \, \mathrm{d}x \right)^{\frac{1}{p}} \tag{C.50}$$

$$= \sup_x \|f_x(y)\|_p \cdot \|f_1(x) - f_2(x)\|_p. \tag{C.51}$$

Similarly,

$$\|f_1(x,y) - f_2(x,y)\|_p = \left(\int \|f_x(y)\|_p^p \, |f_1(x) - f_2(x)|^p \, \mathrm{d}x \right)^{\frac{1}{p}} \tag{C.52}$$

$$\geq \left(\int \inf_x \|f_x(y)\|_p^p \, |f_1(x) - f_2(x)|^p \, \mathrm{d}x \right)^{\frac{1}{p}} \tag{C.53}$$

$$= \inf_x \|f_x(y)\|_p \cdot \|f_1(x) - f_2(x)\|_p. \tag{C.54}$$

□

Corollary C.2.1. *Let $f_1(x)$ and $f_2(x)$ be densities on the same measure space. Let $f_x(y) := f(y|x)$ be a the probability distribution of y conditioned on x, and define the joint distributions $f_i(x,y) = f(y|x)f_i(x)$. Let $p = 1$. Then $\|f_1(x,y) - f_2(x,y)\|_1 = \|f_1(x) - f_2(x)\|_1$.*

Corollary C.2.1 shows that it is sufficient to consider only the accumulating L_1 error of PPRB when interested in the accumulating L_1 error of PPRB-within-Gibbs. PPRB-within-Gibbs at a time

t targets a distribution proportional to

$$p(\boldsymbol{\phi}|\boldsymbol{\theta}, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_t)p(\boldsymbol{y}_t|\boldsymbol{\theta}, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{t-1})F_S^{(t-1)}(\boldsymbol{\theta}) \propto p(\boldsymbol{\phi}|\boldsymbol{\theta}, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_t)A_t,$$

while the true posterior at time t is proportional to

$$p(\boldsymbol{\phi}|\boldsymbol{\theta}, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_t)p(\boldsymbol{\theta}|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_t) = p(\boldsymbol{\phi}|\boldsymbol{\theta}, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_t)T_t.$$

Therefore the PPRB-within-Gibbs target distribution and the true posterior at time t have the form in Lemma C.2.1 and Corollary C.2.1 applies.

C.3 SMCMC and Parallelization Trade-Offs

The sequential nature of the PPRB-within-Gibbs filtering step of Generative Filtering presents a trade-off versus SMCMC. As seen in Figure 4.2 in Section 4.4.1, the filtering step initializes the Generative Filtering ensemble more effectivly than the jumping kernel of SMCMC, resulting in fewer required transition kernel steps to converge to the target distribution. However, the jumping kernel of SMCMC is parallelizable while PPRB-within-Gibbs is not. In Figure C.1 we compare the cumulative runtime required for each method to converge using 8 cores, typical of a personal workstation or laptop. When the jumping kernel and transition kernel are parallelized over 8 cores, in most scenarios, Generative Filtering takes less time to converge than SMCMC. As the number of available cores increases, this trade-off will begin to favor SMCMC in more scenarios. We see an example of this in Section 4.5.

C.4 Pups Sampling Details

The Gibbs sampler uses conjugate full conditional updates for ϕ_s and σ_s^2, and Metropolis-within-Gibbs proposals for $\log(\lambda_{s,t})$, as described in Hooten et al. (2021). We reproduce these full conditional distributions here for convenience. When data have arrived $y_1, \ldots, y_T$ for a time T.

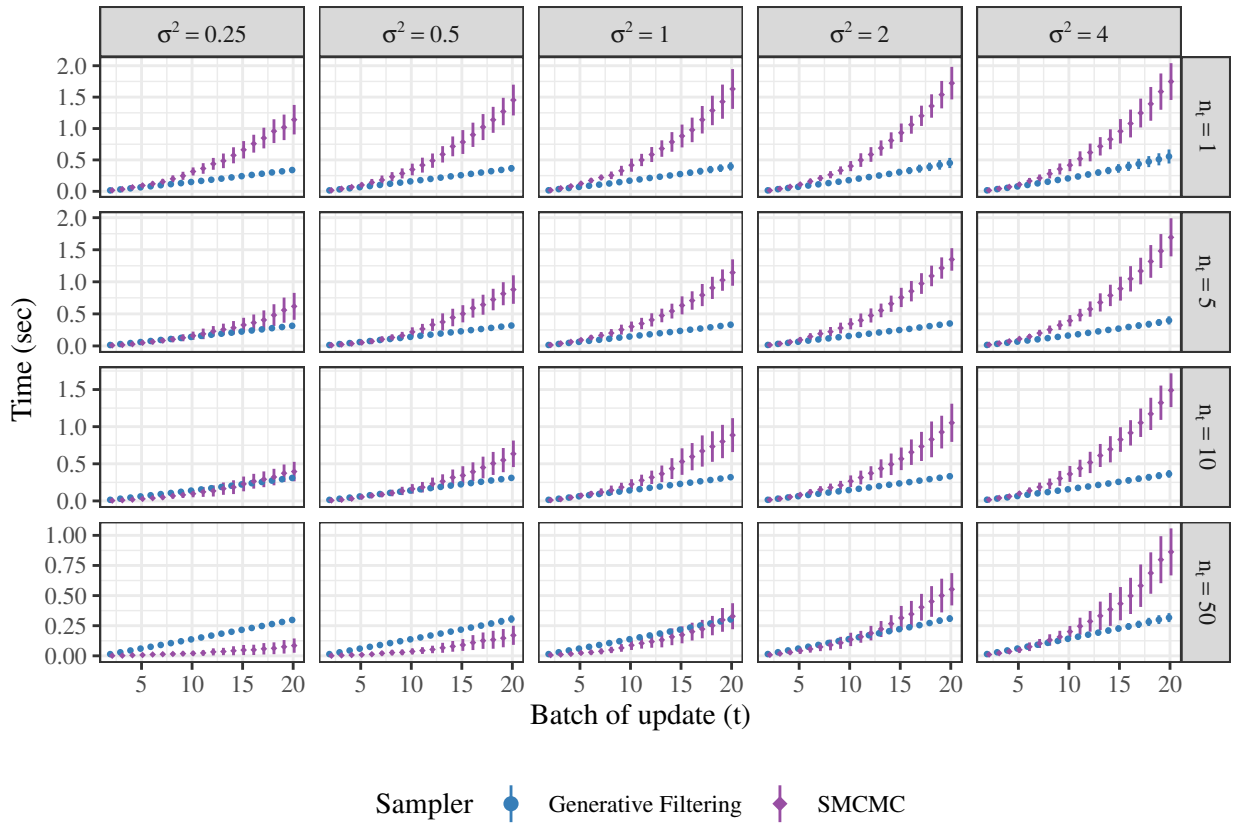


Figure C.1: Cumulative time to reach convergence on 8 cores. Time is shown as mean plus or minus standard deviation across all simulations. This figure illustrates the tradeoff between Generative Filtering and SMCMC. The PPRB step, while creating a better initial value, is not parallelizable. In scenarios where posterior distributions are not strongly affected by new data (e.g., $n_t = 50, \sigma^2 = 0.25$), SMCMC can converge more quickly in time because its jumping kernel is parallelizable.

Then each parameter has full conditional distributions,

$$
\begin{aligned}
\phi_s|\cdot &\sim \mathrm{N}(a^{-1}b, a^{-1}),\\
\sigma_s^2|\cdot &\sim \mathrm{IG}(\tilde{\alpha}, \tilde{\beta}),\\
p(\log(\lambda_{s,t})|\cdot) &\propto \begin{cases}
p(y_{s,1}|\lambda_{s,1})p(\log(\lambda_{s,2})|\phi_s, \sigma_s^2, \log(\lambda_{s,1}))p(\log(\lambda_{s,1})) & \text{for } t = 1,\\
\left(\begin{array}{c} p(y_{s,t}|\lambda_{s,t})p(\log(\lambda_{s,t+1})|\phi_s, \sigma_s^2, \log(\lambda_{s,t})) \\ \times\, p(\log(\lambda_{s,t})|\phi_s, \sigma_s^2, \log(\lambda_{s,t-1})) \end{array}\right) & \text{for } 1 < t < T,\\
p(y_{s,T}|\lambda_{s,T})p(\log(\lambda_{s,T})|\phi_s, \sigma_s^2, \log(\lambda_{s,T-1})) & \text{for } t = T,
\end{cases}
\end{aligned}
$$

where

$$
\begin{aligned}
a &= \frac{T-1}{\sigma_s^2} + \frac{1}{\sigma_\phi^2},\\
b &= \frac{1}{\sigma_s^2}\left(\sum_{t=2}^{T}(\log(\lambda_{s,t}) - \log(\lambda_{s,t-1}))\right) = \frac{\log(\lambda_{s,T}) - \log(\lambda_{s,1})}{\sigma_s^2},\\
\tilde{\alpha} &= \frac{T-1}{2} + \alpha,\\
\tilde{\beta} &= \left(\frac{\sum_{t=2}^{T}(\log(\lambda_{s,t}) - \phi_s - \log(\lambda_{s,t-1}))^2}{2} + \frac{1}{\beta}\right)^{-1}.
\end{aligned}
$$

To update $\log(\lambda_{s,t})$, we draw a proposal $\log(\lambda_{s,t}^*) \sim N(\log(\lambda_{s,t}), \sigma^2_{\mathrm{tune},s,t})$, with proposal variance $\sigma^2_{\mathrm{tune},s,t}$ chosen for each s, t such that the proposals will have an acceptance rate of approximately 0.44.

This Gibbs-style transition kernel is used as the transition kernel for Generative Filtering and SMCMC. The random walk Metropolis proposal for $\log(\lambda_{s,T})$ is used to update this parameter in the PPRB-within-Gibbs sampler.