Scalable Nonparametric Bayes Learning

by

Anjishnu Banerjee

Department of Statistical Science
Duke University

Date: _____

Approved:

_____
David B. Dunson, Supervisor

_____
Surya T. Tokdar

_____
Alan E. Gelfand

_____
Guillermo Sapiro

Dissertation submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in the Department of Statistical Science
in the Graduate School of Duke University
2013

Abstract

Scalable Nonparametric Bayes Learning

by

Anjishnu Banerjee

Department of Statistical Science
Duke University

Date: _____

Approved:

_____
David B. Dunson, Supervisor

_____
Surya T. Tokdar

_____
Alan E. Gelfand

_____
Guillermo Sapiro

An abstract of a dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Statistical Science in the Graduate School of Duke University
2013

Abstract

Capturing high dimensional complex ensembles of data is becoming commonplace in a variety of application areas. Some examples include biological studies exploring relationships between genetic mutations and diseases, atmospheric and spatial data, and internet usage and online behavioral data. These large complex data present many challenges in their modeling and statistical analysis. Motivated by high dimensional data applications, in this thesis, we focus on building scalable Bayesian nonparametric regression algorithms and on developing models for joint distributions of complex object ensembles.

We begin with a scalable method for Gaussian process regression, a commonly used tool for nonparametric regression, prediction and spatial modeling. A very common bottleneck for large data sets is the need for repeated inversions of a big covariance matrix, which is required for likelihood evaluation and inference. Such inversion can be practically infeasible and even if implemented, highly numerically unstable. We propose an algorithm utilizing random projection ideas to construct flexible, computationally efficient and easy to implement approaches for generic scenarios. We then further improve the algorithm incorporating some structure and blocking ideas in our random projections and demonstrate their applicability in other contexts requiring inversion of large covariance matrices. We show theoretical guarantees for performance as well as substantial improvements over existing methods with simulated and real data. A by product of the work is that we discover hitherto

unknown equivalences between approaches in machine learning, random linear algebra and Bayesian statistics. We finally connect random projection methods for large dimensional predictors and large sample size under a unifying theoretical framework.

The other focus of this thesis is joint modeling of complex ensembles of data from different domains. This goes beyond traditional relational modeling of ensembles of one type of data and relies on probability mixing measures over tensors. These models have added flexibility over some existing product mixture model approaches in letting each component of the ensemble have its own dependent cluster structure. We further investigate the question of measuring dependence between variables of different types and propose a very general novel scaled measure based on divergences between the joint and marginal distributions of the objects. Once again, we show excellent performance in both simulated and real data scenarios.

To my parents and my wife.

Contents

List of Tables

List of Figures

List of Abbreviations and Symbols

Symbols

General notes about symbol used in the text.

$\mathbb{R}$	The set of real numbers.
$\mathbb{R}^+$	The set of positive real numbers.
$\mathbb{R}^d$	The set of real valued d dimensional vectors, ie vectors in $\mathbb{R} \times \ldots \times \mathbb{R}$, where the cartesian product is taken.
$\mathbb{N}$	The set of natural numbers, ie integers > 0.
$\mathbb{N}^+$	The set of positive natural numbers.
A^T	Transpose of the matrix A.
$\|\cdot\|_2$	Euclidean norm of a vector or spectral norm of a matrix as the case may be.
$\|\cdot\|_F$	Frobenius norm of a matrix.
$\|\cdot\|_\psi$	Indicating both Frobenius and spectral norms, ie the statement in question is valid when ψ is replaced by 2 or F.
ln	Natural logarithm, ie logarithm to the base e.
$\overline{D}$	The closure of the set D, ie D and all its limit points.

Abbreviations

Some common abbreviations used through the text.

QR	QR decomposition of a matrix, where Q has orthonormal rows and R is upper triangular.

SVD	Singula value decomposition of a matrix, in the special case that the matrix is symmetric positive definite, this corresponds to the spectral decomposition.
GP	Gaussian Process. A stochastic process is said to be a Gaussian process if any finite dimensional realization from it is follows a multivariate Gaussian distribution. More details in the text.
MCMC	Markov chain Monte Carlo. A common method applied often applied in Bayesian settings, of obtaining samples, approximately distributed as the posterior distribution.
iid	Independent and identically distributed random variables.
ESS	Effective sample size from the convergence diagnostics of a Markov Chain Monte Carlo algorithm
MSE	Mean squared error
RMSE	Relative mean squared error. Also used for root mean squared error. Specified at places of usage, what is being used.
PP	Predictive process, an approximation method for Gaussian processes. More details in text.
RP	Random projection, an approximation method by using random matrices. More details in text.

Acknowledgements

First of all, I would like to thank my advisor, Prof David Dunson. He has been a constant source of encouragement and inspiration for me. I admire his unbridled enthusiasm and dedication. I will remain forever grateful to him for having patience with me, when I almost missed conference deadlines, for encouragements when simulations failed to produce desired results or when codes refused to run after days of work. I have hardly ever seen anyone like him who could come up with new ideas in response to every statistical problem that we had to face and I surely would not have been completing this thesis if not for him.

I thank my committee members, Prof Surya Tokdar, Prof Alan Gelfand, Prof Guillermo Sapiro, for firstly agreeing to be on my committee, in spite of their extremely busy schedules and for providing me valuable feedback which has greatly enhanced the quality of my research. On the same note, I must thank Prof Mauro Maggioni, for serving on PhD preliminary examination committee, Prof Scott Schmidler, Prof Sayan Mukherjee, Prof Robert Wolpert for having taught me some wonderful courses. Prof Mike West for having taken care of all of us students here at Duke Statistics. I must mention my collaborator Dr Sandeep Dave, whose data sets were some of the original motivations behind the problems addressed in this thesis.

Thanks go to my fellow PhD colleagues, Jared Murray, for being your ever cheerful self and helping me with the code on the joint modeling project, to Christopher Challis for some wonderful memories and encouragement, to David Mcclure, my

office mate, for helping me get through some intolerable periods of depression, to Silvia Montagna, Fernando Bonassi, Jouchi Nakajima, Andrew Cron, Kai Cui, for some wonderful memories and help at different times during my stay at Duke. It would not have been possible without you all.

An earnest and heartfelt gratitude to my long standing friends and comrades in Chapel Hill and Raleigh. Sayan Dasgupta, for being yourself, Abhishek Pal Majumder, for helping me get through some tough periods, Pourab Roy, for your enthusiasm, Siddhartha Mandal for the fun we had together, Sayantan Banerjee, for the brainstorming sessions and discussions, Pratyaydipta Rudra, Ritwik Chowdhury, Sujatro Chakladar for all the encouragement and help. To Rinku Majumdar and Samarpan Majumdar, you were the closest thing to a family I had in this country, for absorbing all my emotional rants and letting me be myself. I will be missing you all in my next venture, you were the reason I had never felt vulnerable or lonely.

I also express my deep gratitude and thanks to seniors from my alma mater, Indian Statistical Institute, Kolkata, and then here at Duke, Avishek Chakraborty, Chiranjeet Mukherjee, Debdeep Pati and Anirban Bhattacharyya, for all the get togethers, nightly cricket sessions, for helping me when I had first come to Durham in helping me settle in, get a car, for everything you all have done for me and helped me. To Sudhanshu Garg, for being a wonderful flatmate through these four years, Rahul Ghosh, a wonderful friend, and Sambuddha Banerjee - you were like an elder brother for me.

Thanks to my erstwhile batchmates from Indian Statistical Institute, Kolkata, who are doing their PhDs in different universities across the United States, Joyjit Roy, Rajarshi Mukherjee, Apratim Ganguly, Sumit Mukherjee, Anirban Basak, Pallavi Basu, Srijan Sengupta, thanks for all help when I got stuck in with certain proofs, for all the academic and non-acadmeic conversations we have had together, for your hospitality and encouragement, when I have visited you.

Thanks go to the funding agencies, NIH, NSF, who have helped me travel to conferences to learn new stuff, present my work and meet great people. My overall experience as a PhD student was much enhanced by taking part in some of the best statistical meetings around the globe.

Finally, I must thank my parents, Ashoke Kumar Banerjee, Kalyani Banerjee, and my wife, Saptadweepa Banerjee. You have been with me through thick and thin, made endless sacrifices for me. I would have never made this far without you and hope I have been able to fulfill some of the dreams you had for me.

(As a concluding note of thanks, I am sure there are plenty of others to whom I am immensely grateful, but I have forgotten to mention their names here - apologies and thanks to all of you.)

1

Introduction

1.1 Motivation

We are living in a world that is seeing a deluge of information in almost all scientific disciplines, comprising of data of enormous sizes and complex types . In cancer-genome studies, next generation sequencing is becoming commonplace, with ever decreasing costs and increasing accuracy of sequencing tissues and identifying mutations, along with collection of other genetic information, like copy number variation and patient demographics. In the internet domain, we are enable to capture enormous amounts of information regarding peoples' online usage and preferences, from their social networks, shopping patterns, history of webpages visited, among a host of other things. In atmospheric and environmental studies, we are presented with emissions information from very large numbers of monitoring stations placed at dense sets of locations. In astrophysical studies, we try to predict red shifts and galactic movements with information from sky surveys, which involve several gigabytes of data, an example being the Sloan Sky Digital Survey (Stoughton et al., 2007). The sources of large and complex data in modern scientific applications is endless.

These large data present exciting statistical modeling opportunities, while also

having their share of problems for storage and analysis. Traditionally statistical learning algorithms have focused on inference based on small samples and studying the properties of these inference procedures as the sample sizes $n \to \infty$. Traditional statistical procedures also typically focus on probabilistic models for one kind of data. With the enormous amount of data being available in many disciplines, the focus of modern learning algorithms has shifted more towards discovering latent lower dimensional structures in very large data sizes for meaningful and practicable inference strategies. While large data facilitate flexible estimation and complex models, an important concern is to make these flexible but often computationally intensive methods scale up to the enormous sizes of these data. Another important consideration is flexible borrowing of information across different data types, as opposed traditional modeling of one type of data. In this thesis I focus on these two questions:

- strategies for scalable and flexible learning for very large data, while accounting for prior information, and,
- jointly modeling for complex ensembles of data, allowing for differences within and between data types.

1.2 Background, organization and literature review

Broadly, this thesis can be broadly divided as solving two kinds of problems - the first part concerned with scalable methods for nonparametric regression, comprising of chapters $2, 3$ and second part dealing with modeling of joint distributions of objects, comprising chapter 4. Finally, in chapter 5, we describe a cancer-genomic study, which requires methods for both the kinds of problems discussed previously. I now provide some background and relevant literature review for what follows in the thesis.

One of the most common tools of statistics is linear regression (Draper et al., 1966). Modern methods have focused on more flexible forms of regression, making

the shape of predictors in relation to the response as flexible as possible (Friedman et al., 2001). In the simplest setting, the problem of nonparametric regression can be represented as,

$$y_i = \sum_{j=1}^{p} \beta^j f^j(x_i^j) + \epsilon_i, \tag{1.1}$$

where y_i is the response variable, possibly multivariate, β^j are unknown regression coefficients, $f^j(\cdot)$ is the unknown function capturing the shape of the predictor x_i^j in relation to the response y_i, ϵ_i is the error term, for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, p$. The downside of the more flexible modern regression methods is that they involve substantially more complex computations for estimation of $\beta^j, f^j(\cdot)$ and prediction and hence problems in scaling up to large data. In the aforementioned nonparametric setting, we may envision three different scenarios for our large data, viz,

- When the number of observations, or the sample size, n is large, for example, in data from the astrophysical study, Sloan Sky Digital Survey (Stoughton et al., 2007), and our interest is in studying the effect on red-shift of galaxies, given several other covariates. In this example we have billions of observations and usual nonparametric methods do not scale up to such data sizes. The problem is accentuated in Bayesian settings (Box and Tiao, 1973), which allows us to make probability statements about the unknowns $\beta^j, f^j(\cdot)$ instead of just point estimates and also enable incorporation of prior information, but will involve likelihood evaluation involving the large data at every iteration of a large number of iterations of a sampling algorithm.

- When the number of predictors p is large, possibly with $p >> n$. Traditional statistical analysis has always held that the observed sample size n has to be more than the number of predictors p for the unknown coefficients to be estimated properly (Huber et al., 1996). In many modern applications this

does not hold true, as an example, consider aforementioned cancer-genome studies or gene-environment studies, where typically the number of mutations, or polymorphisms (the predictors) are several orders larger than the number of patients (sample size) from whom we have data. Many methods have been proposed to circumvent the problems, some Bayesian methods being stochastic search variable selection (George and McCulloch, 1996), factor models (West, 2003) and more recently, Bayesian sufficient dimension reduction (Tokdar et al., 2010), Bayesian additive regression trees (Chipman et al., 2010), among a host of others.

- When the number of observations n, as well as the number of predictors p are both very large. This is an area that has come into recent focus and common strategies are to use a mix of methods from the $p >> n$ scenario and adjust them to this scenario (Johnstone and Titterington, 2009). Examples of this scenario include prediction in biological settings for sequences of images (the individual image pixels being the response), microarray analysis, among others.

Large n settings settings in nonparametric regression are particularly troublesome due to cumbersome likelihood evaluation and often necessary matrix inversions. The majority of the focus in the literature in dealing with problems of the large data scenarios is to assume an underlying accurate low dimensional representation and to learn this low dimensional representation from the available data.

In the first part of this thesis, we focus instead on randomized projections of the large data onto lower dimensions, instead of trying to explicitly learn about lower dimensional representations. We show that theoretically, under mild conditions, these random lower dimensional projections are minimally different from the full representations with high probability and empirically, they have excellent performance and approximation accuracy. We consider several variants of this idea and present a

unifying theoretical set-up tying them up.

The idea for random lower dimensional projections comes from apparently unrelated ideas in the domain of signal processing, called compressive sensing (Donoho, 2006; Candès et al., 2006). Compressive sensing, (as well as related ideas of manifold learning (Lee and Verleysen, 2007) and graph embedding (Gross and Tucker, 2001)) rely on the Johnson Lindenstrauss lemma (Johnson and Lindenstrauss, 1984), which says that there exists a function, projecting vectors from a higher dimensional space to a lower dimensional one, such that the distances between the vectors are almost preserved. Later developments have shown that a wide variety of random matrices preserve the Johnson Lindenstrauss property, in the sense of preserving norms. For example, let x be a vector in $\mathcal{R}^n$ and Ω be an $m \times n$ random matrix satisfying the conditions required for the Johnson Lindenstrauss lemma, then the euclidean norm, $\|\Omega x\|$ is very close to the norm, $\|x\|$ with high probability (we formalize these ideas in more detail in the later chapters). Typical examples of such matrices Ω are matrices whose entries are sampled independently from some continuous distribution with fast decaying tails, like the standard normal distribution and appropriately scaled.

One of the main contributions of this thesis is to incorporate ideas from apparently unrelated developments in compressive sensing in the context of Bayesian nonparametric regression. Our scalable algorithms use several variants of Johnson Lindenstrauss matrices and other ideas from random matrix theory to come up with very efficient and stable approaches. To begin with, in chapter 2 we focus on approximation algorithms for large n in the context of Gaussian process regression, a very popular tool in Bayesian nonparametric regression. In large n settings, Gaussian process regression becomes inefficient, because likelihood evaluation and inference involves inversion of a covariance matrix of size $n \times n$. This is not only computationally burdensome but extremely numerically unstable, leading to unreliable estimates and predictions. The computational inefficiency associated with Gaussian process re-

gression has received quite a bit of attention in the literature, (partly because of the flexibility of using the Gaussian processes tool), and approximations can be broadly classified into two categories,

- Approximations to the stochastic process itself, which includes the approaches like process convolutions (Higdon, 2002) and the currently popular approach in the Bayesian literature, predictive processes (Banerjee et al., 2008).

- Approximations viewed as a low rank matrix approximation to the covariance matrix, more popular in the machine learning contexts (Quinonero Candela and Rasmussen, 2005).

Another contribution in chapter 2 is to tie up the above two viewpoints for approximate Gaussian process regression. In fact we show that predictive processes and subset of regressors approach in machine learning are equivalent, a fact that has been unreported in the literature. We then build a general approximation scheme, which we call projection approximation and show that all of predictive processes, subset of regressors and other related approaches are in fact special cases of our construction. We then incorporate ideas from random matrix theory and compressive sensing to build algorithms, using randomized projections onto lower dimensional subspaces, that have superior theoretical performance guarantees and good performance in real and simulated data sets.

In chapter 3 we extend ideas from chapter 2 to further improve the projection algorithms. Post multiplying a covariance matrix K by a random matrix Ω and then utilizing approximate decompositions of the product $K\Omega$ is the crux of one of the algorithms in chapter 2. We extend this idea further to consider other types of structured random matrices Ω that enable faster matrix multiplies and blocking schemes, so that the entire decomposition starting from the matrix multiplication may be executed on multicore computing architecture. In addition to this, we look

at other application areas where similar ideas of approximating large positive definite matrices with random projections maybe applied. We also consider a theoretical framework to study the decay rates of eigenvalues, which is why low rank approximations work. We also investigate theoretically the numerical stability of the equivalent linear systems and show that random projection ideas substantially improve this, in terms of matrix conditioning numbers, so that such approximations maybe desirable not only for having better computational efficiency, but actually leading to better estimates.

In chapter 4, we address the other question of jointly modeling complex object ensembles. Surprisingly, as opposed to regression or classification, joint modeling has received very limited attention in the literature. There are some Bayesian approaches for relational data modeling (Airoldi et al., 2006), which consider joint modeling of one type or data, or just exploring the correlation structure. Once again, considering data from a genome-cancer study as an example, we would be interested in relationships between mutation status, copy number variation, drug response etc., without necessarily treating one variable as a response and others as predictors/covariates. Some simplistic joint modeling approaches, assuming same structure across all entities making up the ensemble have been proposed in the literature (Bigelow and Dunson, 2009; Dunson and Bhattacharya, 2010; Dunson, 2009). In chapter 4 we propose a novel model for joint modeling of complex ensembles that allows ensemble entities to have separate but dependent clustering structure. We also devise an innovative sampling scheme based on slice sampling (Walker, 2007), that overcomes the difficulty of having to truncate infinite Dirichlet mixture models. This slice sampling scheme was designed in collaboration with Jared Murray, in a co-authored paper, Banerjee et al. (2013a). We also consider the scenario, when our sole interest is in the dependence structure of the entities and come up with a novel scaled version of divergence between probability distributions that allows us to measure generic

dependencies.

The study presented in chapter 5 was done in collaboration with Dr Sandeep Dave and people in his laboratory. I provided some of statistical insight driving the study, which was subsequently published as Love et al. (2012); Zhang et al. (2013) . I present in this thesis, some of methods employed, which are highly reliant on models described in the previous chapters. We also present some interesting side questions and the methods we use to deal with them, alongwith some highlights of the results from the study. We conclude the thesis with a chapter discussing some current projects and potential future directions.

2

Efficient Gaussian Process Regression for Large Data Sets

2.1 Summary

Gaussian processes (GPs) are widely used in nonparametric regression, classification and spatio-temporal modeling, motivated in part by a rich literature on theoretical properties. However, a well known drawback of GPs that limits their use is the expensive computation, typically $O(n^3)$ in performing the necessary matrix inversions with n denoting the number of data points. In large data sets, data storage and processing also lead to computational bottlenecks and numerical stability of the estimates and predicted values degrades with n. To address these problems, a rich variety of methods have been proposed, with recent options including predictive processes in spatial data analysis and subset of regressors in machine learning. The underlying idea in these approaches is to use a subset of the data, leading to questions of sensitivity to the subset and limitations in estimating fine scale structure in regions that are not well covered by the subset. Motivated by the literature on compressive sensing, we propose an alternative random projection of all the data points onto a

lower-dimensional subspace. We demonstrate the superiority of this approach from a theoretical perspective and through the use of simulated and real data examples.

2.2 Introduction

In many application areas we are interested in modeling an unknown function and predicting its values at unobserved locations. Gaussian processes are used routinely in these scenarios, examples include modeling spatial random effects (Banerjee et al., 2004; Cressie, 1992) and supervised classification or prediction in machine learning (Rasmussen, 2004; Seeger, 2004). Gaussian processes are mathematically tractable, have desirable properties and provide a probabilistic set-up facilitating statistical inference. When we have noisy observations $y_1, \ldots, y_n$ from the unknown function $f : \mathcal{X} \to \Re$ observed at locations $x_1, \ldots, x_n$ respectively, let

$$y_i = f(x_i) + \epsilon_i, \quad \text{for } i = 1, \cdots, n, \tag{2.1}$$

where ϵ_i is the associated idiosyncratic noise. We let $\epsilon_i \sim \mathbb{N}(0, \sigma^2)$ for simplicity. However, for the techniques we develop here, other noise distributions may be used, including heavy tailed ones. The unknown function $f(\cdot)$ is assumed to be a realization from a Gaussian process with mean function $\mu(\cdot)$ and positive definite covariance kernel $k(\cdot, \cdot)$, so that $\mathrm{E}\{f(x)\} = \mu(x)$ and $\mathrm{cov}\{f(x), f(x')\} = k(x, x')$ for all $x, x' \in \mathcal{X}$.

The realizations of $f(\cdot)$ at the sample points $x_1, \ldots, x_n$ have a multivariate Gaussian prior, with evaluations of the resulting posterior and computations involved in calculating predictive means and other summaries involving $O(n^3)$ computation unless the covariance has a special structure that can be exploited. Markov chain Monte Carlo algorithms for posterior computation allowing uncertainty in the residual variance σ^2 and unknown parameters in the mean function $\mu(\cdot)$ and covariance $k(\cdot)$ may require such computations at every one of a large number of iterations. Another concern is declining accuracy of the estimates as the dimension increases, as

matrix inversion becomes more unstable with the propagation of errors due to finite machine precision. This problem is more acute if the covariance matrix is nearly rank deficient, which is often the case when $f(\cdot)$ is considered at nearby points.

The above problems necessitate approximation techniques. Most approaches approximate $f(\cdot)$ with another process $g(\cdot)$ that is constrained to a reduced rank subspace. One popular strategy specifies $g(\cdot)$ as a kernel convolution (Higdon, 2002), with related approaches instead relying on other bases such as low rank splines or moving averages (Wikle and Cressie, 1999; Xia and Gelfand, 2006; Kammann and Wand, 2003). A concern with these approaches is the choice of basis. There are also restrictions on the class of covariance kernels admitting such representations. Banerjee et al. (2008) instead proposed a predictive process method that imputes $f(\cdot)$ conditionally on the values at a finite number of knots, with a similar method proposed by Tokdar (2007) for logistic Gaussian processes. Subset of regressors (Smola and Bartlett, 2001) is a closely related method to the predictive process that was proposed in the machine learning literature and essentially ignored in statistics. Both of these approaches substantially underestimate predictive variance, with Finley et al. (2009) proposing a bias correction in the statistics literature and Snelson and Ghahramani (2006) independently developing an essentially identical approach in machine learning. Alternative methods to adjust for underestimation of predictive variance were proposed in Seeger et al. (2003) and Schwaighofer and Tresp (2002).

Quinonero Candela and Rasmussen (2005) proposed a unifying framework that encompasses essentially all of these subset of regressors-type approximation techniques, showing that they can be viewed as an approximation to the prior on the unknown function, rather than its posterior. While these methods do not require choice of a basis, an equally difficult problem arises in determining the location and spacing of knots, with the choice having a substantial impact. In Tokdar (2007) in the context of density estimation and in unpublished work by Guhaniyogi, Finley,

Banerjee and Gelfand in the context of spatial regression, methods are proposed for allowing uncertain numbers and locations of knots in the predictive process using reversible jump and preferential sampling. Unfortunately, such free knot methods increase the computational burden substantially, partially eliminating the computational savings due to a low rank method. In the machine learning literature, various optimization methods have been proposed for knot selection, typically under the assumption that the knots correspond to a subset of the data points. Such methods include online learning (Csató and Opper, 2002), greedy posterior maximization (Smola and Bartlett, 2001), maximum information criterion (Seeger et al., 2003), and matching pursuit (Keerthi and Chu, 2006) among others.

In this article, we propose a new type of approximation method that bypasses the discrete knot selection problem using random projections. The methodology is straightforward to implement in practice, has a theoretical justification and provides a natural generalization of knot-based methods, with pivoted factorizations and the intuitive algorithm of Finley et al. (2009) arising as special cases. Motivated by Sarlos (2006) and Halko et al. (2011) we use generalized matrix factorizations to improve numerical stability of the estimates, a problem which is typically glossed over. The inspiration for our method arises out of the success of random projection techniques, such as compressed sensing (Candès et al., 2006; Donoho, 2006), in a rich variety of contexts in machine learning. Most of this literature focuses on the ability to reconstruct a signal from compressive measurements, with theoretical guarantees provided on the accuracy of a point estimate under sparsity assumptions. In contrast, our goal is to accurately approximate the posterior distribution for the unknown function in a fundamentally different setting. We also explore how these approximations affect inference on the covariance kernel parameters controlling smoothness of the function, an issue essentially ignored in earlier articles. Our theory suggests that predictive process-type approximations may lead to high correlations between the imputed pro-

cess and parameters, while our method overcomes this problem.

2.3 Random Projection Approximation Methodology

2.3.1 Predictive Processes and Subset of Regressors

As a first step, we place the predictive process and subset of regressors methods under a common umbrella. Consider equation (2.1), with $\mu \equiv 0$ for notational clarity, and let $X^* = \{x_1^*, \ldots, x_m^*\}$ denote a set of knots in $\mathcal{X}$. Letting $f^* = f(X^*) = \{f(x_1^*), \ldots, f(x_m^*)\}^T$ denote the function $f(\cdot)$ evaluated at the knots, the predictive process replaces $f(\cdot)$ by $g(\cdot) = E\{f(\cdot)|f^*\}$, with $g(\cdot)$ a kriged surface in spatial statistics terminology (Stein, 1999). It follows from standard multivariate normal theory that for any $x \in \mathcal{X}, g(x) = (k_{x,*})^T (K_{*,*})^{-1} f^*$, where $k_{x,*}$ is the $m \times 1$ vector $\{k(x, x_1^*), k(x, x_2^*), \ldots, k(x, x_m^*)\}^T$ and $K_{*,*}$ is the $m \times m$ matrix with $k(x_i^*, x_j^*)$ in element i, j.

Subset of regressors is instead obtained via an approximation to $K_{f,f} = \text{cov}\{f(X)\}$. Letting

$$K_{aug} = \text{cov}[\{f(X)^T, (f^*)^T\}^T] = \begin{pmatrix} K_{f,f} & K_{f,*} \\ K_{*,f} & K_{*,*} \end{pmatrix},$$

an optimal (in a sense to be described later) approximation to $K_{f,f}$ is obtained as $Q_{f,f} = K_{f,*}(K_{*,*})^{-1} K_{*,f}$, with $Q_{i,j} = K_{i,*}(K_{*,*})^{-1} K_{*,j}$ denoting cell (i, j) of $Q_{f,f}$.This approximation $Q_{f,f}$ is equivalent to $\text{cov}\{g(X)\}$ obtained from the predictive process approximation, and hence the two approaches are equivalent. As shown in Quinonero Candela and Rasmussen (2005), $g(\cdot)$ is effectively drawn from a Gaussian process with the degenerate covariance kernel

$$q_{SOR}(x, z) = (k_{x,*})^T (K_{*,*})^{-1} k_{*,z},$$

where $k_{*,z} = \{k(x_1^*, z), \ldots, k(x_m^*, z)\}^T$. From equation (2.1), we obtain $Y = (y_1, \ldots, y_n)^T \sim \mathbb{N}(0; \sigma^2 I + K_{f,f})$ in marginalizing out f over the exact prior $f \sim \text{GP}(0, k)$. If we use

the approximated version, we have

$$y_i = g(x_i) + \epsilon_i, \quad \epsilon_i \sim \mathbb{N}(0, \sigma^2). \tag{2.2}$$

Marginalizing out g, we obtain $Y \sim \mathbb{N}(0, \sigma^2 I + Q_{f,f})$.

Let Y_o denote the vector of length n_o of observed values and Y_p the vector of length n_p of values to predict, with $n_o + n_p = n$. Under the above approximation, the conditional predictive distribution is $(Y_p|Y_o) \sim \mathbb{N}\{Q_{p,o}(Q_{o,o} + \sigma^2 I)^{-1} Y_o, Q_{o,o} - Q_{p,o}(Q_{o,o} + \sigma^2 I)^{-1} Q_{o,p}\}$, with $Q_{o,o}, Q_{o,p}, Q_{p,o}$ denoting submatrices of $Q_{f,f}$. Using the Woodbury matrix identity (Harville, 2008) yields $(Q_{o,o} + \sigma^2 I)^{-1} = \sigma^{-2}\{I - K_{o,*}(\sigma^2 K_{*,*} + K_{*,o} K_{o,*})^{-1} K_{*,o}\}$, with calculation involving an $m \times m$ matrix.

Finley et al. (2009) show that the predictive process systematically underestimates variance, since at any $x \in \mathcal{X}$, $\text{var}\{f(x)\} - \text{var}\{g(x)\} = \text{var}\{f(x) \,|\, f^*\} > 0$. To adjust for this underestimation, they replace $g(\cdot)$ by $g(\cdot) + \epsilon_g(\cdot)$, with $\epsilon_g(x) \sim \mathbb{N}\{0, k(x,x) - k_{x,*}^T (K_{*,*})^{-1} k_{x,*}\}$ and $\text{cov}\{\epsilon_g(x_1), \epsilon_g(x_2)\} = 0$ for $x_1 \neq x_2$. Hence, in place of equation (2.2), we have

$$y_i = g(x_i) + \epsilon_g(x_i) + \epsilon_i, \quad \epsilon_i \sim \mathbb{N}(0, \sigma^2).$$

A variety of methods for addressing the variance under-estimation problem were independently developed in the machine learning literature (Quinonero Candela and Rasmussen, 2005), with the fully independent training conditional approximation corresponding exactly to the Finley et al. (2009) approach. Snelson and Ghahramani (2006) also proposed this approach under the sparse Gaussian process with pseudo inputs moniker. In each of these cases, $g_M(\cdot) = g(\cdot) + \epsilon_g(\cdot)$ is effectively drawn from a Gaussian process with the degenerate covariance kernel

$$q_{FITC}(x,z) = q_{SOR}(x,z) + \delta(x,z)\{k(x,z) - q_{SOR}(x,z)\},$$

where $\delta(x,z) = 1$ if $x = z$ and 0 otherwise. Our proposed random projection method

will generalize these knot-based approaches, leading to some substantial practical advantages.

2.3.2 Generalization: Random Projection Method

The key idea for random projection approximation is to use $g_{RP}(\cdot) = E\{f(\cdot)|\Phi f(X)\}$ instead of $g(\cdot) = E\{f(\cdot)|f^*\}$, where Φ is some $m \times n$ matrix. The approximation $g_{RP}(\cdot)$ is drawn from a Gaussian process with covariance kernel,

$$q_{RP}(x, z) = (\Phi k_{x,f})^T (\Phi K_{f,f} \Phi^T)^{-1} \Phi k_{f,z},$$

where $k_{x,f} = \{k(x, x_1), \ldots, k(x, x_n)\}^T$ and $k_{f,z} = \{k(x_1, z), \ldots, k(x_n, z)\}^T$. As in the methods of §2 · 1, we face the variance under-estimation issue with $\text{var}\{f(x)\} - \text{var}\{g_{RP}(x)\} = \text{var}\{f(x) \,|\, \Phi f(X)\} > 0$. Following the same strategy for bias correction as in Finley et al. (2009), we let $g_{RM}(\cdot)$ denote the modified random projection approximation having covariance kernel

$$q_{RM}(x, z) = q_{RP}(x, z) + \delta(x, z)\{k(x, z) - q_{RP}(x, z)\}. \tag{2.3}$$

When Φ is the submatrix formed by m rows of a permutation matrix of order n (Harville, 2008), we revert back to the formulation of §2 · 1, where the knots are an m dimensional subset of the set of data locations X. We consider more generally $\Phi \in \mathcal{C}$, the class of random matrices with full row-rank and with row-norm = 1 to avoid arbitrary scale problems. Before discussing construction of Φ, we consider some properties of the random projection approach.

2.3.3 Properties of the RP method

(1) Limiting Case: When $m = n$, Φ is a square non-singular matrix. Therefore, $(\Phi K_{f,f} \Phi^T)^{-1} = (\Phi^T)^{-1} K_{f,f}^{-1} \Phi^{-1}$, so that $Q_{f,f}^{RP} = K_{f,f}$, and we get back the original process with a full rank random projection.

(2) Optimality in terms of Hilbert space projections: It is well known in the theory of kriging (Stein, 1999) that taking conditional expectation gives the orthogonal projection into the corresponding space of random variables. Let $\mathcal{H}\{f(X), \Phi\}$ denote the Hilbert space spanned by linear combinations of the m random variables $\Phi f(X)$ and equipped with the inner product $\langle f_1, f_2 \rangle = E(f_1 f_2)$ for any $f_1, f_2 \in \mathcal{H}\{f(X), \Phi\}$. The orthogonal projection of f to the Hilbert space is $f^{opt} = argmin_{h \in \mathcal{H}\{f(X,\Phi\}} \|f - h\|$. From kriging theory $f^{opt}(x) = (\Phi k_{x,f})^T (K_{f,f})^{-1} \Phi f(X) = E\{f(x)|\Phi f(X)\}$. Hence, the random projection approximation is optimal in this sense. As f^{opt} is a function of $\Phi \in \mathcal{C}$, the best possible random projection approximation to f could be obtained by choosing Φ to minimize $\|f^{opt} - f\|$. As the predictive process-type approaches in §2 · 1 instead restrict Φ to a subset of $\mathcal{C}$, the best possible approximation under such approaches is never better than that for the random projection. While finding the best Φ is not feasible computationally, §3 proposes a stochastic search algorithm that yields approximations that achieve any desired accuracy level with minimal additional computational complexity.

(3) Relationship with partial matrix decompositions: We briefly discussed in §2·1 that the approximations in the machine learning literature were viewed as reduced rank approximations to the covariance matrices. Here we make an explicit connection between matrix approximation and our random projection scheme, which we build on in the next section. The Nyström scheme (Drineas and Mahoney, 2005) considers the rank m approximations to $n \times n$ positive semidefinite matrix A using $m \times n$ matrix B, by giving an approximate generalized Choleski decomposition of A as CC^T, where $C = (BA)^T (BAB^T)^{-1/2}$. The performance of the Nystöm scheme depends on how well the range of B approximates the range of A. As in property (1), let $Q^{RP}_{f,f}$ be the random projection approximation to $K_{f,f}$. It is easy to see that $Q^{RP}_{f,f}$ corresponds to

a Nyström approximation to $K_{f,f}$, with $C = (\Phi K_{f,f})^T (\Phi K_{f,f} \Phi^T)^{-1/2}$.

The Nyström characterization allows us to obtain a reduced singular value decomposition utilizing the positive definite property as considered in detail in §3. The partial Choleski decompositions for the covariance matrices, advocated in Foster et al. (2009) for approaches in §2 · 1, arise as special cases of the Nyström scheme using permutation submatrices; arguing on the lines of property (2), best case accuracy with the random projection is at least as good as the partial Choleski decomposition. We later show empirically random projection performs substantially better.

(4) Relationship with truncated series expansions: The random projection approximation also arises from a finite basis approximation to the stochastic process f. Under the Karhunen-Loéve expansion (Adler, 1990),

$$f(x) = \sum_{i=1}^{\infty} \eta_i (\lambda_i)^{1/2} e_i(x), \quad x \in \mathcal{X},$$

where $\mathcal{X}$ is compact and λ_i, e_i are eigenvalues and eigenvectors, respectively, of the covariance function k, given by the Fredholm equation of the second kind as (Grigoriu, 2002),

$$\int_{\mathcal{X}} k(x_1, x) e_i(x) dx = \lambda_i e_i(x_1), \quad x \in \mathcal{X}.$$

η_i's are independent $\mathbb{N}(0, 1)$ random variables by virtue of properties of the Gaussian process. Using Mercer's theorem, which is generalization of the spectral theorem for positive definite matrices, we can express the covariance function as (Grigoriu, 2002),

$$k(x_1, x_2) = \sum_{i=1}^{\infty} \lambda_i e_i(x_1) e_i(x_2), \quad x_1, x_2 \in \mathcal{X}.$$

Assume that the eigenvalues in each of the above expansions are in descending order. Let $f_{tr}(x) = \sum_{i=1}^{m} \eta_i (\lambda_i)^{1/2} e_i(x)$ be the approximation to $f(x)$ obtained by finitely

truncating the Karhunen-Loéve expansion, keeping only the m largest eigenvalues. The covariance function for f_{tr} is given by $k_{tr}(x_1, x_2) = \sum_{i=1}^{m} \lambda_i e_i(x_1) e_i(x_2), \quad x_1, x_2 \in \mathcal{X}$, which is, as expected, a corresponding truncation of the expression in Mercer's theorem. If we now evaluate the truncated covariance function on the set of points of interest, X, we get the covariance matrix, $K_{tr} = E\Lambda E^T$, where E is the $n \times m$ matrix with $(i, j)^{th}$ element given by $e_j(x_i)$ and Λ is a $m \times m$ diagonal matrix with the m eigenvalues in its diagonal.

The Karhunen Loéve expansion considers orthogonal functions so that $\int_{\mathcal{X}} e_i(x) e_j(x) dx = 0$ whenever $i \neq j$. If we use the quadrature rule with equal weights for approximation of the integral with the n locations of interest, we have $\sum_{l=1}^{n} e_i(x_l) e_j(x_l) = 0$, which means that the matrix E is approximately row-orthogonal. Assuming that E is exactly orthogonal the truncated Mercer expansion matrix K_{tr} is essentially a reduced rank m spectral decomposition for the actual covariance matrix. The covariance matrix of the random vector $g_{RP}(X)$ is equal to the rank m spectral decomposition when we choose the projection matrix Φ equal to the first m eigenvectors of the actual covariance matrix, as shown in the next section. Therefore $g_{RP}(X)$ has the same probability distribution as $f_{tr}(X)$. In other cases, when $\Phi \neq$ the eigenvectors, as in approaches in section §2 · 1, its easy to show that the random projection corresponds to some other truncated basis expansion in the same way as above. The Karhunen Loéve is however the optimal expansion in the sense that for each m, for any other $h_{tr}(\cdot)$ from some m truncated basis expansion, $\int_{\mathcal{X}} E[\{f(x) - h_{tr}(x)\}^2] dx$ is minimized over $h_{tr}(\cdot)$, for $h_{tr}(\cdot) = f_{tr}(\cdot)$ (Ghanem and Spanos, 2003).

2.4 Matrix Approximations & Projection Construction

2.4.1 Reduced rank matrix approximations

We introduce stochastic matrix approximation techniques that enable us to calculate nearly optimal projections. We start with some key concepts from linear algebra.

Let $\|\cdot\|_2$ and $\|\cdot\|_F$ denote the spectral and Frobenius norm for matrices and let K be any $n \times n$ positive definite matrix. We focus entirely on positive definite matrices. A spectral decomposition of K is given by, $K = UDU^T$, where D is a diagonal matrix whose diagonal elements are the eigenvalues. Since K is positive definite, this is equivalent to the singular value decomposition and the eigenvalues are equal to the singular-values and > 0. U is an orthonormal matrix whose columns are eigenvectors of K. Consider any $n \times n$ permutation matrix P, and since $PP^T = I$ we have,

$$UDU^T = UPP^TDPP^TU^T = (UP)(PDP^T)(UP)^T.$$

Therefore any permutation of the singular values and their respective vectors leads to an equivalent spectral decomposition for K, and it can be shown that the spectral decomposition is unique up to permutations. Henceforth we shall consider only the unique spectral decomposition in which the diagonal elements of D are ordered in increasing order of magnitude, $d_{11} \geqslant d_{22} \ldots \geqslant d_{nn}$. Consider the following partition for the spectral decomposition,

$$K = \left[U_m \, U_{(n-m)}\right] \begin{bmatrix} D_{mm} & 0 \\ 0 & D_{(n-m)(n-m)} \end{bmatrix} \left[U_m \, U_{(n-m)}\right]^T,$$

where D_{mm} is the diagonal matrix containing the m largest eigenvalues of K and U_m is the $n \times m$ matrix of corresponding eigenvectors. Then it follows from the Eckart-Young theorem (Stewart, 1993) that the best rank m approximation to K is given by $K_m = U_m D_{mm} U_m^T$, in terms of both $\|\cdot\|_2$ and $\|\cdot\|_F$. In fact it can be shown that $\|K - K_m\|_F^2 = \sum_{i=m+1}^{n} d_{ii}^2$.

Recall that the crux of our random projection scheme was replacing the covariance matrix K by $(\Phi K)^T(\Phi K \Phi^T)^{-1}(\Phi K)$, where Φ is our random projection matrix. Now if we choose $\Phi = U_m^T$, then,

$$\begin{aligned}(\Phi K)^T(\Phi K \Phi^T)^{-1}(\Phi K) &= (U_m^T K)^T (U_m^T K U_m)^{-1}(U_m^T K) \\ &= \{(D_{mm}0)U^T\}^T (D_{mm})^{-1}\{(D_{mm}0)U^T\} = K_m,\end{aligned}$$

where 0 above is an $m \times (n-m)$ matrix of zeroes. Therefore the best approximation in our scheme is obtained when we have the first m eigenvectors of the SVD forming our random projection matrix.

The problem however is that obtaining the spectral decomposition is as burdensome as computing the matrix inverse, with $O(n^3)$ computations involved. Recent articles in machine learning in the field of matrix approximation and matrix completion have devised random approximation schemes which give near optimal performance with lesser computational cost (Halko et al., 2011; Sarlos, 2006). We can consider these stochastic schemes to address either (i) Given a fixed rank m, what is the near optimal projection for that rank and what is the corresponding error; or (ii) Given a fixed accuracy level $1-\epsilon$, what is the near optimal rank for which we can achieve this and the corresponding projection. We consider each of these questions below.

We first address the fixed rank problem. For any matrix K of order $n \times n$ and a random vector ω of order $n \times 1$, $K\omega$ is a vector in the range of K. For an $n \times r$ random matrix Ω with independent entries from some continuous distribution, $K\Omega$ gives r independent vectors in the range of K with probability 1. There can be at most n such independent vectors, since the dimension of the range $= n$. As we mentioned earlier, when we evaluate the Gaussian process at a fine grid of points, the covariance matrix K is often severely rank deficient and we should be able to accurately capture its range with $m << n$ vectors.

The next question is how to choose the random matrix Ω. The product $K\Omega$ embeds the matrix K from a $\mathcal{R}^{n \times n}$ space into a $\mathcal{R}^{n \times r}$ space. Embeddings with low distortion properties have been well studied and Johnson-Lindenstrauss transforms (Johnson et al., 1986; Dasgupta and Gupta, 2003) are among the most popular low dimensional projections. A matrix Ω of order $n \times r$ is said to be a Johnson-Lindenstrauss transform for a subspace V of $\mathcal{R}^n$ if $| \, \|v\Omega\| - \|v\| \, |$ is small for all $v \in V$ with high

probability. For the precise definition of the transform, we refer the readers to Definition 1, Sarlos (2006). Initially it was shown that Ω with $(i,j)^{th}$ element $= (\frac{1}{\sqrt{r}}\omega_{ij})$, where ω_{ij} independent $\sim \mathbb{N}(0,1)$ would have Johnson-Lindenstrauss property. Later it has been shown that ω_{ij}'s may be considered to be independent Rademacher or coming from a uniform distribution from the corresponding hypersphere (Achlioptas, 2003; Arriaga and Vempala, 2006). The compressive sensing literature has dealt with these choices in some detail and has found no substantial gain in accuracy in signal compression in using one kind over the other (Candès et al., 2006; Donoho, 2006) - our experiments in the present context concur.

Having formed $K\Omega$ the concluding step in our matrix approximation scheme is to find Φ. We first perform a low distortion low dimensional Johnson-Lindenstrauss embedding for the covariance matrix and perform the rank m projection for this embedding to come up with Φ. It is easy to then calculate the approximate spectral decomposition of the covariance based on the Nyström approximation for the random projection. The exact steps are shown below in Algorithm 1 which combines ideas from Sarlos (2006) and algorithm 5.5 in Halko et al. (2011).

Algorithm 1: Approximate spectral decomposition via Nyström method for target rank m

Given a positive definite matrix K of order $n \times n$ and a randomly generated Johnson-Lindenstrauss matrix Ω of order $r \times n$, we find the projection matrix Φ of order $m \times n$ which approximates the range and compute the approximate SVD decomposition via Nyström approximation with Φ. The steps are enumerated below:

(1) Form the matrix product $K\Omega$.

(2) Compute $\Phi^T =$ left factor of the rank m spectral projection of the small matrix $K\Omega$.

(3) Form $K_1 = \Phi K \Phi^T$.

(4) Perform a Choleski factorization of $K_1 = BB^T$.

(5) Calculate the Nyström factor $C = K\Phi^T(B^T)^{-1}$.

(6) Compute a spectral decomposition for $C = UDV^T$.

(7) Calculate the approximate spectral decomposition for $K \approx K_{tr} = UD^2U^T$.

We give the following result for the approximation accuracy of Algorithm 1, which is a modification of theorem 14 in Sarlos (2006).

Theorem 1. *Consider any* $0 < \epsilon \leqslant 1$ *and* $r = \lfloor \frac{m}{\epsilon} \rfloor$. *Obtain* K_{tr} *from Algorithm* 1 *for the positive definite matrix* K *and let* K_m *be the best rank* m *approximation for* K *if terms of* $\|\cdot\|_F$. *Then,*

$$pr\{\|K - K_{tr}\| \leqslant (1+\epsilon)\|K - K_m\|_F\} \geqslant \frac{1}{2}$$

Proof. By construction,

$$\begin{aligned} K_{tr} &= UD^2U^T = UDV^TVDU^T = CC^T \\ &= K\Phi^T(B^T)^{-1}B^{-1}\Phi K = K\Phi^T(BB^T)^{-1}\Phi K \\ &= K\Phi^T K_1^{-1}\Phi K = (\Phi K)^T(\Phi K\Phi^T)^{-1}\Phi K \end{aligned}$$

This shows that the reduced SVD form, K_{tr} produced by Algorithm 1 is indeed equal to the random projection approximation, which is equal to a generalized projection matrix as explained below.

The generalized rank m projection matrix for the projection whose range is spanned by the columns of an $n \times m$ matrix A, with $m \leqslant n$ and whose nullity is the orthogonal complement of the range of $n \times m$ matrix B, is given by $A(B^TA)^{-1}B^T$. This is a generalization of the standard projection matrix formula (Doković, 1991). Therefore, $K_{tr} = PK$, where $P = K\Phi^T\{\Phi(K\Phi^T)\}^{-1}\Phi$ is the generalized projection matrix with range spanned by the columns of $K\Phi^T$ and whose nullity is the orthogonal complement of the range of Φ^T. Again, by construction, range of Φ^T = range of

$K\Omega$ and therefore, range of $K\Phi^T$ = range of $K^2\Omega$ = range of $K\Omega$. Finally since range of $K\Omega$ = row-space of $\Omega^T K$, the result follows by a direct application of theorem 14 in Sarlos (2006). □

With the advances in parallel computing technology and current stress on GPU computing, we may implement a parallel version of Algorithm 1 by running steps 1 & 2 in parallel for several copies of the matrix Ω; with $log(\frac{1}{\eta})$ copies, we can sharpen the probability in theorem 1 to $1-\eta$. In our implementations of algorithm 1 we use $r = m$. The algorithm involves decomposition of the small matrix $\Phi K \Phi^T$ which involves $O(m^3)$ operations. The matrix multiplications involved, for example in computing K_1 are $O(n^2 m)$, which is the additional cost we pay to have the random projection generalization of the algorithms in §2 · 1. Matrix multiplication can be done in parallel, indeed it is the default approach in standard linear algebra packages such as BLAS3 used in Matlab versions 8 and above, and the constants associated with the order of complexity for matrix multiplication is lower than that for inversion. Our results section indicate that added computational complexity in terms of real CPU time is indeed negligible for the random projection algorithm versus techniques in §2 · 1. In fact with the target error algorithm below, we often achieve lower times than predictive process type approaches of §2 · 1, since the rank required to achieve the target error is substantially smaller.

We now answer the fixed accuracy level question. The eigenvector matrix U from the SVD captures the column space/range of the matrix K, in the sense that $K = UU^T K$. In general we consider the error in range approximation $\|K - \Phi^T \Phi K\|_\eta$ (η = 2 or F), as it makes it easier to evaluate the target accuracy. Using simple linear algebra, $U_m U_m^T K = K_m$, so that the best rank m range approximator is the same as the rank m SVD approximation. It suffices to then search for good range approximators, since lemma 4 in Drineas and Mahoney (2005) and discussion in §5.4,

Halko et al. (2011) show that the error with the Nyström approximator is at least as small as the error in range approximation, and empirically is often substantially smaller. We need only find the projection matrix Φ for the range approximation given the target error level and computation of the approximate spectral decomposition using this Φ proceeds as in steps $3-7$ of Algorithm 1. Φ can be obtained to satisfy any target error level by trivial modification of steps from algorithm 4.2 in Halko et al. (2011) in place of steps 1 & 2 in Algorithm 1, summarized below in Algorithm 2.

Finding range satisfying target error condition

Given a positive definite matrix K of order $n \times n$ and target error $\epsilon > 0$, we find the projection matrix Φ of order $m \times n$ which gives $\|K - \Phi^T \Phi K\| < \epsilon$ with probability $1 - \frac{n}{10^r}$. The steps are enumerated below:

(1) Initialize $j = 0$ and $\Phi = []$, the $0 \times n$ empty matrix.

(2) Draw r random vectors $\omega^{(1)}, \ldots, \omega^{(r)}$ each of order $n \times 1$ with independent entries from $\mathbb{N}(0, 1)$.

(3) Compute $\kappa^{(i)} = K\omega^{(i)}$ for $i = 1, \ldots, r$.

(4) Is $\max_{i=1,\ldots,r}(\|\kappa^{(j+i)}\|) < \frac{\epsilon\sqrt{\pi}}{10\sqrt{2}}$? If yes, step 11. If no, step 5.

(5) Recompute $j = j + 1$, $\kappa^{(j)} = [I - \{\Phi^{(j-1)}\}^T \Phi^{(j-1)}]\kappa^{(j)}$ and $\phi^{(j)} = \frac{\kappa^{(j)}}{\|\kappa^{(j)}\|}$.

(6) Set $\Phi^{(j)} = \begin{bmatrix} \Phi^{(j-1)} \\ \{\phi^{(j)}\}^T \end{bmatrix}$.

(7) Draw a $n \times 1$ random vector ω^{j+r} with independent $\mathbb{N}(0, 1)$ entries.

(8) Compute $\kappa^{(j+r)} = [I - \{\Phi^{(j)}\}^T \Phi^{(j)}]K\omega^{(j+r)}$.

(9) Recompute $\kappa^{(i)} = \kappa^{(i)} - \phi^{(j)} \left\langle \phi^{(j)}, \kappa^{(i)} \right\rangle$ for $i = (j + 1), \ldots, (j + r - 1)$.

(10) Back to target error check in step 4.

(11) Output $\Phi = \Phi^{(j)}$.

Step 9 above is not essential, it ensures better stability when κ vectors become

very small. In our implementations of algorithm 2 we use an r such that $\frac{n}{10^r} = 0.1$ to maintain probability of 0.9 of achieving the error level. The computational requirements of Algorithm 2 are similar to that of 1, for more details we refer the reader to §4·4 in Halko et al. (2011). Posterior fit and prediction in Gaussian process regression usually involves integrating out the Gaussian process, as indicated in §4. We end this subsection with another result which shows that target error in prior covariance matrix approximation governs the error in the marginal distribution of the data, integrating out the Gaussian process.

Theorem 2. *Let $Y = (y_1, y_2, \ldots, y_n)^T$ be the observed data points and let $\pi_{full} = \int \pi\{Y, f(X)\} dP\{f(X)\}$, $\pi_{RP} = \int \pi\{Y, g_{RP}(X)\} dP\{g_{RP}(X)\}$ their corresponding marginal distributions. If $\|K_{f,f} - Q^{RP}_{f,f}\|_F \leqslant \epsilon$, which is the error in approximation of the covariance matrix, then the Kullback Leibler divergence between the marginal distributions from the full and approximated Gaussian process,*

$$KL(\pi_{full}, \pi_{RP}) \leqslant \left\{ n + \left(\frac{n}{\sigma}\right)^2 \right\} \epsilon$$

Proof. The Kullback Leibler divergence between two $n-$variate normal distributions $\mathcal{N}_0 = \mathbb{N}(\mu_0, \Sigma_0)$ & $\mathcal{N}_1 = \mathbb{N}(\mu_1, \Sigma_1)$ is given by,

$$\mathrm{KL}(\mathcal{N}_0 \| \mathcal{N}_1) = \frac{1}{2} \left[\mathrm{tr}\left(\Sigma_1^{-1}\Sigma_0\right) - n - \log\left\{ \det\left(\Sigma_1^{-1}\Sigma_0\right) \right\} + (\mu_1 - \mu_0)^T \Sigma_1^{-1} (\mu_1 - \mu_0) \right]$$

In our case, $\mathcal{N}_0 = \pi_{full} = \mathbb{N}(y; 0, K_{f,f} + \sigma^2 I)$ and $\mathcal{N}_1 = \pi_{RP} = \mathbb{N}(y; 0, Q^{RP}_{f,f} + \sigma^2 I)$. Therefore $KL(\pi_{full}, \pi_{RP}) = \frac{1}{2}\left[\mathrm{tr}\left(\Sigma_1^{-1}\Sigma_0\right) - n - \log\left\{ \det\left(\Sigma_1^{-1}\Sigma_0\right) \right\} \right]$, with $\Sigma_0 = K_{f,f} + \sigma^2 I$ and $\Sigma_1 = Q^{RP}_{f,f} + \sigma^2 I$. We have $\|\Sigma_0 - \Sigma_1\|_F = \|K_{f,f} - Q^{RP}_{f,f}\|_F \leqslant \epsilon$.

Break the expression for the Kullback Leibler divergence into 2 parts with the first part,
$\mathrm{tr}\left(\Sigma_1^{-1}\Sigma_0\right) - n = \mathrm{tr}\left\{\Sigma_1^{-1}(\Sigma_0 - \Sigma_1)\right\} = \sum_{i=1}^n \sum_{j=1}^n s_{ij} d_{ji}$ where s_{ij}, d_{ji} are $(ij)^{th}$ & $(ji)^{th}$

elements of $\Sigma_1^{-1}, (\Sigma_0 - \Sigma_1)$ respectively. Then,

$$\operatorname{tr}\left(\Sigma_1^{-1}\Sigma_0\right) - n \leqslant \|\Sigma_1^{-1}\|_{max} \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ji} \leqslant \|\Sigma_1^{-1}\|_{max}\, n^2\epsilon \tag{2.4}$$

In the inequality above we use $\|\Sigma_1^{-1}\|_{max} = \max_{ij} s_{ij}$ and the fact that $\|\Sigma_0 - \Sigma_1\|_F \leqslant \epsilon \implies \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ji} \leqslant n^2\epsilon$. Now $\|\Sigma_1^{-1}\|_{max} \leqslant \|\Sigma_1^{-1}\|_2$. Since Σ_1^{-1} is symmetric postive definite, $\|\Sigma_1^{-1}\|_2$ is the largest eigenvalue of Σ_1^{-1} which is equal to the inverse of the smallest eigenvalue of Σ_1. Recall that $\Sigma_1 = Q_{f,f}^{RP} + \sigma^2 I$ and $Q_{f,f}^{RP}$ is positive semi-definite and has non negative eigenvalues. Therefore all eigenvalues of $\Sigma_1 \geqslant \sigma^2$, and using this in conjunction with inequality (2.4), we have,

$$\operatorname{tr}\left(\Sigma_1^{-1}\Sigma_0\right) - N \leqslant \left(\frac{n}{\sigma}\right)^2 \epsilon \tag{2.5}$$

It remains to bound the second part of the divergence expression. We have $\det\left(\Sigma_1^{-1}\Sigma_0\right) = \left(\prod_{i=1}^{n}\lambda_i^0\right) / \left(\prod_{i=1}^{n}\lambda_i^1\right)$, where λ_i^0, λ_i^1 are eigenvalues of $\Sigma_0 \,\&\, \Sigma_1$ respectively. Since Σ_0, Σ_1 are symmetric, by the Hoffman-Weilandt inequality (Bhatia, 1997), there exists a permutation p such that $\sum_{i=1}^{n}\left\{\lambda_{p(i)}^0 - \lambda_i^1\right\}^2 \leqslant \|\Sigma_0 - \Sigma_1\|_F^2 \leqslant \epsilon^2$. Therefore with the same permutation p, we have for each i, $\left\{\lambda_{p(i)}^0/\lambda_i^1\right\} \in [1-\epsilon, 1+\epsilon]$. Trivial manipulation then yields, $\log\left\{\det\left(\Sigma_1^{-1}\Sigma_0\right)\right\} \in [n\log(1-\epsilon), n\log(1+\epsilon)]$, so that,

$$-\log\left\{\det\left(\Sigma_1^{-1}\Sigma_0\right)\right\} \leqslant n\epsilon \tag{2.6}$$

Combining inequalities (2.5) and (2.6), we have,

$$\mathrm{KL}(\pi_{full}, \pi_{RP}) \leqslant \left\{n + \left(\frac{n}{\sigma}\right)^2\right\}\epsilon$$

which completes the proof. □

This is not an optimal bound, but serves our basic goal of showing that the Kullback Leibler divergence is of the same order as the error in estimation of the covariance matrix in terms of Frobenius norm. Additional assumptions on the eigenspace of the covariance matrix would yield tighter bounds.

2.4.2 Conditioning numbers and examples

The full covariance matrix for a smooth Gaussian process tracked at a dense set of locations will be ill-conditioned and nearly rank-deficient in practice, with propagation of rounding off errors due to finite precision arithmetic, the inverses may be highly unstable and severely degrade the quality of the inference. To see this, consider the simple example with covariance function $k(x, y) = e^{-0.5(x-y)^2}$, evaluated at the points $0.1, 0.2$, which gives the covariance matrix,

$$K = \begin{pmatrix} 1.000 & 0.995 \\ 0.995 & 1.000 \end{pmatrix},$$

which yields the inverse,

$$K^{-1} = \begin{pmatrix} 100.5008 & -99.996 \\ -99.996 & 100.5008 \end{pmatrix}.$$

Perturbing the covariance kernel slightly to $k(x, y) = e^{-0.75(x-y)^2}$, yields a very similar covariance matrix,

$$K_{new} = \begin{pmatrix} 1.0000 & 0.9925 \\ 0.9925 & 1.0000 \end{pmatrix},$$

with $\|K - K_{new}\|_F = 0.0035$. However the inverse of the covariance matrix drastically changes to

$$K_{new}^{-1} = \begin{pmatrix} 67.1679 & -66.6660 \\ -66.6660 & 67.1679 \end{pmatrix},$$

with $\|K^{-1} - K_{new}^{-1}\|_F = 66.6665$. With such a small change in the magnitude of some elements, we have a huge change in its inverse, which would lead to widely different

estimates and predicted values. The problem is obviously much aggravated in large data sets and in Bayesian settings where there the posterior is explored through several rounds of iterations, say in an Gibbs sampling scheme. How well a covariance matrix K is conditioned may be measured by the conditioning number, $\frac{\sigma_l}{\sigma_s}$, where σ_l, σ_s are its largest and smallest eigenvalues respectively (Dixon, 1983). Condition numbers are best when they are close to 1, very large ones indicate numerical instability - in the example above, the condition number of the matrix K is ≈ 400. Condition number arguments imply that low rank approximations may not only be necessitated by computational considerations but may indeed be desirable for better inference over the full covariance matrix. It therefore makes practical sense to choose amongst two low rank approximations of comparable rank or accuracy, the one that is better conditioned. We now show empirically how condition number is improved greatly with the random projection approximation over the knot based schemes, when considering either a fixed rank or target error approach.

We first evaluate with respect to the fixed rank question. Consider a similar covariance kernel as above $k(x, y) = e^{-(x-y)^2}$, and evaluate it over a uniform grid of 1000 points in $[0.1, 100]$, and consider the resulting 1000×1000 covariance matrix K. The condition number of $K \approx 1.0652 \times 10^{20}$, which indicates it is severely ill-conditioned. We now apply Algorithm 1, with $r = m$, for different choices of the target rank m and calculate the error in terms of the Frobenius and spectral norms, conditioning numbers and the time required. For each choice of m, we also consider the approximation as would given by the approaches of §2·1 in two ways, (1) randomly selecting m grid points out of the 1000, call this PP1; and (2) selecting the grid points by the partial Choleski factorization with pivoting, as in Tokdar (2011b), call this PP2, which can be interpreted as a systematic implementation of the suggested approach in Finley et al. (2009). Results are summarized in table 2.1 for some values of m. The random projection approach clearly has better approximation accuracy

than the other methods - this becomes more marked with increase in dimension of the approximation. The condition numbers for the random projection scheme are dramatically better than the other 2 approaches, indicating superior numerical stability and reliable estimates.

Next we compare with respect to achievement of a target error level. For the random projection approach, we implement Algorithm 2. For this comparison it would be useful to know the best possible rank at which the target error would be achieved if we knew the real spectral decomposition. For this purpose we consider matrices of the form $K = EDE^T$, where E is an orthonormal matrix and D is diagonal. The diagonal elements of D, which are the eigenvalues of K are chosen to decay at exponential rates, which holds for smooth covariance kernels (Frauenfelder et al., 2005), with i^{th} element $d_{ii} = e^{-i\lambda}$; for the simulations tabulated, we use $\lambda = 0.5, 0.08, 0.04$ respectively. E is filled with independent standard normal entries and then orthonormalized. Algorithm 2 for random projections, PP1 and PP2 as above are applied to achieve different Frobenius norm error levels ϵ for different values of matrix order n. Results are shown in table 2.2. Clearly random projection achieves the desired target error level with lower ranks for all different values of ϵ and n; also real CPU times required are comparable, in fact the random projection approach has lower time requirements when the rank differences become significant.

Lower target ranks, besides the obvious advantages of computational efficiency and stability, imply lesser memory requirements, which is an important consideration when sample size n becomes very large. Time required for matrix norm calculations for checking target error condition for PP1 or PP2 are not counted in the times shown. All times here as well as in following sections, are in seconds and calculated when running the algorithms in Matlab 7.10 version R2010a on a 64bit CentOS 5.5 Linux machine with a 3.33 Ghz dual core processor with 8Gb of random access memory. The random projection benefits from the default parallel implementation

of matrix multiplication in Matlab. Lower level implementations of the algorithms, for example C/C++ implementations would require parallel matrix multiplication implementation to achieve similar times. With a GPU implementation with parallel matrix multiplication, random projection approximation can be significantly speeded up.

2.5 Parameter Estimation And Illustrations

2.5.1 Bayesian inference for the parameters

An important part of implementing Gaussian process regression is estimation of the unknown parameters of the covariance kernel of the process. Typically the covariance kernel is governed by 2 parameters, characterizing its range and scale. We shall consider the squared exponential kernel used earlier, $k(x,y) = \frac{1}{\theta_2} e^{-\theta_1 \|x-y\|^2}$ for simplicity, but the techniques herein shall be more generally applicable. θ_1 and θ_2 are the range and inverse scale parameters respectively. We shall use Bayesian techniques for inference here to fully explore the posterior over all possible values of these parameters, also applying the random projection scheme for repeated iterations of Markov chain samplers will allow us to fully demonstrate its power.

For Bayesian inference, we have to specify prior distributions for each of the unknown parameters, namely θ_1, θ_2 and σ^2, the variance of the idiosyncratic noise in equation (2.1). In place of (2.1), using the random projection, we have,

$$y_i = g_{RM}(x_i) + \epsilon_i, i = 1, \ldots, n. \tag{2.7}$$

Using the bias corrected form for the random projection approximation the prior for the unknown function is, $[g_{RM}(X)|\theta_1, \theta_2] \sim \mathbb{N}(0, Q^{RM}_{f,f})$, where $Q^{RM}_{f,f} = Q^{RP}_{f,f} + D_M$, with D_M the diagonal matrix as obtained for variance augmentation from equation (2.3). Letting $\tau = \sigma^{-2}$ and choosing conjugate priors, we let $\tau \sim \text{Ga}(a_1, b_1)$, $\theta_2 \sim \text{Ga}(a_2, b_2)$ and $\theta_1 \sim \sum_{h=1}^{t} (1/t)\delta_{c_t}$, denoting a discrete uniform distribution with atoms

$\{c_1, \ldots, c_t\}$. The $\mathrm{Ga}(a, b)$ gamma density is parametrized to have mean a/b and variance a/b^2. The priors being conditionally conjugate, we can easily derive the full conditional distributions necessary to implement a Gibbs sampling scheme for the quantities of interest as follows,

$$
\begin{aligned}
[g_{RM}(X)|-] &\sim \mathbb{N}[\{(Q_{f,f}^{RM})^{-1} + \tau I\}^{-1} Y, \{(Q_{f,f}^{RM})^{-1} + \tau I\}^{-1}] \\
[\tau|-] &\sim \mathrm{Ga}[a_1 + \frac{n}{2}, b_1 + \{Y - g_{RM}(X)\}^T \{Y - g_{RM}(X)\}] \\
[\theta_2|-] &\sim \mathrm{Ga}(b_2 + f^T Q^{-1} f) \\
\mathrm{pr}(\theta_1 = c_i|-) &= c|\mathrm{det} Q_{f,f}^{RM}|^{-\frac{1}{2}} e^{-\frac{1}{2} g_{RM}(X)^T (Q_{f,f}^{RM})^{-1} g_{RM}(X)}
\end{aligned}
$$

where $Q = \theta_2 Q_{f,f}^{RM}$ and c is a constant such that $\sum_{i=1}^{t} \mathrm{Prob}(\theta_1 = c_i|-) = 1$. We can integrate out the Gaussian process $g_{RM}(X)$ from the model to obtain $Y \sim \mathbb{N}(0, \{Q_{f,f}^{RM} + \tau^{-1} I\})$ - this form is useful for prediction and fitting. We show some relevant computational details for the matrix inversion using the Woodbury matrix identity in the appendix.

For computational efficiency, we pre-compute the random projection matrix for each of the discrete grid points for θ_1 and the corresponding matrix inverse required for the other simulations. Changes in the parameter θ_2 do not affect the eigendirections, hence we do not recompute the projection matrix Φ and we can compute the new inverse matrix due to a change in θ_2 by just multiplying with the appropriate scalar. Although other prior specifications are extensively discussed in the literature, we have considered simple cases to illustrate the efficacy of our technique. It is observed that inference for the range parameter θ_1 is difficult and Markov chain Monte Carlo schemes tend to have slow mixing due to high correlation between the imputed functional values and the parameter. The random projection approximation appears to take care of this issue in the examples considered here.

2.5.2 Illustrations

We first consider a simulated data example where we generate data from functions corresponding to a mixture of Gaussian kernels in $[0, 1]$. We consider functions with 3 different degrees of smoothness - an almost flat one, a moderately wavy one and a highly wavy one. For each of these functions, we consider $10,000$ equi-spaced points in $[0, 1]$ and we add random Gaussian noise to each point - this constitutes our observed data set Y. We randomly select $9,000$ points for model fitting and the rest for validation. We now implement random projection with Algorithm 2 with a couple of different target error levels $(0.1, 0.01)$ referred to as RP. We compare it with predictive process with equispaced selection of knots and with the modified version of knot selection by pivoted Choleski factorization (Tokdar, 2011b) explained in §3.2, referred to as PP1 and PP2 respectively. In this simulated example, as well as in the real data examples we use the squared exponential covariance kernel with prior specifications as in the previous section. For the idiosyncratic noise, we use hyperparameters a_1, b_1 such that the mean is approximately equal to estimated noise precision with ordinary least square regression. In particular for the smooth one we use $a_1 = 1, b_1 = 10$. Hyperparameter choices for covariance kernel parameters are guided by some trial runs, we use a grid of 2000 equispaced points in $[0, 2]$ for θ_1 and $a_2 = 2, b_2 = 20$ for θ_2. We run Gibbs samplers for $10,000$ iterations with the first 500 discarded for burn-in. We calculate the predicted values for the held-out set with the posterior means of the parameters from the Gibbs iterations and we also calculate the average rank required to achieve the target accuracy over the iterations. Effective sample size is calculated by using the output for the Markov chains with the CODA package in R. The results are tabulated in table 2.3, whereby random projection has substantial gain in predictive accuracy and in the target rank required, as well as substantially better effective sample sizes for the unknown parameters of the

covariance kernel as well as for the predicted points. With the predictive process type approaches, we would need substantially more Markov chain Monte Carlo iterations to achieve similar effective sample sizes, leading to an increased computational cost.

We finally consider a couple of real data examples, which have been used earlier for reduced rank approaches in Gaussian process regression, of contrasting sizes. The first is the abalone dataset, from the UCI machine learning database (Frank and Asuncion, 2010), where the interest is in modeling the age of abalone, given other attributes, which are thought to be non-linearly related to age. The dataset consists of 4000 training and 177 test cases. We use Euclidean distance between the attributes for our covariance function for the Gaussian process and for the gender attribute, (male/female/infant) is mapped to $\{(1,0,0),(0,1,0),(0,0,1)\}$. The other example we consider is the Sarcos Robot arm, where we are interested in the torque as given by the 22^{nd} column given the other measurements in the remaining 21 columns. This dataset has $44,484$ training and $4,449$ test cases. We once again consider Euclidean distances between the attributes. For each of the experiments, we use Algorithm 2 with target error level 0.01. The hyperparameters for each example is chosen in similar fashion as the simulated example. This leads to choosing $a_1 = 1, b_1 = 0.1$ for the abalone dataset and $a_1 = 2, b_1 = 0.1$ for the Sarcos Robot arm. The grid for θ_1 in either case is 2000 equispaced points in $[0,2]$; for θ_2, in abalone we have $a_2 = 1, b_2 = 1$ while we have $a_2 = 1, b_2 = 0.75$ for Sarcos Robot arm. The Gibbs sampler for the abalone data set is run for $10,000$ iterations with $1,000$ discarded for burn-in, while for the Sarcos Robot arm, it is run for $2,000$ iterations with 500 discarded for burn-in.

The results for both these experiments, tabulated in table 2.4. There is improvement in predictive accuracy when using random projections in both the examples, in particular for the Robot arm dataset predictive accuracy is significantly better. This improvement perhaps is a consequence of the fact that we get better estima-

tion for the parameters when using the random projection approach. In the Sarcos Robot arm data, both the covariance kernel parameters are readily observed to have different posteriors with the random projection approach. This is a consequence of the poor behavior of the Markov chains for these parameters, they exhibit poor mixing. In fact this problem of poor mixing when approximating a stochastic process by imputed points is not unique to Gaussian processes, it have been observed in other contexts too, possibly due to the chains for the imputed points and the unknown parameters being highly correlated with each other (Golightly and Wilkinson, 2006). The random projection approach appears to improve this to a great extent by not considering specific imputed points. The inference is not very sensitive to the choice of hyperparameters, with datasets of this size we are able to overcome prior influence if any. In particular in trial runs with smaller number of iterations, changing the grid for θ_1 to 1000 uniformly spaced points in $[0, 1]$ yielded almost similar results, random projection performing better than the knot based approaches.

2.6 Concluding Remarks

We have developed a broad framework for reduced rank approximations under which almost the entire gamut of existing approximations can be brought in. We have tried to stochastically find the best solution under this broad framework, thereby leading to gains in performance and stability over the existing approaches. Another important contribution has been to connect not only the machine learning and statistical approaches for Gaussian process approximation, but also to relate them to matrix approximations themes - we have shown that the reduced rank Gaussian process schemes are effectively different flavors of approximating the covariance matrix arising therein. The random projection approach has been mainly studied as an approximation scheme in this article, it is also worthwhile considering it from a model based perspective and investigate the added flexibility it offers as an alternative model.

We have not explored the performance of parallel computing techniques in this context, though we have indicated how to go about parallel versions of the algorithms at hand. Further blocking techniques and parallelization remains an area of future interest. We also plan on working out the multivariate version of random projection approximations. In ongoing work, we explore similar approaches in other different contexts - a couple of examples being in the context of functional modeling, where the domain may be discrete and also in the case of parameter estimation for diffusion processes - where similar dimensionality problems are faced sometimes in terms of their discrete Euler approximations. In other ongoing work, we also explore the theoretical rates of convergence of the truncated expressions for different classes of covariance kernels and convergence of the associated posterior distributions of the unknown parameters.

Table 2.1: Comparative performance of the approximations in terms of matrix error norms, with the random projection approach based on Algorithm 1.

For $m = 10$	$\|\|\|_F$	$\|\|\|_2$	Cond No	Time
RP	106.1377	17.6578	1.0556	0.06
PP1	107.6423	17.6776	1.2356	0.04
PP2	106.6644	17.6778	1.2619	0.04

For $m = 25$	$\|\|\|_F$	$\|\|\|_2$	Cond No	Time
RP	82.1550	17.2420	1.7902	0.22
PP1	91.1016	17.5460	230.236	0.18
PP2	85.6616	17.3800	13.8971	0.21

For $m = 50$	$\|\|\|_F$	$\|\|\|_2$	Cond No	Time
RP	50.5356	14.2998	2.9338	0.27
PP1	79.1030	17.0172	2803.5	0.24
PP2	69.5681	15.6815	876.23	0.25

For $m = 100$	$\|\|\|_F$	$\|\|\|_2$	Cond No	Time
RP (Algo1)	6.6119	2.8383	20.6504	0.40
PP1	39.9642	13.1961	1.3815×10^6	0.31
PP2	10.1639	6.3082	1792.1	0.36

Table 2.2: Comparison of the ranks required to achieve specific target errors by the different algorithms, with random projection based on Algorithm 2. We show the best possible low rank m from the Eckart-Young theorem, with the given error level.

		PP1	PP2	RP
$n = 10^2, \epsilon = 0.1, m = 5$	Reqd Rank	17	9	7
	Cond No	298.10	54.59	20.08
	Time	0.03	0.04	0.07
$n = 10^3, \epsilon = 0.01, m = 69$	Reqd Rank	213	97	78
	Cond No	2.30×10^7	2164.6	473.43
	Time	12.1	11.5	36.2
$n = 10^4, \epsilon = 0.01, m = 137$	Reqd Rank	1757	793	174
	Cond No	3.19×10^{19}	2.30×10^9	1012.3
	Time	335	286	214

Table 2.3: Simulated data sets with the target error algorithm for the three different simulations. Different algorithms compared in terms of preditive MSE and various posterior summaries for the unknown parameters. ESS, PV stand for effective sample size, predicted values respectively. The 3 sets, s,w,vw correspond to the cases smooth, wavy and very wavy respectively.

		PP1	PP2	RP
$\epsilon = 0.1$,s	MSPE	11.985	8.447	3.643
	Avg Reqd Rank	1715.6	453.8	117.2
	95% Interval Reqd Rank	[1331,2542]	[377,525]	[97,141]
	Posterior Mean, θ_1	0.09	0.10	0.06
	95% Credible Interval, θ_1	[0.05,0.14]	[0.05,0.15]	[0.04,0.08]
	ESS, θ_1	496	870	1949
	Posterior Mean, θ_2	0.91	1.15	1.25
	95% Credible Interval, θ_2	[0.58,1.58]	[0.85,1.43]	[1.09,1.46]
	ESS, θ_2	2941	3922	4518
	Avg ESS,PV	2190	3131	5377
	Time	39761	29355	32365
$\epsilon = 0.01$,w	MSPE	10.114	6.891	2.265
	Avg Reqd Rank	3927.1	941.5	129.7
	95% Interval Reqd Rank	[2351,5739]	[868,1165]	[103,159]
	Posterior Mean, θ_1	0.07	0.08	0.13
	95% Credible Interval, θ_1	[0.01,0.14]	[0.02,0.15]	[0.09,0.17]
	ESS, θ_1	574	631	1918
	Posterior Mean, θ_2	0.83	0.85	0.79
	95% Credible Interval, θ_2	[0.21,1.74]	[0.40.1.63]	[0.45,1.29]
	ESS, θ_2	3679	4819	5002
	Avg ESS,PV	2875	3781	5769
	Time	78812	47642	33799
$\epsilon = 0.01$,vw	MSPE	17.41	13.82	6.93
	Avg Reqd Rank	4758.5	1412.5	404.5
	95% Interval Reqd Rank	[2871,6781]	[1247,1672]	[312,475]
	Posterior Mean, θ_1	0.11	0.09	0.05
	95% Credible Interval, θ_1	[0.04,0.17]	[0.05,0.13]	[0.03,0.08]
	ESS, θ_1	741	747	1049
	Posterior Mean, θ_2	1.27	1.18	1.19
	95% Credible Interval, θ_2	[1.08,1.43]	[1.12,1.41]	[1.15,1.34]
	ESS, θ_2	1521	2410	2651
	Avg ESS, PV	1263	1415	2422
	Time	89715	57812	47261

Table 2.4: Comparison of the different algorithms based on their performance in the experimental data sets in terms of preditive MSE and various posterior summaries for the unknown parameters. ESS stands for effective sample size. AB and SRA denote the 2 experiments wrt to the Abalone data and the Sarcos Robot Arm data respectively.

		PP1	PP2	RP
AB	MSPE	1.785	1.517	1.182
	Avg Reqd Rank	417.6	328.8	57.2
	95% Interval Reqd Rank	[213,750]	[207,651]	[43,71]
	Posterior Mean, θ_1	0.212	0.187	0.149
	95% Credible Interval, θ_1	[0.112,0.317]	[0.109,0.296]	[0.105,0.207]
	ESS, θ_1	516	715	1543
	Posterior Mean, θ_2	0.981	1.014	1.105
	95% Credible Interval, θ_2	[0.351,1.717]	[0.447,1.863]	[0.638,1.759]
	ESS, θ_2	1352	1427	1599
	Time	19468	21355	15423
SRA	MSPE	0.5168	0.2357	0.0471
	Avg Reqd Rank	4195	2031	376
	95% Interval Reqd Rank	[3301,4985]	[1673,2553]	[309,459]
	Posterior Mean, θ_1	0.496	0.352	0.105
	95% Credible Interval, θ_1	[0.087,0.993]	[0.085,0.761]	[0.042,0.289]
	ESS, θ_1	85	119	147
	Posterior Mean, θ_2	1.411	1.315	1.099
	95% Credible Interval, θ_2	[1.114,1.857]	[1.065,1.701]	[1.002,1.203]
	ESS, θ_2	145	132	227
	Time	57213	53929	20869

3

Parallel inversion of huge covariance matrices

3.1 Summary

An extremely common bottleneck encountered in statistical learning algorithms is inversion of huge covariance matrices, a special case of which was dealt with in chapter 2. We propose general parallel algorithms for inverting positive definite matrices, which are nearly rank deficient. Such matrix inversions are needed in Gaussian process computations, among other settings, and remain a bottleneck even with the increasing literature on low rank approximations. We propose a general class of algorithms for parallelizing computations to dramatically speed up computation time by orders of magnitude exploiting multicore architectures. We implement our algorithm on a cloud computing platform, providing pseudo and actual code. The algorithm can be easily implemented on any multicore parallel computing resource. Some illustrations are provided to give a flavor for the gains and what becomes possible in freeing up this bottleneck.

3.2 Introduction

We consider a symmetric positive definite matrix K of order n where n is very large; on the order of 10,000s to millions or more. Our interest is in evaluation of K^{-1}, which cannot be computed sufficiently quickly using current algorithms. Even if we could compute the inverse, substantial numeric instability and inaccuracies would be expected due to propagation of errors arising from finite machine precision arithmetic (Trefethen and Bau III, 1997).

Typically in statistical applications, one needs to evaluate expressions such as $K^{-1}A$, where A is some matrix of appropriate order. Instead of directly computing K^{-1}, one popular approach is to consider the QR decomposition for $K = QR$ and evaluate the expression, $R^{-1}Q^TA$ (Press et al., 2007). The matrix R is lower triangular and therefore R^{-1} can be evaluated by backward substitution, which requires $O(n^2)$ flops instead of the $O(n^3)$ flops for usual inversion. QR decomposition is known to be relatively more stable and efficient than other standard algorithms (Cox and Higham, 1997), a close competitor being the Cholesky decomposition.

The problem is that QR decomposition for K requires $O(n^3)$ computations, which is of the same order as that of inversion and therefore prohibitively expensive for large n. Even with QR decomposition being relatively stable, for very large n, finite precision numerical errors are large. This is accentuated when the matrix K has a small spectrum, in the sense of fast decaying eigenvalues, as is typically obtained for covariance matrices obtained from smooth covariance kernels and in large least square problems among a host of other areas. However, this small spectrum, while being the bane of full rank algorithms, makes low-rank algorithms highly accurate.

We propose a new class of algorithms combining ideas from recent developments in linear algebra to find approximate low-rank decompositions. These low-rank decompositions provide several orders of magnitude improvement in speed while also

improving accuracy. We propose a general blocking method to enable implementation in parallel on distributed computing architectures. We also consider accuracy of these approximations and provide bounds which are obtained with high probability. Our approach amalgamates ideas from three recent but apparently unrelated developments in numerical linear algebra and machine learning, which we briefly outline below.

There has been an increasing literature on approximating a matrix B of order $n \times n$ by $B\Omega$ where Ω is a matrix with random entries of order $n \times d$ with $d << n$ (Halko et al., 2011). Originally motivated by Johnson Lindenstrauss transforms (Johnson and Lindenstrauss, 1984), these results have been used in a host of application areas, including compressive sensing (Donoho, 2006; Candès et al., 2006) and approximate Gaussian process regression (Banerjee et al., 2013b). Typically Ω is populated by random entries, such as draws from Gaussian or Rademacher distributions. The product $B\Omega$ involves $O(n^2 d)$ flops and can be itself expensive. Recent developments have shown that structured random matrices, such as random Hadamard transforms, may improve the efficiency significantly, bringing down the number of flops required to the order of $n \log d$ (Woolfe et al., 2008). In our case, we show that with certain classes of structured random matrices, Ω of order $n \times d$, we have that $K\Omega$ has approximately the same column space as that of K (in a sense to be made precise later), and that the product $K\Omega$ may be computed efficiently. In addition, such structured random matrices spread the information of the matrix K, so that we obtain bounds on the minimum and maximum eigenvalues of $K\Omega$, implying that approximate decompositions using $K\Omega$ are almost entirely devoid of usual inaccuracies arising from numerical conditioning.

Having efficiently obtained a tall skinny matrix $K\Omega$ from the square positive definite matrix K, we now consider recent developments which show that QR decompositions of such tall skinny matrices may be done efficiently in parallel (Constantine

and Gleich, 2011; Agullo et al., 2010). The key idea is to partition the tall matrix into blocks and efficiently combine QR decomposition of each of these blocks. The main considerations are the choice of the number of blocks and column size d of the projection matrix Ω, which regulate computation time and accuracy. We provide some theoretical pointers while empirically demonstrating gains achieved on modest architectures. Approximate QR decompositions are useful in obtaining other approximate matrix decompositions, such as approximate spectral decompositions or matrix products. Several recent articles have focused on these issues and Halko et al. (2011) provides an excellent review for algorithms which switch between approximate decompositions. For illustration, we focus on the scenario when the principal interest is in evaluating a Gaussian type likelihood and fine tune the algorithms in this context. We finally consider performance versus other competitors.

3.3 Notational Preliminaries

We begin with some notations which we shall use through the rest of the article. In general we will represent matrices in the upper case, A, B etc and row or column vectors in the lower case, a, b etc. We shall use Frobenius and spectral matrix norms: $\|A\|_F = \sqrt{\sum_{i,j} a(i,j)^2}$ and $\|A\|_2 = max\{\|Ax\| : \|x\| = 1\}$ respectively, with the notation $\|A\|_\psi$ to imply the conclusion holds for both norms. We begin with a real-valued positive definite matrix $K \in \mathbb{R}^{n\times n}$ and $k(i,j)$ will denote the (i,j)th element of K. A spectral decomposition of $K = UDU^T$ can be partitioned as,

$$K = \begin{bmatrix} U_m \, U_{(n-m)} \end{bmatrix} \begin{bmatrix} D_m & 0 \\ 0 & D_{(n-m)} \end{bmatrix} \begin{bmatrix} U_m \, U_{(n-m)} \end{bmatrix}^T .$$

Since K is positive definite, the eigenvalues are positive and we assume without loss of generality that the diagonal matrix D of eigenvalues contains them in descending order of magnitude and the eigenvector matrix U is orthonormal. D_m is therefore the

matrix containing the m largest eigenvalues and U_m the corresponding eigenvector matrix. By the Eckart Young theorem (Stewart, 1993), the best rank m approximation K_m to K in terms of minimizing $\|K - K_m\|_\psi$ is given by $K_m = U_m D_m U_m^T$.

We shall focus on the case where the matrix K has fast decaying eigenvalues. Some examples of problems where this happens is in large least squares optimization problems, $\|Ax - b\|$, with $K = A^T A$, or with K being a covariance matrix generated for a Gaussian process at a dense set of locations with a smooth covariance function (Banerjee et al., 2013b; Drineas et al., 2011). For some of our results, we will assume that eigenvalues of K decay at an exponential rate, by which we assume that there exists positive constants λ_1, λ_2 such that $d_{i,i} \leqslant \lambda_1 e^{\lambda_2 i} \forall i$. λ_1, λ_2 shall be referred to as the proportionality and rate constants of exponential decay respectively.

An important consideration in this article will be the numerical stability of the algorithms and decompositions. One way in which the stability of a matrix decomposition can be measured is by how well the matrix in question is conditioned. The condition number of a positive semidefinite matrix A is given by $c(A) = \sigma_l/\sigma_s$, where σ_l, σ_s are the largest and smallest eigenvalues of the matrix respectively. For the positive definite matrix K it would be $c(K) = d(1,1)/d(n,n)$.

3.4 The Main Algorithm

Our fast approximate inversion algorithm uses three different ideas from numerical techniques and randomized linear algebra. The first is to approximate the spectrum of the large matrix K by post-multiplying it with a random matrix Ω, the second is to consider incorporating special structure in the random matrix Ω for faster evaluation of the product and lastly, blocking strategies, to enable computation of decomposition of the product $K\Omega$ on multicore architectures.

3.4.1 Column space approximation

As mentioned, we shall be considering positive definite matrices K with very fast decaying spectrums. In section §4, we provide theoretical justification to show that eigenvalues of positive definite matrices, produced as discrete realizations of a wide class of positive definite kernels, decay very fast. For a positive definite matrix K with fast decaying spectrums, it is reasonable to expect that if Ω is a $n \times r$ matrix with $r \ll n$, populated with independent entries, then the product $K\Omega$ will capture most of the information in K, or approximate the column space of K. Such approximation schemes have been in focus in the rapidly expanding field of *randomized linear algebra* (refer to Halko et al. (2011) for a review). Banerjee et al. (2013b) consider this method in the context of approximations for Gaussian processes for large data sets. In general, it is difficult to measure the amount of information captured in $K\Omega$ from K. One way to do this is to consider the generalized projection of K onto $K\Omega$ and then consider differences. Letting Q be an orthonormal basis for the column space of $K\Omega$, we may consider the error, $\|K - QQ^TK\|_\psi$, where $Q^TQ = I$, by virtue of orthonormality. In general, we may consider an approximation scheme with two objectives in mind.

1. Fix a target rank m and try to minimize the matrix norm error when using Ω having the target rank.

2. Fix a target matrix norm error ϵ and try to achieve that error with high probability with the smallest possible Ω.

We shall consider each of objectives in our projection approximation designs.

The simplest possible choice for the random matrix Ω is a submatrix of columns from the $n \times n$ identity matrix (this submatrix is sometimes called a permutation submatrix, denoted by P). This amounts to choosing r columns at random from the

matrix K. Low rank approximations using a submatrix have been explored in a variety of contexts, including subset of regressors (Quinonero Candela and Rasmussen, 2005) and least squares (Drineas et al., 2011). With the advent of compressive sensing (Donoho, 2006; Candès et al., 2006), it has been shown that instead of restricting Ω to permutation submatrices, more general random matrices often have better performance (Drineas et al., 2011). Common choices of the random matrix Ω, which have been used in the compressive sensing and matrix approximation literature are:

1. elements of Ω are independent and identically distributed Gaussian random variables appropriately scaled,

2. elements of Ω are independent and identically distributed Rademacher variables,

3. elements of Ω are independently sampled from the uniform distribution on the unit sphere.

In general it has been shown that most choices, based on sampling from a centered distribution and then appropriately scaling, work in achieving accurate error bounds with high probability (Candès et al., 2006).

3.4.2 Approximations via structured random matrices

The biggest drawback of the approximation techniques in §3.4.1 is that the matrix product $K\Omega$ may itself be prohibitively expensive $O(n^2 r)$ for very large K. Instead we may form the random matrix Ω such that the product $K\Omega$ can be evaluated faster, exploiting properties of the structured distribution.

Definition 1. *We shall call a random matrix a structured random matrix if it is of the form* $\Omega = cRTP$*, where*

- *c is an appropriate scaling constant, such that the columns of Ω are orthonormal,*
- *R is an $n \times n$ diagonal matrix whose diagonal elements are independent Rademacher entries,*
- *T is an $n \times n$ appropriate orthogonal transform, facilitating the fast multiply,*
- *P is an $n \times r$ permutation submatrix.*

There are different choices of orthogonal transforms we may use, all of which facilitate the fast multiply, examples for T being the discrete Fourier matrix, discrete cosine matrix, discrete Hartley matrix or the Walsh-Hadamard matrix. The Walsh-Hadamard matrix in particular has been in focus in recent literature (Tropp, 2011) and tight bounds on the approximation accuracy have been obtained. The Walsh Hadamard matrix of order $n \times n$ is defined as,

$$H_n = \begin{bmatrix} H_{\frac{n}{2}} & H_{\frac{n}{2}} \\ H_{\frac{n}{2}} & -H_{\frac{n}{2}} \end{bmatrix} \quad \text{with} \quad H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

The Walsh-Hadamard and discrete Fourier transforms have some optimality with respect to preserving matrix coherence (Tropp, 2011; Boutsidis and Gittens, 2012), but a disadvantage is Walsh-Hadamard transforms exist only for positive integers which are powers of 2, while the discrete Fourier transforms may involve complex numbers. The alternatives, Hartley and discrete cosine transforms while being marginally suboptimal do not have these problems (Avron et al., 2010). In the following section, we formalize the theoretical setting and describe probabilistic error bounds for column space approximations using transforms with general orthonormal matrices (therefore valid for any of the aforementioned transforms).

3.4.3 Blocked decompositions for parallelization

Our focus has been on obtaining a low rank rectangular matrix $\hat{K}$ of order $n \times r$, with $r \ll n$, as an approximation to K, such that the column space of $\hat{K}$ is very close to the column space of K. To measure the approximation error, we used the error in column space approximation via projection, measured as, $||K - QQ^T K\|_\psi$, where Q is an orthogonal basis for the column space of $\hat{K}$. Such a Q is typically obtained from the QR decomposition of $\hat{K}$, where Q is an $n \times r$ matrix with orthonormal columns and R is an upper triangular matrix (lower triangular in some conventions). Obtaining a QR decomposition of an $n \times r$ matrix has computational cost $O(nr^2)$. With the matrix multiplication for forming $K\Omega$, the eventual computational cost is $O(n^2 r)$ (for $n \gg r$), which may be too expensive for very large n.

To further reduce computational burden, we may consider parallelizing the computations. Matrix multiplication is trivially parallelizable, while most matrix decompositions are not (Choi et al., 1994). State-of-the-art linear algebra algorithms, for example in ScaLAPACK routines (Blackford et al., 1997), QR, Cholesky or SVD decompositions, involve dense manipulations of the whole matrix, which are not parallelizable or cannot be blocked. In addition to computational cost, storing the full matrix $\hat{K}$ may be problematic in terms of memory requirements for very large n. However, in our case, we are interested in the QR decomposition of a tall matrix, number of rows being much greater than the number of columns, for which blocking strategies have been recently developed, exploiting modern parallel computing platforms like MapReduce. We elaborate on the blocking strategies next.

For notational clarity assume that $n = 8r$ and consider a partition of $\hat{K}$,

$$\hat{K} = \begin{bmatrix} \frac{K_1}{K_2} \\ \hline \frac{K_3}{K_4} \end{bmatrix}$$

In the above each K_i is of size $2r \times r$. Denote the QR factorization of the small matrix $K_i = Q_i R_i$. Along the lines of the $TSQR$ factorization presented in Constantine and Gleich (2011) we consider the following blocked factorization,

$$\begin{bmatrix} K_1 \\ K_2 \\ K_3 \\ K_4 \end{bmatrix} = \begin{bmatrix} Q_1 & & & \\ & Q_2 & & \\ & & Q_3 & \\ & & & Q_4 \end{bmatrix} \begin{bmatrix} R_1 \\ R_2 \\ R_3 \\ R_4 \end{bmatrix} = Q_1^b R_1^b \tag{3.1}$$

Each of the small matrices Q_i being orthogonal, the left factor of the decomposition in (3.1), Q_1^b, is formed by stacking the small matrices Q_i as diagonal blocks, in an orthogonal matrix. It is also an orthogonal basis for the column space of $\hat{K}$ and in many application areas it suffices to work with Q_1^b. In some other applications, we may require evaluation of the product $Q_1^b x$, where x is some vector of order $r \times 1$, which can be evaluated in a similar blocked fashion, utilizing the small Q_i's. The matrix R_1^b, formed by stacking the small upper triangular matrices R_i, is not upper triangular and hence the decomposition $Q_1^b R_1^b$ is not a QR decomposition of $\hat{K}$. To achieve the QR decomposition of $\hat{K}$ consider the following sequence of blocked decompositions for R_1^b,

$$\begin{bmatrix} R_1 \\ \hline R_2 \\ \hline R_3 \\ R_4 \end{bmatrix} = \left[\begin{array}{c|c} Q_5 & \\ \hline & Q_6 \end{array}\right] \left[\begin{array}{c} R_5 \\ \hline R_6 \end{array}\right] = Q_2^b R_2^b, \tag{3.2}$$

and finally the QR decomposition of the $2r \times r$ matrix R_2^b,

$$\left[\begin{array}{c} R_5 \\ \hline R_6 \end{array}\right] = Q_3^b R_3^b, \tag{3.3}$$

where we can give the final QR decomposition of $\hat{K}$ as,

$$\hat{K} = (Q_1^b Q_2^b Q_3^b) R_3^b = \hat{Q} R_3^b. \tag{3.4}$$

R_3^b by virtue of the construction is an upper triangular matrix, while $\hat{Q}$, a product of orthogonal matrices is itself orthogonal. A similar iteration can be carried out for any n, r by appropriately adjusting the block sizes. In some instances, if only $(R_3^b)^{-1}$ is required, the matrix product for forming $\hat{Q}$ need not be evaluated at all. On the other hand, if a product of the form $\hat{Q}^T x$ is required for some some vector x, the entire product maybe evaluated via blocking. We give more details about such strategies in our examples.

There are other ways of constructing a sequence of small blocked QR decompositions leading to a QR decomposition of the tall matrix $\hat{K}$. The scheme presented in (3.4) is the one allowing for maximum assimilation of computations at the same time. Depending on the architecture being used, this may not be the most optimal choice. Below we present another analogous blocked QR scheme, which allows for lesser number of computations to be carried on simultaneously, but possibly uses lesser communication between units of a multicore architecture. Assume, again for notational clarity that $n = 8r$. The matrix $\hat{K}$ is now partitioned into 8 blocks, $\{K_i\}_{i=1}^8$ of size $r \times r$ each. We operate on $\{K_i\}_{i=1}^4$ and $\{K_i\}_{i=5}^8$ simultaneously, via a sequence of small QR decompositions in the following manner,

$$\begin{bmatrix} K_1 \\ K_2 \\ K_3 \\ K_4 \\ \hline K_5 \\ K_6 \\ K_7 \\ K_8 \end{bmatrix} = \begin{array}{l} \begin{bmatrix} Q_1 & & & \\ & I & & \\ & & I & \\ & & & I \end{bmatrix} \begin{bmatrix} Q_2 & & & \\ & I & & \\ & & I & \end{bmatrix} \begin{bmatrix} Q_3 & & \\ & I & \\ & & I \end{bmatrix} Q_4 R_4 \\ \hline \begin{bmatrix} Q_5 & & & \\ & I & & \\ & & I & \\ & & & I \end{bmatrix} \begin{bmatrix} Q_6 & & & \\ & I & & \\ & & I & \end{bmatrix} \begin{bmatrix} Q_7 & & \\ & I & \\ & & I \end{bmatrix} Q_8 R_8, \end{array} \tag{3.5}$$

where, $Q_1 R_1$ is the orthogonal matrix from the QR decomposition of K_1, Q_2 is from the QR decomposition of $\begin{bmatrix} R_1 \\ K_2 \end{bmatrix}$, etc and I denotes identity matrix of appropri-

ate size. The matrices $R_1, R_2 \ldots$ are intentionally omitted from (3.5) for clarity. Analogously $Q_5, Q_6, \ldots$ are obtained from the partition $\{K_i\}_{i=5}^{8}$. To complete the decomposition, we make another QR decomposition of $\begin{bmatrix} R_4 \\ R_8 \end{bmatrix}$. This alternative blocking scheme involves minimum communication between units of the multicore architecture. The optimal schemes will usually be a mixture of the strategies (3.2) and (3.5). For more variants of blocking algorithms to get QR decompositions for tall matrices, we refer the readers to Constantine and Gleich (2011); Agullo et al. (2010).

We now consider the theoretical gains possible from the blocked algorithm, while considering block size. Consider the ideal scenario, when there is no overhead or communication cost and each unit of the multicore architecture has equal speed. We consider the strategy corresponding to (3.2). Each of the small QR decompositions involve an $2r \times r$ matrix, which has $O(r^3)$ cost. At the first iteration, as in (3.2), using the same number of units as the number of blocks, say b, we can perform all the computations in $O(r^3)$. In fact, it is easy to see that the computational order of $O(r^3)$ holds true for each of the subsequent iterations. The total number of iterations needed, following the first strategy would be $\lfloor \log_2(b) \rfloor + 1$, which brings the total computational cost to $O(\log_2(b) r^3)$ for the QR of $\hat{K}$. Forming the product $K\Omega$ and additional matrix multiplications (which will almost always be needed in applications after the QR decomposition), bring the cost to $O(n^2 r)$ without parallelization, whereas with the blocking algorithm, provided we perform the matrix multiplications with the same blocked structure, we achieve the whole computation in $O(\log_2(b) r^3)$. Following the first strategy and from the above discussion, in the ideal scenario, the number of blocks to be used is $b = \lfloor \frac{n}{2r} \rfloor + 1$. The theoretical possible maximum speed-up, assuming no communication cost, is $O(\frac{b^2}{\log_2(b)})$, where we measure the speed-up from Amdahl's equation (Mattson et al., 2005), as pro-

portion of speed of the unparallelized computations to the speed of the parallelized computations.

3.5 Theory: Motivating results and approximation error bounds

3.5.1 Fast decay of spectrum of large positive definite matrices

In this subsection we derive some motivating results justifying our assertion in §3.1 that positive definite matrices obtained as dense discrete realizations of positive definite kernels will have a very fast decaying spectrum, enabling us to obtain good approximations using much lower rank matrices. The ideas are not entirely new, we adapt abstract results from theory of integral equations and stochastic series expansions to our context. To prepare the background, we start with the well-known Mercer's theorem for positive definite functions.

Definition 2. *Let D be a compact metric space and C a function, $C : D \times D \to \mathbb{R}^+ \cup \{0\}$. C is said to be positive definite if for all n, scalars $c_1, \ldots, c_n \in \mathbb{R}$ and $x_1, \ldots, x_n \in D$, we have $\sum_i \sum_j c_i\, c_j\, C(x_i, x_j) > 0$. It follows trivially from the definition that if K is an $n \times n$ matrix with $k(i, j) = c(x_i, x_j)$ for $x_1, \ldots, x_n \in D$, then K is a positive definite matrix.*

With this definition, we give the following version of Mercer's theorem for the positive definite function $C(\cdot, \cdot)$ (Kühn, 1987),

Theorem 3 (Mercer's theorem)**.** *For every positive definite function $C(\cdot, \cdot)$ defined from $D \times D \to \mathbb{R}^+ \cup \{0\}$, where D is a compact metric space, there exists a real valued scalar sequence, $\{\lambda_i\} \in l_1$, $\lambda_1 \geqslant \lambda_2 \geqslant \ldots \geqslant 0$ and an orthonormal sequence of functions, $\{\phi_i(\cdot)\} \in L^2(D)$, $\phi_i(\cdot) : D \to \mathbb{R}$, such that,*

$$C(x_1, x_2) = \sum_{i \in \mathbb{N}} \lambda_i \phi_i(x_1) \phi_i(x_2),$$

$\forall x_1, x_2 \in D$ and this sequence converges in the mean square sense and uniformly in D.

We omit the the proof of this theorem and refer the readers to (Kühn, 1987).Mercer's theorem is a generalization of the spectral decomposition for matrices extended to functions. $\{\lambda_i\}$, $\{\phi(\cdot)\}$ are referred to as the eigenvalues and eigenfunctions respectively of $C(\cdot,\cdot)$. Alternatively, suppose $D \subset \mathbb{R}^d$ and considering Lebesgue measure, we have the following integral equation for the eigenvalues and eigenfunctions, analogous to the definition of matrix eigenvalues and eigenfunctions,

$$\lambda_i \phi_i(y) = \int_D C(x,y)\phi_i(x)dx, \tag{3.6}$$

$\forall x \in D, i \in \mathbb{N}$.

The eigenvalues and eigenfunctions of $C(\cdot,\cdot)$ can be approximated using the eigenvalues and eigenvectors of the positive definite matrix K, obtained by evaluating $C(\cdot,\cdot)$ at any set of n locations $x_1, \ldots x_n \in D$. To see this, consider a discrete approximation of the integral equation (3.6), using the points $x_1, \ldots, x_n$, with equal weights in the quadrature rule,

$$\lambda_i \phi_i(y) \approx \frac{1}{n} \sum_{j=1}^{n} C(x_j, y)\phi_i(x_j) \tag{3.7}$$

Substituting $y = x_1, \ldots, x_n$ in (3.7) we get a system of linear equations which is equivalent to the matrix eigenproblem, $KU^i = d(i,i)U^i, i = 1, \ldots, n$, where U^i is the i^{th} row of the eigenvector matrix U. This approximation is related to the Galerkin method for integral equations and the Nystrom approximation (Baker and Baker, 1977; Delves and Mohamed, 1988). It also corresponds to finite truncation of the expansion from the Mercer's theorem. In general it can be shown that $d(i,i)/n$ and $\sqrt{n}\, u(i,j)$ converge to λ_i and $\phi_i(x_j)$ respectively as $n \to \infty$ (Baker and Baker, 1977).

The accuracy of the finite truncation of the Mercer expansion has been in recent focus and error bounds have been obtained depending on the degree of smoothness of the positive definite function (Todor, 2006; Schwab and Todor, 2006), in the context of stochastic uncertainty quantification. To borrow from these ideas and adapt the

abstract results in our case, we begin with definitions quantifying the smoothness of the covariance functions.

Definition 3. *Let $C(\cdot,\cdot)$ be a positive definite function on $D \times D$, where D is a compact metric space. We call $C(\cdot,\cdot)$ piecewise Sobolev (p,q) smooth, for some $p,q \in \mathbb{N}$, if there exists a finite disjoint partition $\{D_j, j \in J\}$ of D, with $\overline{D} \subset \cup_{j\in J}\overline{D_j}$, such that for any pair $(j_1,j_2) \in J$, $C(\cdot,\cdot)$ is Sobolev (p,q) on $D_{j_2} \times D_{j_2}$.*

Definition 4. *Let $C(\cdot,\cdot)$ be a positive definite function on $D \times D$, where D is a compact metric space. We call $C(\cdot,\cdot)$ piecewise analytic smooth, for some $p,q \in \mathbb{N}$, if there exists a finite disjoint partition $\{D_j, j \in J\}$ of D, with $\overline{D} \subset \cup_{j\in J}\overline{D_j}$, such that for any pair $(j_1,j_2) \in J$, $C(\cdot,\cdot)$ is analytic on $D_{j_2} \times D_{j_2}$.*

Commonly used covariance functions such as the squared exponential, $C(x,y) = c_1 exp(-c_2\|x-y\|^2)$ and the Matern function, depending on the value of the parameter ν, fall into one of the above categories. In fact the squared exponential function is Sobolev smooth on any compact domain and we can give stronger results for decay of its eigenvalues.

Using the couple of definitions categorizing smoothness, we give the following result, which is analogous to a functional version of the Eckart-Young theorem.

Lemma 4. *Let $D \subset \mathbb{R}^d$. For any $m \in \mathbb{N}$, and let $\mathcal{F}_m$ be an m dimensional closed subspace of positive definite functions on $D \times D$, where by m dimensional we mean that $\exists$ sequence of orthonormal function $\{\psi_j(\cdot)\}_{j=1}^m$ on D such that$\{\psi_j^2(\cdot)\}_{j=1}^m$ spans $\mathcal{F}_m$, with positive coefficients by virtue of positive definiteness. Let $C_{\mathcal{F}_m}(\cdot,\cdot)$ be the projection of $C(\cdot,\cdot)$ onto $\mathcal{F}_m$, and infimum of the errors in approximating $C(\cdot,\cdot)$ by m dimensional functions $\in \mathcal{F}_m$, ie $E_m = \inf_{\mathcal{F}_m}\{\|C - C_{\mathcal{F}_m}\|^2, C_m \in \mathcal{F}_m\}$. Then,*
(i) If $C(\cdot,\cdot)$ is piecewise Sobolev (p,q) $\forall(p,q)$ $\exists$ for each $s > 1$, a positive constant c_s depending only on $C(\cdot,\cdot), s$, such that $E_m \leqslant c_s \sum_{j\geqslant m} j^{-s}$. Henceforth we call this

bound E_m^s.

(ii) If $C(\cdot,\cdot)$ *is piecewise analytic* $\exists$ *, positive constants* c_1, c_2 *depending only on* $C(\cdot,\cdot)$*, such that* $E_m \leqslant c_s \sum_{j\geqslant m} c_1 \exp^{c_2 m^{1/d}}$*. Henceforth we call this bound* E_m^a.

Proof. Let the Mercer expansion for $C(x,y)$ be $\sum_{j\geqslant 1} \lambda_j \phi_j(x)\phi_j(y)$. Define a random function, $f(x) = \sum_{j\geqslant 1} \eta_j \lambda_j \phi_j(x)$, where η_j are independent standard normal random variables. Let $C_{\mathcal{F}_m}(x,y) = \sum_{j=1}^m \theta_j \psi_j(x)\psi_j(y)$ corresponding to the basis $\{\psi^2(\cdot)\}_{j=1}^m$ for $\mathcal{F}_m$. Similar to f, corresponding to $C_{\mathcal{F}_m}(\cdot,\cdot)$, define a random function, $f_{\mathcal{F}_m}(x) = \sum_{j\geqslant 1} \zeta_j \theta_j \psi_j(x)$, where ζ_j's are independent random normal variables. Defining $\|f - f_{\mathcal{F}_m}\| = E(f - f_{\mathcal{F}_m})^2$, and applying theorem 2.7 in Schwab and Todor (2006), we get that, $E_m = \sum_{j\geqslant m} \lambda_j$. Then the result follows by applications of Corollary 3.3 and Proposition 3.5 in Todor (2006). □

The above lemma quantifies the finite truncation accuracy of the Mercer theorem for smooth kernels. The bounds obtained are optimal and in general cannot be improved. The spaces $\mathcal{F}_m$'s can be made more general to encompass all square integrable bivariate functions, but the proof becomes more involved in that case and we omit it for the sake of brevity. In case of the squared exponential kernel, which is smooth of all orders, the finite truncation is even sharper and we give the following corollary for it:

Corollary 5. *Let* $C(x,y) = \theta_1 \exp^{-\theta_2 \|x-y\|^2}$*, for* $x, y \in D$*, compact* $\subset \mathbb{R}^d$ *and* $\theta_1, \theta_2 > 0$*. Using all notations as in lemma 4,* $\exists$ *positive constant* c_{θ_1,θ_2} *such that* $E_m \leqslant c_{\theta_1,\theta_2} \sum_{j\geqslant m} \frac{\theta_2^{j^{1/d}}}{\Gamma(j^{1/d}/2)}$*. Henceforth we shall call this bound* E_r^e.

The proof is exactly similar to the proof of Lemma 4 and an application of Proposition 3.6 in Todor (2006).

We now describe some results relating to the matrix eigenvalues, using the strength of Lemma 4 above.

Theorem 6. *Let K be the $n \times n$ positive definite matrix with $k(i,j) = C(x_i, x_j)$, with $x_i, x_j \in \{x_1, \ldots, x_n\} \subset D \subset \mathbb{R}^d$, compact. Also let $K = UDU^T$ be the spectral decomposition of K. Then,*

(i) If $C(\cdot,\cdot)$ is piecewise Sobolev smooth $\forall (p,q)$, then $\exists$ for each $s > 1$, a positive constant c_s depending only on $C(\cdot,\cdot), s$, such that $d(m,m) \leqslant n\, c^{-s} m^{-s}$.

(ii) If $C(\cdot,\cdot)$ is piecewise analytic, then $\exists$ positive constants c_1, c_2 depending only on $C(\cdot,\cdot)$, such that $d(m,m) \leqslant n\, c_1 \exp^{c_2 m^{1/d}}$

(iii) In particular, for a squared exponential kernel, $C(x,y) = \theta_1 \exp^{-\theta_2 \|x-y\|^2}$ $\exists$ positive constant c_{θ_1,θ_2} such that $d(m,m) \leqslant n\, c_{\theta_1,\theta_2} \frac{\theta_2^{m^{1/d}}}{\Gamma(m^{1/d}/2)}$.

Proof. We begin with the discrete approximate solution of the integral equation, using the Galerkin technique, described in equation (3.7). Note that this method coincides with the Raleigh-Ritz method with the identity matrix in Baker and Baker (1977), since $C(\cdot,\cdot)$ is symmetric positive definite. Applying then theorem 3.31 on page 322 of Baker and Baker (1977), we have $d(m,m) \leqslant n\lambda_m$, where λ_m is the m^{th} eigenvalue from the Mercer expansion of $C(\cdot,\cdot)$. The result then follows by a straight-forward application of Lemma 4 above and Corollary 3.3, Proposition 3.5 in Todor (2006). For the squared exponential kernel, it is straightforward application of Corollary 5 above and Proposition 3.6 in Todor (2006). □

This theorem shows that for most covariance functions, the positive definite matrices that are generated by their finite realizations have eigenvalues that decay extremely fast, which was our assertion at the start of the section. In the simulated experiments we compute eigenvalues of some such covariance kernels and show that empirical results support our theory.

3.5.2 Approximation accuracy and condition numbers

We now present some results regarding the accuracy of our approximation algorithms. We shall be concerned with the column space approximation error, when we use $\hat{K} = \Omega$ instead of K, measured as $\|K - QQ^T K\|_\psi$, where Ω is a structured random matrix as in definition 1 and Q is orthonormal basis for the columns of $\hat{K}$. Such a Q can be obtained from the QR decomposition of $\hat{K}$.

Theorem 7. *Let K be the $n \times n$ positive definite matrix with $k(i,j) = C(x_i, x_j)$, with $x_i, x_j \in \{x_1, \ldots, x_n\} \subset D \subset \mathbb{R}^d$, compact. Let Ω be an $n \times r$ structured random matrix as formulated in definition 1 and Q be the left factor from the QR decomposition of $K\Omega$. Choose r, k such that $4[\sqrt{k} + \sqrt{8 \ln kn}]^2 \ln K \leqslant r \leqslant n$. With probability at least $(1 - O(1/k))$, the following hold true,*

(i) If $C(\cdot,\cdot)$ is piecewise Sobolev smooth $\forall (p,q)$, then (a) $\|K - QQ^T K\|_2 \leqslant (1 + \sqrt{7n/r}) c_s r^{-s}$, where c_s is a positive constant depending on s for any $s > 1$; (b) $\|K - QQ^T K\|_F \leqslant (1 + \sqrt{7n/r}) E_r^s$.

(ii) If $C(\cdot,\cdot)$ is piecewise analytic, then (a) $\|K - QQ^T K\|_2 \leqslant n(1 + \sqrt{7n/r}) c_1 \exp^{c_2 m^{1/d}}$, where c_1, c_2 are positive constants ; (b) $\|K - QQ^T K\|_F \leqslant n(1 + \sqrt{7n/r}) E_r^a$.

(iii) In particular, for a squared exponential kernel, $C(x,y) = \theta_1 \exp^{-\theta_2 \|x-y\|^2}$, then (a) $\|K - QQ^T K\|_2 \leqslant (1 + \sqrt{7n/r}) c_{\theta_1,\theta_2} \frac{\theta_2^{m^{1/d}}}{\Gamma(m^{1/d}/2)}$, where c_{θ_1,θ_2} is a positive constant depending on θ_1, θ_2; (b) $\|K - QQ^T K\|_F \leqslant (1 + \sqrt{7n/r}) E_r^e$,

where E_r^s, E_r^a, E_r^e are bounds as in Theorem 4 and Corollary 5.

Proof. First note that,

$$P_{\hat{K}} = ((\hat{K})^T \hat{K})^+ (\hat{K})^T,$$

where $((\hat{K})^T \hat{K})^+$ denotes the Moore-Penrose inverse of $((\hat{K})^T \hat{K})$. Let $Q_k R_k$ denote

the QR decomposition of $\hat{K}$, so that we have,

$$\begin{aligned} P_{\hat{K}}K &= (R_k^T Q_k^T Q_k R_k)^{-1} Q_k R_k K \\ &= R_k^{-1} Q_k^T K. \end{aligned}$$

Let K_r be the best rank r approximation to K and let the spectral decomposition of $K = UDU^T$. Then from the Eckart-Young theorem, $\|K - K_m\|_2 = d(r, r)$ and $\|K - K_m\|_2 = \sum_{j=r}^{n} d(j, j)$. The result then follows by an application of our Theorem 6 and Theorem 11.2 in Halko et al. (2011). □

In addition to providing accurate column space approximations, these orthogonal transforms spread the information of the matrix K and improve its conditioning, while preserving its geometry. By preserving the geometry, we mean preserving the norms of the eigenvectors - we explore more of this aspect empirically in the simulations. Any low rank approximation improves the conditioning, but it has been shown (Boutsidis and Gittens, 2012; Avron et al., 2010) that projections using the orthogonal transforms as above, improve conditioning substantially beyond what is achieved by just using any low rank approximation with high probability in special cases. Halko et al. (2011); Boutsidis and Gittens (2012) obtain bounds on the eigenvalues of $A\Omega$ where Ω is a structured random matrix and A is orthogonal. We are in interested in the stability of the numerical system, $\hat{K}R_k^{-1}$, as explained in the introduction of this chapter, where the QR decomposition of $\hat{K} = Q_k R_k$. We present the following result, without proof, trivially modifying results from Boutsidis and Gittens (2012); Avron et al. (2010),

Theorem 8. *Fix* $0 < \epsilon < (1/3)$ *and choose* $0 < \delta < 1, r$ *such that* $6\epsilon^2[\sqrt{(}r) + \sqrt{8 \ln n/\delta}]^2 \ln 2n/\delta \leqslant r \leqslant n$. *Then with probability at least* $1 - 2\delta$, *we have the condition number of the linear system of interest,* $c(KR_k^{-1}) \leqslant \sqrt{\frac{1+\epsilon}{1-\epsilon}}$.

This improvement in conditioning number causes huge improvement in learning algorithms over other competing approaches as we demonstrate later.

3.6 Illustrations

3.6.1 Simulation Examples

We begin with empirical investigation into the eigenvalue decays of matrices realized from commonly used covariance kernels. We consider an equispaced grid of 100 points in $[0, 1]$, to generate the 100×100 covariance matrices with 100 positive eigenvalues. We start with such a moderate size, 100, to illustrate that even for this small size, eigenvalues decay extremely fast, as was hypothesized and theoretically proven earlier.

This first one we use is the squared exponential kernel,

$$C(x, y) = \theta_1 \exp^{-\theta_2 \|x-y\|^2}$$

The parameter θ_1 is a scaling parameter and does not change the eigendirections. In our present simulation, we set $\theta_1 = 1$. The parameter θ_2 controls the smoothness of the covariance, as it decays with the distance $\|x - y\|$. We consider a range of θ_2 values, $0.05, 0.5, 1, 1.5, 2, 10$, respectively, considering negligible decay with distance and very fast decay with distance. The plot of the eigenvalues is shown in figure 3.1. The figure shows similar rates of decay across the range of values of the smoothness parameter θ_2. Depending on the smoothness parameter, the value of the largest eigenvalue changes by several orders, but in all cases, by the $10th$ largest eigenvalue, the sequence has approximately converged to 0, which indicates that for a low rank approximation, a choice of rank 10 would be sufficient for our purposes.

This second one we use is the Matern covariance kernel (Williams and Rasmussen, 1996),

$$C(x, y) = \theta_1 \frac{1}{\Gamma(\nu) 2^{\nu-1}} \left(\sqrt{2\nu} \frac{\|x - y\|}{\theta_2} \right)^{\nu} B_\nu \left(\sqrt{2\nu} \frac{\|x - y\|}{\theta_2} \right),$$

where B_ν is the modified Bessel function of the second kind. The parameter θ_1 is a scaling parameter and does not change the eigendirections, analogous to θ_1 of the squared exponential kernel. In our present simulation, we set $\theta_1 = 1, \theta_2 = 1$. The parameter ν controls the smoothness of the covariance, analogous to the parameter θ_2 of the squared exponential kernel. As a special case, as $\nu \to \infty$, we obtain the squared exponential kernel. In typical spatial applications, ν is considered to be within $[0.5, 3]$, as the data are rarely informative about the smoothness levels beyond these. We consider a range of ν values, $0.5, 1, 1.5, 2, 2.5, 3$, respectively, considering negligible decay with distance and very fast decay with distance. The plot of the eigenvalues is shown in figure 3.2. The figure shows similar rates of decay across the range of values of the smoothness parameter ν, exactly as exhibited with the squared exponential kernel. Once again, a low rank approximation of rank 10 would suffice in this case for approximating the 100×100 matrix.

Using the same set of simulations, we compute conditioning numbers, in turn for each of the covariance kernels. We compute the conditioning number of the the full 100×100 covariance matrix and its best rank m approximations for $m = 5, 10, 15, 20, 50$ respectively. Results for the squared exponential kernel are tabulated in table 3.1 and for the Matern covariance kernel in table 3.2. The full covariance matrix is extremely ill-conditioned in each case, the best rank m approximations improve the conditioning, as is to be expected. Even with the best rank m approximations, the conditioning numbers are still very large indicating very unstable linear systems. In the table we omit the exact digits for the very large numbers and just indicate their orders in terms of powers of 10. In case of the squared exponential kernel, condition numbers decrease from left to right and from top to bottom. This pattern is in general not true for the condition numbers of the Matern kernel, whose condition numbers are several orders smaller than that for the squared exponential but still to large for it to be stable linear systems. In general for the Matern kernel,

as ν becomes larger, the kernel becomes smoother, the condition numbers are larger and the eigenvalues decay faster.

We next move to the simulations to consider the effect of blocking and gain in efficiency from parallelization. For large sample sizes, we will get very poor and time consuming inverse estimates, as demonstrated by the large conditioning numbers from the simulations of the previous section. To circumvent this, we start therefore with a known spectral decomposition and apply our approximations, pretending that the decomposition was unknown. We consider the known spectral decomposition as $K = EDE^T$, where E is an orthonormalized matrix with randomly generated iid Gaussian entries. D is the diagonal matrix of eigenvalues in decreasing order or magnitude and for this simulation example we assume exponential decay with scale = 1 and rate 0.01. We generate this covariance matrix K for sample sizes $n = 1000, 5000, 10000, 50000$ respectively and apply our blocked approximate inversion algorithm. For the random matrix Ω used for the projection, we use three different choices, a matrix with scaled iid standard Gaussian entries, a structured random Hartley transform and a structured random discrete cosine transform. We have a total 64 cores at our disposal and to get a flavor of the gains possible by parallelization, we use $8, 16, 32, 64$ cores in turn and measure the gain in efficiency as the ratio of time taken to the time for the algorithm without parallelization.

We report the gain in efficiency in figure 3.3. The figure reveals the efficiency gain by using the blocked algorithm. It shows that efficiency gain increases with number of cores in the larger sample sizes. Specifically when the sample size is $50,000$, it seems that more than 64 cores could have further increased the efficiency. The gains reported are obviously lower than the theoretical maximum possible gains, but are still substantial and have potential of further increase with larger number of cores in huge sample sizes.

3.6.2 Real data examples

We consider a real data examples with a very large data set, to demonstrate the large data handling power and efficiency gain via our approach. We consider subsets of the actual data available and variables such that the data fit into a Gaussian process framework.

The example we consider is a subset of the Sloan sky digital survey (Stoughton et al., 2007). In the survey, redshift and photometric measurements were made for a very large number of galaxies. The photometric measurements consist of 5 band measurements and are available for most of the galaxies, while the redshift measurements are not available for a large number of galaxies due to spectroscopy constraints. The main interest is therefore in predicting the red-shifts, given the other 5 measurements. For this example we consider a subset comprising of $180,000$ galaxies, whose photometric measurements and red-shift measurements are known. We hold out $30,000$ galaxies and pretend that redshifts are unknown for these galaxies, while using the remaining $150,000$ to train our Gaussian process model.

For the computations we use both a squared exponential kernel and a Matern kernel. The data are scaled and centered before calculations so that we are not concerned with inference regarding the θ_1 parameters for the covariance kernel. For the Matern kernel, we use a discrete uniform prior for the parameter ν based on 100 equispaced points on the grid $[0.5, 3]$. For the squared exponential kernel, we use a 100 trials, each having a random selection $1,000$ points from the training set to set a band for θ_2, to estimate range of covariance of the data. It appears that it is sufficient to consider θ_2 in the range $[0.05, 1]$. We therefore use a grid on 100 uniformly spaced points in $[0.05, 1]$ and analogous to the prior for ν, place a discrete uniform prior on θ_2 of the squared exponential kernel.

For data sets of this size it becomes extremely inefficient to use predictive pro-

cesses or knot based approaches per se, without blocking or parallelization and knot selection is almost practically infeasible to attempt. We work with the modified predictive processes approach of Finley et al. (2009), without knot selection as competitors to our approach. The modified predictive processes is conceptually equivalent to the fully independent training conditional approach (Quinonero Candela and Rasmussen, 2005) or the Gaussian process approximation with pseudo inputs, with (Snelson and Ghahramani, 2006), with the same priors for the parameters as our approach, so that the only difference is in the way the covariance inversion is done.

For each of the approaches, including ours, one issue is how to select rank of the approximate projections. We use an approximate estimate of the covariance decay to come up with the low rank projection to be used. To calculate this we select $1,000$ data points spread out across the training set, in the following approximate manner, first select the pair of points which are most distant from each other, then select the point which is almost equidistant from the pair selected, and so on, till we have a well spread out selection. Here distance is measured as the Euclidean distance between the 5 variate photometric measurement vectors. Using these $1,000$ data points, we calculate the prior covariance matrix, using the Gaussian covariance kernel and the Matern kernel for each value of the smoothness parameter on its grid of values (Our simulation examples have revealed that the decay of eigenvalues depends on this smoothing parameter). We then estimate the rank such that Frobenius norm error in using the best low rank approximation from the Eckart Young theorem for the full covariance matrix would be no more than 0.001 (bearing in mind that this is an estimate of the best possible error, actual error in different projections will be more). In the computations for our approach, we use a different random projection to the targeted rank for each grid value of the smoothing parameter, while for the knot based approaches, a different random selection of knots is used.

For our approach, we use 3 different choices of the projection matrix Ω, a matrix

with scaled iid standard Gaussian entries, a structured random matrix with the Hartley transform and a structure random matrix with the discrete cosine transform. We use blocking as in our main algorithm for each of the three types of projection matrices on all 64 cores of the distributed computing structure. The knot based approach has no obvious method of blocking or for it to be used on a grid computing environment. The MCMC algorithm in each case is run for $10,000$ iterations, with the first $2,000$ discarded as burn-in. Usual diagnostics do not show evidence of non-convergence of the MCMC algorithms and we report the effective sample size for the smoothing parameter as obtained from the CODA package in R Plummer et al. (2006). Time taken by each method is computed and reported. The time calculations are made by recording the exact CPU times for iterations, excluding the time for Frobenius norm accuracy and conditioning number calculations and also excluding the time taken to set up parallel Matlab workers via matlabpool. We measure predictive accuracy, of the predictions in the hold out set, based on the relative mean squared error.

We tabulate the results from the experiment in the table 3.3. The performance of any of the projection based approaches is substantially better than the knot based PP in terms of any of the parameters reported, predictive accuracy, time taken or the effective sample size of the smoothing parameter. Time taken by the PP in case of this large dataset was several days, as compared to a few hours for the blocked projection approaches. The improvement in predictive accuracy can probably be attributed to much better conditioned stable linear systems being used inherently and hence improved inference. The discrete cosine transform and Hartley transform perform marginally better than the random Gaussian projection, however the structured random transforms show marked improvement in the time taken, possibly due to the faster matrix multiplies. There is little to choose between the discrete cosine transform or the discrete Hartley transform.

3.7 Discussion

In this chapter we present a new method of blocked approximate inversion for large positive definite matrices, using subsampled random orthogonal transforms, motivated by several recent developments in random linear algebra. The blocking strategy comes from development of parallel QR algorithms for tall matrices and in our case tall matrices are obtained from large positive definite matrices as the first projection step of a low rank approximation. We have also presented a comprehensive set of theoretical results quantifying the decay of eigenvalues of covariance matrices, which in turns helps bound the errors from the low rank projections. The examples show marked improvement in terms of numerical stability, prediction accuracy and efficiency and should be routinely applicable in a wide variety of scenarios.

One interesting future direction is the investigation of several structured random projections taken together. One may break up a target rank into several parts, the first of which is spent on a random projection, and the remaining pieces are learnt through the direction of maximum accuracy. Both theoretical (finding projections in the directions of maximum accuracy, which may be the direction of maximum information descent) and empirical investigation of this issue is currently being investigated.

Table 3.1: Table of condition numbers for the squared exponential kernel for different values of the smoothing parameter θ_2 versus truncation levels, for truncating to the best possible approximation according to the Eckart-Young theorem. The rows represent the levels of truncation and the columns, the values of the smoothing parameter.

	$\theta_1 = 0.05$	$\theta_2 = 0.5$	$\theta_2 = 1$	$\theta_2 = 1.5$	$\theta_2 = 2$	$\theta_2 = 10$
full, $m = 100$	$O(10^{20})$	$O(10^{19})$	$O(10^{19})$	$O(10^{19})$	$O(10^{18})$	$O(10^{18})$
$m = 50$	$O(10^{17})$	$O(10^{17})$	$O(10^{17})$	$O(10^{17})$	$O(10^{17})$	$O(10^{16})$
$m = 20$	$O(10^{17})$	$O(10^{17})$	$O(10^{16})$	$O(10^{16})$	$O(10^{16})$	$O(10^{15})$
$m = 15$	$O(10^{17})$	$O(10^{16})$	$O(10^{16})$	$O(10^{16})$	$O(10^{16})$	$O(10^{10})$
$m = 10$	$O(10^{16})$	$O(10^{15})$	$O(10^{13})$	$O(10^{11})$	$O(10^{10})$	$O(10^{5})$
$m = 5$	$O(10^{9})$	$O(10^{6})$	$O(10^{4})$	$O(10^{4})$	$O(10^{3})$	29.89

Table 3.2: Table of condition numbers for the Matern kernel for different values of the smoothing parameter ν versus truncation levels, for truncating to the best possible approximation according to the Eckart-Young theorem. The rows represent the levels of truncation and the columns, the values of the smoothing parameter.

	$\nu = 0.5$	$\nu = 1$	$\nu = 1.5$	$\nu = 2$	$\nu = 2.5$	$\nu = 3$
full, $m = 100$						
full, $m = 100$	$O(10^3)$	$O(10^5)$	$O(10^7)$	$O(10^9)$	$O(10^{11})$	$O(10^{13})$
$m = 50$	$O(10^3)$	$O(10^5)$	$O(10^6)$	$O(10^8)$	$O(10^9)$	$O(10^{11})$
$m = 20$	$O(10^2)$	$O(10^4)$	$O(10^5)$	$O(10^6)$	$O(10^7)$	$O(10^9)$
$m = 15$	$O(10^2)$	$O(10^3)$	$O(10^4)$	$O(10^5)$	$O(10^6)$	$O(10^7)$
$m = 10$	$O(10^2)$	$O(10^3)$	$O(10^3)$	$O(10^4)$	$O(10^5)$	$O(10^5)$
$m = 5$	38.18	$O(10^2)$	$O(10^2)$	$O(10^2)$	$O(10^3)$	$O(10^3)$

Table 3.3: Results from the real data experiment. Columns are the type of experiment, PP corresponds to the modified predictive process approach, RP corresponds to the projection method with a Gaussian projection matrix, HP corresponds to a structured random projection with the Hartley transform, HC corresponds to a structured random projection with the discrete cosine transform. Time taken is measured as relative time, taking the time taken by HC to be 1. RMSE is relative mean squared error, ESS stands for effective sample size.

	PP	RP	HP	HC
RMSE, Sq Exp	70.63	19.87	14.64	15.72
Relative time, Sq Exp	$O(10^3)$	10.79	1.57	1
Avg Condition No, Sq Exp	$O(10^7)$	$O(10^3)$	$O(10^2)$	$O(10^2)$
Avg Frobenius norm error, Sq Exp	0.55	0.04	0.07	0.05
ESS, Sq Exp	237	833	1991	1875
RMSE, Matern	35.61	21.27	20.83	20.97
Relative time, Matern	$O(10^4)$	32.30	0.91	1
Avg Condition No, Matern	$O(10^4)$	$O(10^3)$	$O(10^2)$	$O(10^2)$
Avg Frobenius norm error, Matern	1.37	0.62	0.18	0.39
ESS, Matern	569	1322	1219	1794

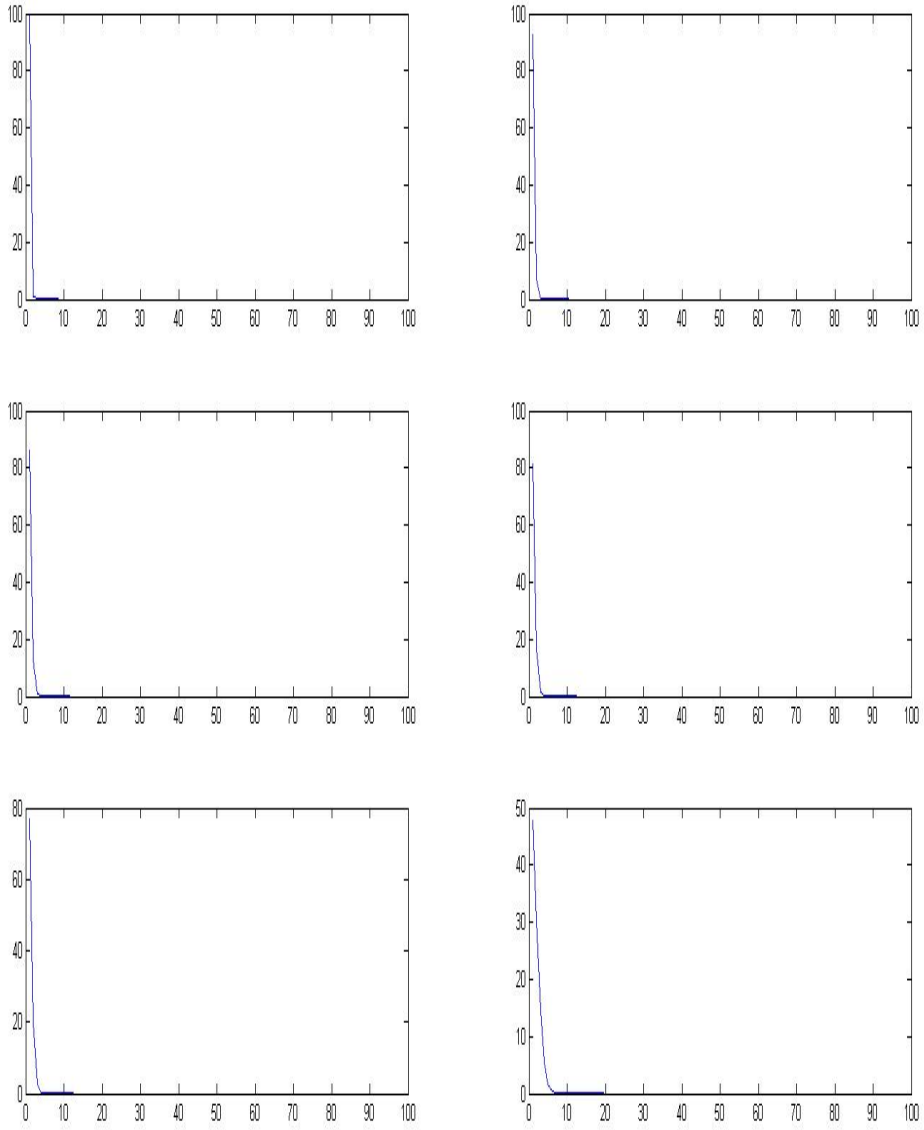

FIGURE 3.1: Decay of Eigenvalues: Panel representing decay of eigenvalues in the squared exponential covariance kernel. We plot the 100 eigenvlaues in decreasing order of magnitude, the x-axes represent the indices, the y-axes the eigenvalues. Top left panel, top right, middle left, middle right, bottom left, bottom right are for values of the smoothness parameter $\theta_2 = 0.05, 0.5, 1, 1.5, 2, \&10$ respectively.

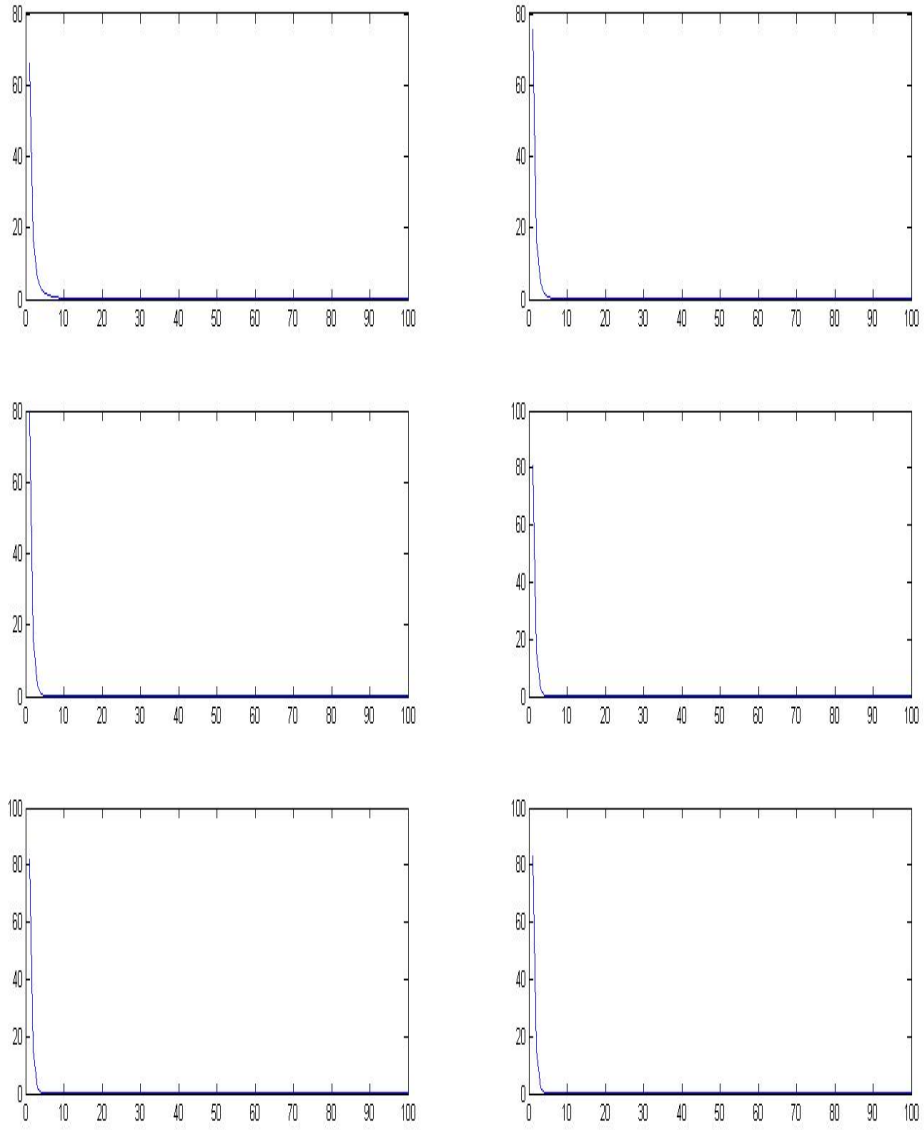

FIGURE 3.2: Decay of Eigenvalues: Panel representing decay of eigenvalues in the Matern covariance kernel. We plot the 100 eigenvlaues in decreasing order of magnitude, the x-axes represent the indices, the y-axes the eigenvalues. Top left panel, top right, middle left, middle right, bottom left, bottom right are for values of the smoothness parameter $\nu = 0.5$, 1, 1.5, 2, 2.5, &3 respectively.

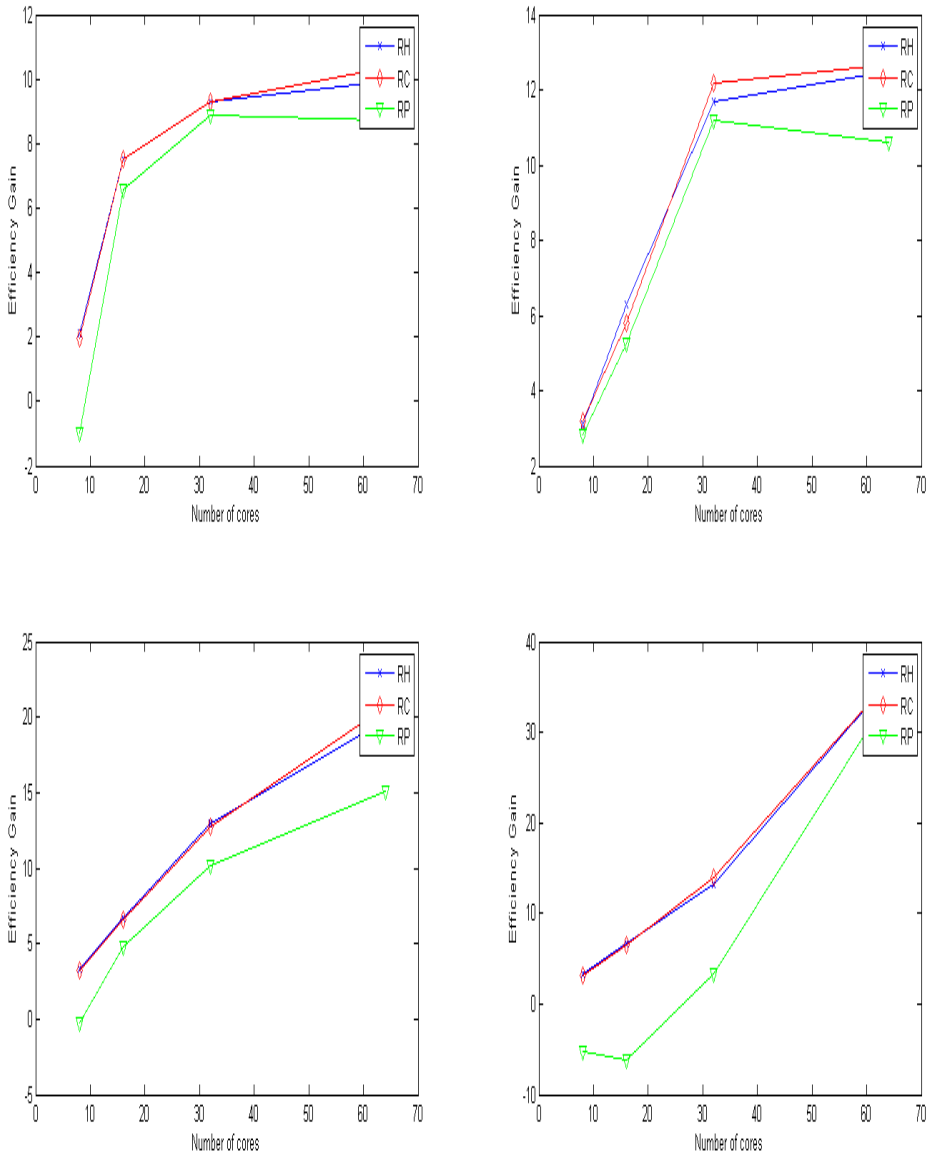

FIGURE 3.3: Gain in efficiency by using increasing number of cores in a parallel computing environment: Top left panel, top right, bottom left, bottom right are for sample sizes $n = 1000$, 5000, 10000, 50000 respectively. Superimposed on each panel are the gains by using a Hartley transform (RH), discrete cosine transform (RC) and a scaled random Gaussian projection (RP).

4

Infinite tensor factorization priors for joint modeling

4.1 Summary

There is increasing interest in broad application areas in defining flexible joint models for data having a variety of measurement scales, while also allowing data of complex types, such as functions, images and documents. We consider a general framework for nonparametric Bayes joint modeling through mixture models that incorporate dependence across data types through a joint mixing measure. The mixing measure is assigned a novel infinite tensor factorization (ITF) prior that allows flexible dependence in cluster allocation across data types. The ITF prior is formulated as a tensor product of stick-breaking processes. Focusing on a convenient special case corresponding to a Parafac factorization, we provide basic theory justifying the flexibility of the proposed prior and resulting asymptotic properties. Focusing on ITF mixtures of product kernels, we develop a Gibbs sampling algorithm for routine implementation relying on slice sampling. The methods are compared with alternative joint mixture models based on Dirichlet processes and related approaches through simulations and real data applications.

4.2 Introduction

There has been increasing emphasis in a broad variety of application areas on joint modeling data of widely disparate types, including not only real numbers, counts and categorical data but also more complex *objects*, such as functions, shapes, and images. We refer to this general problem as *mixed domain modeling* (MDM), and a rich variety of relevant methods have been proposed in the literature. Until recently, the emphasis in the statistics literature was almost entirely on parametric hierarchical models for joint modeling of mixed discrete and continuous data without considering more complex *object data*. The two main strategies are to rely on underlying Gaussian variable models (Muthen, 1984) or exponential family models, which incorporate shared latent variables in models for the different outcomes (Sammel et al., 1997; Dunson, 2000, 2003). Recently, there have been a number of articles using these models as building blocks in discrete mixture models relying on Dirichlet processes (DPs) or closely-related variants (Cai et al., 2011; Song et al., 2009; Yang and Dunson, 2010). DP mixtures for mixed domain modeling were also considered by citetHB11,SN09,BD10 among others. Related approaches are increasingly widely-used in broad machine learning applications, such as for joint modeling of images and captions (Li et al., 2011), and have rapidly become a standard tool for MDM. Although such models are quite flexible, and can accommodate joint modeling with complicated objects such as functions (Bigelow and Dunson, 2009), we argue in this article that they suffer from a key disadvantage that can be overcome with *next generation* extensions of the DP to accommodate dependent object type-specific clustering.

As motivation, we start by considering a simple bivariate setting $p = 2$ in which data for subject i consist of $y_i = (y_{i1}, y_{i2})' \in \mathcal{Y}$, with $\mathcal{Y} = \mathcal{Y}_1 \otimes \mathcal{Y}_2$, $y_{i1} \in \mathcal{Y}_1$, and $y_{i2} \in \mathcal{Y}_2$ for $i = 1, \ldots, n$. We desire a joint model in which $y_i \sim f$, with f a

probability measure characterizing the joint distribution. In particular, letting $\mathcal{B}(\mathcal{Y})$ denote an appropriate sigma-algebra of subsets of $\mathcal{Y}$, f assigns probability $f(B)$ to each $B \in \mathcal{B}(\mathcal{Y})$. We assume $\mathcal{Y}$ is a measurable Polish space, as we would like to keep the domains $\mathcal{Y}_1$ and $\mathcal{Y}_2$ as general as possible to encompass not only subsets of Euclidean space and the set of natural numbers but also function spaces that may arise in modeling curves, surfaces, shapes and images. In many cases, it is not at all straightforward to define a parametric joint measure, but there is typically a substantial literature suggesting various choices for the marginals $y_{i1} \sim f_1$ and $y_{i2} \sim f_2$ separately.

If we only had data for the jth variable, y_{ij}, then one possible strategy is to use a mixture model in which

$$f_j(B) = \int_{\Theta_j} \mathcal{K}_j(B; \theta_j) dP_j(\theta_j), \quad B \in \mathcal{B}(\mathcal{Y}_j), \tag{4.1}$$

where $\mathcal{K}_j(\cdot; \theta_j)$ is a probability measure on $\{\mathcal{Y}_1, \mathcal{B}(\mathcal{Y}_1)\}$ indexed by parameters $\theta_j \in \Theta_j$, $\mathcal{K}_j$ obeys a parametric law (e.g., Gaussian), and P_j is a probability measure over $\{\Theta_j, \mathcal{B}(\Theta_j)\}$. A nonparametric Bayesian approach is obtained by treating P_j as a random probability measure and choosing an appropriate prior. By far the most common choice is the Dirichlet process (Ferguson, 1973), which lets $P_j \sim DP(\alpha P_{0j})$. Under the Sethuraman (1994) stick-breaking representation, one then obtains

$$f_j(B) = \sum_{h=1}^{\infty} \pi_h \mathcal{K}_j(B; \theta_h^*), \quad \pi_h = V_h \prod_{l<h} (1 - V_l), \quad \theta_h^* \sim P_{0j}, \tag{4.2}$$

and $V_h \sim \text{Be}(1, \alpha)$, so that f_j can be expressed as a discrete mixture. This discrete mixture structure implies the following simple hierarchical representation, which is crucially used for efficient computation:

$$\begin{aligned} y_{ij} &\sim \mathcal{K}_j(\theta_{C_i}^*), \quad \theta_h^* \sim P_{0j} \\ \text{pr}(C_i = h) &= \pi_h, \end{aligned} \tag{4.3}$$

where C_i is a cluster index for subject i. The great success of this model is largely attributable to the *divide and conquer* structure in which one allocates subjects to clusters probabilistically, and then can treat the observations within each cluster as separate instantiations of a parametric model. In addition, there is a literature showing appealing properties, such as minimax optimal adaptive rates of convergence for DPMs of Gaussians (Shen and Ghosal, 2011; Tokdar, 2011a).

The standard approach to adapt expression (4.1) to accommodate mixed domain data is to simply let $f(B) = \int_\Theta \mathcal{K}(B;\theta)dP(\theta)$, for all $B \in \mathcal{B}(\mathcal{Y})$, where $\mathcal{K}(\cdot;\theta)$ is an appropriate joint probability measure over $\{\mathcal{Y}, \mathcal{B}(\mathcal{Y})\}$ obeying a parametric law. Choosing such a joint law is straightforward in simple cases. For example, Hannah et al. (2011) rely on a joint exponential family distribution formulated via a sequence of generalized linear models. However, in general settings, explicitly characterizing dependence within $\mathcal{K}(\cdot;\theta)$ is not at all straightforward and it becomes convenient to rely on a product measure (Dunson and Bhattacharya, 2010):

$$\mathcal{K}(B;\theta) = \prod_j \mathcal{K}(B_j;\theta_j), \quad B = \bigotimes_{j=1}^{p} B_j, \quad B_j \in \mathcal{B}(\mathcal{Y}_j). \tag{4.4}$$

If we then choose $P \sim DP(\alpha P_0)$ with $P_0 = \bigotimes_{j=1}^{p} P_{0j}$, we obtain an identical hierarchical specification to (4.3), but with the elements of $y_i = \{y_{ij}\}$ *conditionally independent* given the cluster allocation index C_i.

This conditional independence assumption given a single latent class variable is the nemesis of the joint DPM approach leading to practically poor performance in many moderate to high-dimensional settings even when one is not faced with the complication of multi-modal data analysis. For example, as motivated in (Dunson and Xing, 2009; Dunson, 2010), the DP and related approaches imply that two subjects i and i' are either allocated to the same cluster ($C_i = C_{i'}$) *globally* for all their parameters or are not clustered. The soft probabilistic clustering of the DP

is appealing in leading to substantial dimensionality reduction, but a single global cluster index conveys several substantial practical disadvantages. Firstly, to realistically characterize joint distributions across many variables, it may be necessarily to introduce many clusters, degrading the performance in the absence of large sample sizes. Secondly, as the DP and the intrinsic Bayes penalty for model complexity both favor allocation to few clusters, one may over cluster and hence obscure important differences across individuals, leading to misleading inferences and poor predictions. Often, the posterior for $\{C_i\}$ may be largely driven by certain components of the data, particularly when more data are available for those components, at the expense of poorly characterizing components for which less, or more variable, data are available.

For these reasons, we seek to define general classes of nonparametric Bayes models, which introduce separate but dependent cluster indices for each type of data, so that instead of a single $C_i \in \{1, \ldots, \infty\}$, we have a multivariate $C_i = (C_{i1}, \ldots, C_{ip})^T \in \{1, \ldots, \infty\}^p$ with

$$\text{pr}(C_{i1} = h_1, \ldots, C_{ip} = h_p) = \pi_{h_1 \cdots h_p}, \quad h_j = 1, \ldots, \infty, j = 1, \ldots, p, \tag{4.5}$$

where $\pi = \{\pi_{h_1 \cdots h_p}\} \in \Pi_p^\infty$ is an infinite p-way *probability tensor* characterizing the joint probability mass function of the multivariate cluster indices. A related problem was addressed by Petrone et al. (2009) who proposed a hybrid Dirichlet process motivated by functional data applications. They focused on functional data models in which individual surfaces were formulated as a locally-selected patchwork of a global collection of surfaces. This corresponds to letting $C_i(s) \in \{1, \ldots, \infty\}$, with $C_i(s)$ the cluster allocation at location s. Substantial challenges result in the implementation. Alternatively, Dunson (2009) proposed a local partition process prior, which treats the elements of C_i as either equal to a global cluster index or to independent local cluster indices. Dunson (2010) developed a formulation that allows dependence in

clustering to be driven by known covariates, such as a time index.

The fundamental limitation of these and other methods is that they do not allow *learning* of a flexible dependence structure in the cluster indices. For example, subjects allocated to $C_{i1} = 1$ for the first data type may be more likely to be allocated to $C_{i4} = 3$ for the fourth data type, but we may have no prior information or covariates to include characterizing that relationship. Indeed, a fundamental goal of the nonparametric Bayes joint modeling analysis is to infer unanticipated dependencies in general types of data, while exploiting those dependencies for dimensionality reduction in characterizing the joint distribution, performing inferences and conducting predictions. With this in mind, we need a general purpose framework that can accommodate a broad variety of mixed data types and hierarchical modeling settings beyond simply product kernel models, while leading to straightforward posterior computation using existing MCMC and other tools developed for Dirichlet process-type models.

4.3 Probabilistic Tensor Factorizations for Dependent Clustering

4.3.1 Tensor Factorizations for Categorical Data

We begin by reviewing tensor factorizations for a finite p-way probability tensor. In particular, suppose that $C_{ij} \in \{1, \ldots, d_j\}$, with d_j the number of possible levels of the jth cluster index. Then, assuming that C_i are observed unordered categorical variables, Dunson and Xing (2009) proposed a probabilistic Parafac factorization of the tensor π:

$$\pi = \sum_{h=1}^{k} \lambda_h \psi_h^{(1)} \otimes \cdots \otimes \psi_h^{(p)}, \tag{4.6}$$

where $\lambda = \{\lambda_h\}$ follows a stick-breaking process, $\psi_h^{(j)} = (\psi_{h1}^{(j)}, \ldots, \psi_{hd_j}^{(j)})^T$ is a probability vector specific to component h and outcome j, $\otimes$ denotes the outer product and

one can potentially let $k = \infty$. The acronym Parafac stands for parallel factorization, analogous to the factor models for matrix valued data. This approach characterizes the probability tensor π as a weighted sum of rank one tensors expressed as outer products of probability vectors, with the weights decreasing rapidly towards zero to favor shrinkage towards a low rank characterization. The characterization in (4.6) is quite useful practically in modeling high dimensional categorical data and sparse contingency tables in that it can be used to characterize any joint probability mass function, while favoring a representation having few effective degrees of freedom and leading to simple and efficient computation even as p increases.

An alternative to the Parafac tensor factorizations is the so-called Tucker or higher-order SVD factorization that replaces (4.6) with

$$\pi = \sum_{h_1=1}^{k} \cdots \sum_{h_p=1}^{k} \lambda_{h_1 \cdots h_p} \psi_{h_1}^{(1)} \otimes \cdots \otimes \psi_{h_j}^{(p)}, \tag{4.7}$$

where $\{\lambda_{h_1 \cdots h_p}\} \in \Pi_p^k$ is a p-way *core tensor*. Bhattacharya and Dunson (2012) recently proposed a simplex factor model, which leads to a computationally convenient Tucker factorization of the joint probability mass function for multivariate categorical data. Although the simplex factor model has some advantages over (4.6) in terms of parsimoniously characterizing dependence in high-dimensional categorical data, we focus primarily on Parafac-type representations for simplicity in generalizing to the case in which C_i is unobserved and can take infinitely-many different levels.

4.3.2 Infinite Tensor Factorizations

Our emphasis is on mixture models that incorporate multivariate, dependent cluster indices $C_i = (C_{i1}, \ldots, C_{ip})^T$ with the joint probability mass function expressed as in (4.5) so that each C_{ij} can take one of infinitely-many different levels. We can be flexible in terms of exactly where these cluster indices appear within a hierarchi-

cal Bayesian semi- or nonparametric model, but the key initial question is how to choose a prior for the infinite probability tensor π that leads to a flexible dependence structure in the cluster indices, favors a parsimonious structure utilizing relatively few clusters (and hence having relatively few effective degrees of freedom), and leads to efficient posterior computation utilizing tools developed in Dirichlet process-type mixture models. In addition, we would certainly like to inherit the appealing large support and asymptotic properties of Dirichlet mixtures even in extending the framework to accommodate multivariate, dependent underlying clustering. Our primary focus is not on clustering for its own sake but in using soft probabilistic clustering to induce flexible, parsimonious joint models with good practical performance. However, that said, one does in the process infer a posterior distribution over a multivariate cluster index as well as the joint probability distribution for these indices, and such quantities may be of interest in many settings.

Focusing an a Parafac factorization, which extends (4.6) to infinitely-many levels, we let

$$\begin{aligned} \pi_{c_1 \cdots c_p} &= \text{pr}(C_1 = c_1, \ldots, C_p = c_p) = \sum_{h=1}^{\infty} \lambda_h \prod_{j=1}^{p} \psi_{hc_j}^{(j)} \\ \lambda_h &= V_h \prod_{l<h} (1 - V_l), \quad V_h \sim \text{Be}(1, \alpha) \\ \psi_{hr}^{(j)} &= U_{hr}^{(j)} \prod_{s<r} (1 - U_{hs}^{(j)}), \quad U_{hr}^{(j)} \sim \text{Be}(1, \beta_j), \end{aligned} \tag{4.8}$$

A more compact notation for this factorization of the infinite probability tensor π is

$$\pi = \sum_{h=1}^{\infty} \lambda_h \bigotimes_{j=1}^{p} \psi_h^{(j)}, \quad \lambda \sim \text{Stick}(\alpha), \ \psi_h^{(j)} \sim \text{Stick}(\beta_j), \tag{4.9}$$

which takes the form of a stick-breaking mixture of outer products of stick-breaking processes. This form is carefully chosen so that the elements of π are stochastically

larger in those cells having the smallest indices, with rapid decreases towards zero as one moves away from the upper right corner of the tensor. In this sense, we obtain a type of array stick-breaking process, which has a very different form from the matrix stick-breaking process of Dunson et al. (2008). As shorthand notation, we use $\pi \sim \text{ITF}(\alpha, \beta)$ to denote the infinite tensor factorization prior in (4.9), with $\beta = (\beta_1, \ldots, \beta_p)^T$.

The hyperparameter α controls the overall dependence between ensemble components. As α decreases towards zero, we obtain independent Dirichlet process clustering for each data type, while as α increases there will be dependence in clustering. As β decreases, we favor fewer clusters within each data type, with separate β_js allowing certain data types to have more clusters than others. In practice, we will place hyperpriors on these key hyperparameters for greater data adaptivity. The prior ensures that the elements of π are close to zero except in the upper left corner, suggesting that one can approximate π by $\pi^{(k)}$, where $\pi^{(k)}$ sets $U_{hk}^{(j)} = 1$ so that $\pi_{c_1 \cdots c_p}^{(k)} = 0$ if $c_j > k$ for any j. This is formalized in Lemma 2 in §2.3. Hence, even though we allow an infinite-dimensional tensor, it can typically be accurately approximated by a low dimensional tensor, favoring parsimony in joint modeling.

4.3.3 Properties

Let $\mathcal{Q}$ denote a probability measure on Π_p^∞ such that any realization $\pi \sim \mathcal{Q}$ can be expressed as $\pi = \sum_{h=1}^\infty \lambda_h \bigotimes_{j=1}^p \psi_h^{(j)}$, where $\lambda = \{\lambda_h\} \in \mathcal{S}_\infty$ and $\psi_h^{(j)} \in \mathcal{S}_\infty$, with $\mathcal{S}_k$ the k-simplex. To define $\mathcal{Q}$ as a general class of tensor products of stick-breaking processes, we further assume

$$\begin{aligned} \lambda_h &= V_h \prod_{l<h}(1 - V_l), \quad V_h \sim \text{Be}(a_h, \alpha_h), \\ \psi_{hr}^{(j)} &= U_{hr}^{(j)} \prod_{s<r}(1 - U_{hs}^{(j)}), \quad U_{hr}^{(j)} \sim \text{Be}(b_{rj}, \beta_{rj}), \end{aligned} \tag{4.10}$$

The prior ITF(α, β) is special case of this formulation as are other stick breaking constructions like the Pitman-Yor process (Pitman and Yor, 1997).

For any p-way tensor $\pi \in \Pi_p^\infty$ to be a valid probability measure, we need its components to sum to 1 almost surely. This will not hold for all priors $\mathcal{Q}$ in the general class defined above, but will hold for a subclass satisfying the sufficient conditions in Lemma 9.

Lemma 9. *Assume that,*

$$\sum_{h=1}^{\infty} log\left(1 + \frac{a_h}{\alpha_h}\right) = \infty \text{ and } \sum_{r=1}^{\infty} log\left(1 + \frac{b_{rj}}{\beta_{rj}}\right) = \infty, \text{ for } j = 1, \ldots, p.$$

Then for any $\pi \sim \mathcal{Q}$, *(i)* $E\left\{\sum_{c_1=1}^{\infty} \cdots \sum_{c_p=1}^{\infty} \pi_{c_1,\ldots,c_p}\right\} = 1$ *and (ii)* $\sum_{c_1=1}^{\infty} \cdots \sum_{c_p=1}^{\infty} \pi_{c_1,\ldots,c_p} = 1$ *almost surely.*

It is easy to verify that conditions of Lemma 9 hold for ITF(α, β). Therefore, we have the following corollary,

Corollary 10. *Let* $\pi \sim ITF(\alpha, \beta)$. *Then (i)*$E\left\{\sum_{c_1=1}^{\infty} \cdots \sum_{c_p=1}^{\infty} \pi_{c_1,\ldots,c_p}\right\} = 1$ *and (ii)* $\sum_{c_1=1}^{\infty} \cdots \sum_{c_p=1}^{\infty} \pi_{c_1,\ldots,c_p} = 1$ *almost surely.*

Having ensured validity of the proposed prior we discuss its support properties. We first show in Lemma 11 that any probability tensor can be approximated by a finite Parafac factorization.

Lemma 11. *Consider any* $\pi \in \Pi_p^\infty$. *Then, for any* $\epsilon > 0$, *there exists a* $\pi^* \in \Pi_p^\infty$ *such that* $\|\pi - \pi^*\|_1 < \epsilon$ *where* $\pi^* = \sum_{h=1}^{k} \lambda_h \bigotimes_{j=1}^{p} \psi_h^{(j)*}$ *with* $\lambda = (\lambda_1, \ldots, \lambda_k)^T \in \mathcal{S}_{k-1}$, $\psi_h^{(j)*} \in \mathcal{S}_\infty$, *and* $\psi_{hc_j}^{(j)*} = 0$ *for any* $c_j > d_j$, *with* $d_j < \infty$ *for* $j = 1, \ldots, p$.

With the aid of Lemma 11 we prove that the proposed ITF prior has support on the whole of $\mathcal{P}$.

Lemma 12. *Consider any* $\pi^0 \in \Pi_p^\infty$. *Define the* ϵ $L1$ *neighborhood of* π^0 *as* $N_\epsilon(\pi^0) = \{\pi \in \Pi_p^\infty : \|\pi^0 - \pi\|_1 < \epsilon\}$. *Then for any* $\epsilon > 0$ *we have* $ITF\{N_\epsilon(\pi^0)\} > 0$.

In the next lemma we summarize expectations of various clustering properties of draws from the ITF(α, β) prior.

Lemma 13. *Let* $\pi \sim ITF(\alpha, \beta)$ *and* π^m *be respective marginals, that is,* $\pi^m(1) = \left\{\sum_{c_2=1}^\infty \cdots \sum_{c_p=1}^\infty \pi_{c_1,\ldots,c_p}\right\}$. *Let* $(C_1, \ldots, C_p)$ *be cluster indices, with* $pr(C_1 = c_1, \ldots, C_p = c_p) = \pi_{c_1,\ldots,c_p}$ *and* I *any index set,* $I \subset \{1, \ldots, p\}$. *Then for any* $i, j \in \{1, \ldots, p\}$,

(i) $E\{pr(C_j = c_j)\} = E\{\pi^m(c_j)\} = \frac{\beta_i^{c_j - 1}}{(1+\beta_j)^{c_j}}$.

(ii) $E\{pr(C_1 = c_1, \ldots, C_p = c_p)\} = \prod_{j=1}^p \frac{\beta_j^{c_j - 1}}{(1+\beta_j)^{c_j}}$.

(iii) $E\{pr(C_j$*'s are equal, for* $j \in I)\} = \frac{1}{1+\sum_{j \in I} \beta_j}$.

(iv) $\lim_{\beta_j \to 0, \forall j \in I} \{pr(C_j$*'s are equal, for* $j \in I)\} = 1$ *almost surely. On the other hand,* $\lim_{\beta_j \to \infty, \text{for any } j \in I} \{pr(C_j$*'s are equal, for* $j \in I)\} = 0$ *almost surely.*

Therefore, by (ii) the prior is centered on independent clustering across components. Also, property (iv) shows how the hyperparameters β control variability in clustering, with probabilities of components being clustered together ranging from zero to one in limiting cases. In practice, hyperpriors on β allows the data to inform about the appropriate values.

4.4 Infinite Tensor Factorization Mixtures

Assume that for each individual i we have a data ensemble $(y_{i1}, \ldots, y_{ip}) \in \mathcal{Y}$ where $\mathcal{Y} = \bigotimes_{j=1}^p \mathcal{Y}_j$. Let $\mathcal{B}(\mathcal{Y})$ be the sigma algebra generated by the product sigma algebra $\mathcal{B}(\mathcal{Y}_1) \times \cdots \times \mathcal{B}(\mathcal{Y}_p)$. Consider any Borel set $B = \bigotimes_{j=1}^p B_j \in \mathcal{B}(\mathcal{Y})$. Given

cluster indices ($C_{i1} = c_{i1}, \ldots, C_{ip} = c_{ip}$), we assume that the ensemble components are independent with

$$f(y_{i1} \in B_1, \ldots, y_{ip} \in B_p \,|\, C_{i1} = h_1, \ldots, C_{ip} = h_p) = \prod_{j=1}^{p} \mathcal{K}_j(B_j; \theta_{j,h_j}). \tag{4.11}$$

$\mathcal{K}_j(\cdot; \theta_{j,h})$ is an appropriate probability measure on $\{\mathcal{Y}_j, \mathcal{B}(\mathcal{Y}_j\}$ as in equation (4.1). Marginalizing out the cluster indices, we obtain

$$f(y_{i1} \in B_1, \ldots, y_{ip} \in B_p) = \sum_{h_1=1}^{\infty} \cdots \sum_{h_p=1}^{\infty} \pi_{h_1,\ldots,h_p} \prod_{j=1}^{p} \mathcal{K}_j(B_j; \theta_{j,h_j}), \tag{4.12}$$

where, $\pi_{h_1,\ldots,h_p} = \text{pr}(C_{i1} = h_1, \ldots, C_{ip} = h_p)$. We let $\pi \sim \text{ITF}(\alpha, \beta)$ and we call the resulting mixture model an infinite tensor factorization mixture, $f \sim \text{ITM}(\alpha, \beta)$. To complete the model specification, we let $\theta_{j,h_j} \sim P_{0j}$ independently as in (4.2).

The model $y_i \sim f$, $f \sim \text{ITM}(\alpha, \beta)$, can be equivalently expressed in hierarchical form as

$$\begin{aligned} y_{ij} &\sim \mathcal{K}_j(\theta^*_{ij}),\ \theta^*_i = P \sum_{h_1=1}^{\infty} \cdots \sum_{h_p=1}^{\infty} \pi_{h_1,\ldots,h_p} \prod_{j=1}^{p} \delta_{\theta_{j,h_j}}, \\ \pi &\sim \text{ITF}(\alpha, \beta), \quad \theta_{j,h_j} \sim P_{0j}, \end{aligned} \tag{4.13}$$

Here, P is a joint mixing measure across the different data types and is given a infinite tensor process prior, $P \sim \text{ITP}(\alpha, \beta, \bigotimes_{j=1}^{p} P_{0j})$. Marginalizing out the random measure P, we obtain the same form as in (4.12). The proposed infinite tensor process prior provide a much more flexible generalization of existing priors for discrete random measures, such as the Dirichlet process (Shahbaba and Neal, 2009) or Pitman Yor process (Pitman and Yor, 1997).

It is trivial to verify that any draw from the ITM is a valid probability measure on $\mathcal{B}(\mathcal{Y})$, using the result for validity of draws from ITF in lemma 9.

A crucial component of the proposed ITM model is the flexible borrowing of information across the ensemble components through dependent clustering. A priori, the ITM model is centered on independent ensemble components, however this no longer holds true for the a posteriori expectation in general.

4.5 Posterior Inference

4.5.1 Markov Chain Monte Carlo Sampling

Infinite tensor factorization mixtures admit simple and efficient exact MCMC posterior inference, utilizing blocked and partially collapsed steps. We adapt ideas from Walker (2007) and Papaspiliopoulos and Roberts (2008) to derive slice sampling steps with label switching moves, avoiding truncation approximations. Begin by defining the augmented joint likelihood for an observation y_i, cluster labels $c_i = (c_{i0}, c_{i1}, \ldots, c_{ip})$ and slice variables $u_i = (u_{i0}, u_{i1}, \ldots, u_{ip})$ as

$$p(y_i, c_i, u_i \mid \lambda, \Psi, \Theta) = \mathbf{1}\left(u_{i0} < \lambda_{c_{i0}}\right) \prod_{j=1}^{p} \mathcal{K}_j(y_{ij}; \theta^{(j)}_{c_{ij}}) \mathbf{1}\left(u_{ij} < \psi^{(j)}_{c_{i0}c_{ij}}\right) \tag{4.14}$$

It is straightforward to verify that on marginalizing u_i the model is unchanged, but including u_i induces full conditional distributions for the cluster indices with finite support. Let $m_{0h} = \sum_{i=1}^{n} \mathbf{1}\left(c_{i0} = h\right)$ and $\mathcal{D}_0 = \{h : m_{0h} > 0\}$. Similarly define $m_{jhk} = \sum_{i=1}^{n} \mathbf{1}\left(c_{i0} = h\right) \mathbf{1}\left(c_{ij} = k\right)$ and $\mathcal{D}_j = \{k : \sum_{h=1}^{\infty} m_{jhk} > 0\}$, and let $k_j^* = \max(\mathcal{D}_j)$ for $0 \leqslant j \leqslant p$. Define $\mathcal{U}_0 = \{u_{i0} : 1 \leqslant i \leqslant n\}$, $\mathcal{C}_0 = \{c_{i0} : 1 \leqslant i \leqslant n\}$, $\mathcal{U}_1 = \{u_{ij} : 1 \leqslant i \leqslant n, 1 \leqslant j \leqslant p\}$ and $\mathcal{C}_1 = \{c_{ij} : 1 \leqslant i \leqslant n, 1 \leqslant j \leqslant p\}$. The superscript $(-i)$ denotes that the quantity is computed excluding observation i.

1. Block update $(\mathcal{U}_0, \lambda, \alpha)$

   (a) Sample $(\alpha \mid \mathcal{C}_0)$. Standard results (Antoniak 1974) give

   $$p(\alpha \mid \mathcal{C}_0) \propto p(\alpha) \alpha^{\tilde{c}} \frac{\Gamma(\alpha)}{\Gamma(\alpha + n)}$$

   for $\tilde{c} = |\mathcal{D}_0|$ which can be sampled via Metropolis-Hastings or using auxiliary

variables when $p(\alpha)$ is a mixture of Gamma distributions (Escobar & West, 1995).

(b) Sample $(\lambda \mid \alpha, \mathcal{C}_0)$ by drawing $V_h \sim Beta(1 + m_{0h}, \alpha + \sum_{l=h+1}^{k_0^*} m_{0l})$ for $1 \leqslant h \leqslant k_0^*$ and setting $\lambda_h = V_h \prod_{l<h}(1 - V_l)$

(c) Label switching moves:

i. From $\mathcal{D}_0$ choose two elements h_1, h_2 uniformly at random and change their labels with probability $\min(1, (\lambda_{h_1}/\lambda_{h_2})^{m_{0h_2} - m_{0h_1}})$

ii. Sample a label h uniformly from $1, 2, \ldots, k_0^*$ and propose to swap the labels $h, h+1$ and corresponding stick breaking weights V_h, V_{h+1}. Accept with probability $\min(1, a)$ where

$$a = \left(\frac{k_0^*}{k_0^* + 1}\right)^{\mathbf{1}\left(h = k_0^*\right)} \frac{(1 - V_h)^{m_{0(h+1)}}}{(1 - V_{h+1})^{m_{0h}}}$$

(d) Sample $(u_{i0} | c_{i0}, \lambda) \sim U(0, \lambda_{c_{i0}})$ independently for $1 \leqslant i \leqslant n$

2. Update $\mathcal{C}_0$. From (4.14) the relevant probabilities are

$$Pr(c_{i0} = h | u_i, c_i, \Psi, \lambda) \propto \mathbf{1}\left(u_{i0} < \lambda_h\right) \prod_{j=1}^{p} \mathbf{1}\left(u_{ij} < \psi_{hc_{ij}}^{(j)}\right)$$

However, it is possible to obtain more efficient updates through *partial collapsing*, which allows us to integrate over the lower level slice variables and Ψ instead of conditioning on them. Then we have

$$Pr(c_{i0} = k \mid u_{i0}, \mathcal{C}_1, \mathcal{C}_0^{(-i)}, \lambda)$$

$$\propto \mathbf{1}\left(u_{i0} < \lambda_h\right) \prod_{j=1}^{p} \frac{\left(1 + m_{jkc_{ij}}^{(-i)}\right) \prod_{l<c_{ij}} \left(\beta_k^{(j)} + \sum_{s>l} + m_{jks}^{(-i)}\right)}{\prod_{l \leqslant c_{ij}} \left(1 + \beta_k^{(j)} + \sum_{s \geqslant c_{ij}} + m_{jks}^{(-i)}\right)} \tag{4.15}$$

To determine the support of (4.15) we need to ensure that $u_0^* = \min\{u_{i0} : 1 \leqslant i \leqslant n\}$ satisfies $u_0^* > 1 - \sum_{l=1}^{k_0^*} \lambda_l$ If $\sum_{l=1}^{k_0^*} \lambda_l < 1 - u_0^*$ then draw additional stick breaking weights $V_{k_0^*+1}, \ldots, V_{k_0^*+d}$ independently from $Beta(1, \alpha)$ until $\sum_{l=1}^{k_0^*+d} \lambda_l > 1 - u_0^*$,

ensuring that $\sum_{l=k_0^*+d+1}^{\infty} \mathbf{1}\left(u_{i0} < \lambda_h\right) = 0$ for all $1 \leqslant i \leqslant n$. Then the support of (4.15) is contained within $1, 2, \ldots, k_j^* + d$ and we can compute the normalizing constant exactly.

3. Block update $(\mathcal{U}_1, \Psi, \beta)$:

   (a) Update $(\beta_r^{(j)} \mid \{c_{ij} : c_{i0} = r\}, \mathcal{C}_0)$ for $1 \leqslant j \leqslant p$, $1 \leqslant r \leqslant k_0^*$. If the concentration parameter is shared across global clusters (that is, $\beta_r^{(j)} \equiv \beta^{(j)}$) then a straightforward conditional independence argument gives

$$p(\beta^{(j)} \mid \{c_{ij} : c_{i0} = r\}, \mathcal{C}_0) \propto p(\beta^{(j)}) \prod_{r \in \mathcal{D}_0} \left(\beta^{(j)}\right)^{\tilde{c}_{jr}} \frac{\Gamma(\beta^{(j)})}{\Gamma(\beta^{(j)} + n_r)}$$

   where $n_r = |\{i : c_{i0} = r\}|$ and $\tilde{c}_{jr} = |\{h : m_{jrh} > 0\}|$. Note that terms with $n_r = 1$ (corresponding to top-level singleton components) do not contribute, since $\beta^{(j)}\Gamma(\beta^{(j)}) = \Gamma(\beta^{(j)} + 1)$. The updating scheme of Escobar & West (1995) is simple to adapt here using $|\mathcal{D}_0|$ independent auxiliary variables.

   (b) For $r \in \mathcal{D}_0$ update $(\psi_r^{(j)} \mid \mathcal{C}_0, \mathcal{C}_1, \beta_r^{(j)})$ by drawing $U_{rh}^{(j)} \sim Beta(1 + m_{jrh}, \beta_r^{(j)} + \sum_{l=h+1}^{k_0^*} m_{jlh})$ for $1 \leqslant h \leqslant k_j^*$

   (c) Label switching moves: For $1 \leqslant j \leqslant p$,

      i. From $\mathcal{D}_j$ choose two elements h_1, h_2 uniformly at random and change their labels with probability $\min(1, a)$ where

$$a = \prod_{h_0 \in \mathcal{D}_0} \left(\frac{\psi_{h_0 h_1}^{(j)}}{\psi_{h_0 h_2}^{(j)}}\right)^{m_{jh_0h_2} - m_{jh_0h_1}}$$

      ii. Sample a label h uniformly from $1, 2, \ldots, k_j^*$ and propose to swap the labels $h, h+1$ and corresponding stick breaking weights. Accept with probabiliy $min(1, a)$ where

$$a = \left(\frac{k_j^*}{k_j^* + 1}\right)^{\mathbf{1}\left(h = k_j^*\right)} \prod_{h_0 \in \mathcal{D}_0} \frac{\left(1 - U_{rh}^{(j)}\right)^{m_{jr(h+1)}}}{\left(1 - U_{r(h+1)}^{(j)}\right)^{m_{jrh}}}$$

(d) Sample $(u_{ij}|c_i, \Psi) \sim U(0, \psi^{(m)}_{c_{i0}c_{ij}})$ independently for $1 \leqslant j \leqslant p$, $1 \leqslant i \leqslant n$.

4. Update $\mathcal{C}_j$ for $1 \leqslant j \leqslant p$ independently. We have

$$Pr(c_{ij} = k \mid y, \Theta, u_{ij}, c_{i0}, \Psi) \propto \mathcal{K}_j(y_{ij}; \theta^{(j)}_k) \mathbf{1}\left(u_{ij} < \psi^{(j)}_{c_{i0}k}\right)$$

As in step 2 we determine the support of the full conditional distribution as follows: Let $u^*_j = \min\{u_{ij} : 1 \leqslant i \leqslant n\}$. For all $r \in \mathcal{D}_0$, if $\sum_{h=1}^{k^*_j} \psi^{(j)}_{rh} < 1 - u^*_j$ then extend the stick breaking measure $\psi^{(j)}_r$ by drawing d_r new stick breaking weights from the prior so that $\sum_{h=1}^{k^*_j + d_r} \psi^{(j)}_{rh} > 1 - u^*_j$. Draw $\theta^{(j)}_{k^*_j+1}, \ldots, \theta^{(j)}_{k^*_j+d} \sim p(\theta^{(j)})$ independently (where $d = \max\{d_r : r \in \mathcal{D}_j\}$). Then update c_{ij} from

$$Pr(c_{ij} = k \mid y, \Theta, u_{ij}, c_{i0}, \Psi) = \frac{\mathcal{K}_j(y_{ij}; \theta^{(j)}_k) \mathbf{1}\left(u_{ij} < \psi^{(j)}_{c_{i0}k}\right)}{\sum_{h=1}^{k^*_j + d} \mathcal{K}_j(y_{ij}; \theta^{(j)}_k) \mathbf{1}\left(u_{ij} < \psi^{(j)}_{c_{i0}k}\right)}$$

5. Update $(\Theta|-)$ by drawing from

$$p(\theta^{(j)}_h \mid y, \mathcal{C}_j) \propto p(\theta^{(m)}_h) \prod_{\{i:c_{ij}=h\}} \mathcal{K}_j(y_{ij}; \theta^{(j)}_h)$$

for each $1 \leqslant j \leqslant p$ and $1 \leqslant h \leqslant k^*_j$

Remark 1: An alternative augmented likelihood using two slice variables instead of $p + 1$ is

$$p(y_i, c_i, u_i \mid \lambda, \Psi, \Theta) = \mathbf{1}\left(u_{i1} < \prod_{j=1}^{p} \psi^{(j)}_{c_{i0}c_{ij}}\right) \mathbf{1}\left(u_{i0} < \lambda_{c_{i0}}\right) \prod_{j=1}^{p} \mathcal{K}_j(y_{ij}; \theta^{(j)}_{c_{ij}}) \tag{4.16}$$

While there is some reason to believe that MCMC in this lower-dimensional space may have better convergence behavior, (4.16) is difficult to handle in two ways. First it would require checking the condition $1 - \sum_{h_1=1}^{k^*_1} \cdots \sum_{h_p=1}^{k^*_p} \prod_{j=1}^{p} \psi^{(j)}_{c_{i0}h_j} < \min\{u_{i1}\}$. If this fails it is not obvious how to efficiently compute the slice (that is, along which margin(s) the

probability tensor should be extended). Second, the lower cluster indices $(c_{i1} \ldots, c_{ip})$ are artificially coupled when conditioning on u_{i1}, even when also conditioning on c_{i0}.

Remark 2: Marginalizing a variable out of some full conditionals and not others can result in a set of incompatible conditional distributions, yielding a Markov chain that is no longer ergodic. Van Dyk and Park (2008) give a general recipe for partially collapsed Gibbs sampling, and demonstrate that like blocked and marginalized Gibbs samplers, partially collapsed Gibbs samplers dominate naive Gibbs in terms of maximal autocorrelation. To verify that step 2 maintains the correct stationary distribution first observe that if we sampled instead from the blocked conditional $p(\Psi, \mathcal{U}_1, c_{i0}|-) = p(\Psi, \mathcal{U}_1, | c_{i0}, -)p(c_{i0}|u_{i0}, \mathcal{C}_1, \mathcal{C}_0^{(-i)}, \lambda)$ then it is trivial to confirm that the stationary distribution is correct. This would require n redundant samples of $(\Psi, \mathcal{U}_1)$. But since $(\Psi, \mathcal{U}_1)$ are updated again in step 3 before they are conditioned upon we can safely skip their updates in step 2 without altering the stationary distribution of the chain. Note that step three must immediately follow step two, unlike a vanilla Gibbs sampler which would be invariant to permuting the steps.

Remark 3: It is possible to more efficiently compute the slice in step 4 by replacing u_j^* with $u_{rj}^* = \min\{u_{ij} : c_{i0} = r\}$. We present the simpler method here for clarity.

4.5.2 Inference

Given samples from the MCMC scheme above we can estimate the predictive distribution as

$$\hat{f}(y_{n+1} \mid y_n) = \frac{1}{T} \sum_{t=1}^{T} \sum_{h_0=1}^{k_0^*} \sum_{h_1=1}^{k_1^*} \cdots \sum_{h_p=1}^{k_p^*} \lambda_{h_0}^{(t)} \prod_{j=1}^{p} \psi_{h_0 h_j}^{(t)} \mathcal{K}_j \left(y_{(n+1)j}; \theta_{h_j}^{(j)(t)} \right) \tag{4.17}$$

Each of the inner sums in (4.17) is a truncation approximation, but it can be made arbitrarily precise by extending the stick breaking measures with draws from the prior and drawing corresponding atoms from $p(\theta^{(j)})$. In practice this usually isn't necessary as any error in the approximation is small relative to Monte Carlo error.

The other common inferential question of interest in the MDM settings is the dependence between components, for example testing whether component $j1$ and $j2$ are

independent of each other. As already noted, the dependance between the components comes in through the dependance between the cluster allocations and therefore, tests for independence between $j1$ and $j2$ is equivalent to testing for independence between their latent cluster indicators C_{j1} and C_{j2}. Such a test can be constructed in terms of the divergence between the joint and marginal posterior distributions of C_{j1} and C_{j2}. The Monte Carlo estimate of the Kulback Leibler divergence between the joint and marginal posterior distributions is given as,

$$\begin{aligned} I(j1,j2) \quad &= \tfrac{1}{T}\sum_{t=1}^{T}\sum_{h_{j1}=1}^{k_{j1}^*}\sum_{h_{j2}=1}^{k_{j2}^*}\left\{\left(\sum_{h_0=1}^{k_0^*}\lambda_{h_0}^{(t)}\psi_{h_0h_{j1}}^{(t)}\psi_{h_0h_{j2}}^{(t)}\right)\right. \\ &\qquad \left.\log\left(\frac{\sum_{h_0=1}^{k_0^*}\lambda_{h_0}^{(t)}\psi_{h_0h_{j1}}^{(t)}\psi_{h_0h_{j2}}^{(t)}}{\left[\sum_{h_0=1}^{k_0^*}\lambda_{h_0}^{(t)}\psi_{h_0h_{j1}}^{(t)}\right]\left[\sum_{h_0=1}^{k_0^*}\lambda_{h_0}^{(t)}\psi_{h_0h_{j1}}^{(t)}\right]}\right)\right\} \end{aligned} \tag{4.18}$$

Under independence, the divergence should be 0. Testing each such pairwise independence allows us to represent graphically the dependance structure, where nodes represent the components and conditional independence implies absence of the corresponding edge. Analogous divergences can be considered for testing other general dependancies, like 3 -way, 4-way independences.

4.6 Experiments

Our approach can be used for two different objectives in the context of mixed domain data - for prediction and for inference on the dependence structure between different data types. We outline results of experiments with both simulated and real data that show the performance of our approach with respect to both the objectives.

4.6.1 Simulated Data Examples

To the best of our knowledge, there is no standard model to jointly predict for mixed domain data as well as evaluate the dependence structure, so as a competitor, we use a joint DPM. To keep the evaluations fair, we use two scenarios. In the first the ground truth is close to that of the joint DPM, in the sense that all the components of the mixed data have the same cluster structure. The other simulated experiment considers the case when the ground truth is close to the ITF, where different components of the mixed data

ensemble have their own cluster structure but clustering is dependent. The goal here in each of the scenarios is to compare joint DPM vs ITF in terms of recovery of dependence structure and predictive accuracy.

For scenario 1, we consider a set of 1,000 individuals from whom an ensemble comprising of T, a time series R, a multivariate real-valued response ($\in \Re^4$) and C1,C2,C3, 3 different categorical variables have been collected, to emulate the type of data collected from patients in cancer studies and other medical evaluations. For the purposes of scenario 1, we simulate T, R, C1, C2, C3 each from a mixture of 3 clusters. For example, R is simulated from a two-component mixture of multivariate normals with different means, R is simulated from a mixture of two autoregressive kernels and each of the categorical variables from a mixture of two multinomial distributions. If we label the clusters as 1 and 2, for each simulation, either all of the ensemble (T,R,C1,C2,C3) comes from 1 or all of it comes from 2. After simulation we randomly hold out R in 50 individuals, C1, C2 in 10 each, for the purposes of measuring prediction accuracy. For the categorical variables prediction accuracy is considered with a $0-1$ loss function and is expressed as a percent missclassification rate. For the multivariate real variable R, we consider squared error loss and accuracy is expressed as relative predictive error. We also evaluate for some of the pairs their dependence via estimated mutual information.

For scenario 2, the same set-up as in scenario 1 is used, except for the cluster structure of the ensemble. Now simulations are done such that T falls into three clusters and this is dependent on R and C1. C2 and C3 depend on each other and are simulated from two clusters each but their clustering is independent of the other variables in the ensemble. We measure prediction accuracy using a hold out set of the same size as in scenario 1 and also evaluate the dependence structure from the ITF model.

In each case, we take 100,000 iterations of the MCMC scheme with the first few 1,000 discarded as a burn-in. These are reported in table 4.1 (left). We also summarize the recovered dependence structure in table 4.1 and in table 4.2. In scenario 1, the prediction accuracy of ITF and DPM are comparable, with DPM performing marginally better in a couple of cases. Note that the recovered dependence structure with the ITF is exactly accurate which shows that the ITF can reduce to joint co-clustering when that is the truth. In scenario 2, however there is significant improvement in using the ITF over the DPM

with predictive accuracy. In fact the predictions from the DPM for the categorical variable are close to noise. The dependence structure recovered the ITF almost reflects the truth as compared to that from the DPM which predicts every pair is dependent, by virtue of its construction.

4.6.2 Real Data Examples

For generic real mixed domain data the dependence structure is wholly unknown. To evaluate how well the ITF does in capturing pairwise dependencies, we first consider a network example in which recovering dependencies is of principal interest and prediction is not relevant. We consider data comprising of 105 political blogs (Adamic and Glance, 2005) where the edges in the graph are composed of the links between websites. Each blog is labeled with its ideology, and we also have the source(s) which were used to determine this label. Our model includes the network, ideology label, and binary indicators for 7 labeling sources (including "manually labeled", which are thought to be the most subject to errors in labelings). We assume that ideology impacts links through cluster assignment only, which is a reasonable assumption here. We collect 100,000 MCMC iterations after a short burn-in and save the iterate with the largest complete-data likelihood for exploratory purposes.

Fig. 4.1 shows the network structure, with nodes colored by ideology. It is immediately clear that there is significant clustering, apparently driven largely by ideology, but that ideology alone does not account for all the structure present in the graph. Joint DPM approach would allow for only one type of clustering and prevent us from exploring this additional structure. The recovered clustering in fig. 4.2 reveals a number of interesting structural properties of the graph; for example, we see a tight cluster of conservative blogs which have high in- and out- degrees but do not link to one another (green) and a partitioning of the liberal blogs into a tightly connected component (purple) and a periphery component with low degree (blue). The conservative blogs do not exhibit the same level of assortative mixing (propensity to link within a cluster) as the liberal blogs do, especially within the purple component.

To get a sense for how stable the clustering is, we estimate the posterior probability that nodes i and j are assigned to the same cluster by recording the number of times this

event occurs in the MCMC. We observe that the clusters are generally quite stable, with two notable exceptions. First, there is significant posterior probability that points 90 and 92 are assigned to the red cluster rather than the blue cluster. This is significant because these two points are the conservative blogs which are connected only to liberal blogs (see fig. 4.1). While the graph topology strongly suggests that these belong to the blue cluster, the labels are able to exert some influence as well. Note that we do not observe the same phenomenon for points 7, 15, and 25, which are better connected. We also observe some ambiguity between the purple and blue clusters. These are nodes 6, 14, 22, 33, 35 and 36, which appear at the intersection of the purple/blue clusters in the graph projection because they are not quite as connected as the purple "core" but better connected than most of the blue clusters.

Finally, we examine the posterior probability of being labeled "conservative" (fig. 4.3). Most data points are assigned very high or low probability. The five labeled points stand out as having uncharacteristic labels for their link structure (see fig 4.1). Since the observed label doesn't agree with the graph topology, the probability is pulled away from 0/1 toward a more conservative value. This effect is most pronounced in the three better-connected liberal blogs (lower left) versus the weakly connected conservative blogs (upper right).

For the second example, we use data obtained from the Osteoarthritis Initiative (OAI) database, which is available for public access at `http://www.oai.ucsf.edu/`. The question of interest for this data is investigate relationships between physical activity and knee disease symptoms. For this example we use a subset of the baseline clinical data, version 0.2.2. The data ensemble comprises of variables including biomarkers, knee joint symptoms, medical history, nutrition, physical exam and subject characteristics. In our subset we take an ensemble of size 120 for 4750 individuals. We hold out some of the biomarkers and knee joint symptoms and consider prediction accuracy of the ITF versus the joint DPM model. For the real variables, mixtures of normal kernels are considered, for the categorical, mixtures of multinomials and for the time series, mixtures of fixed finite wavelet basis expansion.

Results for this experiment are summarized in table 4.3 for 4 held-out variables. ITF outperforms the DPM in 3 of these 4 cases and marginally worse prediction accuracy in case of the other variable. It is also interesting to note that ITF helps to uncover useful

relationships between medical history, physical activity and knee disease symptoms, which has a potential application for clinical action and treatments for the subsequent patient visits.

4.7 Discussion

We have developed a general model to accommodate complex ensembles of data, along with a novel algorithm to sample from the posterior distributions arising from the model. Theoretically, extension to any number of levels of stick breaking processes should be possible, the utility and computational feasibility of such extensions is being studied. Also under investigation is connections with random graph/network models and theoretical rates of posterior convergence.

Table 4.1: Simulation Example, Scenario 1: Prediction error (top), tests of independence (bottom)

	ITF	DPM
T	1.79	1.43
C2	31%	23 %
C3	37%	36 %

	ITF	DPM	"Truth"
C1 vs T	Yes	Yes	Yes
C2 vs T	Yes	Yes	Yes
C3 vs T	Yes	Yes	Yes
C2 vs R	Yes	Yes	Yes

Table 4.2: Simulation Example, Scenario 2: Prediction error (top), tests of independence (bottom)

	ITF	DPM
T	4.61	10.82
C2	27%	55 %
C3	34%	57 %

	ITF	DPM	"Truth"
C1 vs T	Yes	Yes	Yes
C2 vs T	No	Yes	No
C3 vs T	No	Yes	No
C2 vs R	No	Yes	No

Table 4.3: OAI Data example: Relative Predictive Accuracy. The variables are respectively, left knee baseline pain, isometric strength left knee extension,left knee paired X ray reading, left knee baseline radiographic OA.

	ITF	DPM
P01BL12SXL	31.21	100.92
V00LEXWHY1	7.94	7.56 %
V00XRCHML	23.01	31.84 %
P01LXRKOA	65.78	90.30 %

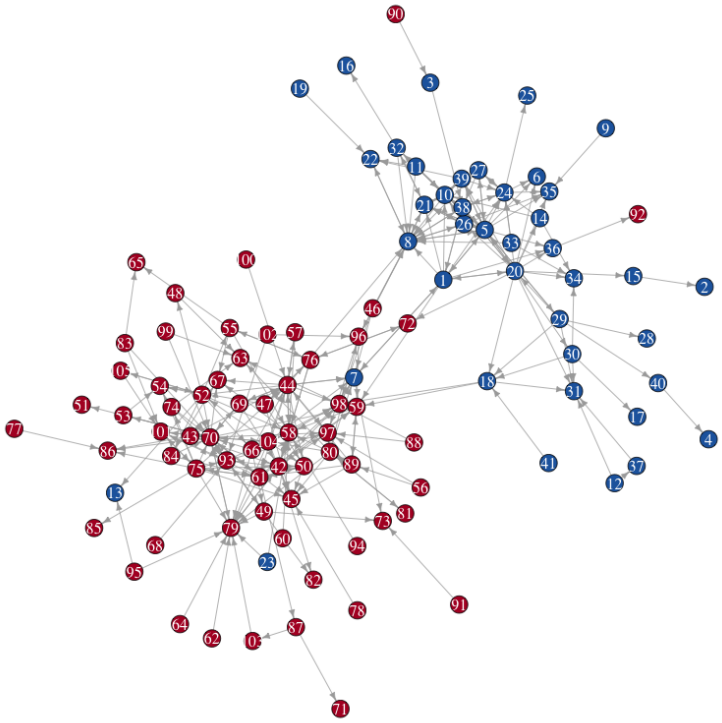

FIGURE 4.1: Network Example: True Clustering

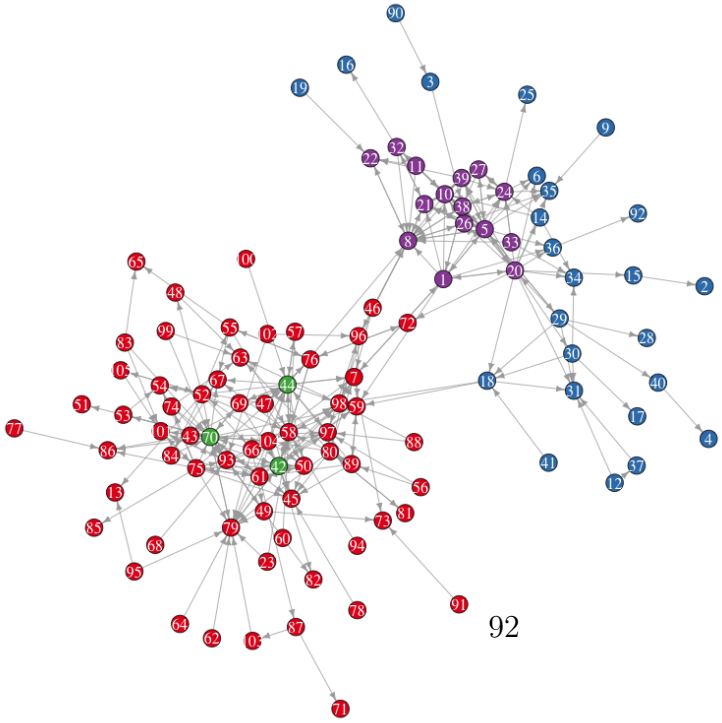

FIGURE 4.2: Network Example: Recovered Clustering

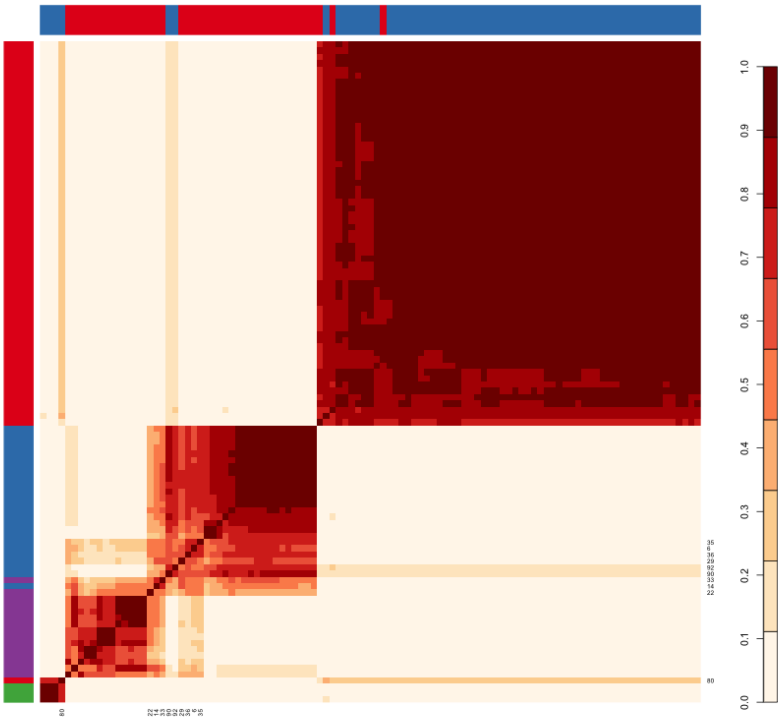

FIGURE 4.3: Network Example: Pairwise cluster assignment probability. Left bars correspond to clustering in Fig. 4.2, top bars correspond to clustering on the ideology label.

5

Applications in cancer studies: Studying genetic basis of lymphomas

5.1 Introduction

Two lymphomas, believed to characterized by unknown genetic factors, are Diffuse Large B Cell lymphoma (DLBCL) and Burkitt lymphoma. DLBCL is one of the most common forms of lymphomas in adults, while Burkitt lymphoma, a related kind of lymphoma, is less common, believed to be caused by the deregulation of the MYC gene. DLBCL is curable in about half the number of patients who have it, but has frequent relapses. Understanding some of the genetic factors associated with is important for development of novel targeted treatment. Gene expression profiles have shown that there exist substantial molecular differences between the two diseases, along with some other similarities (Dave et al., 2006; Hummel et al., 2006). The genetic basis of either of these diseases though, is largely unknown and has hitherto been unexplored. In this chapter, we describe statistical methods applied for a study which sequences tissues from patients affected with DLBCL and Burkitt's lymphoma and try to discover some of the underlying genetic associations. The sequencing involved in this study includes next generation sequencing of exomes from Burkitt lymphoma tumors and DLBCL tumors, including one instance of a whole exome

sequencing of a Burkitt lymphoma tumor. Other related data are collected and we try to answer some of the following questions by novel statistical analysis: (i) Can we estimate sample sizes necessary for identifying important mutations? (ii) Can we identify relations between gene expressions, molecular level information and mutations of the tumor tissues as well as mutations of normal tissues? (iii) What are genetic differences, if any between the two kinds of lymphomas? (iii) Can we meaningfully cluster the mutations? Some statistical challenges involved in these questions include jointly modeling different types of data (gene expression - real valued, mutations - categorical variables, etc.), identifying detection powers of common and rare variants as functions of sample sizes, performing high dimensional regression. The tissues for the study were obtained from patients of the Hematologic Malignancies Research Consortium (HMRC), anonymized and then shipped to Duke University, in accordance with the guidelines laid down by the Institutional Review Board at Duke University.

5.2 Description of the data

In case of the DLBCL, a total of 94 tumor tissues were sequenced via next generation sequencing. Of the 94 tissues, 73 were were obtained from sequencing exomes from DLBCL primary tumors. The remaining 21 are obtained from sequencing DLBCL cell lines. Amongst the 73 primary tumors, 34 matched normal tissues were also sequenced. By matched normals we mean sequencing of non-tumor tissue, obtained from the bone marrow of the patients suffering from lymphoma. These were collected by using Agilent reagents on the Illumina sequencing platforms. The accuracy of sequencing was further enhanced by verifying it against Sanger sequencing results. To further bolster detection of potential variants associated with lymphoma, we consider sequencing data from 256 additional normal tissue from healthy patients in recently published studies (Ng et al., 2009; Yi et al., 2010; Li et al., 2010) and additional data from dbSNP database (Sherry et al., 2001) and the 1000 genome project (Siva, 2008). In all, $121,589$ variants were identified in the DLBCL samples, including 23214 somatic mutations throughout the DLBCL genome. This

the largest of study of its kind in sequencing in an attempt to identify potential genetic associations with DLBCL.

In case of the Burkitt's lymphoma, a total of 59 tumor tissues were sequenced via next generation sequencing. Of the 59 tissues, 51 were were obtained from sequencing exomes from Burkitt primary tumors. The remaining 8 are obtained from sequencing Burkitt lymphoma cell lines. Amongst the 59 primary tumors, 14 matched normal tissues were also sequenced. In this case an additional 19 tissues were sequenced from patients unaffected with Burkitt lymphoma. These sequencing data, as in the DLBCL data, were collected by using Agilent reagents on the Illumina sequencing platforms. The accuracy of sequencing was further enhanced by verifying it against Sanger sequencing results. In the sequencing study, nearly all known mutations characterizing Burkitt lymphoma were identified, including mutation of MYC gene. As in the DLBCL case, we use additional data from Ng et al. (2009); Yi et al. (2010); Li et al. (2010); Sherry et al. (2001); Siva (2008) to augment the pool of variants from healthy patients.

Some justification of the sample sizes are considered in the following section on statistical methods, showing that the number of mutations are sufficient to capture any common variant, but that new rare variants will be discovered irrespective of the sample size, because of the inherent stochastic nature or endogenous processes and evolving genetic diversity.

5.3 Methods and results

5.3.1 Variant detection as a function of sample size

We want to estimate the number of new variants that will be observed for each additional sample. Suppose for the existing data with n samples (with $n = 94, 59$ for DLBCL and Burkitt's lymphoma) respectively, we have a combined pool of V variants. At any smaller sample size r, a simple estimate of the number of variants that would have been observed, is by considering the number of variants for each possible choice of r variants and then averaging these values over all the $\binom{n}{r}$ possible choices. The difficulty is that with $n = 94$

say, for many r, $\binom{n}{r}$ will be too large to enumerate all the samples. As an example, we can compute $\binom{94}{24} = O(10^{21})$, which is too large to be practically dealt with. An alternative is to consider sampling from these $\binom{n}{r}$ possible choices and obtain a bootstrap style point estimate. To better calibrate variability, we instead choose a Bayesian approach, placing a Poisson process prior on $v(r)$, the number of mutations observed at sample size r, such that prior has enforces stochastic ordering, ie, $v(r) < v(r-1)$ with 0 probability. We estimate the $v(r)$ and compute the differential increment at each sample size $d(r) = v(r) - v(r-1)$, for, $r = 2, 3, \ldots, 94$. We then use a non-parametric functional model for $\{d(r), r\}$, for predicting the additional number of mutations that would be discovered with each new sample. The model has excellent accuracy for RMSE using several hold out samples. The estimated $d(r)$ and the posterior credible intervals are presented in figure 5.1, while the total number of unique variants at each sample size, $v(r)$ is presented in figure 5.2. Overall it was discovered that a sample size of about 25 is enough to capture about 99% of variants, in both DLBCL and Burkitt's lyphoma. However if we exclude known common variants from the data and estimate the rates of discovery using the same procedure, the story changes completely. The rate of discovery of rare variants is found to be roughly linear with the sample size, even when increasing sample sizes upto 250, by considering sequencing data from the healthy patients in other studies. Most variants associated with cancer will be rare and therefore some will remain undiscovered, irrespective of the size of the study.

5.3.2 Association of variants with lymphoma, a new dependence measure

The discovery of new variants which are possibly associated with lymphoma is extremely difficult because of the rarity of the variants, as explained by the sample size calculations previously. A way of circumventing this issue is possibly to cluster the variants according to some parameters and then jointly look at each cluster's relationship with lymphoma status. In this way, if clusters contain known oncogenes (mutations of which are believed

to cause cancer), they maybe more likely to be associated with lymphoma and should be flagged accordingly, pointing towards a Bayesian statistical analysis of the problem, with the presence of prior information. It then becomes important to decide which parameters to choose, to cluster the genes and which method of clustering to use. The genetic parameters of interest in this case are gene size (ordinal scalars), background non synonymous mutation rates in normal samples (real scalars), somatically acquired variants (ordinal scalars), the rate of these events in carriers (ordinal vectors), among others. Each of these parameters, of different types, have their own marginal distribution. It is therefore a tailor-made situation for application of our joint modeling techniques from chapter 4 - we use the infinite tensor factorization priors in the analysis and the slice sampling algorithm as described.

One issue of concern is calibrating dependence between these variables. The mutual dependence estimator as given in equation (4.18) is not well behaved, particularly due to it being unbounded. We use a new measure, which is scaled version of mutual information, bivariate special case of which was described by Lu (2011). Let $X_i \in \mathcal{X}_i, i = 1, 2, \ldots, p$ be p random variables and let $H(X_i), I(X_i)$ represent the entropy and marginal mutual information of X_i. Then a scaled version of mutual information can be be given as,

$$I_s(X_1, X_2, \ldots, X_p) = \sqrt{\left[1 - exp\left(\frac{-2I(X_1, X_2, \ldots, X_p)}{1 - I(X_1, X_2, \ldots, X_p)/\max\{H(X_1), H(X_2), \ldots, H(X_p)\}}\right)\right]}$$

As in equation (4.18), this may be estimated in terms of the stick breaking weights and the cluster indices of the infinite tensor factor mixture model. In addition, I_s has the following nice properties, (i) is always between $[0, 1]$, (ii) is 1 only when there exists a pair X_i, X_j which such that one is a function of the other; (iii) is 0 only when all the X_i's are independent. It also reduces to the usual correlation coefficient in case of multivariate normal random variables. With this measure we characterize two genes to be in the same cluster if the dependence between their parameters, as estimated by I_s is outside the 95% posterior credible region. We use these clusters to come up with driver mutations possibly associated with lymphoma.

From the analysis, we identify 322 genes whose mutations have strong evidence in favor of their association with DLBCL and 70 genes whose mutations have strong evidence of being associated with Burkitt's lymphoma. Amongst the 322 genes discovered for DLBCL, are 12 known oncogenes, including ARIDIA, SETD2, CARD11, PIK3R1, PIK3CD. Of these only CARD11 was annotated earlier as possibly being associated with DLBCL. Further investigations were carried on with PIK3CD, whose alterations were experimentally validated as having relationship with DLBCL. Similarly for Burkitt's lymphoma certain known oncogenes, which were previously not known to have a relationship with Burkitt's lymphoma, were flagged by this analysis. ID3, flagged by our analysis, was experimentally validated presenting strong evidence that its mutations are associated with Burkitt's lymphoma.

Another issue addressed is the prediction of missing copy number variation based on known copy number variation of related genes. We use a nonparametric Gaussian process regression, but the sample size being very large here, we use the approximation techniques from chapter 2&3, successful predicting missing copy numbers and having high accuracy in hold-out samples.

5.3.3 Predicting overlaps with other studies

Three other studies were regarding DLBCL, on a much lower scale, were also conducted by other research groups which were published very recently (Lohr et al., 2012; Pasqualucci et al., 2011; Morin et al., 2011), flagging variants with potential association with DLBCL. It is of interest to consider the intersections of the genes reported by these variants with our own and whether these the numbers in these intersections are to expected or are too large or too low. We consider the model,

$$y_{ig} \sim Binomial(n_i, p_{ig}), \text{ for } i = 1, 2, 3, 4,$$

where y_{ig} is number of patients with mutation g in study i, n_i is the number of patients in study i, p_{ig} is the probability of mutation g in study i. Then, we would choose a hierarchical model for the p_{ig}'s, as,

$$p_{ig} \sim P_g,$$

with P_g the distribution of the mutation rate across studies (reflecting the differences in mutation rate across the different populations/ethnic groups represented in the different studies). Now the problem is that we have only 4 realizations from P_g for each gene, the mutations are so rare that most studies have no individuals with the variant and hence it is very difficult to learn the parameters in P_g from the data. We then use the following prior specification to overcome this,

$$p_{ig} \sim \beta(p_{0g}n_g, (1-p_{0g})n_g),$$

where p_{0g} = prior expectation for the rate of mutation in gene g, we fix this in advance at their expected value. Then, the n_g, prior sample size, controls the variability among studies. We use an empirical Bayes approach to learn about n_g.

In table 5.1 we present the actual and predicted overlaps between the studies. The actual numbers are not entirely unexpected, close to the ones predicted by our hierarchical model under fairly flexible assumptions.

Table 5.1: Table of actual and predicted two way overlaps between the recent studies of the DLBCL genome. In the row headers, $1, 2, 3, 4$ represent out study, and the studies Lohr et al. (2012); Pasqualucci et al. (2011); Morin et al. (2011) respectively. The predicted values are calculated from the model described in the text.

	Predicted	Actual
1 & 2	15	18
1 & 3	11	17
1 & 4	63	31
2 & 3	21	13
2 & 4	23	21
3 & 4	22	19

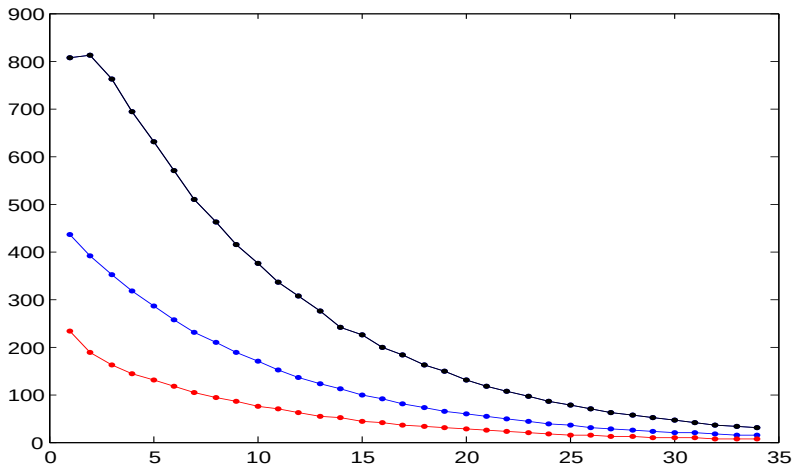

FIGURE 5.1: Sample size calculations: The figure represents estimate of the new number of variants $d(r)$ discovered for each sample size r along with its 95% credible interval, where blue gives the posterior median, black and red are the upper and lower credible limits respectively. The x-axis represents sample size, and has been truncated to a maximum 34, because the three lines essentially merge beyond this size. The y-axis represents additional number of variants being discovered, numbers to be read as $\times 10^2$.

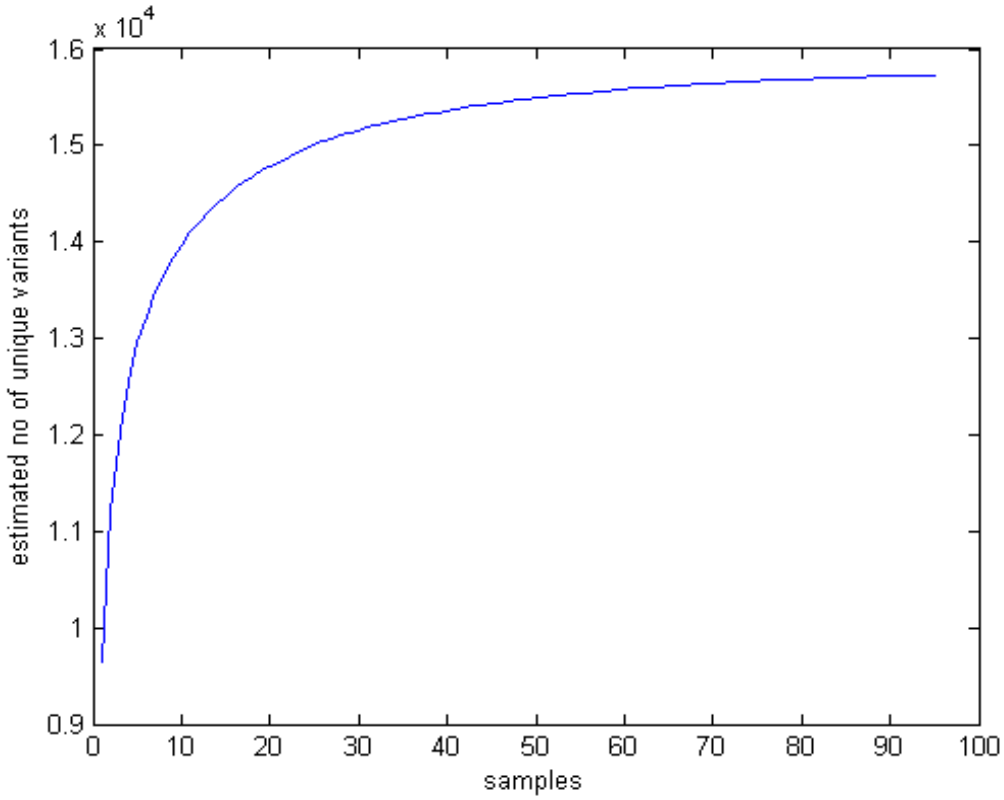

FIGURE 5.2: Sample size calculations: The figure represents estimate of the total number of unique variants $v(r)$ discovered for each sample size r.

6

Final remarks, current and future directions

6.1 Thesis summary

To summarize, we have dealt with two problems in this thesis. For the first problem, we have used random projections to come up with approximation methods for Bayesian nonparametric regression. This has several advantages, we are spared the need to learn a lower dimension and hence the extra computational and modeling effort for it. As shown with theory and examples in chapter 3, random projection can be done in several flavors. Instead of using vanilla random projection matrices, whose matrices are iid entries, we used more structured random matrices, which give us better computational speed and numerical stability. An important aspect is the parallelization of approximate matrix decomposition, with the recent focus of grid computing architecture, our blocked algorithm for large positive definite matrices can be used whenever an application needs inversion or pseudo-inversion of an almost rank deficient positive matrix. The other problem, we have dealt with in this thesis, is joint modeling of complex ensembles of objects. Once again, we have proposed a model which captures complex dependencies while retaining the flexibility in clustering for individual ensemble entities. In addition to this, we developed a novel sampling algorithm for simulating samples from the posterior, which circumvents

the need to need to finitely truncate a Dirichlet process, as is done in standard Dirichlet process mixture models. This saves us the trouble of having to keep track of many more weight parameters, which can get increasingly troublesome as the dimension of ensemble increases. The methods developed here to tackle each of the problems, large dimensional nonparametric Bayesian regression and joint modeling, are quite general and can be used off the box, without much modification or tuning for general scenarios. The cancer study is an interesting practical application of the methods presented, along with some other innovations, required by the idiosyncrasies of the problem. We now present some avenues for new research and development in each of the two contexts.

6.2 Further research for large nonparametric regression

6.2.1 A unifying framework for large n and large p regression

One of our current projects is to attempt to provide a framework which unifies regression via random projections for large n and compressed regression for large p. Consider the usual framework of learning about the response $y_i \in \mathcal{Y}, i = 1, 2, \ldots, n$, where n is the sample size from a set of p predictors, $x_i^j \in \mathcal{X}_j, i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, p$. We assume that there exists an unknown function $f : \mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_p \to \mathcal{Y}$, which we are trying to estimate. We are primarily interested in flexible prediction of unobserved response values given the predictors, via estimation of the unknown function. Consider the special case when there exists a sequence of functions, akin to a basis expansion, $\{\psi_k(\cdot)\} : \mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_p \to \mathcal{Y}$, such that $f(\cdot)$ can be expressed as a unique linear combination of these functions $f(x_1, x_2 \ldots, x_p) = \sum_k \psi_k \beta_k$ or in other words, $f(\cdot)$ has a unique representation in the span of $\{\psi_k\}_{k=1}^K$. We let $K \in \mathbb{N} \cup \{\infty\}$.

Most regression methods, ordinary least squares, functional and nonparametric, fall under this framework. Examples are simple linear regression, when we would take $K = p$ and $\psi_k(x_1, x_2, \ldots, c_p) = x_k, k \in \{1, 2, \ldots, K\}$. In case of Gaussian process regression, we can consider the Karhunen-Loéve expansion, when $\psi_k(\cdot)$ is the product of the k^{th} eigenfunction and eigenvalue, of the the covariance function of $f(\cdot)$. Note that, we do not

require explicit specification of the form of $\psi_k(\cdot)$, we just require the expansion for $f(\cdot)$ in its terms to be unique. The standard Bayesian approach is to go via prior specifications for β_k's. Once again, in case or ordinary linear regression, we specify priors for the regression coefficients. In case of Gaussian process regression, we specify that β_k's are iid Gaussian, which automatically leads to a Gaussian process specification for f. Inference procedures may or may not require the learning of coefficients β_k. In a Gaussian process, inference typically proceeds through learning the covariance function parameters, which are inherent in $\psi_k(\cdot)$ and not β_k's.

We have already explored different flavors of random projection methods for large n in chapters 2&3 in detail, in the Gaussian process setting. In large p settings, in case of ordinary linear regression, Zhou et al. (2007, 2009); Guhaniyogi and Dunson (2013) use random projections. Let Φ be a random projection matrix, as motivated in chapters 2&3. In case of Zhou et al. (2007) and Zhou et al. (2009), compressed regression is motivated for privacy preservation and they explore compressed regression via compression of the entire data space, by considering $\Phi y = \Phi X\beta + \epsilon$, instead of $y = X\beta + \epsilon$. Guhaniyogi and Dunson (2013) consider compressed regression as competitors of penalized methods for variable selection, via compression of the predictor space, by replacing the $n \times p$ predictor X with $X\Phi$ in $y = X\beta + \epsilon$, where Φ is an $p \times r$ with $r << p$. Both these approaches, compression of the entire data space and compression only of the predictor, show remarkable performance in real and simulated data settings.

We now attempt to describe a framework that encompasses the random projection approach for large n and large p under a single umbrella, having as its special cases compressed regression as in Zhou et al. (2009); Guhaniyogi and Dunson (2013) and projection approximation for Gaussian processes. For initial purposes, assume that K is finite. Let Φ be a $K \times r$ matrix with $r << K$ and let Ψ be a $n \times K$ matrix, with $\Psi(i,j) = \psi_j(x_i)$. Instead of Ψ, if we now consider $\Psi\Phi$, we get a random subspace projection. Alternatively, consider the functions defined as $\phi_j(x) = \sum_{i=1}^{K} \Phi(i,j)\psi_i(x)$ and consider finding the regression set-up where we are trying to estimate the unknown function in the span of $\{\phi_j(\cdot)\}_{j=1}^{r}$.

We may write this model equation as,

$$y_i = \sum_{j=1}^{r} \theta_j \phi_j(x_i) + \epsilon,$$

where θ_j's are the new regression coefficient in the random subspace. It is trivial to see that this setup encompasses the approach of Guhaniyogi and Dunson (2013) as a special case, which is for large p. To see that this also encompasses the Gaussian process approximations described in chapters 2&3 for large n, let $K = n$ and consider the Galerkin integral approximation as described in property (4) of §2.3.3 and §3.5.1. The covariance of $E(f|\Phi f)$ as in chapter 2 corresponds to a finite truncation of the Karhunen Loéve expansion of the Gaussian process $f(\cdot)$ and then find the eigenvectors and eigenvalues of its covariance function by using the Galerkin method with the matrix Φ. The approximate process corresponds to a stochastic process with eigenvalues and eigenvectors of its covariance as given by the Galerkin approximation. Letting our Φ be identical to the random matrix Φ of chapter 2, the Galerkin method corresponds to approximating eigenvalues and eigenvectors with the set $\{\{\phi_j(x_i)\}_{i=1}^n\}_{j=1}^r$. It is easy to see from this discussion that predictive processes and the knot based approaches also come under this framework, when Φ is a random permutation matrix. Both theoretical implications and general practical applications of this framework is under investigation.

Some flavors of current results are:

- If prior has large support on reduced space, then the implied prior has large support on the original space.
- Difference in posterior predictive expectation is at most $O(\frac{log(r/\delta)}{n})$ with probability $1 - \delta$
- Difference in posterior predictive density in terms of a class of $f - divergences$ is at most, $O(log(r^2\delta)/n\|\beta\|^2 max_i E(\|\psi(x)\|^2))$ with probability $1 - \delta$ etc..

We discuss more details in forthcoming submission.

6.2.2 1-bit compressive sensing

Another interesting direction in the context of compressed regression, via very recent developments called sign compression or 1-bit compressive sensing (Laska et al., 2011; Jacques et al., 2011). Consider the scenario when the response, y or the predictors x are binary. Using Φy or Φx, where the entries of Φ are real numbers (as when using a standard random projection or a structured random matrix) may not be the best idea. In the recent developments of 1-bit compressive sensing, one tries to make a signal recovery by using just $sgn(\Phi x)$, where x is the signal, Φ is a projection or measurement matrix and $sgn(\cdot)$ is the sign function, mapping to the binary set $\{1, -1\}$, as in Laska et al. (2011). Yan et al. (2012) uses similar ideas for outlier detection and classification problems. We are investigating use of 1-bit compressive in the context high dimensional Bayesian logistic regression.

6.3 Further research for Bayesian joint modeling

In the context of joint modeling we are currently considering estimators of scaled divergences. Póczos and Schneider (2012) consider nonparametric estimation of conditional information and divergences. In our experience, conditional information can be misleading when not scaled appropriately, so we are investigating scaled estimators for I_s as presented in chapter 5 on the lines of Póczos and Schneider (2012). The other way, which is more appealing from a Bayesian context, is to place priors on the mutual information or its scaled versions directly, instead of approaching its estimation via a joint model. We are exploring appropriate prior distributions that can be used in this context.

Another interesting direction we are pursuing in this regard, related to our methods for high dimensional nonparametric regression, is estimation of tensor factorizations via random projections. Instead of considering a random projection matrix $(\phi(i, j)))$, we may as well consider a random projection tensor $\phi(i, j, k)$. Some similar ideas in the context of non-negative tensor factorization and sparse tensor factorization have been pursued by Lim and Comon (2010).

As a final remark, with increasingly complex data collection being the order of the day, it is an exciting time to be a statistician. I end here with my favorite quote,

"A day full of possibilities! It's a magical world, Hobbes, ol' buddy ... let's go exploring!"

- in Calvin & Hobbes by Bill Watterson.

Appendix A

Example of inversion with the Woodbury matrix identity

Either of the algorithms, 1 or 2, in this chapter 2 yield $Q^{RP}_{f,f} = UD^2U^T$, with $U^TU = I$. We would be interested in calculating $\Sigma_1^{-1} = (Q^{RP}_{f,f} + \sigma^2 I)^{-1}$, in the marginalized form for inference or prediction. Using the Woodbury matrix identity (Harville, 2008) we have,

$$\begin{aligned}\Sigma_1^{-1} &= \sigma^{-2}I - \sigma^{-2}U(D^{-2} + \sigma^{-2}U^TU)^{-1}U^T\sigma^{-2} \\ &= \sigma^{-2}I - \sigma^{-4}U(D^{-2} + \sigma^{-2}I)^{-1}U^T\end{aligned}$$

In the above $D^{-2} + \sigma^{-2}I$ is a diagonal matrix whose inverse can be obtained by just taking reciprocals of the diagonals. Thus direct matrix inversion is entirely avoided with the decomposition available from the algorithms.

In a similar vein we can utilize the QR decomposition and the small R^{-1}, which is itself evaluated through backward subsitution and not direct inversion, in chapter 3, for computing a the matrix inverse for prediction in case of a GP, using the similar formulation as above with the Woodbury identity.

Bibliography

Achlioptas, D. (2003), "Database-friendly random projections: Johnson-Lindenstrauss with binary coins," *Journal of Computer and System Sciences*, 66, 671–687.

Adamic, L. and Glance, N. (2005), "The political blogosphere and the 2004 US election: divided they blog," in *Proceedings of the 3rd international workshop on Link discovery*, pp. 36–43, ACM.

Adler, R. J. (1990), "An introduction to continuity, extrema, and related topics for general Gaussian processes," *IMS Lecture Notes-Monograph Series*, 12, 75–76.

Agullo, E., Coti, C., Dongarra, J., Herault, T., and Langem, J. (2010), "QR factorization of tall and skinny matrices in a grid computing environment," in *Parallel & Distributed Processing (IPDPS), 2010 IEEE International Symposium on*, pp. 1–11, IEEE.

Airoldi, E. M., Blei, D. M., Fienberg, S. E., Xing, E. P., and Jaakkola, T. (2006), "Mixed membership stochastic block models for relational data with application to protein-protein interactions," in *Proceedings of the international biometrics society annual meeting*.

Arriaga, R. I. and Vempala, S. (2006), "An algorithmic theory of learning: Robust concepts and random projection," *Machine Learning*, 63, 161–182.

Avron, H., Maymounkov, P., and Toledo, S. (2010), "Blendenpik: Supercharging LAPACK's least-squares solver," *SIAM Journal on Scientific Computing*, 32, 1217–1236.

Baker, C. T. and Baker, C. (1977), *The numerical treatment of integral equations*, vol. 13, Clarendon press Oxford.

Banerjee, A., Murray, J., and Dunson, D. B. (2013a), "Bayesian learning of joint distributions of objects," in *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics (AISTATS), Scottsdale, AZ, USA*.

Banerjee, A., Dunson, D. B., and Tokdar, S. T. (2013b), "Efficient Gaussian process regression for large datasets," *Biometrika*, 100, 75–89.

Banerjee, S., Carlin, B. P., and Gelfand, A. E. (2004), *Hierarchical modeling and analysis for spatial data*, Chapman and Hall.

Banerjee, S., Gelfand, A. E., Finley, A. O., and Sang, H. (2008), "Gaussian predictive process models for large spatial data sets," *J. R. Statist. Soc. B*, 70, 825–848.

Bhatia, R. (1997), *Matrix analysis*, Springer Verlag, New York.

Bhattacharya, A. and Dunson, D. B. (2012), "Simplex factor models for multivariate unordered categorical data," *Journal of the American Statistical Association*, 107, 362–377.

Bigelow, J. and Dunson, D. (2009), "Bayesian semiparametric joint models for functional predictors," *Journal of the American Statistical Association*, 104, 26–36.

Blackford, L. S., Choi, J., Cleary, A., DAzevedo, E., Demmel, J., Dhillon, I., Dongarra, J., Hammarling, S., Henry, G., Petitet, A., et al. (1997), *ScaLAPACK user's guide*, vol. 3, Siam Philadelphia.

Boutsidis, C. and Gittens, A. (2012), "Improved matrix algorithms via the subsampled randomozied Hadamard transform," *arxiv.org*.

Box, G. E. and Tiao, G. C. (1973), "Bayesian inference in statistical analysis," Tech. rep., DTIC Document.

Cai, J., Song, X., Lam, K., and Ip, E. (2011), "A mixture of generalized latent variable models for mixed mode and heterogeneous data," *Computational Statistics & Data Analysis*, 55, 2889–2907.

Candès, E. J., Romberg, J., and Tao, T. (2006), "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Info. Theory*, 52, 489–509.

Chipman, H. A., George, E. I., and McCulloch, R. E. (2010), "BART: Bayesian additive regression trees," *The Annals of Applied Statistics*, 4, 266–298.

Choi, J., Walker, D. W., and Dongarra, J. J. (1994), "PUMMA: Parallel universal matrix multiplication algorithms on distributed memory concurrent computers," *Concurrency: Practice and Experience*, 6, 543–570.

Constantine, P. G. and Gleich, D. F. (2011), "Tall and skinny QR factorizations in MapReduce architectures," in *Proceedings of the second international workshop on MapReduce and its applications*, pp. 43–50, ACM.

Cox, A. J. and Higham, N. J. (1997), "Stability of Householder QR factorization for weighted least squares problems," *Numerical Analysis*, pp. 57–73.

Cressie, N. (1992), "Statistics for spatial data," *Terra Nova*, 4, 613–617.

Csató, L. and Opper, M. (2002), "Sparse on-line Gaussian processes," *Neural Computation*, 14, 641–668.

Dasgupta, S. and Gupta, A. (2003), "An elementary proof of a theorem of Johnson and Lindenstrauss," *Random Structures and Algorithms*, 22, 60–65.

Dave, S. S., Fu, K., Wright, G. W., Lam, L. T., Kluin, P., Boerma, E.-J., Greiner, T. C., Weisenburger, D. D., Rosenwald, A., Ott, G., et al. (2006), "Molecular diagnosis of Burkitt's lymphoma," *New England Journal of Medicine*, 354, 2431–2442.

Delves, L. M. and Mohamed, J. (1988), *Computational methods for integral equations*, Cambridge University Press.

Dixon, J. D. (1983), "Estimating extremal eigenvalues and condition numbers of matrices," *SIAM J. Numerical Anal.*, 20, 812–814.

Doković, D. (1991), "Unitary similarity of projectors," *Aequationes Mathematicae*, 42, 220–224.

Donoho, D. L. (2006), "Compressed sensing," *IEEE Trans. Info. Theory*, 52, 1289–1306.

Draper, N. R., Smith, H., and Pownell, E. (1966), *Applied regression analysis*, vol. 3, Wiley New York.

Drineas, P. and Mahoney, M. W. (2005), "On the Nyström method for approximating a Gram matrix for improved kernel-based learning," *J. Mach. Learn. Res.*, 6, 2153–2175.

Drineas, P., Mahoney, M. W., Muthukrishnan, S., and Sarlós, T. (2011), "Faster least squares approximation," *Numerische Mathematik*, 117, 219–249.

Dunson, D. (2000), "Bayesian latent variable models for clustered mixed outcomes," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62, 355–366.

Dunson, D. (2003), "Dynamic latent trait models for multidimensional longitudinal data," *Journal of the American Statistical Association*, 98, 555–563.

Dunson, D. (2009), "Nonparametric Bayes local partition models for random effects," *Biometrika*, 96, 249–262.

Dunson, D. (2010), "Multivariate kernel partition process mixtures," *Statistica Sinica*, 20, 1395.

Dunson, D. and Bhattacharya, A. (2010), "Nonparametric Bayes regression and classification through mixtures of product kernels," *Bayesian Stats.*

Dunson, D. and Xing, C. (2009), "Nonparametric Bayes modeling of multivariate categorical data," *Journal of the American Statistical Association*, 104, 1042–1051.

Dunson, D. B., Xue, Y., and Carin, L. (2008), "The matrix stick-breaking process," *Journal of the American Statistical Association*, 103, 317–327.

Ferguson, T. (1973), "A Bayesian analysis of some nonparametric problems," *The annals of statistics*, pp. 209–230.

Finley, A. O., Sang, H., Banerjee, S., and Gelfand, A. E. (2009), "Improving the performance of predictive process modeling for large datasets," *Comp. Statist. Data Anal.*, 53, 2873–2884.

Foster, L., Waagen, A., Aijaz, N., Hurley, M., Luis, A., Rinsky, J., Satyavolu, C., Way, M. J., Gazis, P., and Srivastava, A. (2009), "Stable and efficient Gaussian process calculations," *J. Mach. Learn. Res.*, 10, 857–882.

Frank, A. and Asuncion, A. (2010), *UCI Machine Learning Repository*, School of Information and Computer Sciences, University of California, Irvine, California.

Frauenfelder, P., Schwab, C., and Todor, R. A. (2005), "Finite elements for elliptic problems with stochastic coefficients," *Computer Meth. Appl. Mechanics and Engineering*, 194, 205–228.

Friedman, J., Hastie, T., and Tibshirani, R. (2001), *The elements of statistical learning*, vol. 1, Springer Series in Statistics.

George, E. and McCulloch, R. (1996), "Stochastic search variable selection," *Markov chain Monte Carlo in practice*, pp. 203–214.

Ghanem, R. and Spanos, P. D. (2003), *Stochastic Finite Elements: A Spectral Approach*, Dover Publications, New York, revised edn.

Golightly, A. and Wilkinson, D. J. (2006), "Bayesian sequential inference for nonlinear multivariate diffusions," *Statist. Comp.*, 16, 323–338.

Grigoriu, M. (2002), *Stochastic Calculus: Applications in Science and Engineering*, Birkhauser, Boston, illustrated edn.

Gross, J. L. and Tucker, T. W. (2001), *Topological graph theory*, Dover Publications.

Guhaniyogi, R. and Dunson, D. B. (2013), "Bayesian Compressed Regression," *arXiv preprint arXiv:1303.0642*.

Halko, N., Martinsson, P. G., and Tropp, J. A. (2011), "Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions," *SIAM Rev.*, 53, 217–288.

Hannah, L. A., Blei, D. M., and Powell, W. B. (2011), "Dirichlet Process Mixtures of Generalized Linear Models," *The Journal of Machine Learning Research*, 12, 1923–1953.

Harville, D. A. (2008), *Matrix algebra from a statistician's perspective*, Springer Verlag, New York, reprint edn.

Higdon, D. (2002), "Space and space-time modeling using process convolutions," *Quantitative methods for current environmental issues*, pp. 37–56.

Huber, P. J., Huber, P., Huber, P., Statisticien, M., Suisse, E. U., Huber, P., and Statistician, M. (1996), *Robust statistical procedures*, vol. 68, SIAM.

Hummel, M., Bentink, S., Berger, H., Klapper, W., Wessendorf, S., Barth, T. F., Bernd, H.-W., Cogliatti, S. B., Dierlamm, J., Feller, A. C., et al. (2006), "A biologic definition of Burkitt's lymphoma from transcriptional and genomic profiling," *New England Journal of Medicine*, 354, 2419–2430.

Jacques, L., Laska, J. N., Boufounos, P. T., and Baraniuk, R. G. (2011), "Robust 1-bit compressive sensing via binary stable embeddings of sparse vectors," *arXiv preprint arXiv:1104.3160*.

Johnson, W. B. and Lindenstrauss, J. (1984), "Extensions of Lipschitz mappings into a Hilbert space," *Contemporary mathematics*, 26, 1.

Johnson, W. B., Lindenstrauss, J., and Schechtman, G. (1986), "Extensions of Lipschitz maps into Banach spaces," *Israel J. Math.*, 54, 129–138.

Johnstone, I. M. and Titterington, D. M. (2009), "Statistical challenges of high-dimensional data," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 367, 4237–4253.

Kammann, E. E. and Wand, M. P. (2003), "Geoadditive models," *J. R. Statist. Soc. C*, 52, 1–18.

Keerthi, S. and Chu, W. (2006), "A matching pursuit approach to sparse Gaussian process regression," *Advances in Neural Information Processing Systems*, 18, 643–650.

Kühn, T. (1987), "Eigenvalues of integral operators generated by positive definite Hölder continuous kernels on metric compacta," in *Indagationes Mathematicae (Proceedings)*, vol. 90, pp. 51–61, Elsevier.

Laska, J. N., Wen, Z., Yin, W., and Baraniuk, R. G. (2011), "Trust, but verify: Fast and accurate signal recovery from 1-bit compressive measurements," *Signal Processing, IEEE Transactions on*, 59, 5289–5301.

Lee, J. A. and Verleysen, M. (2007), *Nonlinear dimensionality reduction*, Springer.

Li, L., Zhou, M., Wang, E., and Carin, L. (2011), "Joint dictionary learning and topic modeling for image clustering," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pp. 2168–2171, IEEE.

Li, Y., Vinckenbosch, N., Tian, G., Huerta-Sanchez, E., Jiang, T., Jiang, H., Albrechtsen, A., Andersen, G., Cao, H., Korneliussen, T., et al. (2010), "Resequencing of 200 human exomes identifies an excess of low-frequency non-synonymous coding variants," *Nature genetics*, 42, 969–972.

Lim, L.-H. and Comon, P. (2010), "Multiarray signal processing: Tensor decomposition meets compressed sensing," *Comptes Rendus Mécanique*, 338, 311–320.

Lohr, J. G., Stojanov, P., Lawrence, M. S., Auclair, D., Chapuy, B., Sougnez, C., Cruz-Gordillo, P., Knoechel, B., Asmann, Y. W., Slager, S. L., et al. (2012), "Discovery and prioritization of somatic mutations in diffuse large B-cell lymphoma (DLBCL) by whole-exome sequencing," *Proceedings of the National Academy of Sciences*, 109, 3879–3884.

Love, C., Sun, Z., Jima, D., Li, G., Zhang, J., Miles, R., Richards, K. L., Dunphy, C. H., Choi, W. W., Srivastava, G., Banerjee, A., et al. (2012), "The genetic landscape of mutations in Burkitt lymphoma," *Nature genetics*, 44, 1321–1325.

Lu, S. (2011), "Measuring Dependence Via Mutual Information," .

Mattson, T. G., Sanders, B. A., and Massingill, B. (2005), *Patterns for parallel programming*, Addison-Wesley Professional.

Morin, R. D., Mendez-Lago, M., Mungall, A. J., Goya, R., Mungall, K. L., Corbett, R. D., Johnson, N. A., Severson, T. M., Chiu, R., Field, M., et al. (2011), "Frequent mutation of histone-modifying genes in non-Hodgkin lymphoma," *Nature*, 476, 298–303.

Muthen, B. (1984), "A general structural equation model with dichotomous, ordered categorical, and continuous latent variable indicators," *Psychometrika*, 49, 115–132.

Ng, S. B., Turner, E. H., Robertson, P. D., Flygare, S. D., Bigham, A. W., Lee, C., Shaffer, T., Wong, M., Bhattacharjee, A., Eichler, E. E., et al. (2009), "Targeted capture and massively parallel sequencing of 12 human exomes," *Nature*, 461, 272–276.

Papaspiliopoulos, O. and Roberts, G. (2008), "Retrospective Markov chain Monte Carlo methods for Dirichlet process hierarchical models," *Biometrika*, 95, 169–186.

Pasqualucci, L., Trifonov, V., Fabbri, G., Ma, J., Rossi, D., Chiarenza, A., Wells, V. A., Grunn, A., Messina, M., Elliot, O., et al. (2011), "Analysis of the coding genome of diffuse large B-cell lymphoma," *Nature genetics*, 43, 830–837.

Petrone, S., Guindani, M., and Gelfand, A. E. (2009), "Hybrid Dirichlet mixture models for functional data," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71, 755–782.

Pitman, J. and Yor, M. (1997), "The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator," *The Annals of Probability*, 25, 855–900.

Plummer, M., Best, N., Cowles, K., and Vines, K. (2006), "CODA: Convergence diagnosis and output analysis for MCMC," *R News*, 6, 7–11.

Póczos, B. and Schneider, J. (2012), "Nonparametric estimation of conditional information and divergences," in *15th International Conference on Artificial Intelligence and Statistics*, vol. 22, pp. 914–923.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (2007), *Numerical recipes 3rd edition: The art of scientific computing*, Cambridge University Press.

Quinonero Candela, J. and Rasmussen, C. E. (2005), "A Unifying View of Sparse Approximate Gaussian Process Regression," *J. Mach. Learn. Res.*, 6, 1939–1959.

Rasmussen, C. E. (2004), "Gaussian processes in machine learning," *Advanced Lectures on Machine Learning*, pp. 63–71.

Sammel, M., Ryan, L., and Legler, J. (1997), "Latent variable models for mixed discrete and continuous outcomes," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59, 667–678.

Sarlos, T. (2006), "Improved approximation algorithms for large matrices via random projections," *47th Annual IEEE Symposium on Foundations of Computer Science*, pp. 143–152.

Schwab, C. and Todor, R. A. (2006), "Karhunen–Loève approximation of random fields by generalized fast multipole methods," *Journal of Computational Physics*, 217, 100–122.

Schwaighofer, A. and Tresp, V. (2002), "Transductive and inductive methods for approximate Gaussian process regression," *Advances in Neural Information Processing Systems*, 15, 953–960.

Seeger, M. (2004), "Gaussian processes for machine learning," *International Journal of Neural Systems*, 14, 69–106.

Seeger, M., Williams, C. K. I., and Lawrence, N. D. (2003), "Fast forward selection to speed up sparse Gaussian process regression," *Workshop on AI and Statistics*, 9, 2003–2010.

Sethuraman, J. (1994), "A constructive definition of Dirichlet priors," *Statistica Sinica*, 4, 639–650.

Shahbaba, B. and Neal, R. (2009), "Nonlinear models using Dirichlet process mixtures," *The Journal of Machine Learning Research*, 10, 1829–1850.

Shen, W. and Ghosal, S. (2011), "Adaptive Bayesian multivariate density estimation with Dirichlet mixtures," *Arxiv preprint arXiv:1109.6406*.

Sherry, S., Ward, M.-H., Kholodov, M., Baker, J., Phan, L., Smigielski, E., and Sirotkin, K. (2001), "dbSNP: the NCBI database of genetic variation," *Nucleic acids research*, 29, 308–311.

Siva, N. (2008), "1000 Genomes project," *Nature biotechnology*, 26, 256–256.

Smola, A. J. and Bartlett, P. (2001), "Sparse greedy Gaussian process regression," *Advances in Neural Information Processing Systems 13*, pp. 619–625.

Snelson, E. and Ghahramani, Z. (2006), "Sparse Gaussian processes using pseudo-inputs," *Advances in Neural Information Processing Systems*, 18, 1257–1264.

Song, X., Xia, Y., and Lee, S. (2009), "Bayesian semiparametric analysis of structural equation models with mixed continuous and unordered categorical variables," *Statistics in medicine*, 28, 2253–2276.

Stein, M. (1999), *Interpolation of Spatial Data: Some Theory for Kriging*, Springer Verlag, New York, illustrated edn.

Stewart, G. W. (1993), "On the early history of the singular value decomposition," *SIAM review*, 35, 551–566.

Stoughton, C., Lupton, R. H., Bernardi, M., Blanton, M. R., Burles, S., Castander, F. J., Connolly, A. J., Eisenstein, D. J., Frieman, J. A., Hennessy, G. S., et al. (2007), "Sloan digital sky survey: early data release," *The Astronomical Journal*, 123, 485.

Todor, R. A. (2006), "Robust eigenvalue computation for smoothing operators," *SIAM journal on numerical analysis*, 44, 865–878.

Tokdar, S. (2011a), "Adaptive Convergence Rates of a Dirichlet Process Mixture of Multivariate Normals," *Arxiv preprint arXiv:1111.4148*.

Tokdar, S. T. (2007), "Towards a faster implementation of density estimation with logistic Gaussian process priors," *J. Comp. Graph. Statist.*, 16, 633–655.

Tokdar, S. T. (2011b), "Adaptive Gaussian Predictive Process Approximation," *Duke Statistical Science Discussion Paper*, 11-13.

Tokdar, S. T., Zhu, Y. M., and Ghosh, J. K. (2010), "Bayesian density regression with logistic Gaussian process and subspace projection," *Bayesian analysis*, 5, 319–344.

Trefethen, L. N. and Bau III, D. (1997), *Numerical linear algebra*, no. 50, Society for Industrial Mathematics.

Tropp, J. A. (2011), "Improved analysis of the subsampled randomized Hadamard transform," *Advances in Adaptive Data Analysis*, 3, 115–126.

Van Dyk, D. and Park, T. (2008), "Partially Collapsed Gibbs Samplers," *Journal of the American Statistical Association*, 103, 790–796.

Walker, S. (2007), "Sampling the Dirichlet mixture model with slices," *Communications in StatisticsSimulation and Computation®*, 36, 45–54.

West, M. (2003), "Bayesian factor regression models in the large p, small n paradigm," *Bayesian statistics*, 7, 723–732.

Wikle, C. K. and Cressie, N. (1999), "A dimension-reduced approach to space-time Kalman filtering," *Biometrika*, 86, 815.

Williams, C. K. I. and Rasmussen, C. E. (1996), "Gaussian Processes for Regression," pp. 514–520.

Woolfe, F., Liberty, E., Rokhlin, V., and Tygert, M. (2008), "A fast randomized algorithm for the approximation of matrices," *Applied and Computational Harmonic Analysis*, 25, 335–366.

Xia, G. and Gelfand, A. E. (2006), "Stationary process approximation for the analysis of large spatial datasets," Tech. rep., Citeseer.

Yan, M., Yang, Y., and Osher, S. (2012), "Robust 1-bit compressive sensing using adaptive outlier pursuit," *Signal Processing, IEEE Transactions on*, 60, 3868–3875.

Yang, M. and Dunson, D. (2010), "Bayesian semiparametric structural equation models with latent variables," *Psychometrika*, 75, 675–693.

Yi, X., Liang, Y., Huerta-Sanchez, E., Jin, X., Cuo, Z. X. P., Pool, J. E., Xu, X., Jiang, H., Vinckenbosch, N., Korneliussen, T. S., et al. (2010), "Sequencing of 50 human exomes reveals adaptation to high altitude," *Science Signalling*, 329, 75.

Zhang, J., Grubor, V., Love, C. L., Banerjee, A., Richards, K. L., Mieczkowski, P. A., Dunphy, C., Choi, W., Au, W. Y., Srivastava, G., et al. (2013), "Genetic heterogeneity of diffuse large B-cell lymphoma," *Proceedings of the National Academy of Sciences*, 110, 1398–1403.

Zhou, S., Lafferty, J., and Wasserman, L. (2007), "Compressed regression," *arXiv preprint arXiv:0706.0534*.

Zhou, S., Lafferty, J., and Wasserman, L. (2009), "Compressed and privacy-sensitive sparse regression," *Information Theory, IEEE Transactions on*, 55, 846–866.

Biography

Anjishnu Banerjee was born in Asansol, West Bengal, India on March 1st, 1986. He completed his bachelor degree with honors from Indian Statistical Institute, Kolkata, India in 2007 and continued there for a masters degree, specializing in Mathematical Statistics and Probability, in 2009. He then came to Duke University in Durham, NC, United States, pursuing a PhD degree in the department of Statistical Science, advised by Professor David B. Dunson. He earned an MS in Statistical Science *en route* to his PhD in 2012. His publications during the PhD program include Banerjee et al. (2013b,a); Zhang et al. (2013); Love et al. (2012). He received the notable paper award for his publication in the proceedings of 16th International Conference on Artificial Intelligence and Statistics (AISTATS) 2013, Scottsdale, AZ, USA. After completion of his PhD, he is slated to join Amazon LLC, as a research scientist.