Graphs and Combinatorial Representations of Stochastic Processes

by

Daniel Shaw Ting

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Statistics

and the Designated Emphasis

in

Communication, Computation, and Statistics

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Michael I. Jordan, Chair
Professor Pieter Abbeel
Professor Peter Bickel
Professor Martin Wainwright

Fall 2012

Graphs and Combinatorial Representations of Stochastic Processes

Abstract

Graphs and Combinatorial Representations of Stochastic Processes

by

Daniel Shaw Ting

Doctor of Philosophy in Statistics

with the Designated Emphasis in

Communication, Computation, and Statistics
University of California, Berkeley

Professor Michael I. Jordan, Chair

This thesis covers two distinct topics connected by their use of graphs. First is a theoretical analysis of graph Laplacians and locally linear embedding (LLE) on manifolds using tools for diffusion processes. The implications of this analysis are (1) a better understanding of the relationship between graph Laplacians and LLE, (2) understanding how a graph construction method affects the limit operator, and (3) obtaining a graph has nice properties such as sparsity or a well-behaved spectrum given a desired limit.

In the second topic we examine random graphs and their relationship to nonparametric Bayesian methods. We give combinatorial processes describing several nonparametric hierarchical Bayesian models. These processes lead to the development of new MCMC samplers and provide a new perspective on the models. We introduce the idea of discrete coagulation and fragmentation processes to describe various hierarchical models and identify a particular model of interest using coagulation-fragmentation duality. We consider these random graphs in the more general context of random combinatorial objects and give an application of random trees to drawing a random sample without replacement from a distributed stream.

To mom, dad, family and friends, and my advisor Mike Jordan

dedication txt

Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank all those that have helped guide me through these last few years of research and study. Because of all the help, knowledge, and skills I have received from others, I can look forward to applying that knowledge for the rest of my life and to applying the skills gained for exploring new, interesting ideas.

To my advisor, Michael Jordan, thank you for the always on point advice, whether it be on academic, career, or life, as well as the freedom to explore a multitude of topics. I thank Steve Fienberg for his guidance and encouragement during my masters at CMU and Larry Wasserman for developing my initial interest in statistics as an undergraduate. To Ling Huang, thanks for the guidance, support, and being a patient sounding board for ideas. To my committee, Martin Wainwright, Peter Bickel, and Pieter Abbeel, thank you for the time digging through a dense thesis. And last but not least, I thank my family and friends for their love and support and my dad, who nurtured my interest in mathematics.

Chapter 1

Introduction

This dissertation presents a few topics using graphs in statistics and machine learning. The applications of graphs may be broken down abstractly into three categories:

1. Graph Laplacians and an analysis of their asymptotic properties,
2. Representations of combinatorial objects and their applications to nonparametric hierarchical Bayesian models, and
3. Exploiting graphical structure in general Markov Chain Monte Carlo procedures.

For all three categories, the goals are two-fold. First is to develop theory or insights which can present a unified perspective of multiple existing methods. This allows one to relate as well as distinguish amongst the different methods. Second is to apply these insights to develop new methods or models that improve upon existing ones.

The first category contributes to understanding manifold learning methods as well as in empirically constructed smoothness penalties for use in semi-supervised learning. Specifically, we examine how the choice of graph construction method affects the limiting graph Laplacian. We import theory for the approximation of diffusion processes to allow us to (1) analyze both Laplacian based methods and local linear embedding (LLE), (2) identify and sometimes correct for deficiencies in methods, and (3) define a graph construction method that can empirically construct a desired first order smoothness functional and how to do it in a way that gives attractive theoretical or computational properties. From a technical perspective, the contributions are a method for analyzing kNN graphs and other non-smooth kernels.

The second category focuses on finite combinatorial representations for stick-breaking processes used in nonparametric Bayesian mixture modeling when the data itself is finite. The contributions to this area are two-fold. The first contribution is in computation for MCMC methods. Using representations of mixture models with random forests, we develop novel samplers for nonparametric Bayesian models like the Dirichlet process (DP) mixture model or the hierarchical Dirichlet process (HDP) mixture model which outperform existing samplers. Developing samplers for other models such as the nested Dirichlet process (nDP) and tree stick-breaking process is straightforward using the representation. The second contribution is that the combinatorial representations present another perspective and interesting insights for many hierarchical Bayesian models. In particular, the hierarchical structure of many nonparametric Bayesian models may be codified by coagulation and fragmentation operations. This aids in understanding the relationship among the different hierarchical nonparametric Bayesian models as well as suggesting a new hierarchical model where the marginal distributions at every level form a Pitman-Yor process. We also examine some side problems in random permutations and sampling and show how the combinatorial representation yields a novel reservoir sampling algorithm for computing a random sample without replacement in a map-reduce or distributed setting.

The third category introduces the idea of using augmenting a Markov chain on a single variable to a Markov chain where the states are themselves are graphs. This ideas is used to propose some beneficial modifications to the new MCMC algorithms developed in the second category.

Chapter 2

Graph Constructions and Asymptotics of the Graph Laplacian

Graph Laplacians have become a core technology in machine learning. They have appeared in clustering (Kannan et al., 2004, von Luxburg et al., 2008), dimensionality reduction (Belkin and Niyogi, 2003, Nadler et al., 2006), and semi-supervised learning (Belkin and Niyogi, 2004, Zhu et al., 2003).

While graph Laplacians are but one member of a broad class of methods that use local neighborhood graphs to model data lying on a low-dimensional manifold embedded in a high-dimensional space, they are distinguished by their appealing mathematical properties, notably: (1) the graph Laplacian is the infinitesimal generator for a random walk on the graph, and (2) it is a discrete approximation to a weighted Laplace-Beltrami operator on a manifold, an operator which has numerous geometric properties and induces a smoothness functional. These mathematical properties have served as a foundation for the development of a growing theoretical literature that has analyzed learning procedures based on the graph Laplacian. To review briefly, Bousquet et al. (2003) proved an early result for the convergence of the unnormalized graph Laplacian to a regularization functional that depends on the squared density p^2. Belkin and Niyogi (2005) demonstrated the pointwise convergence of the empirical unnormalized Laplacian to the Laplace-Beltrami operator on a compact manifold with uniform density. Lafon (2004) and Nadler et al. (2006) established a connection between graph Laplacians and the infinitesimal generator of a diffusion process. They further showed that one may use the degree operator to control the effect of the density. Hein et al. (2005) combined and generalized these results for weak and pointwise (strong) convergence under weaker assumptions as well as providing rates for the unnormalized, normalized, and random walk Laplacians. They also make explicit the connections to the weighted Laplace-Beltrami operator. Singer (2006) obtained improved convergence rates for a uniform density. Giné and Koltchinskii (2005) established a uniform convergence result and functional central limit theorem to extend the pointwise convergence results. von Luxburg et al. (2008) and Belkin and Niyogi (2006) presented spectral convergence results for the eigenvectors of graph Laplacians in the fixed and shrinking bandwidth cases respectively.

Although this burgeoning literature has provided many useful insights, several gaps remain between theory and practice. Most notably, in constructing the neighborhood graphs underlying the graph Laplacian, several choices must be made, including the choice of algorithm for constructing the graph, with k-nearest-neighbor (kNN) and kernel functions providing the main alternatives, as well as the choice of parameters (k, kernel bandwidth, normalization weights). These choices can lead to the graph Laplacian generating fundamentally different random walks and approximating different weighted Laplace-Beltrami operators. The existing theory has focused on one specific choice in which graphs are generated with smooth kernels with shrinking bandwidths. But a variety of other choices are often made in practice, including kNN graphs, r-neighborhood graphs, and the "self-tuning" graphs of Zelnik-Manor and Perona (2004). Surprisingly, few of the existing convergence results apply to these choices (see Maier et al. (2008) for an exception).

This chapter provides a general theoretical framework for analyzing graph Laplacians and operators that behave like Laplacians. Our point of view differs from that found in the existing literature; specifically, our point of departure is a stochastic process framework that utilizes the characterization of diffusion processes via drift and diffusion terms. This yields a general kernel-free framework for analyzing graph Laplacians with shrinking neighborhoods. We use it to extend the pointwise results of Hein et al. (2007) to cover non-smooth kernels and introduce location-dependent bandwidths. Applying these tools we are able to identify the asymptotic limit for a variety of graphs constructions including kNN, r-neighborhood, and "self-tuning" graphs. We are also able to provide an analysis for Locally Linear Embedding (Roweis and Saul, 2000).

A practical motivation for our interest in graph Laplacians based on kNN graphs is that these can be significantly sparser than those constructed using kernels, even if they have the same limit. Our framework allows us to establish this limiting equivalence. On the other hand, we can also exhibit cases in which kNN graphs converge to a different limit than graphs constructed from kernels, and that this explains some cases where kNN graphs perform poorly. Moreover, our framework allows us to generate new algorithms: in particular, by using location-dependent bandwidths we obtain a class of operators that have nice spectral convergence properties that parallel those of the normalized Laplacian in von Luxburg et al. (2008), but which converge to a different class of limits.

2.1 The Framework

Our work exploits the connections among diffusion processes, elliptic operators (in particular the weighted Laplace-Beltrami operator), and stochastic differential equations (SDEs). This builds upon the diffusion process viewpoint in Nadler et al. (2006). Critically, we make the connection to the drift and diffusion terms of a diffusion process. This allows us to present a kernel-free framework for analysis of graph Laplacians as well as giving a better intuitive understanding of the limit diffusion process.

We first give a brief overview of these connections and present our general framework for the asymptotic analysis of graph Laplacians as well as providing some relevant background material. We then introduce our assumptions and derive our main results for the limit operator for a wide range of graph construction methods. We use these to calculate asymptotic limits for specific graph constructions.

Relevant Differential Geometry

Assume $\mathcal{M}$ is a m-dimensional manifold embedded in $\mathbb{R}^b$. To identify the asymptotic infinitesimal generator of a diffusion on this manifold, we will derive the drift and diffusion terms in normal coordinates at each point. We refer the reader to Boothby (1986) for an exact definition of normal coordinates. For our purposes it suffices to note that normal coordinates are coordinates in $\mathbb{R}^m$ that behave roughly as if the neighborhood was projected onto the tangent plane at x. The extrinsic coordinates are the coordinates $\mathbb{R}^b$ in which the manifold is embedded. Since the density, and hence integration, is defined with respect to the manifold, we must relate to link normal coordinates s around a point x with the extrinsic coordinates y. This relation may be given as follows:

$$y - x = H_x s + L_x(ss^T) + O(||s^3||), \tag{2.1}$$

where H_x is a linear isomorphism between the normal coordinates in R^m and the m-dimensional tangent plane T_x at x. L_x is a linear operator describing the curvature of the manifold and takes $m \times m$ positive semidefinite matrices into the space orthogonal to the tangent plane, $T_x^\perp$. More advanced readers will note that this statement is Gauss' lemma and H_x and L_x are related to the first and second fundamental forms.

We are most interested in limits involving the weighted Laplace-Beltrami operator, a particular second-order differential operator.

Weighted Laplace-Beltrami operator

Definition 1 (Weighted Laplace-Beltrami operator)**.** *The weighted Laplace-Beltrami operator with respect to the density q is the second-order differential operator defined by $\Delta_q := \Delta_{\mathcal{M}} - \frac{\nabla q^T}{q}\nabla$ where $\Delta_{\mathcal{M}} := div \circ \nabla$ is the unweighted Laplace-Beltrami operator.*

It is of particular interest since it induces a smoothing functional for $f \in C^2(\mathcal{M})$ with support contained in the interior of the manifold:

$$\langle f, \Delta_q f \rangle_{L(q)} = ||\nabla f||^2_{L_2(q)}\,. \tag{2.2}$$

Note that existing literature on asymptotics of graph Laplacians often refers to the s^{th} weighted Laplace-Beltrami operator as Δ_s where $s \in \mathbb{R}$. This is Δ_{p^s} in our notation. For more information on the weighted Laplace-Beltrami operator see Grigor'yan (2006).

Equivalence of Limiting Characterizations

We now establish the promised connections among elliptic operators, diffusions, SDEs, and graph Laplacians. We first show that elliptic operators define diffusion processes and SDEs and vice versa. An elliptic operator $\mathcal{G}$ is a second order differential operator of the form

$$\mathcal{G}f(x) = \sum_{ij} a_{ij}(x)\frac{\partial^2 f(x)}{\partial x_i \partial x_j} + \sum_i b_i(x)\frac{\partial f(x)}{\partial x_i} + c(x)f(x),$$

where the $m \times m$ coefficient matrix $(a_{ij}(x))$ is positive semidefinite for all x. If we use normal coordinates for a manifold, we see that the weighted Laplace-Beltrami operator Δ_q is a special case of an elliptic operator with $(a_{ij}(x)) = I$, the identity matrix, $b(x) = \frac{\nabla q(x)}{q(x)}$, and $c(x) = 0$. Diffusion processes are related via a result by Dynkin which states that given a diffusion process, the generator of the process is an elliptic operator.

The (infinitesimal) generator $\mathcal{G}$ of a diffusion process X_t is defined as

$$\mathcal{G}f(x) := \lim_{t \to 0} \frac{\mathbb{E}_x f(X_t) - f(x)}{t}$$

when the limit exists and convergence is uniform over x. Here $\mathbb{E}_x f(X_t) = \mathbb{E}(f(X_t)|X_0 = x)$. A converse relation holds as well. The Hille-Yosida theorem characterizes when a linear operator, such as an elliptic operator, is the generator of a stochastic process. We refer the reader to Kallenberg (2002) for proofs.

A time-homogeneous stochastic differential equation (SDE) defines a diffusion process as a solution (when one exists) to the equation

$$dX_t = \mu(X_t)dt + \sigma(X_t)dW_t,$$

where X_t is a diffusion process taking values in $\mathbb{R}^d$. The terms $\mu(x)$ and $\sigma(x)\sigma(x)^T$ are the *drift* and *diffusion* terms of the process.

By Dynkin's result, the generator $\mathcal{G}$ of this process defines an elliptic operator and a simple calculation shows the operator is

$$\mathcal{G}f(x) = \frac{1}{2}\sum_{ij} \left(\sigma(x)\sigma(x)^T\right)_{ij} \frac{\partial^2 f(x)}{\partial x_i \partial x_j} + \sum_i \mu_i(x)\frac{\partial f(x)}{\partial x_i}.$$

In such diffusion processes there is no absorbing state and the term in the elliptic operator $c(x) = 0$. We note that one may also consider more general diffusion processes where $c(x) \leq 0$. When $c(x) < 0$ then we have the generator of a diffusion process with killing where $c(x)$ determines the killing rate of the diffusion at x.

To summarize, we see that a SDE or diffusion process define an elliptic operator, and importantly, the coefficients are the drift and diffusion terms, and the reverse relationship holds: An elliptic operator defines a diffusion under some regularity conditions on the coefficients.

All that remains then is to connect diffusion processes in continuous space to graph Laplacians on a finite set of points. Diffusion approximation theorems provide this connection. We state one version of such a theorem .

Theorem 2 (Diffusion Approximation). *Let $\mu(x)$ and $\sigma(x)\sigma(x)^T$ be drift and diffusion terms for a diffusion process defined on a compact set $S \subset \mathbb{R}^b$, and let and G be the corresponding infinitesimal generator. Let $\{Y_t^{(n)}\}_t$ be Markov chains with transition matrices P_n on state spaces $\{x_i\}_{i=1}^n$ for all n, and let $c_n > 0$ define a sequence of scalings. Put*

$$\begin{aligned}\hat{\mu}_n(x_i) \ &= c_n\mathbb{E}(Y_1^{(n)} - x_i | Y_0^{(n)} = x_i)\\ \hat{\sigma}_n(x_i)\hat{\sigma}_n(x_i)^T &= c_n \mathbb{V}ar(Y_1^{(n)} | Y_0^{(n)} = x_i).\end{aligned}$$

Let $f \in C^2(S)$. If for all $\epsilon > 0$

$$\begin{gathered}\hat{\mu}_n(x_i) \to \mu(x_i),\\ \hat{\sigma}_n(x_i)\hat{\sigma}_n(x_i)^T \to \sigma(x_i)\sigma(x_i)^T,\\ c_n \sup_{i \leq n} \mathrm{P}\left(\left|\left| Y_1^{(n)} - x_i \right|\right| > \epsilon \middle| Y_0^{(n)} = x_i \right) \to 0,\end{gathered}$$

then the generators $A_n f = c_n(P_n - I)f \to Gf$ Furthermore, for any bounded f and $t_0 > 0$ and the continuous-time transition kernels $T_n(t) = exp(tA_n)$ and T the transition kernel for G, we have $T_n(t)f \to T(t)f$ uniformly in t for $t < t_0$.

Proof. We first examine the case when $f(x) = x$. By assumption,

$$\begin{aligned}A_n\pi_n x = c_n(P_n - I)x &= c_n\mathbb{E}(Y_1^{(n)} - x_i | Y_0^{(n)} = x_i)\\ &= \mu_n(x) \to \mu(x) = Ax.\end{aligned}$$

Similarly if $f(x) = xx^T$, $||A_n\pi_n f - Af||_\infty \to 0$. If $f(x) = 1$, then $A_n\pi_n f = \pi_n Af = 0$. Thus, by linearity of A_n, $A_n\pi_n f \to Af$ for any quadratic polynomial f.

Taylor expand f to obtain $f(x+h) = q_x(h) + \delta_x(h)$ where $q_x(h)$ is a quadratic polynomial in h. Since the second derivative is continuous and the support of f is compact, $\sup_{x\in\mathcal{M}} \delta_x(h) = o(||h||^2)$ and $\sup_{x,h} \delta_x(h) < M$ for some constant M.

Let $\Delta_n = Y_1^{(n)} - x_i$. We may bound A_n acting on the remainder term $\delta_x(h)$ by

$$\begin{aligned}\sup_x A_n\delta_x &= c_n\mathbb{E}(\delta_x(\Delta_n)|Y_0^{(n)} = x)\\ &\leq \sup_x c_n\mathbb{E}(\delta_x(\Delta_n)\mathbb{I}(||\Delta_n|| \leq \epsilon)|Y_0^{(n)} = x) +\\ &\qquad\qquad M \sup_x c_n\mathbf{P}(||\Delta_n|| > \epsilon|Y_0^{(n)} = x)\\ &= o(c_n\mathbb{E}(||\Delta_n||^2 \, |Y_0^{(n)} = x)) + M\sup_x c_n\mathbf{P}(||\Delta_n|| > \epsilon|Y_0^{(n)} = x)\\ &= o(1)\end{aligned}$$

where the last equality holds by the assumptions on the uniform convergence of the diffusion term $\hat{\sigma}_n\hat{\sigma}_n^T$ and on the shrinking jumpsizes. Thus, $A_n\pi_n f \to Af$ for any $f \in C^2(\mathcal{M})$.

The class of functions $C^2(\mathcal{M})$ is dense in $L_\infty(\mathcal{M})$ and form a core for the generator A. Standard theorems give equivalence between strong convergence of infinitesimal generators on a core and uniform strong convergence of transition kernels on a Banach space (e.g. Theorem 1.6.1 in Ethier and Kurtz (1986)). □

We remark that though the results we have discussed thus far are stated in the context of the extrinsic coordinates $\mathbb{R}^b$, we describe appropriate extensions in terms of normal coordinates in section 2.8.

2.2 Assumptions

We describe here the assumptions and notation for the rest of the chapter. The following assumptions we will refer to as the *standard assumptions*. Unless stated explicitly otherwise, let f be an arbitrary function in $C^2(\mathcal{M})$.

Manifold assumptions

Assume $\mathcal{M}$ us a smooth m-dimensional manifold isometrically embedded in $\mathbb{R}^b$ via the map $i : \mathcal{M} \to \mathbb{R}^b$. The essential conditions that we require on the manifold are

1. Smoothness, the map i is a smooth embedding.
2. A single radius h_0 such that for all $x \in supp(f)$, $\mathcal{M} \cap B(x, h_0)$ is a neighborhood of x with normal coordinates, and
3. Bounded curvature of the manifold over $supp(f)$, i.e. that the second fundamental form is bounded .

When the manifold is smooth and compact, then these conditions are satisfied.

Assume points $\{x_i\}_{i=1}^\infty$ are sampled i.i.d. from a density $p \in C^2(\mathcal{M})$ with respect to the natural volume element of the manifold, and that p is bounded away from 0.

Notation

For brevity, we will always use $x, y \in \mathbb{R}^b$ to be points on $\mathcal{M}$ expressed in extrinsic coordinates and $s \in \mathbb{R}^m$ to be normal coordinates for y in a neighborhood centered at x. Since they represent the same point, we will also use y and s interchangeably as function arguments, i.e. $f(y) = f(s)$. Whenever we take a gradient,it is with respect to normal coordinates.

Generalized kernel

Though we use a kernel free framework, our main theorem utilizes a kernel, but one that is generalizes previously studied kernels by 1) considering non-smooth base kernels K_0, 2) introducing location dependent bandwidth functions $r_x(y)$, and 3) considering general weight functions $w_x(y)$. Our main result also handles 4) random weight and bandwidth functions.

Given a bandwidth scaling parameter $h > 0$, define a new kernel by

$$K(x,y) = w_x(y) K_0 \left(\frac{||y-x||}{h r_x(y)} \right). \tag{2.3}$$

Previously analyzed constructions using smooth kernels with compact support are specific instances of this more general kernel. In those constructions, the bandwidth scaling is fixed so $r_x = 1$ and the weighting function takes the particular form $w_x(y) = d(x)^{-\lambda} d(y)^{-\lambda}$ where $d(x)$ is the degree function and $\lambda \in \mathbb{R}$ is some constant.

The directed kNN graph is obtained if $K_0(x,y) = \mathbb{I}(||x-y|| \leq 1)$, $r_x(y) =$ distance to the k^{th} nearest neighbor of x, and $w_x(y) = 1$ for all x, y. We note that the kernel K is not necessarily symmetric; however, if $r_x(y) = r_y(x)$ and $w_x(y) = w_y(x)$ for all $x, y \in \mathcal{M}$ then the kernel is symmetric and the corresponding unnormalized Laplacian is positive semi-definite.

Kernel assumptions

We now introduce our assumptions on the choices K_0, h, w_x, r_x that govern the graph construction. Assume that the base kernel $K_0 : \mathbb{R}_+ \to \mathbb{R}_+$ has bounded variation and compact support and $h_n > 0$ form a sequence of bandwidth scalings. For (possible random) location dependent bandwidth and weight functions $r_x^{(n)}(\cdot) > 0, w_x^{(n)}(\cdot) \geq 0$, assume that they converge to $r_x(\cdot), w_x(\cdot)$ respectively and the convergence is uniform over $x \in \mathcal{M}$. Further assume they have Taylor-like expansions for all $x, y \in \mathcal{M}$ with $||x-y|| < h_n$

$$\begin{aligned} r_x^{(n)}(y) &= r_x(x) + (\dot{r}_x(x) + \alpha_x sign(u_x^T s) u_x)^T s + \epsilon_r^{(n)}(x,s) \\ w_x^{(n)}(y) &= w_x(x) + \nabla w_x(x)^T s + \epsilon_w^{(n)}(x,s) \end{aligned} \tag{2.4}$$

where the approximation error is uniformly bounded by

$$\sup_{x \in \mathcal{M}, ||s|| < h_n} |\epsilon_r^{(n)}(x,s)| = O(h_n^2)$$
$$\sup_{x \in \mathcal{M}, ||s|| < h_n} |\epsilon_w^{(n)}(x,s)| = O(h_n^2)$$

We briefly motivate the choice of assumptions. The bounded variation condition allows for non-smooth base kernels but enough regularity to obtain limits. The Taylor-like expansions give conditions where the limit is tractable to analytically compute as well as allowing for randomness in the remainder term as long as it is of the correct order. The particular expansion for the location dependent bandwidth allows one to analyze undirected kNN graphs, which exhibit a non-differentiable location dependent bandwidth (see section 2.4). Note that we do not constrain the general weight functions $w_x^{(n)}(y)$ to be a power of the degree function, $d_n(x)^\alpha d_n(y)^\alpha$ nor impose a particular functional form for location dependent bandwidths r_x. This gives us two degrees of freedom, which allows the same asymptotic limit be obtained for an entire class of parameters governing the graph construction. In section 2.6, we discuss one may choose a graph construction that has more attractive finite sample properties than other constructions that have the same limit.

Functions and convergence

We define here what we mean by convergence when the domains of the functions are changing. When take $g_n \to g$ where $domain(g_n) = \mathcal{X}_n \subset \mathcal{M}$, to mean $||g_n - \pi_n g||_\infty \to 0$ where $\pi_n g = g|_{\mathcal{X}_n}$ is the restriction of g to $\mathcal{X}_n$. Likewise, for operators T_n on functions with domain $\mathcal{X}_n$, we take $T_n g = T_n \pi_n g$. Convergence of operators $T_n \to T$ means $T_n f \to Tf$ for all $f \in C^2(\mathcal{M})$. When $\mathcal{X}_n = \mathcal{M}$ for all n, this is convergence in the strong operator topology under the L_∞ norm.

We consider the limit of the random walk Laplacian defined by as $L_{rw} = I - D^{-1}W$ where I is the identity, W is the matrix of edge weights, and D is the diagonal degree matrix.

2.3 Main Theorem

Our main result is stated in the following theorem.

Theorem 3. *Assume the standard assumptions hold eventually with probability 1. If the bandwidth scalings h_n satisfy $h_n \downarrow 0$ and $nh_n^{m+2}/\log n \to \infty$, then for graphs constructed using the kernels*

$$K_n(x,y) = w_x^{(n)}(y) K_0\left(\frac{||y-x||}{h_n r_x^{(n)}(y)}\right) \tag{2.5}$$

there exists a constant $Z_{K_0,m} > 0$ depending only on the base kernel K_0 and the dimension m such that for $c_n = Z_{K_0,m}/h^2$,

$$-c_n L_{rw}^{(n)} f \to Af$$

where A is the infinitesimal generator of a diffusion process with the following drift and diffusion terms given in normal coordinates:

$$\mu_s(x) = r_x(x)^2\left(\frac{\nabla p(x)}{p(x)} + \frac{\nabla w(x)}{w(x)} + (m+2)\frac{\dot{r}_x(x)}{r_x(x)}\right),$$
$$\sigma_s(x)\sigma_s(x)^T = r_x(x)^2 I$$

where I is the $m \times m$ identity matrix.

Proof. We apply the diffusion approximation theorem (Theorem 2) to obtain convergence of the random walk Laplacians. Since $h_n \downarrow 0$, the probability of a jump of size $> \epsilon$ equals 0 eventually. Thus, we simply need to show uniform convergence of the drift and diffusion terms and identify their limits. We leave the detailed calculations in section 2.8 and present the main ideas in the proof here.

We first assume that K_0 is an indicator kernel. To generalize, we note that for kernels of bounded variation, we may write $K_0(x) = \int \mathbb{I}(|x| < z)d\eta_+(z) - \int \mathbb{I}(|x| < z)d\eta_-(z)$ for some finite positive measures η_-, η_+ with compact support. The result for general kernels then follows from Fubini's theorem. We also initially assume that we are given the true density p. After identifying the desired limits given the true density, we show that the empirical version converges uniformly to these limits.

The key calculation is lemma 7 in the section 2.8 which establishes that integrating against an indicator kernel is like integrating over a sphere re-centered on $h_n^2 \dot{r}_x(x)$. Given this calculation and by Taylor expanding the non-kernel terms, one obtains the infinitesimal first and second moments and the degree operator.

$$\begin{aligned}
M_1^{(n)}(x) &= \frac{1}{h_n^m}\int sK_n(x,y)p(y)ds \\
&= C_{K_0,m}h_n^2 r_x(x)^{m+2}\left(w_x(x)\frac{\nabla p(x)}{m+2} + p(x)\frac{\nabla w_x(x)}{m+2} + w_x(x)p(x)\dot{r}_x(x) + o(1)\right) \\
M_2^{(n)}(x) &= \frac{1}{h_n^m}\int ss^T K_n(x,y)p(y)ds \\
&= \frac{C_{K_0,m}}{m+2}h_n^2 r_x(x)^{m+2}\left(w_x(x)p(x)I + O(h_n)\right), \\
d_n(x) &= \frac{1}{h_n^m}\int K_n(x,y)p(y)ds \\
&= C'_{K_0,m} r_x(x)^m \left(w_x(x)p(x) + O(h_n)\right)
\end{aligned}$$

where $C_{K_0,m} = \int u^{m+2}d\eta$, $C'_{K_0,m} = \int u^m d\eta$ and η is the signed measure $\eta = \eta_+ - \eta_-$. A more detailed expansion of the moment calculations is given in section 2.8

Let $Z_{K_0,m} = (m+2)\frac{C'_{K_0,m}}{C_{K_0,m}}$ and $c_n = Z_{K_0,m}/h_n^2$. Since K_n/d_n define Markov transition kernels, taking the limits $\mu_s(x) = \lim_{n\to\infty} c_n M_1^{(n)}(x)/d_n(x)$ and $\sigma_s(x)\sigma_s(x)^T = \lim_{n\to\infty} c_n M_2^{(n)}(x)/d_n(x)$ and applying the diffusion approximation theorem gives the stated result.

To more formally apply the diffusion approximation theorem we may calculate the drift and diffusion in extrinsic coordinates. In extrinsic coordinates, we have

$$\begin{aligned}
\mu(x) &= r_x(x)^2 H_x\Big(\frac{\nabla p(x)}{p(x)} + \frac{\nabla w_x(x)}{w_x(x)} + (m+2)\frac{\dot{r}_x(x)}{r_x(x)}\Big) \\
&\qquad\qquad\qquad\qquad + r_x(x)^2 L_x(I), \\
\sigma(x)\sigma(x)^T &= r(x)^2 \Pi_{T_x},
\end{aligned}$$

where Π_{T_x} is the projection onto the tangent plane at x, and H_x and L_x are the linear mappings between normal coordinates and extrinsic coordinates defined in Eqn (2.1).

To prove the convergence of the empirical quantities, we proceed in two steps. First, examine the behavior of a non-random kernel where the bandwidth and weight functions are fixed. The a.s. uniform convergence of the moments can be shown using Bernstein's inequality and Borel-Cantelli. In the second step, we show that the moments using the random bandwidth and weight functions may be eventually bounded above and below using appropriate non-random functions. These function shrink to the limit bandwidth and weight functions at an appropriate rate and the squeeze theorem establishes the a.s. uniform convergence. Further details are given in section 2.8

Since p, w_x, r_x are all assumed to be bounded away from 0, the scaled degree operators d_n are eventually bounded away from 0 with probability 1, and the continuous mapping theorem applied to $\frac{M_i^{(n)}/h_n^2}{d_n}$ gives a.s. uniform convergence of the drift and diffusion.

□

Unnormalized and Normalized Laplacians

While our results are for the infinitesimal generator of a diffusion process, that is, for the limit of the random walk Laplacian $L_{rw} = I - D^{-1}W$, it is easy to generalize them to the unnormalized Laplacian $L_u = D - W = DL_{rw}$ and symmetrically normalized Laplacian $L_{norm} = I - D^{-1/2}WD^{-1/2} = D^{1/2}L_{rw}D^{-1/2}$.

Corollary 4. *Take the assumptions in Theorem 3, and let A be the limiting operator of the random walk Laplacian. The degree terms $d_n(\cdot)$ converge uniformly a.s. to a function $d(\cdot)$, and*

$$-c_n' L_u^{(n)} f \to d \cdot Af \quad \textit{a.s.}$$

where $c_n' = c_n/h^m$. Furthermore, under the additional assumptions $nh_n^{m+4}/\log n \to \infty$, $\sup_{x,y} |w_x^{(n)} - w_x| = o(h_n^2)$, $\sup_{x,y} |r_x^{(n)} - r_x| = o(h_n^2)$, and $d, w_x, r_x \in C^2(\mathcal{M})$, we have

$$-c_n L_{norm}^{(n)} f \to d^{1/2} \cdot A(d^{-1/2} f) \quad \textit{a.s.}$$

Proof. For any two functions $\phi_1, \phi_2 : \mathcal{M} \to \mathbb{R}$, define $g_u(\phi_1, \phi_2) = (\phi_1(\cdot), f_1(\cdot)\phi_2(\cdot))$. We note that g_u is a continuous mapping in the L_∞ topology and

$$(d_n, c_n' L_u^n f) = g_u(d_n, c_n L_{rw} f).$$

By the continuous mapping theorem, if $d_n \to d$ a.s. and $c_n L_{rw}^{(n)} f \to Lf$ a.s. in the then

$$c_n' L_u^{(n)} \to d \cdot Lf.$$

Thus, convergence of the random walk Laplacians implies convergence of the unnormalized Laplacian under the very weak condition of convergence of the degree operator to a bounded function.

Convergence of the normalized Laplacian is slightly trickier. We may write the normalized Laplacian as

$$\begin{aligned} L_{norm}^{(n)} f &= d_n^{1/2} L_{rw}^{(n)} (d_n^{-1/2} f) && (2.6) \\ &= d_n^{1/2} L_{rw}^{(n)} (d^{-1/2} f) + d_n^{1/2} L_{rw}^{(n)} ((d_n^{-1/2} - d^{-1/2}) f). && (2.7) \end{aligned}$$

Using the continuous mapping theorem, we see that convergence of the normalized Laplacian, $c_n L_{norm}^{(n)} f \to d^{-1/2} L_{rw}(d^{-1/2} f)$, is equivalent to showing $c_n L_{rw}^{(n)}((d_n^{-1/2} - d^{-1/2})f) \to 0$. A Taylor expansion of the inverse square root gives that showing $c_n L_{rw}^{(n)}(d_n - d) \to 0$ is sufficient to prove convergence.

We now verify conditions which will ensure that the degree operators will converge at the appropriate rate. We further decompose the empirical degree operator into the bias $\mathbb{E} d_n - d$ and empirical error $d_n - \mathbb{E} d_n$.

Simply carrying out the Taylor expansions to higher order terms in the calculation of the degree function d_n in Eq. 2.24, and using the refined calculation of the zeroth moment in lemma 8 in section 2.8, the bias of the degree operator is $d_n - d = h_n^2 b + o(h_n^2)$ for some uniformly bounded, continuous function b.

Thus we have,

$$c_n L_{rw}^{(n)}(d_n - d) = c_n h_n^2 \, ||(I - P_n)b||_\infty + o(1) = o(1) \tag{2.8}$$

since $c_n h_n^2$ is constant and $||(I - P_n)\phi||_\infty \to 0$ for any continuous function ϕ.

We also need to check that the empirical error $||d_n - \mathbb{E}d_n||_\infty = O(h_n^2)$ a.s.. If $nh_n^{m+4}/\log n \to \infty$ then using the Bernstein bound in equation 2.28 with ϵ replaced by h_n^2 and applying Borel-Cantelli gives the desired result.

□

Limit as weighted Laplace-Beltrami operator

Under some regularity conditions, the limit given in the main theorem (Theorem 3) yields a weighted Laplace-Beltrami operator.

For convenience, define $\gamma(x) = r_x(x)$, $\omega(x) = w_x(x)$.

Corollary 5. *Assume the conditions of Theorem 3 and let $q = p^2\omega\gamma^{m+2}$. If $r_x(y) = r_y(x), w_x(y) = w_y(x)$ for all $x, y \in \mathcal{M}$ and $r_{(\cdot)}(\cdot), w_{(\cdot)}(\cdot)$ are twice differentiable in a neighborhood of (x, x) for all x, then for $c_n' = Z_{K_0,m}/h^{m+2}$*

$$-c_n' L_u^{(n)} \to \frac{q}{p}\Delta_q. \tag{2.9}$$

Proof. Note that $\nabla|_{y=x}\, \gamma(y) = 2\ \nabla|_{y=x}\, r_x(y)$. The result follows from application of Theorem 3, Corrollary 4, and the definition of the weighted Laplace-Beltrami operator. □

2.4 Application to Specific Graph Constructions

To illustrate Theorem 3, we apply it to calculate the asymptotic limits of graph Laplacians for several widely used graph construction methods. We also apply the general diffusion theory framework to analyze LLE.

r-Neighborhood and Kernel Graphs

In the case of the r-neighborhood graph, the Laplacian is constructed using a kernel with fixed bandwidth and normalization. The base kernel is simply the indicator function $K_0(x) = I(|x| < r)$. The radius $r_x(y)$ is constant so $\dot{r}(x) = 0$. The drift is given by $\mu_s(x) = \nabla p(x)/p(x)$ and the diffusion term is $\sigma_s(x)\sigma_s(x)^T = I$. The limit operator is thus

$$\frac{1}{2}\Delta_{\mathcal{M}} + \frac{\nabla p(x)^T}{p(x)}\nabla = \frac{1}{2}\Delta_2$$

as expected. This analysis also holds for arbitrary kernels of bounded variation. One may also introduce the usual weight function $w_x^{(n)}(y) = d_n(x)^{-\alpha} d_n(y)^{-\alpha}$ to obtain limits of the form $\frac{1}{2}\Delta_{p^{2-2\alpha)}}$. These limits match those obtained by Hein et al. (2007) and Lafon (2004) for smooth kernels.

Directed k-Nearest Neighbor Graph

For kNN-graphs, the base kernel is still the indicator kernel, and the weight function is constant 1. However, the bandwidth function $r_x^{(n)}(y)$ is random and depends on x. Since the graph is directed, it does not depend on y so $\dot{r}_x = 0$.

By the analysis in section 2.4, $r_x(x) = cp^{-1/m}(x)$ for some constant c. Consequently the limit operator is proportional to

$$\frac{1}{p^{2/m}}(x)\left(\Delta_{\mathcal{M}} + 2\frac{\nabla p^T}{p}\nabla\right) = \frac{1}{p^{2/m}}\Delta_{p^2}.$$

Note that this is generally *not* a self-adjoint operator in $L(p)$. The symmetrization of the graph has a non-trivial affect to make the graph Laplacian self-adjoint.

Undirected k-Nearest Neighbor Graph

We consider the OR-construction where the nodes v_i and v_j are linked if v_i is a k-nearest neighbor of v_j *or* vice-versa. In this case $h_n^m r_x^{(n)}(y) = \max\{\rho_n(x), \rho_n(y)\}$ where $\rho_n(x)$ is the distance to the k_n^{th} nearest neighbor of x. The limit bandwith function is non-differentiable, $r_x(y) = \max\{p^{-1/m}(x), p^{-1/m}(y)\}$, but a Taylor-like expansion exists with $\dot{r}_x(x) = \frac{1}{2m}\frac{\nabla p(x)^T}{p(x)}$. The limit operator is

$$\frac{1}{p^{2/m}}\Delta_{p^{1-2/m}}.$$

which is self-adjoint in $L_2(p)$. Surprisingly, if $m = 1$ then the kNN graph construction induces a drift *away* from high densiy regions.

Conditions for kNN convergence

To complete the analysis, we must check the conditions for kNN graph constructions to satisfy the assumptions of the main theorem. This is a straightforward application of existing uniform consistency results for kNN density estimation.

Let $h_n = \left(\frac{k_n}{n}\right)^{1/m}$. The condition we must verify is

$$\sup_{y\in\mathcal{M}} \left|\left|r_x^{(n)} - r_x\right|\right|_\infty = O(h_n^2) \text{ a.s.}$$

We check this for the directed kNN graph, but analyses for other kNN graphs are similar. The kNN density estimate of Loftsgaarden and Quesenberry (1965) is

$$\hat{p}_n(x) = \frac{V_m}{n(h_n r_x^{(n)}(x))^m} \tag{2.10}$$

where $h_n r_x^{(n)}(x)$ is the distance to the k^{th} nearest neighbor of x given n data points. Taylor expanding equation 2.10 shows that if $||\hat{p}_n - p||_\infty = O(h_n^2)$ a.s. then the requirement on the location dependent bandwidth for the main theorem is satisfied.

Devroye and Wagner (1977)'s proof for the uniform consistency of kNN density estimation may be easily modified to show this. Take $\epsilon = (k_n/n)^2$ in their proof. One then sees that $h_n = k_n/n \to 0$ and $\frac{nh_n^{m+2}}{\log n} = \frac{k_n^{2+2/m}}{n^{1+2/m}\log n} \to \infty$ are sufficient to achieve the desired bound on the error.

"Self-Tuning" Graphs

The form of the kernel used in self-tuning graphs is

$$K_n(x,y) = \exp\left(\frac{-\,||x-y||^2}{\sigma_n(x)\sigma_n(y)}\right).$$

where $\sigma_n(x) = \rho_n(x)$, the distance between x and the k^{th} nearest neighbor. The limit bandwidth function is $r_x(y) = \sqrt{p^{-1/m}(x)p^{-1/m}(y)}$. Since this is twice differentiable, corollary 5 gives the asymptotic limit, which is the same as for undirected kNN graphs,

$$p^{-2/m}\Delta_{p^{1-2/m}}.$$

Locally Linear Embedding

Locally linear embedding (LLE), introduced by Roweis and Saul (2000), has been noted to behave like (the square of) the Laplace-Beltrami operator Belkin and Niyogi (2003).

Using our kernel-free framework we will show how LLE differs from weighted Laplace-Beltrami operators and graph Laplacians in several ways. 1) LLE has, in general, *no well-defined asymptotic limit* without additional conditions on the weights. 2) It can only behave like an *unweighted* Laplace-Beltrami operator. 3) It is affected by the curvature of the manifold, and the curvature can cause LLE to not behave like any elliptic operator (including the Laplace-Beltrami operator).

The key observation is that LLE only controls for the drift term in the extrinsic coordinates. Thus, the diffusion term has freedom to vary. However, if the manifold has curvature, the drift in extrinsic coordinates constrains the diffusion term in normal coordinates.

The LLE matrix is defined as $(I-W)^T(I-W)$ where W is a weight matrix which minimizes reconstruction error $W = argmin_{W'}\,||(I-W')y||^2$ under the constraints $W'1 = 1$ and $W'_{ij} \neq 0$ only if j is one of the k^{th} nearest neighbors of i. Typically $k > m$ and reconstruction error $= 0$. We will analyze the matrix $M = I - W$.

Suppose LLE produces a sequence of matrices $M_n = I - W_n$. The row sums of M_n are 0. Thus, we may decompose $M_n = A_n^+ - A_n^-$ where A_n^+, A_n^- are generators for finite state Markov processes obtained from the positive and negative weights respectively. Assume that there is some scaling c_n such that $c_n A_n^+, c_n A_n^-$ converge to generators of diffusion processes with drifts μ_+, μ_- and diffusion terms $\sigma_+\sigma_+^T, \sigma_-\sigma_-^T$. Set $\mu = \mu_+ - \mu_-$ and $\sigma\sigma^T = \sigma_+\sigma_+ - \sigma_-\sigma_-$.

No well-defined limit

We first show there is generally no well-defined asymptotic limit when one simply minimizes reconstruction error. Suppose $rank(L_x) < m(m+1)/2$ at x. This will necessarily be true if the extrinsic dimension $b < m(m+1)/2 + m$. For simplicity assume $rank(L_x) = 0$. Minimizing the LLE reconstruction error does not constrain the diffusion term, and $\sigma(x)\sigma(x)^T$ may be chosen arbitrarily. Choose asymptotic diffusion $\sigma\sigma^T$ and drift μ terms that are Lipschitz so that a corresponding diffusion process necessarily exists. A diffusion with terms $2\sigma\sigma^T$ and μ will also exist in that case.

One may easily construct graphs for the positive and negative weights with these asymptotic diffusion and drift terms by solving highly underdetermined quadratic programs. Furthermore, in the interior of the manifold, these graphs may be constructed so that the finite sample drift terms are exactly equal by adding an additional constraint. Thus, $A_n^+ \to 2G_0 + \mu^T\nabla$ and $A_n^- \to G_0 + \mu^T\nabla$ where G_0 is the generator for a diffusion process with zero drift and diffusion term $\sigma_-(x)\sigma_-(x)^T$. We have $c_nM_n = A_n^+ - A_n^- \to G_0$. Thus, we can construct a sequence of LLE matrices that have 0 reconstruction error but have an arbitrary limit. It is trivial to see how to modify the construction when $0 < rank(L_x) < m(m+1)/2$.

No drift

Since $\mu_s(x) = 0$, if the LLE matrix does behave like a Laplace-Beltrami operator, it must behave like an unweighted one, and the density has no affect on the drift.

Curvature and limit

We now show that the curvature of the manifold affects LLE and that the LLE matrix may not behave like any elliptic operator. If the manifold has sufficient curvature, namely if the extrinsic coordinates have dimension $b \geq m+m(m+1)/2$ and $rank(L_x) = m(m+1)/2$, then the diffusion term in the normal coordinates is fully constrained by the drift term in the extrinsic coordinates.

Recall from equation 2.1 that the extrinsic coordinates as a function of the normal coordinates are $y = x + H_xs + L_x(ss^T) + O(||s||^3)$. By linearity of H_x and L_x, the asymptotic drift in the extrinsic coordinates is $\mu(x) = H_x\mu_s(x) + L_x(\sigma_s(x)\sigma_s(x)^T)$.

Since reconstruction error in the extrinsic coordinates is 0, we have in normal coordinates

$$\mu_s(x) = 0 \quad \text{and} \quad L_x(\sigma_s(x)\sigma_s(x)^T) = 0.$$

In other words, the asymptotic drift and diffusion terms of A_n^+ and A_n^- must be the same, and $c_nM_n \to G_0 - G_0 = 0$.

This implies that the scaling c_n where LLE can be expected to behave like an elliptic operator gives the trivial limit 0. If another scaling yields a non-trivial limit, it may include higher-order differential terms. It is easy to see when L_x is not full rank, the curvature affects LLE by partially constraining the diffusion term.

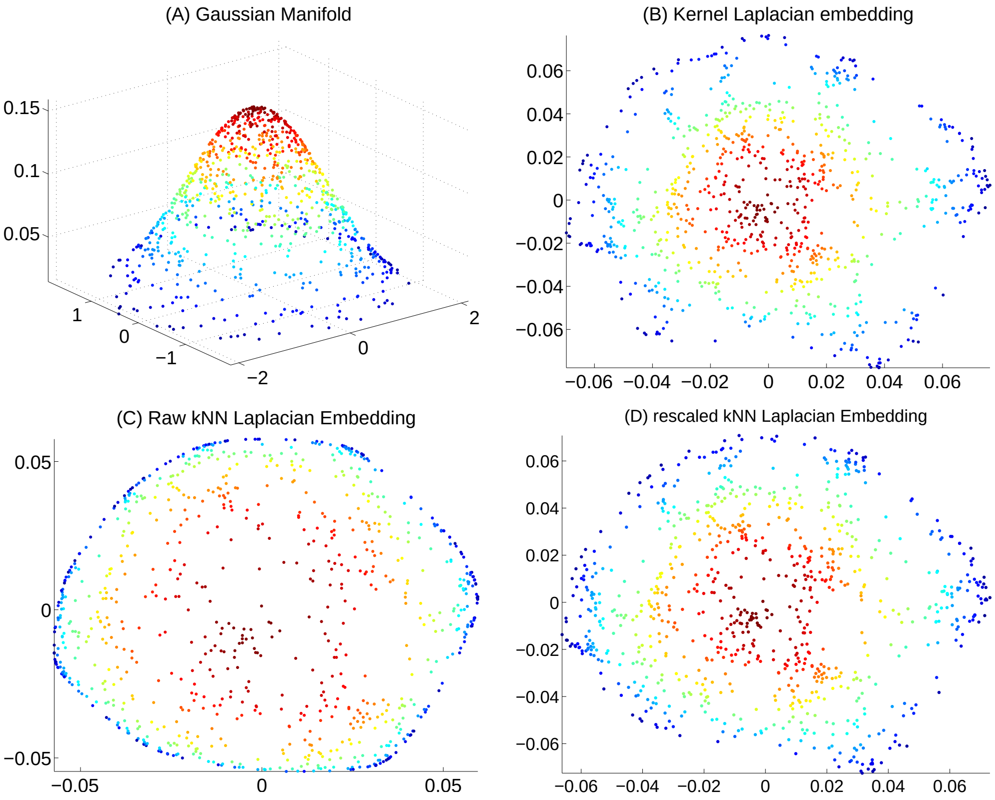


Figure 2.1: (A) shows a 2D manifold where the x and y coordinates are drawn from a truncated standard normal distribution. (B-D) show embeddings using different graph constructions. (B) uses a normalized Gaussian kernel $\frac{K(x,y)}{d(x)^{1/2}d(y)^{1/2}}$, (C) uses a kNN graph, and (D) uses a kNN graph with edge weights $\sqrt{\hat{p}(x)\hat{p}(y)}$. The bandwidth for (B) was chosen to be the median standard deviation from taking 1 step in the kNN graph.

Regularization and LLE

We note that while the LLE framework of minimizing reconstruction error can yield ill-behaved solutions, practical implementations add a regularization term when constructing the weights. This causes the reconstruction error to be non-zero in general and gives unique solutions for the weights which favor equal weights (and asymptotic behavior like kNN graphs).

2.5 Experiments

To illustrate the theory, we show how to correct the bad behavior of the kNN Laplacian for a synthetic data set. We also show how our analysis can predict the surprising behavior of LLE.

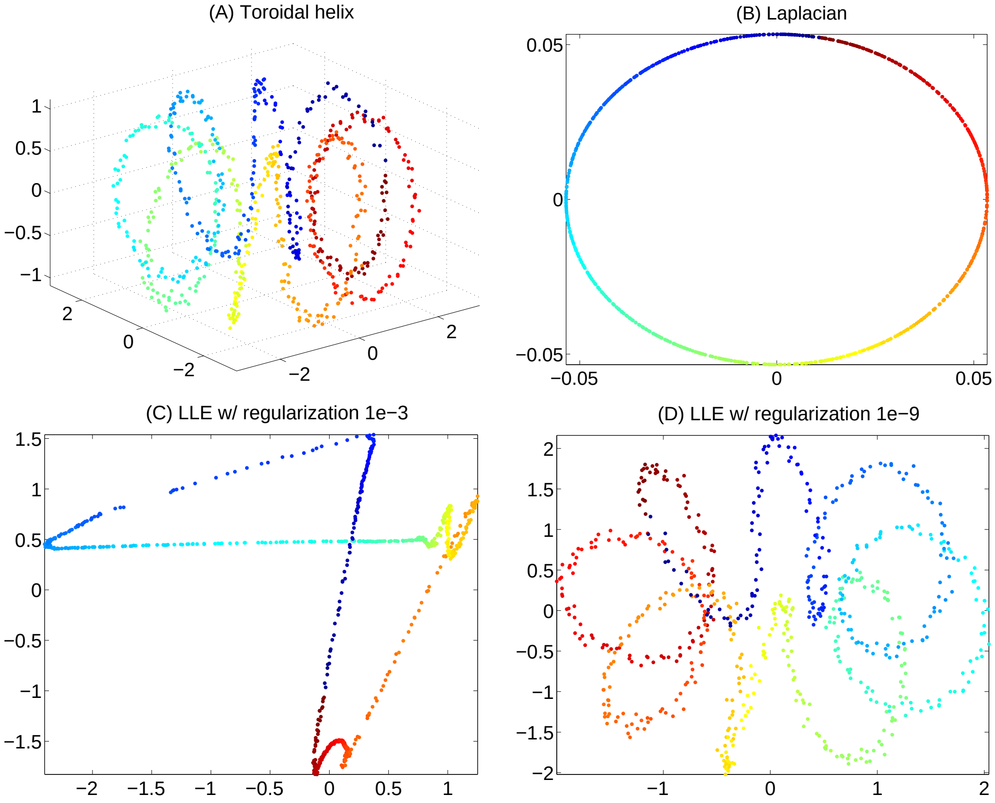


Figure 2.2: (A) shows a 1D manifold isometric to a circle. (B-D) show the embeddings using (B) Laplacian eigenmaps which correctly identifies the structure, (C) LLE with default regularization 1e-3, and (D) LLE with negligible regularization 1e-6.

kNN Laplacian

We consider a non-linear embedding example which almost all non-linear embedding techniques handle well but the kNN graph Laplacian performs poorly. Figure 2.1 shows a 2D manifold embedded in 3 dimensions and embeddings using different graph constructions. The theoretical limit of the normalized Laplacian L_{knn} for a kNN graph is $L_{knn} = \frac{1}{p}\Delta_1$. while the limit for a graph with Gaussian weights is $L_{gauss} = \Delta_p$. The first 2 coordinates of each point are from a truncated standard normal distribution, so the density at the boundary is small and the effect of the $1/p$ term is substantial. This yields the bad behavior shown in Figure 2.1 (C). We may use the relationship between the k^{th}-nearest neighbor and the density in Eqn (2.10) to obtain a pilot estimate $\hat{p}$ of the density. Choosing $w_x(y) = \sqrt{\hat{p}_n(x)\hat{p}_n(y)}$, gives a weighted kNN graph with the same limit as the graph with Gaussian weights. Figure 2.1 (D) shows that this change yields the roughly desired behavior but with fewer “holes” in low density regions and more in high density regions.

LLE

We consider another synthetic data set, the toroidal helix, in which the manifold structure is easy to recover. Figure 2.5 (A) shows the manifold which is clearly isometric to a circle, a fact picked up by the kNN Laplacian in Figure 2.5 (B).

Our theory predicts that the heuristic argument that LLE behaves like the Laplace-Beltrami operator will *not* hold. Since the total dimension for the drift and diffusion terms is 2 and the global coordinates also have dimension 2, that there is forced cancellation of the first and second order differential terms and the operator should behave like the 0 operator or include higher order differentials. In Figure 2.5 (C) and (D), we see this that LLE performs poorly and that the behavior comes closer to the 0 operator when the regularization term is smaller.

2.6 Remarks and Discussion

Non-shrinking neighborhoods

In this dissertation, we have presented convergence results using results for diffusion processes without jumps. Graphs constructed using a fixed, non-shrinking bandwidth do not fit within this framework, but approximation theorems for diffusion processes with jumps still apply (see Jacod and Širjaev (2003)). Instead of being characterized by the drift and diffusion pair $\mu(x), \sigma(x)\sigma(x)^T$, the infinitesimal generators for a diffusion process with jumps is characterized by the "Lêvy-Khintchine" triplet consisting of the drift, diffusion, and "Lêvy measure." Given a sequence of transition kernels K_n, the additional requirement for convergence of the limiting process is the existence of a limiting transition kernel K such that $\int K_n(\cdot, dy)g(y)dy \rightarrow \int K(\cdot, dy)g(y)dy$ locally uniformly for all C^1 functions g. This establishes an impossibility result, that no method that only assigns positive mass on shrinking neighborhoods can have the same graph Laplacian limit as a a kernel construction method where the bandwidth is fixed.

Convergence rates

We note that one missing element in our analysis is the derivation of convergence rates. For the main theorem, we note that it is, in fact, not necessary to apply a diffusion approximation theorem. Since our theorem still uses a kernel (albeit one with much weaker conditions), a virtually identical proof can be obtained by applying a function f and Taylor expanding it. Thus, we believe that similar convergence rates to Hein et al. (2007) can be obtained. Also, while our convergence result is stated for the strong operator topology, the same conditions as in Hein give weak convergence.

Relation to density estimation

The connection between kernel density estimation and graph Laplacians is obvious, namely, any kernel density estimation method using a non-negative kernel induces a random walk graph Laplacian and vice versa.

In this dissertation, we have shown that as a consequence of identifying the asymptotic degree term, we have shown consistency of a wide class of adaptive kernel density estimates on a manifold. We also have shown that on compact sets, the the bias term is uniformly bounded by a term of order h^2, and a small modification to the Bernstein bound (Eqn 2.28) gives that the variance is bounded by a term of order h^{-m}. Both of which one would expect. This generalizes previous work on manifold density estimation by Pelletier (2005) and Ozakin (2009) to adaptive kernel density estimation.

The well-studied field of kernel density estimation may also lead to insights on how to choose a good location dependent bandwidth as well. We compare the form of our density estimates to other well-known adaptive kernel density estimation techniques. The balloon estimator and sample smoothing estimators as described by Terrell and Scott (1992) are respectively given by

$$\hat{f}_1(x) = \frac{1}{nh(x)^d} \sum_i K\left(\frac{||x_i - x||}{h(x_i)}\right) \tag{2.11}$$

$$\hat{f}_2(x) = \frac{1}{n} \sum_i \frac{1}{h(x_i)^d} K\left(\frac{||x_i - x||}{h(x_i)}\right). \tag{2.12}$$

In the univariate case, Terrell and Scott (1992) show that the balloon estimators yield no improvement to the asymptotic rate of convergence over fixed bandwidth density estimates. The sample smoothing estimator gives a density estimate which does not necessarily integrate to 1. However, it can exhibit better asymptotic behavior in some cases. The Abramson square root law estimator (Abramson, 1982) is an example of a sample smoothing estimator and takes $h(x_i) = hp(x_i)^{-1/2}$. On compact intervals, this estimator has bias of order h^4 rather than the usual h^2 (Silverman, 1998), and it achieves this bias reduction without resorting to higher order kernels, which necessarily negative in some region. However, the bias in the tail for univariate Gaussian data is of order $(h/\log h)^2$ (Terrell and Scott, 1992), which is only marginally better than h^2.

While we do not make claims of being able to reduce bias in the case of density estimation a manifold, in fact, we do not believe bias reduction to the order of h^4 is possible unless one makes some use of manifold curvature information, the existing density estimation literature suggests what potential benefits one may achieve over different regions of a density.

Eigenvalues/Eigenvectors

We find our location dependent bandwidth results to be of interest in the context of the negative result in von Luxburg et al. (2008) for unnormalized Laplacians with a fixed bandwidth. Their results state that for unnormalized graph Laplacians, the eigenvectors of the discrete approximations do not converge if the corresponding eigenvalues lie in the range of the asymptotic degree operator $d(x)$, whereas for the normalized Laplacian, the "degree operator" is the identity and the eigenvectors converge if the corresponding eigenvalues stay away from 1. Our results suggest that even with unnormalized Laplacians, one can obtain convergence of the eigenvectors by manipulating the range of the degree operator through the use of a location dependent bandwidth function. For example, with kNN graphs we have that the degree operator is essentially 1. For self-tuning graphs, the degree operator also converges to 1, and since the kernels form an equicontinuous family of functions, the theory for compact integral operators may be rigorously applied when the bandwidth scaling is fixed.

Thus we can obtain unnormalized and normalized graph Laplacians that (1) have spectra that converges for fixed (non-decreasing) bandwidth scalings and (2) converge to a limit that is different from that of previously analyzed normalized Laplacians when the bandwidth decreases to 0.

Corollary 6. *Assume the standard assumptions. Further assume that* $\left\{K_0\left(\frac{\|y-x\|}{h}\right) : h > h_0\right\}$ *forms an equicontinuous family of functions for some* $h_0 > 0$*. Let* $q, g \in C^2(\mathcal{M})$ *be bounded away from 0 and* ∞*. Set*

$$\gamma = \sqrt{\frac{q}{pg}} \qquad\qquad r_x(y) = \sqrt{\gamma(x)\gamma(y)} \tag{2.13}$$

$$\omega = \left(\frac{pg}{q}\right)^{m/2}\frac{g}{p} \qquad\qquad w_x(y) = \sqrt{\omega(x)\omega(y)}. \tag{2.14}$$

If $h_n = h_1$ *for all* n*, then the eigenvectors of the normalized Laplacians converge in the sense given in von Luxburg et al. (2008). If* $h_n \downarrow 0$ *satisfy the assumptions of theorem 3, then the limit rescaled degree operator is* $d = g$ *and*

$$-c_n L_{norm} f \to g^{-1/2}\frac{q}{p}\Delta_q(g^{-1/2}f) \tag{2.15}$$

which induces the smoothness functional

$$\left\langle f, g^{-1/2}\frac{q}{p}\Delta_q(g^{-1/2}f)\right\rangle_{L_2(p)} = \left\langle \nabla(g^{-1/2}f), \nabla(g^{-1/2}f)\right\rangle_{L_2(q)}. \tag{2.16}$$

Proof. Assume the $h_n \downarrow 0$ case. Use corollary 5 and solve for ω and γ in the system of equations: $q = p^2\omega\gamma^{m+2}$, $g = p\omega\gamma^m$. In the $h_n = h_1$ case, the conditions satisfy those given in von Luxburg et al. (2008) with the modification that the kernel is not bounded away from 0 and the additional assumption that p is bounded away from 0. Thus, the asymptotic degree operator d is bounded away from 0, and the proofs in von Luxburg et al. (2008) may be applied without additional modification. □

We note that the restriction to an equicontinuous family of kernel functions excludes kNN graph constructions. However, one may get around this by considering the two-step transition kernels $K_2(x, y) = K(x, \cdot) * K(\cdot, y)$, where $*$ denotes the convolution operator with respect to the underlying density. For indicator kernels like those used in kNN graph constructions, K_2 will be Lipschitz and hence form an equicontinuous family. Thus, if one handles the potential issues with the random bandwidth function, one may apply the theory of compact integral operators to obtain convergence of the spectrum and eigenvectors for kNN graph Laplacians when k grows appropriately.

Reasons for choosing a graph construction method

We highlight how our more general kernel can yield advantageous properties. In particular, it yields graphs constructions where one can (1) control the sparsity of the Laplacian matrix, (2) control connectivity properties in low density regions, (3) give asymptotic limits that cannot be attained using previous graph construction methods, and (4) give Laplacians with good spectral properties in the non-shrinking bandwidth case.

One way to control (1) and (2) is to make the binary choice of using kNN or a kernel with uniform bandwidth to construct the graph. Our results show that, by using a pilot estimate of the density, one can obtain sparsity and connectivity properties in the continuum between these two choices.

For (3) and (4), we note that the limits for previously analyzed unnormalized Laplacians were of the form $p^{\alpha-1}\Delta_{p^\alpha} f$. Using corollary 5, one see that limits of the form $\frac{q}{p}\Delta_q$ for any smooth, bounded density q on the manifold can be obtained. Equivalently, one can approximate the smoothness functional $||\nabla f||^2_{L_2(q)}$ for any almost any q, not just p^α.

For normalized Laplacians, which have good spectral properties, the previously known limits induced smoothness functionals of the form $\left|\left|\nabla(p^{(1-\alpha)/2}f)\right|\right|^2_{L_2(p^\alpha)}$. With our more general kernel and any $g, q \in C^2(\mathcal{M})$, we may induce a smoothness functional of the form $||\nabla(gf)||^2_{L_2(q)}$. In particular, in the case where the smoothness functional is just a norm on the gradient of f, $||\nabla f||^2_{L_2(q)}$, q may be chosen to be almost any density, not just $q = p^1$.

2.7 Conclusions

We have introduced a general framework that enables us to analyze a wide class of graph Laplacian constructions. Our framework reduces the problem of graph Laplacian analysis to the calculation of a mean and variance (or drift and diffusion) for any graph construction method with positive weights and shrinking neighborhoods. Our main theorem extends existing strong operator convergence results to non-smooth kernels, and introduces a general location-dependent bandwidth function. The analysis of a location-dependent bandwidth function, in particular, significantly extends the family of graph constructions for which an asymptotic limit is known. This family includes the previously unstudied (but commonly used) kNN graph constructions, unweighted r-neighborhood graphs, and “self-tuning” graphs.

Our results also have practical significance in graph constructions as they suggest graph constructions that (1) can produce sparser graphs than those constructed with the usual kernel methods, despite having the same asymptotic limit, and (2) in the fixed bandwidth regime, produce normalized Laplacians that have well-behaved spectra but converge to a different class of limit operators than previously studied normalized Laplacians. In particular, this class of limits include those that induce the smoothness functional $||\nabla f||^2_{L_2(q)}$ for almost any density q. The graph constructions may also (3) have better connectivity properties in low-density regions.

2.8 Proofs

Main lemma

Lemma 7 (Integration with location dependent bandwidth)**.** *Let $\mathbb{I}$ be the indicator function and $h > 0$ be a constant. Let r_x be a location dependent bandwidth function that satisfies the standard assumptions, i.e. it has a Taylor-like expansion*

$$\tilde{r}_x(y) = r_x(x) + (\dot{r}_x(x) + \alpha_x \text{sign}(u_x^T s) u_x)^T s + \epsilon_r(x, s).$$

Let $V_m = \frac{\pi^{m/2}}{\Gamma\left(\frac{m}{2}+1\right)}$ be the volume of the unit m–sphere.

Then

$$M_0 = \frac{1}{V_m h^m} \int \mathbb{I}\left(\frac{||y-x||}{\tilde{r}_x(s)} < h\right) ds = r_x(x)^m + h^2 \epsilon_0(x, h)$$
$$M_1 = \frac{1}{V_m h^m} \int s\mathbb{I}\left(\frac{||y-x||}{\tilde{r}_x(s)} < h\right) ds = h^2 r_x(x)^{m+2} \dot{r}(x) + h^3 \epsilon_1(x, h)$$
$$M_2 = \frac{1}{V_m h^m} \int ss^T\mathbb{I}\left(\frac{||y-x||}{\tilde{r}_x(s)} < h\right) ds = \frac{2h^2}{m+2} r_x(x)^{m+2} I + h^3 \epsilon_2(x, h)$$

where $\sup_{x\in\mathcal{M}, h<h_0} ||\epsilon_i(x, h)|| < C_\epsilon$ *for some constant* $C_\epsilon > 0$.

Proof. Let $v(s) = \dot{r}(x) + \text{s}ign(s^T u_x)\alpha u_x$. We will show that the set on which the indicator function is approximately a sphere shifted by $v/r_x(x)$ with radius $hr_x(x)$.

$$\begin{aligned}
\mathbb{I}\left(\frac{||y-x||}{r_x(s)} < h\right) &= \mathbb{I}\Big(||s||^2 + \left|\left|L(ss^T)\right|\right|^2 < h^2(r_x(x) + v(s)^T s + O(||s||^2))^2\Big) \\
&= \mathbb{I}\left(||s||^2 < h^2 r_x(x)^2 (1 + 2v(s)^T s + O(h^2))\right) \\
&= \mathbb{I}\left(||s||^2 - 2h^2 \frac{v(s)^T s}{r_x(x)} + \frac{h^4 v(s)^T v(s)}{r_x(x)^2} < h^2 r_x(x)^2 + O(h^4)\right) \\
&= \mathbb{I}\left(\left|\left|s - \frac{v(s)}{r_x(x)}\right|\right| < hr_x(x) + h^3 \delta_x(s)\right)
\end{aligned}$$

for some function $\delta_x(s)$. Furthermore, the assumptions on the bounded curvature of the manifold and uniform bounds on the bandwidth function remainder term $\epsilon_r(x, s)$ give that the perturbation term $\delta_x(s)$ may be uniformly bounded by $\sup_{x\in\mathcal{M}} |\delta_x(s)| \leq C_\delta(||s||^2)$ for some constant C_δ.

The result for the zeroth moment follows immediately from this. The results for the first and second moments we calculate in lemma 10. □

Refined analysis of the zeroth moment

For convergence of the normalized Laplacian, we need a more refined result for the zeroth moment.

Lemma 8. *Assume*

$$\tilde{r}_x(y) = r_x(s) + \epsilon_r(x, s).$$

where $r_x(s)$ is twice continuously differentiable as a function of x and s and and ϵ_r is bounded. Then

$$\int \frac{1}{V_m h^m} \mathbb{I}\left(\frac{||y-x||}{\tilde{r}_x(s)} < h\right) ds = r_x(x)^m + h^2 b(x) + h^2 \epsilon_0(x, h)$$

where b is continuous and $\sup_x |\epsilon_0(x,h)| \to 0$ as $h \to 0$.

Proof. We first sketch idea behind the proof and leave the details to interested readers. One may convert the integral in normal coordinates to an integral in polar coordinates (R, θ). One may then apply the implicit function theorem to obtain that the unperturbed radius function R is a twice continuously differentiable function of h. This gives a Taylor expansion of the zeroth moment with respect to h. $\epsilon_r(x, s)$ gives the desired result.

We may express the integral for the zeroth moment in polar coordinates

$$\begin{aligned} Z_x(h) &= \int \frac{1}{V_m h^m} \mathbb{I}\left(\frac{||y-x||}{\tilde{r}_x(s)} < h\right) ds \\ &= \int R_x(\theta, h) d\mu_\theta \end{aligned}$$

where μ_θ is the uniform measure on the surface of the unit m-sphere and $\tilde{s} = s/h = R_x(\theta, h))\theta$ solves the equation

$$||\tilde{s}||^2 + L(\tilde{s}\tilde{s}^T) = \left(r_x(x) + h\nabla r_x(x)^T \tilde{s} + h^2 \tilde{s}^T \mathcal{H}_{r_x(0)} \tilde{s}\right)^2.$$

and $\mathcal{H}_{r_x(0)}$ is the Hessian of $r_x(\cdot)$ evaluated at 0.

By the implicit function theorem, the solutions $\tilde{s}$ define a twice continuously differentiable function of x, h. For sufficiently small $h \geq 0$, $\tilde{s}$ is bounded away from 0 since r_x is bounded away from 0 and $||s/h||$ is bounded away from ∞ by the bound in lemma 7. Thus, $R_x(\theta, h)$ and $Z_x(h)$ are twice continuously differentiable with bounded second derivatives.

$Z_x(h)$ then has a second-order Taylor expansion $Z_x(h) = Z_x(0) + Z_x'(0)h + Z_x''(0)h^2 + o(h^2)$.

By the less refined analysis in lemma 7, we have that $Z_x(0) = r_x(x)^m$ and $Z_x'(0^+) = 0$. One may apply a squeeze theorem to obtain that the contribution of the error term $\epsilon_r(x, s)$ to the zeroth moment is bounded by $C_r \sup_{x,s} |\epsilon_r(x, s)|$ for some constant C_r, and the result follows. □

Moments of the indicator kernel / Integrating over the centered sphere in normal coordinates

Here we calculate the first three moments of the normalized indicator kernel where $V_m = \int \mathbb{I}(||u|| < 1)du = \int_{S_m} du$ is the volume of the m-dimensional unit sphere in Euclidean space.

Lemma 9 (Moments for the sphere)**.** *Let* $K(||s|| / h) = \frac{1}{h^m V_m}\mathbb{I}(||s|| < h)$*. Then the first two moments are given by:*

$$M_0 = \int K(||s|| / h) ds = \frac{1}{h^m V_m}\int_{S_m} ds = 1 + O(h^3)$$
$$M_1 = \int sK(||s|| / h) ds = \frac{1}{h^m V_m}\int_{S_m} s ds = 0 + O(h^4)$$
$$M_2 = \int ss^T K(||s|| / h) ds = \frac{1}{h^m V_m}\int_{S_m} ss^T ds = \frac{1}{m+2}\mathbb{I} + O(h^4).$$

Proof. The error terms $O(h^i)$ arise trivially after converting normal coordinates to tangent space coordinates. Thus, we may simply treat the integrals as integrals in m–dimensional Euclidean space to obtain the leading term. The values for M_0 and M_1 follow immediately from the definition of the volume V_m and by symmetry of the sphere. We obtain the second moment result by calculating the values on the diagonal and off-diagonal. On the off-diagonal

$$\frac{1}{V_m}\int_{S_m} s_i s_j ds = 0$$

for $i \neq j$ due to symmetry of the sphere.

On the diagonal

$$\frac{1}{V_m}\int_{S_m} s_i^2 ds = \frac{V_{m-1}}{V_m}\int_{-1}^{1} s_i^2 (1 - s_i^2)^{(m-1)/2} ds_i \tag{2.17}$$
$$= \frac{V_{m-1}}{V_m}\int_{-1}^{1} s_i \times s_i (1 - s_i^2)^{(m-1)/2} ds_i \tag{2.18}$$
$$= 0 + \frac{V_{m-1}}{V_m}\int_{-1}^{1} \frac{1}{m+1}(1 - s_i^2)^{(m+1)/2} ds_i \tag{2.19}$$
$$= \frac{1}{m+1}\frac{V_{m-1}}{V_m V_{m+1}}\int_{-1}^{1} V_{m+1}(1 - s_i^2)^{(m+1)/2} ds_i \tag{2.20}$$
$$= \frac{1}{m+1}\frac{V_{m-1}}{V_{m+1}}\frac{V_{m+2}}{V_m} \tag{2.21}$$
$$= \frac{1}{m+2} \tag{2.22}$$

where the last equality uses the recurrence relationship $V_{m+2} = \frac{2\pi}{m+2}V_m$. □

Integrating the shifted and peturbed sphere

Here we calculate the moments used in Lemma 7.

The integrals in lemma 7 essentially involve integrating over sphere with (1) a shifted center $h^2\dot{r}_x(x)$, (2) a symmetric shift by $sign(s^T u)h^2\alpha_x u$ on two half-spheres, and (3) a small perturbation $h^3\delta_x(s)$.

Lemma 10 (Moments of the shifted and perturbed sphere). *Let $v_c \in \mathbb{R}^m$, u be a unit vector in $\mathbb{R}^m$, $\beta \in \mathbb{R}$, and $h > 0$. Define $\tilde{K}(s) = \mathbb{I}(\left|\left|s - v_c + \mathrm{s}ign(s^T u)\beta u\right|\right| < h + h^3\delta)$, so that the support of $\tilde{K}$ is a shifted and perturbed sphere with center v_c, symmetric shift $\mathrm{s}ign(s^T u)\beta u$, and radius perturbation $h^3\delta$.*

Assume $\left|\left|v_c\right|\right|, |\beta| < Ch^2$ and $\delta < \min\{C, 1\}$ for some constant C, and put $h_{max} = h + h^3\delta$ Then

$$M_0 = \frac{1}{V_m}\int_{\mathbb{R}^m} \tilde{K}(s)ds = h^m + \epsilon_0$$
$$M_1 = \frac{1}{V_m}\int_{\mathbb{R}^m} s\tilde{K}(s)ds = h^{m+2}v_c + \epsilon_1$$
$$M_2 = \frac{1}{V_m}\int_{\mathbb{R}^m} ss^T\tilde{K}(s)ds = \frac{h^{m+2}}{m+2}\mathbb{I} + \epsilon_2.$$

where $\epsilon_1 < \kappa C h_{max}^{m+1}$ and $\epsilon_i < \kappa C h_{max}^{m+3}$ for $i = 1, 2$ and κ is some universal constant that does not depend on δ, v_c, or β.

Proof. Set $H_+ = \{s \in \mathbb{R}^m : u^T s > 0\}$ and $H_- = H_+^C$ to be the half-spaces defined by u. For a set $H \subset \mathbb{R}^m$, let $H + v_c := \{w + v_c : w \in H\}$.

We first bound the error introduced by the perturbation $h^3\delta$. Define

$$\mathcal{A} := supp(\tilde{K}) = \{s \in \mathbb{R}^m : \left|\left|s - v_c + \mathrm{s}ign(s^T u)\beta u\right|\right| < h + h^3\delta\}$$
$$\overline{\mathcal{A}} := \{s \in \mathbb{R}^m : \left|\left|s - v_c + \mathrm{s}ign(s^T u)\beta u\right|\right| < h\}$$

so that $\overline{\mathcal{A}}$ gets rid of the dependence on the perturbation.

For any function Q, we have a trivial bound

$$\begin{aligned}\left|\int_{\mathcal{A}} Q(s)ds - \int_{\overline{\mathcal{A}}} Q(s)ds\right| &< Q_{max}|Vol(A) - Vol(\bar(A))| \\ &< Q_{max}V_m|h_{max}^m - h^m| \\ &< Q_{max}V_m(mh_{max}^{m-1})(h^3\delta) \\ &= O(h^{m+2}Q_{max}) \qquad (2.23)\end{aligned}$$

where $Q_{max} = \sup_{||s||<h_{max}} Q(s)$ and mV_{m-1} is the surface area of the m-dimensional sphere. For $Q(s) = 1/V_m$, s/V_m, or ss^T/V_m, the corresponding Q_{max} are $1/V_m$, h_{max}/V_m, and h_{max}^2/V_m. The error induced by the perturbation is thus of the right order.

We now consider the integral over the unperturbed but shifted sphere. Denote by $B_h(v)$ the ball of radius h centered on v. Note that the function $\mathbb{I}(s \in \overline{\mathcal{A}}) = \mathbb{I}(\left|\left|s - v_c + sign(s^T u)\beta u\right|\right| < h)$ is symmetric around v_c. Thus, for a function $Q(s - v_c + \beta u)$ which is symmetric around v_c,

$$\begin{aligned}
\int_{\overline{\mathcal{A}}} Q(s-v_c)ds &= 2\int_{\overline{\mathcal{A}}\cap H^+} Q(s-v_c)ds \\
&= 2\int_{H^+} Q(s-v_c)\mathbb{I}(||s-v_c|| < h)ds - \\
&\quad 2\int_{H^+} Q(s-v_c)(\mathbb{I}(||s-v_c|| < h) - \mathbb{I}(||s-v_c+\beta u|| < h))ds \\
&= \int Q(s)\mathbb{I}(||s|| < h)ds - \\
&\quad 2\int_{H^+} Q(s-v_c)(\mathbb{I}(s \in B_h(v_c)) - \mathbb{I}(s \in B_h(v_c - \beta u)))ds.
\end{aligned}$$

For $Q(s) = 1/V_m$ or ss^T/V_m, lemma 9 gives that the value of the main term $\int Q(s)\mathbb{I}(||s|| < h)ds$ is h^m or $\frac{h^{m+2}}{m+2}I$ respectively. The error term is bounded by

$$\begin{aligned}
&2\int_{H^+} Q(s-v_c)(\mathbb{I}(s \in B_h(v_c)) - \mathbb{I}(s \in B_h(v_c - \beta u)))ds \\
&\quad \leq 2Q_{max}\int_{H^+} |\mathbb{I}(s \in B_h(v_c)) - \mathbb{I}(s \in B_h(v_c - \beta u))|ds \\
&\quad < 2Q_{max}|\beta|Area(H^+ \cap B_h(v_c)) \\
&\quad < 2Q_{max}|\beta|(mV_{m-1}h^{m-1}) \\
&\quad < 2mV_{m-1}CQ_{max}h^{m+1}
\end{aligned}$$

where $Area(H^+ \cap B_h(v_c))$ is the surface area of a half-sphere of radius h. Plugging in $Q_{max} = 1/V_m$ and h^2/V_m give that the error terms for the zeroth and second moment calculations are of the right order.

By another symmetry argument, we have for the first moment calculation $\int_{\overline{\mathcal{A}}} \frac{1}{V_m}(s - v_c)ds = 0$ or equivalently,

$$\begin{aligned}
\frac{1}{V_m}\int_{\overline{\mathcal{A}}} sds &= \frac{v_c}{V_m}\int_{\overline{\mathcal{A}}} ds \\
&= h^m v_c + O(h^{m+3})
\end{aligned}$$

where the last equality holds from the calculation of the zeroth moment above. More precisely, the error term is bounded by $2mV_{m-1}CQ_{max}h^{m+1}v_c$.

□

Details of proof the main theorem

Expansion of moment calculations

We expand the moment calculations in the proof of the main theorem. Each step is a consequence of an assumption or from the lemmas computing the moments using an indicator kernel.

$$
\begin{aligned}
M_1^{(n)}(x) &= \frac{1}{h_n^m}\int sK_n(x,y)p(y)ds \\
&= \frac{1}{h_n^m}\int sw_x^{(n)}(s)K_0\left(\frac{||y-x||}{h_n r_x^{(n)}(s)}\right)p(s)ds \\
&= \frac{1}{h_n^m}\int s\left(w_x(x)+\nabla w_x(x)^T s+O(h_n^2)\right)\left(p(x)+\nabla p(x)^T s+O(h_n^2)\right)\times \\
&\quad \times K_0\left(\frac{||y-x||}{h_n r_x^{(n)}(s)}\right)ds \\
&= C_{K_0,m}h_n^2 r_x(x)^{m+2}\left(w_x(x)\frac{\nabla p(x)}{m+2}+p(x)\frac{\nabla w_x(x)}{m+2}+w_x(x)p(x)\dot{r}_x(x)+o(1)\right)
\end{aligned}
$$

$$
\begin{aligned}
M_2^{(n)}(x) &= \frac{1}{h_n^m}\int ss^T K_n(x,y)p(y)ds \\
&= \frac{1}{h_n^m}\int ss^T w_x^{(n)}(s)K_0\left(\frac{||y-x||}{h_n r_x^{(n)}(s)}\right)p(s)ds \\
&= \frac{1}{h_n^m}\int ss^T\left(w_x(x)+O(h_n)\right)\left(p(x)+O(h_n)\right)K_0\left(\frac{||y-x||}{h_n r_x^{(n)}(s)}\right)ds \\
&= \frac{C_{K_0,m}}{m+2}h_n^2 r_x(x)^{m+2}\left(w_x(x)p(x)I+O(h_n)\right),
\end{aligned}
$$

$$
\begin{aligned}
d_n(x) &= \frac{1}{h_n^m}\int K_n(x,y)p(y)ds && (2.24)\\
&= \frac{1}{h^m}\int w_x^{(n)}(s)K_0\left(\frac{||y-x||}{h_n r_x^{(n)}(s)}\right)p(s)ds && (2.25)\\
&= \frac{1}{h^m}\int \left(w_x(x)+O(h_n)\right)\left(p(x)+O(h_n)\right)K_0\left(\frac{||y-x||}{h_n r_x^{(n)}(s)}\right)ds && (2.26)\\
&= C'_{K_0,m}r_x(x)^m\left(w_x(x)p(x)+O(h_n)\right) && (2.27)
\end{aligned}
$$

Almost sure uniform convergence of empirical quantities

Proof. For non-random $r_x^{(n)} = r_x, w_x^{(n)} = w_x$, the uniform and almost sure convergence of the empirical quantities to the true expectation follows from an application of Bernstein's inequality. In particular, the value of $F_n(x,S) = S_i K\left(\frac{||Y-x||}{h_n r_x(Y)}\right)$ is bounded by $K_{max}h_n$, where S is Y in normal coordinates and K_{max} depends on the kernel and the maximum curvature of the manifold.

Furthermore, the second moment calculation for $M_2^{(n)}$ gives that the variance $\mathbb{V}ar(F_n(x,S))$ is bounded by ch_n^{m+2} for some constant c that depends on K and the max of p, and does not depend on x. By Bernstein's inequality and a union bound, we have

$$\begin{aligned}
&Pr\left(\sup_{i\leq n}\left|\mathbb{E}_n\frac{1}{h_n^{m+2}}F_n(x_i,Y)-\frac{1}{h_n^2}M_1^{(n)}\right|>\epsilon\right)\\
&=Pr\left(\sup_{i\leq n}|\mathbb{E}_nF_n(x_i,Y)-\mathbb{E}F_n(x_i,Y)|>\epsilon h_n^{m+2}\right)\\
&<2n\exp\left(-\frac{\epsilon^2}{2c/(nh_n^{m+2})+2K_{max}\epsilon/(3nh_n^{m+1})}\right). \qquad (2.28)
\end{aligned}$$

The uniform convergence a.s. of the first moment follows from Borel-Cantelli. Similar inequalities are attained for the empirical second moment and degree terms.

Now assume $r_x^{(n)}, w_x^{(n)}$ are random and define F_n as before. To handle the random weight and bandwidth function case, we first choose deterministic weight and bandwidth functions to maximize the first moment under a constraint that is satisfied eventually a.s.. Define

$$\begin{aligned}
\overline{w}_x^{(n)}(y)&=w_x(y)+\kappa h_n^2 sign(s_i)\\
\overline{r}_x^{(n)}(y)&=r_x(x)+(\dot{r}_x(x)+\alpha_x\mathrm{sign}(u_x^Ts)u_x)^Ts-\kappa h_n^2 sign(s_i)\\
\overline{F}_n(y)&=s_i\overline{w}_x^{(n)}(y)K_0\left(\frac{||y-x||}{h_n\overline{r}_x^{(n)}(y)}\right)
\end{aligned}$$

for some constant κ such that $\overline{r}_x^{(n)} < r_x^{(n)}$ and $\overline{w}_x^{(n)} > w_x^{(n)}$ eventually. This is possible since the perturbation terms $\epsilon_r^{(n)}(x,s), \epsilon_w^{(n)}(x,s) = O(h_n^2)$. Thus, we have $\overline{F}_{\kappa,n}(x,y) > F_n(x,y)$ for all $x,y \in \mathcal{M}$ eventually with probability 1. Since $\overline{F}_{\kappa,n}(x,Y)$ uses deterministic weight and bandwidth functions, we obtain i.i.d. random variables and may apply the Bernstein bound on $\overline{F}_{\kappa,n}(x,y)$ to obtain an upper bound on the empirical quantities, namely $\mathbb{E}_n\overline{F}_{\kappa,n}(x,Y) > \mathbb{E}_nF_n(x,Y)$ for all $x \in \mathcal{M}$ eventually with probability 1. We may similarly obtain a lower bound. By lemma 10, the difference between the expectation of the upper bound and the is $\mathbb{E}\overline{F}_{\kappa,n}(x,Y) - \mathbb{E}\overline{F}_{0,n}(x,Y) = o(\kappa h_n^{m+2})$. Applying the squeeze theorem gives a.s. uniform convergence of the empirical first moment $M_1^{(n)}/h_n^2$. The degree and second moment terms are handled similarly. □

Chapter 3

Combinatorial Structures: Distributions and Representations

In this chapter we examine the problem of representating combinatorial structures and describing natural distributions on the representations. By combinatorial structures, we mean graphs, permutations, partitions, or other discrete objects. The primary motivation for studying these combinatorial structures is in their relationship to stick-breaking processes and nonparametric Bayesian hierarchical mixture models.

We give a finite combinatorial representation many nonparametric hierarchical Bayesian models using random graphs. In other words, we give a *finite* random cluster model which describes the clustering behavior of a hierarchy of *infinite* stick-breaking processes on any finite subset of a set of points. The implications of the representations are two-fold. In terms of new algorithms, we obtain new Markov Chain Monte Carlo (MCMC) samplers for nonparametric hierarchical Bayesian models. In particular, we obtain two samplers for the hierarchical Dirichlet process (HDP). In the experimental results in chapter 4, both samplers empirically show substantially better performance over a Chinese Restaurant Franchise sampler. Furthermore, we also present an informal argument that one of the new samplers should never be more than 3 times worse than the usual Chinese restaurant franchise Gibbs sampler in the worst case. The representations also lead to better understanding of hierarchical models. In particular, graph representations of hierarchical Bayesian models lend themselves to descriptions as coagulation and fragmentation processes. These hierarchical models, including the HDP, nested Dirichlet Process (nDP), nested Chinese restaurant process (nCRP), and tree-structured stick-breaking process, may be described by the sequence of coagulation and fragmentation operations. Using coagulation-fragmentation duality, one also identifies a hierarchical model of particular interest where the marginal distribution at each level of the hierarchy is from a Pitman-Yor process.

Beyond the applications to Bayesian models, this chapter describes the relationships among random graphs, permutations, stick-breaking, and coalescent processes. For example, the "reversed" Chinese restaurant process provides an immediate connection between the Kingman coalescent and the CRP. By exploiting the relationship between graphs and permutations, we devise a merge operation which generalizes reservoir sampling algorithms for drawing a random sample without replacement from a single stream to an distributed algorithm.

This chapter address a fairly wide range of topics which fall into the following categories: 1) the representation of combinatorial structures, 2) natural distributions and data generating processes on these structures and their relation to existing nonparametric models, 3) insights gained from examining alternate representations 4) practical implications of alternate representations, such as new MCMC methods.

3.1 Representations for mixture models

Partitions are a combinatorial structure of particular interest due to their important role in Bayesian statistics. Any Bayesian mixture model has a latent structure which is described by a partition. For hierarchical mixture models the latent structure may be described by a nested partition. This partition, along with the data, forms a sufficient statistic for the complete data likelihood.

The most common way to represent the latent structure is to introduce a latent class membership variable z_i for each data point x_i, where the latent variable assigns a cluster ID to the corresponding point. In this representation, changing a single latent variable z_i only affects the class membership of a single point.

Instead of latent class membership variables, one may use combinatorial structures to represent partitions. Examples of combinatorial structures include graphs, forests, permutations, and mappings from a finite set to itself. Such structures have been used as data augmentation schemes in the past. The general scheme of using the connected components of a random graph to represent a partition is referred to as a random cluster model. The Swendsen-Wang sampler for the Ising model is an example of a random cluster model used to sample partitions. Recently, Blei and Frazier (2010) use functional digraphs to give the distance-dependent Chinese Restaurant Process (ddCRP) which gives both a new Gibbs sampler for Dirichlet process mixture models as well as a non-exchangeable prior on partitions.

The representation of a partition in terms of a combinatorial structure defines an alternate set of variables on which one can perform Gibbs sampling. We present two applications to samplers for Hierarchical Dirichlet Process Mixture Models using the insights gained from examining the combinatorial structures. The first extends the ddCRP sampler to Hierarchical Dirichlet Processes (HDPs) and demonstrates how exploiting the combinatorial structure via dynamic programming makes one iteration as fast as the usual Gibbs sampler. The second extends the split-merge sampler for Dirichlet Processes to HDPs. Chapter 4 introduces further optimizations to split-merge samplers using ideas that are generally applicable to Markov Chain Monte Carlo methods.

We also extend the existing connections between combinatorial structures and certain stick-breaking processes to a wider range of processes including the HDP, the Nested Dirichlet Processes (nDP) of Rodriguez et al. (2008), the Nested Chinese Restaurant Processes (nCRPs) of Blei et al. (2010), and Tree-structured Stick Breaking Process of Adams et al. (2010). The underlying combinatorial structure makes clear that all of these processes may be described as compositions of coagulation and fragmentation processes. We briefly discuss how the processes of coagulation and fragmentation on combinatorial structures lead to hierarchical models and sharing of parameters.

We also point out that one coagulation-fragmentation process which is not currently used for data modeling has particularly nice properties.

3.2 Notation

For the reader's convenience, we list the common symbols we will use in this chapter and their descriptions.

Symbol	Description
$\mathcal{X}$	set $\{x_1, ..., x_n\}$ of n data points
$[n]$	the set $\{1, ..., n\}$
$\mathcal{B}$	a partition of $[n]$ with blocks $B_1, ..., B_k$
$\mathcal{X}(B_i)$	the set $\{x_j : j \in B_i\}$ of points in $\mathcal{X}$ corresponding to B_i
$f(x_i\|\mathcal{X}(B_j))$	the posterior predictive probability of x_i for a mixture component given $\mathcal{X}(B_j)$ belongs to that component
π	a permutation
b	a base for the strong generating set representation i.e. a permutation
F	a recursive forest on vertices $[n]$ and ordering defined by b or the equivalent SGS representation
S_n	the permutation group on $[n]$

3.3 Arborescence Forests, Random Recursive Forests, and the Chinese Restaurant Process

We introduce a series of bijections between combinatorial structures. In particular, we relate combinatorial structures with a graph representation. These bijections allow one to develop samplers in a representation of one's choice.

We start by relating a commonly used Bayesian nonparametric prior, the Chinese Restaurant Process (CRP), to a random combinatorial structure, a random recursive forest, through its sequential construction. This relationship is well-known in the probability community but has only recently implicitly made its way into the machine learning community via the work of Blei and Frazier (2010).

An arborescence forest F is a directed acyclic graph in which every node has outdegree 1 except for the roots of the forest which have outdegree 0. The weakly connected component containing node i consists of all nodes j such that, after replacing all directed edges with undirected edges, there is an undirected path between i to j. An arborescence tree is defined in an analogous manner.

A tree T with totally ordered vertices labeled $v_1 < ... < v_n$ is a recursive tree if all paths to the root v_1 are decreasing. A labeled forest F is a recursive forest if each subtree is a recursive tree. A permutation b defines a total ordering by $b_1 < b_2 < ... < b_n$. We will refer to the permutation b as a base and say F respects the base b if F is a recursive forest on $b_1 < ... < b_n$.

To draw from a recursive tree uniformly at random with respect to a base b, one may use the following sequential procedure. Designate b_1 as a root. For subsequent points where $1 < i \leq n$, connect b_i uniformly at random to one of the previous vertices $\{b_1, ..., b_{i-1}\}$. It is easy to see that any sequence of choices yields a unique recursive tree and any recursive tree may be constructed with some sequence of choices. To draw a recursive forest uniformly at random, introduce a dummy vertex $b_0 < b_1$. Draw a random recursive tree on the $n + 1$ vertices. One is left with a random recursive forest after removing b_0. From this construction it is clear that for a fixed base, there are $(n - 1)!$ recursive trees and $n!$ recursive forests. One may tilt the distribution of forest to contain more or fewer trees by changing the probability of connecting to the dummy vertex b_0.

The construction of a random recursive forest is intimately related to the sequential procedure that describes the CRP. In the CRP, the i^{th} customer chooses to sit to the left of the j^{th} customer, $j < i$, with probability $\propto 1$ and chooses to sit at a new table with probability $\propto \theta$. If one identifies the j^{th} customer with vertex b_j and b_0 with the action of sitting at a new table, then the relationship is clear.

3.4 Permutations, Strong Generating Sets, and Bases

The CRP also has a well-known relationship to the cycle representation of permutations. Each table in the CRP represents a cycle. Furthermore, each sequence of choices yields a unique seating arrangement. For i to sit down to the left of j means that one inserts i immediately to the left of j in the cycle representation. Since the edges of a recursive forest represent the "sit to the left" choices in a CRP, it shares the same cycle representation induced by the CRP. We further examine this relationship and show how the recursive forest representation naturally results from the action of the permutation group and is the representation of a permutation in terms of a strong generating set (SGS).

We show how considering both the left and right group actions leads to a "reversed" Chinese restaurant process. This "reversed" CRP clearly establishes the connection between the CRP and coalescent model in a simple, direct manner. We also examine connections to random permutations and sampling which are applied to a distributed random sampling problem in section 3.8

Consider the (right) action of the permutation group S_n on the ordered list $b = (1, \cdots, n)$. Let $G_i \leq S_n$ be the stabilizer of $\{1, ..., i\}$. In other words, G_i is the subgroup of permutations of $\{i + 1, ..., n\}$ which fix $\{1, ..., i\}$. Then, $G_n \leq G_{n-1} \leq \cdots G_1 \leq G_0 = S_n$ is a stabilizer chain with respect to the base b where G_n is the trivial subgroup containing only the identity. This gives a unique representation of the symmetric group via a set of generators defined by right transversals T_i of G_i in G_{i-1}. In other words, for any permutation $\pi \in S_n$ and set of right transversals $\{T_i\}$, we may uniquely write

$$\pi = g_n \cdots g_1 g_0$$

where $g_i \in T_i$. The set $\cup_i T_i$ defines a strong generating set (SGS) with respect to the base b.

One particular set of right transversals is of interest. The right transversals $T_i = \{(a\, i) : a \leq i\}$ define an SGS with respect to $b = (1, \cdots, n)$. In other words, T_i consists of the identity and the

$i-1$ transpositions of i with a smaller number. For example the permutation $(152)(34)$ may be written as $(12)(3)(34)(25)$. Clearly, the choice of base $b=(1,\cdots,n)$ is made for convenience, and we may consider arbitrary ordered lists of $\{1,\cdots,n\}$. By identifying the transposition $(a\,i)$ with the directed edge $i \to a$, the connection between permutations and random recursive forests is clear.

We note that this representation of a permutation with a SGS is exactly the logic behind the Knuth (Fisher-Yates) shuffle which gives an $O(n)$ algorithm for generating a permutation on n elements. The base is the initial value of the list that is shuffled. A random permutation $\pi = (a_2b_2)...(a_nb_n)$ is generated in the SGS representation. The Knuth shuffle generates a permutation by applying the transpositions using the left group action so that (a_nb_n) is applied first.

We note that the right action, which applies (a_2b_2) first, may also be used to generate a random permutation. It has the desirable property that it lets one draw a sequence of uniformly (but not independently) distributed random permutations on $S_1, S_2, ...$. However, the left action allows one to obtain a random sample without replacement of size k in $O(k)$ time by applying k transpositions from the SGS representation.

The left action also provides a different perspective to the CRP by yielding a “reversed” Chinese restaurant process. In this case, the sequence of transpositions in the SGS representation are applied in reverse order, or equivalently in the recusive representation, the edges are examined in reverse order with respect to the base. Suppose n customers arrive in some order at a restaurant, but no one is seated until the last customer arrives. The last person to arrive either chooses to sit at a new table with probability proportional to some parameter θ or makes a decision to befriend and sit with one of the previous $n-1$ customers with probability proportional to 1. When a customer sits down, that person and all his friends sit at the same table. In this representation the connection to the coalescent model is clear. If the customers arrive in random order, then the first choice will either coalesce a random pair with probability proportional to $n-1=\binom{n}{2}\times(2/n)$ or sit at a new table (mutate) with probability proportional to $\theta=(n\theta/2)\times(2/n)$.

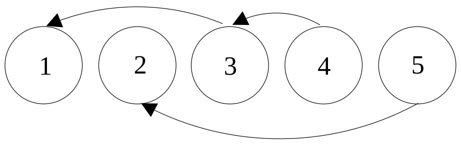
Figure 3.4 shows the relationships among a permutation in its SGS representation, random recursive forests, and the action of left and right multiplication by a permutation in its SGS representation.

One can make a more explicit connection between the permutations and the recursive forest using the cycle representation of a permutation. Iterate through nodes in the order specified by the base. If a node is a root of the forest, then it denotes the start a new cycle. Otherwise, insert the node to the right of the node it points to. This corresponds to left multiplication by a transposition. If one inserts to the left, then it corresponds to right multiplication.

A simple application of the SGS representation is an algorithm for generating permutations with k cycles in S_n. Choose a parameter θ to solve $k=\sum_{i=0}^{n-1}\frac{\theta}{\theta+i}$. This gives a $CRP(\theta)$ distribution where k is the number of expected tables and the equivalent cycle representation has k cycles. Until k is the number of self-transpositions in the SGS representation, run a Gibbs step to update one transposition in the SGS representation. Output the final permutation. The usual approach involves calculating Sterling numbers of the first kind as part of its recursive procedure (Wilf and Nijenhuis (1989)).

1	5	3	4	2
1	5	4	3	2
4	5	1	3	2
4	5	1	3	2
4	5	1	3	2

(1 4 3) (2 5)



1	*	*	*	*
1	2	*	*	*
3	2	1	*	*
3	2	4	1	*
3	5	4	1	2

(1 3 4) (2 5)

Figure 3.1: These figures illustrate the relationships with the permutation $(11)(22)(13)(34)(25)$ which is in the SGS representation with respect to the base $(1, 2, 3, 4, 5)$. The left and right figures represent the action of left and right multiplication by the permutation with the cycle representation of the permutation given at the bottom. Left multiplication corresponds to the Knuth shuffle. The two actions have different properties. Left multiplication maintains the invariant that after applying the k^{th} transposition, the last k elements are a uniform random sample without replacement. Right multiplication maintains the invariant that the first k elements are a uniformly drawn random permutation of the first k elements. In both of the left figures, the final state contains two cycles when regarded as a permutation in one-line notation, (25) and a cycle containing $1, 3, 4$. The figure on the right shows the random recursive forest representation. The connected components of the forest correspond to cycles of the related permutations.

Change of base and invariance of the connected components

As noted above, given a fixed base b, a permutation π has an SGS representation F and vice versa. The forest F may be regarded as a permutation with respect to the base $(1, ..., n)$. and the base b itself may be regarded as permutation. When we do so, the relationship is codified by the group action of conjugation $b \circ F = bFb^{-1}$.

This relationship gives a method for changing the base. To change from base b to b', one solves the equation $b \circ F = b' \circ F'$ to obtain $F' = b'^{-1}bFb^{-1}b' = (b'^{-1}b) \circ F$. While F' is regarded abstractly as a permutation, one simply needs to write down its SGS representation with respect to the base $(1, ..., n)$ to obtain a forest.

The change of base has one particularly important property with respect to the forest representation, namely that the vertex sets of the connected components of the forest do not change. This is obvious from the cycle representation since the cycles represent the connected components of the forest. Since $b' \circ F' = b \circ F$, the cycles are identical, and the connected components must the same vertices. Thus, samplers which operate on the forest representation with respect to one base can change the base and obtain a new forest representation which induces the same partition structure as the old forest.

We note that this is not the only choice which preserves the partition structure under a change of base.

Polya Urn Processes

When there is a Polya Urn process that generates a partition, the SGS or recursive forest representation is often a particularly useful representation of the object. Imagine that the balls in the Polya urn are numbered in addition to being colored. The base describes the order in which the balls or points appear while the color denotes the cluster that the point belongs to. The edges give which numbered ball was drawn at each time step. Thus, the recursive forest describes a process on the numbered balls. One may color the balls the urn is initialized with, and the edges will determine the colors of the rest of the balls.

These Polya urn processes along with the notion of exchangeability lead to Gibbs samplers for many distributions on partitions of interest. To resample the color of a random ball, one first removes that ball. By exchangeability, this ball may be treated as if it was the last ball. One then chooses from the remaining balls in the urn and replaces the removed ball with a ball of the same color as the newly selected ball.

The forest representation leads to a generalization of this process. To obtain the Gibbs step described by the Polya Urn, one chooses a new base where the selected ball corresponds to the last element in the base. One then randomly selects a new value for the last element in the SGS representation. This step is equivalent to updating the color of the selected ball.

Beta-Binomial Distribution and Subtrees

We have described sequential processes that generate the combinatorial structure of interest. We now describe a generalization of the stick-breaking process that generates the combinatorial structure. When viewed as a hierarchical processes, it is related to the tree structured stick breaking process of Adams et al. (2010). Using the process, one can obtain coarse to fine grained information about the combinatorial structure without sampling the entire structure. For example, in the case of mixture models, one is only interested in the cluster sizes, or equivalently, the sizes of the maximal subtrees in an arborescence forest and not the trees themselves. In that case, the stick-breaking process allows one to draw a sequence of Beta-Binomial random variables to obtain the tree sizes for the processes defining a CRP.

The hierarchical processes recursively iterates stick-breaking processes to find the sizes of all subtrees, not just the maximally connected ones. The process is interesting not only because of the hierarchical nature but also that it yields interesting marginal distributions for the sizes of subtrees given the location of the subtree's root within the base.

We proceed by describing a hierarchical method for generating arborescence trees, and the associated hierarchical stick-breaking process is a natural consequence. Suppose one is given a set of n_0 points and wishes to draw arborescence forests corresponding to a $CRP(\theta)$ distribution. First, apply the standard stick-breaking process to find the sizes of the maximal subtrees of the arborescence forest. In other words, start with two points, a dummy point b_0 and the first point b_1 in the base and remove b_1 from the set of unassigned points. The first step of the process samples the number of points that will be attached to the first point b_1. Draw $s_1 \sim Beta-Binomial(n_0-1, 1, \theta)$ and randomly sample s_1 points from the set of unassigned points. All these points belong to the arborescence tree rooted at b_1. The remaining unassigned points belong to the dummy node b_0. Update the number of remaining points $n_1 = n_0 - s_1 - 1$. Repeat the process, starting with the first point in the base that is in the unassigned set, and continue until there are no more unassigned points.

Conditional on the points that belong to a maximal subtree, it is trivial to verify that the joint probability factorizes so that the structure of the subtrees are independent of each other and that each subtree is simply a random arborescence tree, so one need not worry about having multiple components. To draw a random arborescence tree on n_s points, note that the second point in the tree must be connected to the first. Thus we can break the arborescence tree into points attached to the second point and points attached to the first point but not the second. This is trivial to do by drawing $Beta - Binomial(n_s - 2, 1, 1) = Uniform(n_s - 2)$ to determine the size of the subtree rooted at the second point.

While not obvious from the hierarchical process, the sequential construction of arborescence forests gives the marginal distribution for the size of the subtree rooted at the i^{th} regardless of whether or not it the root of a maximally connected component. It is easy to see from the sequential procedure that the size of the subtree rooted at i is independent of all points before i and that the size is $Beta - Bernoulli(n_0 - i, 1, i + \theta - 1)$ distributed.

Since all the necessary draws are $Beta - Binomial$, it follows from Kolmogorov's extension theorem that there exists an underlying process giving $Beta$ distributed weights at every node. Thus, if one wishes to draw the number of points that are in a given set S that also belong to the subtree rooted at i, then one simply needs to sample from a $Beta - Binomial(|S \cap \{i + 1, ..., n_0\}|, 1, \theta + i - 1)$. The parameters of the $Beta$ distribution do not change, only the number of trials in the binomial part changes.

We also note that this observation yields a Markov chain for sampling from a CRP law. Choose an random index i from any distribution, and let B be the block associated with i. Divide the points in B into those occurring before i or after i by sampling from a $Hypergeometric(n_0 - 1, |B| - 1, i - 1)$ distribution. Similarly, find the number of points occurring before i for all other blocks as well. Find the size of the subtree rooted at i, and propose to create a new block with probability proportional to θ with i and its subtree or to attach it to block B_j with probability proportional to s_j where s_j is the number of points in B_j occurring before i.

The random split version of this sampler is inefficient for mixture models since a random split takes time proportional to the size of the subtree that is split off is unlikely to be accepted, and a sequential allocation approach takes time proportional to the size of the entire block and will split off small blocks. It is of mathematical interest since it defines a new finite state space Markov chain with a CRP law as its stationary distribution and, as far as the authors know, there is no existing counterpart in the continuous state space with a Poisson-Dirichlet process law. By considering the process on an infinite base, one obtains the continuous state space equivalent.

3.5 Other combinatorial structures

Thus far, we have focused on permutations, representations of permutations, and their related combinatorial structures: partitions and recursive forests. In particular, we have been interested in how the weakly connected components of a recursive forest defines a partition. Clearly, the (weakly) connected components of any (di)graph defines a partition. We need not restrict ourselves to recursive forests. However, some graphs are of greater interest than others due to computational reasons or due to links to processes on infinite, continuous spaces.

We give a few examples of other interesting combinatorial structures.

Swendson-Wang, the Ising model, and the random cluster model

The random cluster model draws a random graph by randomly sampling edges from a graph. The connected components of the graph form clusters. The Swendson-Wang algorithm is a sampler for the Ising model. It draws from a random cluster model given the current configuration of spins. One then randomly labels the clusters of the induced partition as $-1, +1$ with equal probability. Figure 3.2 illustrates the Swendson-Wang algorithm.

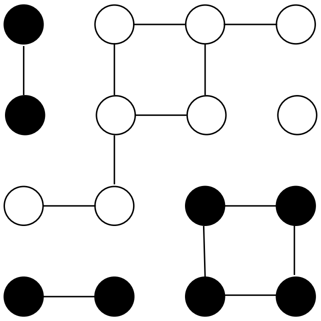


Figure 3.2: Ising model and Swendson-Wang: One alternates between sampling bonds and assigning spins to the connected components formed by the bonds.

Functional digraphs and the ddCRP

The Distance Dependent CRP introduced by Blei and Frazier (2010) defines a distribution on functional digraphs, i.e. digraphs where every node has outdegree 1 including self-loops. They are "functional" digraphs since the edges define a mapping from $[n] \to [n]$, and likewise any function $f : [n] \to [n]$ defines a functional graph. Random recursive forests are a specific type of functional graph in which the mapping f must obey the constraint $f(a) = b \implies a > b$. As usual, the weakly connected components of the graph define a partition.

Undirected graphs, subforests, and matrices

Random recursive forests may be regarded as a random sub-forest of a directed acyclic graph. One may also consider random subforests of undirected graphs. Undirected graphs are of interest due to their connection to Laplacian based semi-supervised learning methods, in particular, the harmonic function solution of Zhu (2005). Given $0-1$ labels, the value of the harmonic function solution at node v is the probability that v is in the same connected component as a 1 label for a random arborescence forest with roots at the labelled points and probability proportional to the product of its edge weights. This fact may be derived from Cramer's rule and an extension to Kirchhoff matrix tree theorem given by Chaiken (1982). Instead of mixture modeling approaches which require sampling or non-convex optimization, the harmonic function solution or the Tikhonov regularized harmonic function solution, has a simple solution as the solution to a linear system. Figure 3.3 gives an illustration of a partition induced by an arborescence forest.

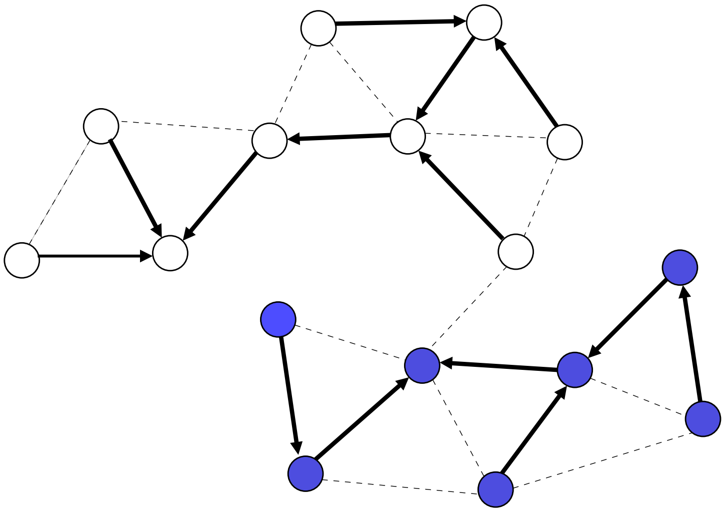


Figure 3.3: Laplacian and arborescence forests: Each arborescence forest on the underlying graph induces a partition. This figure shows one example of an arborescence forest partitioning the vertices into two components.

3.6 Hierarchical Models

Thus far, the discussion has centered around distributions over simple partitions. Beyond simple partitions, nested partitions are of interest as they correspond to hierarchical models. A nested partition is a sequence of partitions $\Pi_1, \Pi_2, ...$ such that Π_i is a refinement of Π_{i-1}. The natural way to generalize recursive forests in order to represent nested partitions is to add colors to the edges where the color indicates at which level of the nested partition the two adjacent points separate into different blocks. Each partition in the nested partition is then represented by the tree induced by the appropriate set of colors.

We start by describing the Hierarchical Dirichlet Process (HDP) and give a corresponding combinatorial representation. This representation leads to a generalization to the ddCRP sampler of Blei and Frazier (2010) to HDPs. We use the notion of a base and exploit the forest structure to introduce further improvements.

More generally, we describe how coagulations and fragmentations lead to natural hierarchical models. We use this to describe several nonparametric hierarchical Bayesian models: the HDP, the Nested Dirichlet Processes (nDP) of Rodriguez et al. (2008), the Nested Chinese Restaurant Processes (nCRPs) of Blei et al. (2010), and Tree-structured Stick Breaking Process of Adams et al. (2010).

Hierarchical Dirichlet Process

The HDP is a hierarchical model in which the data points are already grouped into a set of predefined groups $\mathcal{G}$. The simple two-level HDP has the following generative process for latent class membership variables.

$$\beta \sim GEM(\theta_0) \tag{3.1}$$

$$\mathbf{w}_g \sim DP(\theta\beta) \quad \text{for } g \in \mathcal{G} \tag{3.2}$$

$$z_{ig} \sim \mathbf{w}_g \quad \text{for } i \in \mathcal{I}_g \tag{3.3}$$

where GEM refers to the Griffith-Engen-McCloskey law.

We will regard the GEM process as defining a random measure on $\mathcal{N}$, namely the measure $\sum_{i\in\mathcal{N}} \beta_i \delta_i$. The per group stick weights $\mathbf{w}_g$ define random measures on $\mathcal{N}$ as well. Points x_{ig} and $x_{i'g'}$ belonging to groups g and g' respectively belong to the same cluster if and only if $z_{ig} = z_{i'g'}$.

The corresponding combinatorial process on partitions of n points is the Chinese Restaurant Franchise (CRF) described by Teh et al. (2006). In the restaurant analogy for the CRF, there are $|\mathcal{G}|$ franchised restaurants with n_g customers in the g^{th} restaurant. The franchise has an infinite number of dishes to choose from, and each restaurant has an infinite number of tables, but only one dish is served at each table. Whenever a customer orders a dish, he/she is able to see the popularity of the dish across all restaurants in the franchise. The customers arrive in sequence where the i^{th} customer to arrive at restaurant g chooses to sit at an occupied table t with probability proportional to the number of customers already at that table or chooses to sit at a new table with probability proportional to some parameter θ. If the customer chooses to sit a new table, he/she orders a dish ϕ_t with probability proportional to the number of tables which serve that dish across all restaurants, or he/she orders a dish which has not been ordered yet with probability proportional to some parameter θ_0.

Just like the HDP is a composition of Dirichlet processes, the Chinese Restaurant Franchise is the composition of multiple Chinese Restaurant Processes. We may use this observation to obtain a natural forest representation. Within each group, draw a forest on the customers in each group corresponding to a CRP. Putting all the individual forests together gives a single forest with black edges. The roots of this forest correspond to tables in the CRF analogy. Draw a red colored forest on the roots of this black forest. The induced partition on the roots of the black forest correspond to the partitioning of tables by the dish served on each table.

We note that while we describe everything for the HDP using the one-parameter GEM distribution, it can easily be generalized to the two-parameter version as well. Section 3.7 gives more details.

Sequential procedure for drawing Forests

The CRF gives a natural sequential procedure for drawing forests. Given a new point x_{jg} in group g, connect it to a previous point x_{ig} in the same group g where $i < j$ and color it black with probability $\propto 1$. Otherwise denote it as a root of the black forest with probability $\propto \theta$. If the point is black root, then connect it with a red edge to the k^{th} previous black root with probability $\propto 1$ or designate it a root of the red forest with probability $\propto \theta_0$. The connected components of the forest including both the red and black edges defines the same distribution over clusters as Equation 3.3. Figure 3.4 illustrates the forest representation.

This gives the p.m.f. for the augmented forest representation containing the forest F and a coloring of the edges as red or black.

$$\begin{aligned} p(F) &= \frac{\theta_0^k \theta^r \Gamma(\theta_0)}{\Gamma(r+\theta_0)} \prod_{g \in \mathcal{G}} \frac{\Gamma(\theta)}{\Gamma(\theta+n_g)} \\ &\propto \frac{\theta_0^k \theta^r \Gamma(\theta_0)}{\Gamma(r+\theta_0)} \end{aligned}$$

where k is the number of roots of the red forest, r is the number of roots of the black forest, and n_g is the number of points in group g.

We may use this representation to propose both a new Gibbs sampler for the HDP in the colored recursive forest representation as well as proposing a split-merge sampler for the HDP.

Backward-Forward Forest Gibbs sampler

In the usual Gibbs sampler for a HDP mixture model based on the CRF, each Gibbs step updates the class membership of one point. In the forest representation, a Gibbs step updates the class membership of a point and all points in its subtree. If one had to touch all the points in the subtree in each iteration, this would be a potentially expensive operation. However, one may exploit the underlying forest structure and apply dynamic programming to obtain a sampler that can move multiple points yet each iteration has a cost similar to the usual Gibbs sampler.

For each point we store the sufficient statistic for that point and all the points in its subtree. Thus when the outgoing edge of a point is updated, rather than updating the labels of all of the points in its subtree, one updates the sufficient statistic along with some other bookkeeping variables. The variables needed are given in table 3.6. We note, however, that the descriptions of the varibles are not strict invariants that are maintained at all times.

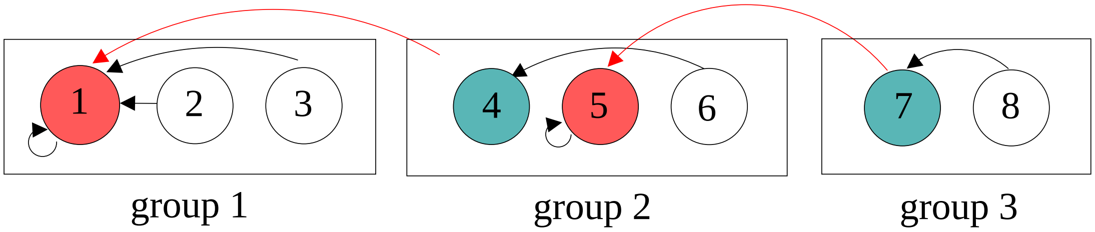


Figure 3.4: Forest representation of a Hierarchical Dirichlet Process (HDP). This figure shows the HDP forest representation for a clustering of eight data points into two blocks where the points are pre-classified into three groups. The black edges form a recursive forest for each of the groups while the red edges form a recursive forest on the roots of the forest induced by the black edges. The red circles are the roots of the forest induced by the red and black edges while the blue-green colored circles are roots of the forest induced by black edges.

variable	description
b	permutation used as the base of the recursive forest
F	array containing the random recursive forest
F_{degree}	array containing the number of red incoming and outgoing edges for each point
F_{stats}	array containing the sufficient statistics of the subtrees
z	array containing the assignment of points to clusters
i	location of the point that the Gibbs sampler is currently updating
L_{cg}	list of points in cluster c and group g that come before the current point i
L_c^{root}	list of points in cluster c that come before i and are roots of F restricted to black edges
n_{cg}	number of points $\lvert L_{cg}\rvert$
n_c^{root}	number of points $\lvert L_c^{root}\rvert$
R	the number of roots of the black forest

Like the usual Chinese Restaurant Franchise sampler, this requires a nontrivial number of bookkeeping variables. This setup, however, does not require keeping track of the complicated assignment of points to individual tables and groupings of tables at each level of the hierarchy. The CRF requires one to keep track of a nested partition with unknown structure formed by the assignments to tables at various levels. In the forest representation, the structure of the bookkeeping variables is fixed by the structure induced by the groups.

The sampler and variables presented here are for the simple two-level HDP, but additional levels will only require changes to F_{degree}, L_{cg}, and R. F_{degree} will be a $\ell \times n$ matrix containing the degrees for each of the ℓ colors corresponding to levels in the hierarchy. Likewise, R will be an array containing the number of roots for each color. In the case where there are more than two levels, the groups themselves form a nested partition. Thus while the description of L_{cg} is still the same, updating the cluster assignment of one point will require updating L_{cg} for multiple groups. There is an alternative representation for L_{cg} which only stores the list of points for groups that are at the leaves of the nested partition and counts for the internal nodes, but one still needs to update counts for every level of the hierarchy when the cluster membership of a point changes.

The sampler also does not distinguish between updating table versus point assignments. One always operates on single points and pulls all of its children along in an update. The information encoded by F_{degree} is whether or not a Gibbs update can give the i^{th} point an outgoing black edge. If there are any incoming red edges then the outgoing edge must also be red.

The sampler proceeds in two stages, a backward sweep followed by a forward one. The backward sweep does not require an explicit forest; it instead builds one up. The forward pass will start off with only the forest structure and reconstruct the explicit assignment of points to clusters. For convenience in exposition, we will take the base to always be $(1, ..., n)$.

Algorithm 1 $BackwardSweep$

```
Require: Parameters $\theta_0, \theta$ for the HDP and a prior $\pi$ on the mixture component parameters. The
    variables in table 3.6 except $F, F_{stats}, F_{degree}$. Instead of containing the actual red degree of
    each point, $F_{degree}$ only contains the outgoing red degree. Instead of $F_{stats}$, one simply needs
    the sufficient statistic $S_c$ for each cluster $c$.
Ensure: $F, F_{stats}, F_{degree}$ match the descriptions in table 3.6.
Initialize $F_{stats}$ to contain the sufficient statistic for each singleton.
for $i = n \to 1$ do
    Let $g$ be the group $x_i$ belongs to and $c = z_i$ be the current cluster allocation for $x_i$
    Remove $F_{stats}(i)$ from $S_c$ as well as $x_i$ from $L_{cg}$
    If $F_{degree}(i) > 0$ decrement $F_{degree}(i)$ and $R$ by one and remove $x_i$ from $L_c^{root}$.
    Calculate the Gibbs probabilities of attaching to cluster $c'$ via a black edge via

        $p(c') \propto n_{c'g} f(x_i|S_{c'}, \pi) 1(F_{degree}(i) = 0)$

    Calculate the Gibbs probabilities of attaching to cluster $c'$ via a red edge via

        $p(c') \propto \frac{\theta}{R + \theta_0} n_{c'}^{root} f(x_i|S_{c'}, \pi)$

    Calculate the Gibbs probabilities of creating a new cluster $c'$ and adding a red self-loop

        $p(c') \propto \frac{\theta\theta_0}{R + \theta_0} n_{c'}^{root} f(x_i|S_{c'}, \pi)$

    Choose a cluster $c_{new}$ and color $\ell$ according to the Gibbs probabilities
    Choose a destination point $j$ for the outgoing edge from the set $L_{c_{new}g}$ or $L_{c_{new}}^{root}$ depending
        on the color.
    Update $F_{stats}(j)$ with $F_{stats}(i)$, $F(i) = j$, $F_{degree} += 1(\ell = red)$, $R += 1(\ell = red)$
end for
```

We note that the calculation of Gibbs probabilities and selection of an outgoing edge are identical in the two sweeps, but the bookkeeping maintains different invariants. The backward pass

Algorithm 2 $ForwardSweep$

```
Require: Parameters $\theta_0, \theta$ for the HDP and a prior $\pi$ on the mixture component parameters.
    $F, F_{stats}, F_{degree}$ match the descriptions in table 3.6.
Ensure: The variables in table 3.6 except $F_{stats}$ match the descriptions given.
Start a new cluster $c = 1$ with sufficient statistic $S_1 = F_{stats}(1)$.
Set $z_1 = 1$ and initialize $L_{1g} = L_1^{root} = \{x_1\}$ where $g$ is the group $x_1$ belongs to.
for $i = 2 \to n$ do
    Let $g$ be the group $x_i$ belongs to
    If $F(i)$ is null and $i$ does not start a new cluster, let $c = z_{F(i)}$, the cluster that $i$'s parent belongs to, and remove $F_{stats}(i)$ from $S_c$.
    If $F_{degree}(i) > 0$ decrement $F_{degree}(i)$ and $R$ by one.
    Calculate the Gibbs probabilities of attaching to cluster $c'$ via a black edge via
        $$p(c') \propto n_{c'g} f(x_i|S_{c'}, \pi) 1(F_{degree}(i) = 0)$$
    Calculate the Gibbs probabilities of attaching to cluster $c'$ via a red edge via
        $$p(c') \propto \frac{\theta}{R + \theta_0} n_{c'}^{root} f(x_i|S_{c'}, \pi)$$
    Calculate the Gibbs probabilities of creating a new cluster $c'$ and adding a red self-loop
        $$p(c') \propto \frac{\theta\theta_0}{R + \theta_0} n_{c'}^{root} f(x_i|S_{c'}, \pi)$$
    Choose a cluster $c_{new}$ and color $\ell$ according to the Gibbs probabilities
    Choose a destination point $j$ for the outgoing edge from the set $L_{c_{new}g}$ or $L_{c_{new}}^{root}$ depending on the color.
    Add $x_i$ to $L_{c_{new}g}$ and if $ell$ is red then also add it to $L_{c_{new}}$.
    Update $F(i) = j$, $F_{degree}+ = 1(\ell = red)$
    Update $z_i = c_{new}$ and update $S_{c_{new}}$ with $F_{stats}(i)$.
end for
```

ensures that for all points after i which the sweep has already covered, all the variables concerning the forest are correct, and these will not be further updated in the backward pass. The forward pass ensures all the variables except F_{stats} are valid for the points before the current step i which have already been covered by the sweep.

Split-merge sampler

We now present a sequentially allocated split-merge algorithm for the HDP. Since this is a Metropolis-Hastings procedure, the only detail that needs to be filled in is the proposal distribution. We do this by simply adapting the $ForwardSweep$ procedure. In the version of the split-merge sampler we present, the main changes to the procedure are that the number of clusters is fixed at 1 or 2 depending on whether a merge or split is proposed, and we start with an “empty” allocation for each proposal similar to the sampler given by Dahl (2003).

Algorithm 3 $ProposeSplit$

```
Require: A connected component $B$ of the forest with both red and black edges.
Permute the order of points in $B$ uniformly at random. WLOG label them as points $x_1, ..., x_{|B|}$.
Denote $x_1, x_2$ as the first two points of new blocks $B_0, B_1$ respectively, and update the relevant bookkeeping variables.
Initialize $F_{stats}$ to sufficient statistics for singletons and $F_{degree}(j) = 0$ for all $j > 2$.
Perform a forward Gibbs sweep starting at the third point with the following differences. One cannot propose a new cluster, and one does not need to remove $F_{stats}(i)$ from the parent cluster's sufficient statistic. The sequential allocation probabilities are stored.
```

A $ProposeMerge$ step is identical except one starts with two components B_0, B_1 and lets $B = B_0 \cup B_1$ be their union, and instead of starting two clusters with x_0 and x_1, one only starts one cluster x_0. If a proposal is accepted, then the chain simply overwrites the state of the components of the forest that are affected by the proposal.

We note that Wang and Blei (2012) have recently proposed a split merge sampler for the HDP. However, our split-merge sampler is substantially different from theirs. In the context of the CRF, their formulation applies split-merge moves to the tables but their split-merge steps cannot separate customers sitting at a common table. Furthermore, since for tables that share the same dish or mixture component parameter, the assignment of customers to tables is random, the split-merge moves pick from a restricted set of random splits. In our formulation, a split-merge step both reassigns customers to tables as well as tables to dishes. We believe this addresses the fundamental problem with split-merge samplers for HDPs.

Choosing to split or merge

We have now specified the proposal distribution given a choice to split or merge has already been made. We briefly discuss the possible choices for choosing whether to split or merge and suggest a possible improvement to the existing choice.

Existing split-merge samplers for the CRP make a choice to split or merge by choosing a random transposition uniformly at random and checking whether the two elements in the transposition are in the same or different blocks. This proposal is based on the transposition random walk which yields a $CRP(1, 0)$ distribution and ignores the role of the parameter θ. One may introduce a simple change to the proposal to include the effect of θ. First, take a size-biased pick B_0 for a block. Propose to split with probability $\propto \theta|B_0|$, or propose to merge with block B_i with probability $\propto |B_i|$. This change is especially relevant for HDP split-merge samplers since probability of creating a new root is $\propto \frac{\theta\theta_0}{R+\theta_0}$. As more roots are created, the probability of creating a new cluster becomes smaller, and too many splits may be proposed using the naive method. In our experiments, we used the simple rule based on the transposition random walk. We also note that an alternate move to splitting and merging is to exchange the elements of two blocks. This has been proposed before (Thibaux (2008)) though not implemented.

Evaluation

We implemented a Gibbs sampler for the usual CRF representation, a Gibbs sampler in the forest representation, and a split-merge sampler with various optimizations introduced in chapter 4. The implementations were all done in python. We take the same comparison metric as Dahl (2003) to compare the samplers, namely effective sample size on the clustering entropy statistic. To compare the computation cost of the different samplers, we use the number of likelihood calculations performed. This is reasonable since the time spent in each sampler was dominated by the time spent on likelihood calculations. Since we introduce a few further optimizations for the split-merge samplers in chapter 4, we briefly summarize the results here and leave the full comparison for the next chapter.

We find that the Gibbs sampler on the forest representation generally performs substantially better than the usual Gibbs sampler, both in terms of effective sample size per unit of computation as well as finding local modes faster in the burn in phase. However, it did not dominate the usual Gibbs sampler for every data set. The split-merge sampler with the additional optimization of early rejection for merge moves typically improved upon the Gibbs sampler in the forest representation. Without early rejection, datasets with well defined clusters did not see a benefit or performed worse with split-merge moves.

Discussion on different HDP samplers

We have presented two new algorithms for sampling from HDP mixture models. The first is a Gibbs sampler based on the random recursive forest representation, and is a cousin to the ddCRP sampler of Blei and Frazier (2010) for CRP mixture models. A key feature of the algorithm is that it exploits the tree representation to make each Gibbs iteration take constant time by using dynamic programming. The second algorithm is an extension to the split-merge algorithms for CRP mixture models described by Jain and Neal (2004) and Dahl (2003). It goes beyond the naive approach of Wang and Blei (2012) which applies the usual CRP split-merge to one level of the hierarchy conditional on the rest.

Both algorithms propose moves that can split off more than one point. The split-merge sampler uses a sequential allocation rule to propose a good split while the forest-based Gibbs sampler proposes random splits of various sizes. As Jain and Neal (2004) point out, splitting uniformly at random yields a poor proposal for a split-merge sampler since the splits are unlikely to be accepted but still take time proportional to the merged blocks. Although split proposals in the forest-based Gibbs sampler do not make use of the value of the data points, the randomness allows the splits to be performed cheaply at constant cost if the computational cost of evaluating the likelihood is constant given a sufficient statistic. Furthermore, the forest-based Gibbs sampler does not perform a uniform random split and instead proposes smaller splits.

We give a heuristic justification that the size of the splits proposed by the forest-based Gibbs sampler and ddCRP are of an appropriate size for CRP mixture models. There are two problems associated with the split size. The first is if the split size is large enough to overcome the effect of the prior and adding additional parameters to the model. The second is that the proposed split actually reflects the true desired split. The usual CRP Gibbs sampler only moves one point at a time and suffers from the first problem. The split-merge sampler using uniform random splits proposes large, expensive random splits in which the resulting blocks from the split are too similar to each other. To address the first problem, we note that the choice of whether or not to split a block is akin to the problem of model selection. If k is the number of parameters to be added and n is the size of the block, then the Bayesian Information Criterion (BIC) adds a model selection penalty of $\frac{k}{2}\log n$ to the maximized log-likelihood. It is a Laplace approximation to the marginal likelihood and will pick the true model as $n \to \infty$ if one of the models is true. Given a fixed base, the size of the proposed split for the i^{th} element of the base is distributed $1 + Beta - Binomial(n-i, 1, i-1)$. Thus, the expected split size Z of a randomly chosen element in the block is $\mathbb{E}Z = \frac{1}{n}\sum_{i=1}^{n}\left(1 + \frac{n-i}{i}\right) \approx \log n$. Thus the split sizes are appropriately sized to overcome the penalty of adding additional parameters.

If one large block was generated from a two component mixture with mixing weights $p, (1-p)$, then if the split size is large, then asymptotics dictate that the two blocks from the split will be indistinguishable since they are, in fact, drawn from the same distribution. In the block that is split off is of size $\log n$, then the probability that those $\log n$ points all belong to the component with weight p is approximately $p^{\log n} = \frac{1}{pn}$. Since one sweep of the forest-based Gibbs sampler proposes n random splits, if all splits were of size $\log n$, then the expected number of splits is approximately a constant $1/p$. For a split-merge sampler with a random split proposal, one sweep consists of only one split proposal, so even if it proposed splits of size $\log n$, it would require $O_p(n)$ sweeps before all the points being split off come from the same cluster. Thus, both the relatively small size of the splits plus the efficiencies gained from dynamic programming give a heuristic justification that the forest-based Gibbs sampler chooses splits that are not too large.

We can also heuristically show that the forest-based Gibbs sampler will never do much worse than the CRF Gibbs sampler. This is since, for the regular CRP case, half of the moves proposed by the forest-based Gibbs sampler only move a singleton. For a random recursive tree of size n, the size of the subtree rooted at the i^{th} node is distributed $1 + Beta - Binomial(n-i, 1, i-1)$. Thus, the probability of the subtree being a singleton is $\frac{i}{n}$. Summing over i from 2 to n, one finds that the expected number of singleton subtrees is $(n+1)/2$. Since subtrees can only attach to nodes occurring earlier in the base, each proposed singleton move in the forest Gibbs sampler is not directly comparable to a move in the usual CRP Gibbs sampler. However, since the forest-based Gibbs sampler chooses a base uniformly at random, the proposed singleton moves for nodes occurring in the second half of the base are roughly comparable to CRP Gibbs moves. The number of singleton move proposals in the second half of the base is $\approx 3n/8 > n/3$. Thus, one may expect that a forest-based Gibbs sampler will never do much more than 3 times worse than a Gibbs sampler. This number is also consistent with our experimental results.

Forest representations, sequential constructions, and MCMC samplers

In this section, we have exploited two properties of HDPs to create new samplers. First, we exploited the forest representation of an HDP to obtain a Gibbs sampler. Second, we exploited a sequential procedure for drawing a recursive forest to obtain a sequentially allocated split-merge sampler. Furthermore, with the forest Gibbs sampler, one may construct an alternative split-merge sampler using the restricted Gibbs sampling idea in Jain and Neal (2004) and Jain and Neal (2007). These ideas are not specific to the HDP. They may be applied to any distribution over forests or recursive forests with sequential constructions.

We will present forest representations and sequential constructions for the priors of several nonparametric Bayesian models. Using the same ideas presented in this section, one can obtain forest Gibbs samplers and split-merge samplers for these other models as well.

3.7 Other models: Fragmentations and coagulations

With the forest representation for a HDP, we have shown how a alternate combinatorial representation leads to new samplers. The goal of this section is to demonstrate how similar ideas may be applied to other processes which are described as stick-breaking processes, but have an underlying combinatorial representation. In general, most processes encoding a hierarchical model can be described as some composition of fragmentation and coagulation kernels.

Coalescent and coagulations

The main property of a coagulation is obvious from its name, given a coagulation process with partition Π_t at time t, then $\Pi_{t+\Delta}$ is a coarsening of Π_t for $\Delta > 0$, or equivalently, Π_t is a refinement of $\Pi_{t+\Delta}$. The time-reversal of a coagulation yields a fragmentation, a process on partitions such that $\Pi_{t+\Delta}$ is a refinement of Π_t for $\Delta > 0$.

The canonical example of a coagulation is Kingman's coalescent (Kingman (1982)) which is commonly used in population genetics for describing the ancestry of a set of alleles. At each step of the jump chain of Kingman's coalescent, two blocks of the partition are randomly chosen and merged. The connections between Kingman's coalescent and the Chinese restaurant process are well known. See, for example, Pitman and Picard (2006). If one builds a coalescent tree and marks the branches using a $Poisson(\theta)$ process, then the partitioning of the leaves of the tree into distinct alleles is $CRP(\theta)$ distributed. Thus, one can obtain a sequence of partitions $\{\Pi_\theta\}$ indexed by the mutation rate θ which is a realization of a fragmentation process. The limits Π_0 and Π_∞ give a partitioning into all singletons or a single large block. We may view the CRP as either a fragmentation of a partition with all points in a single block, or as a coagulation of n points. Teh et al. (2008) use this construction of a nested partition as the basis for a hierarchical clustering method.

HDP as a fragmentation-coagulation

We now cast the HDP as a fragmentation followed by a coagulation. With the HDP, we start with an a priori partitioning Π of points into groups. The first step of the HDP independently fragments each block $B_i \in \Pi$ using a CRP to obtain a refinement of Π . The combinatorial representation after this stage is the black forest with no red edges or in the case of the CRF, the partitioning of customers to tables. The second step of the HDP is the key idea which allows different groups to share parameters and borrow strength from each other in mixture modeling. This step is a coagulation in which the blocks of the refinement are coagulated with a CRP. In the forest representation, this corresponds to the red edges connecting components of the black forest. In the CRF representation, this correspondence is less obvious.

Nested Dirichlet Process

The nested Dirichlet Process (nDP) is another nonparametric Bayesian model which allows different groups to share parameters. In the original formulation given by Rodriguez et al. (2008), a family $\mathcal{F} = \{F_1, ..., F_g\}$ of distributions gives the law of a nested Dirichlet Process mixture $nDP(\theta_0, \theta_1, H)$ if

$$\begin{aligned}
G_k^* &\sim DP(\theta_0 H) \\
\pi* &\sim GEM(\theta_1) \\
G_j &\sim \sum_k \pi_k^* \delta_{G_k^*} \\
F_j(\cdot|\phi) &= \int f(\cdot|\phi, \eta) dG_j(\eta).
\end{aligned}$$

where f is a density function parameterized by ϕ, η.

We first reformulate the model specification in a way that allows easy identification of a prior over nested partitions.

$$\begin{aligned}
q^{(k)} &\sim GEM(\theta_0) \\
\pi^* &\sim GEM(\theta_1) \\
A_j &\sim \pi_k^* \\
B_i &\sim q^{(A_{J(i)})} \\
\eta_{ab} &\sim H \\
G_j &= \sum_b q_b^{(A_j)} \delta_{\eta_{A_j b}} \\
F_j(\cdot|\phi) &= \int f(\cdot|\phi, \eta) dG_j(\eta)
\end{aligned}$$

where as before, we consider GEM as inducing a random measure on $\mathcal{N}$. The latent variable A_j for group j chooses the mixture component that G_j belongs to. In this case each mixture component is a random measure itself. The latent variable B_i picks out a component of the random measure G_j where $j = J(i)$ is the group that point i belongs to.

The result of this process yields a coagulation-fragmentation process. One may partition the points by three different cluster membership variables. The first partition Π_1 is based on $J(i)$, the fixed assignment of points to groups. The second partition Π_2 is based on $A_{J(i)}$ which is a coagulation of the partition formed by groups. The third partition Π_3 is based on the pair $(A_{J(i)}, B_i)$. It is a fragmentation of second partition since the B_i refine the partition determined by the $A_{J(i)}$.

This proves that the nDP may be constructed via a coagulation-fragmentation process. One forest representation for the nDP is obvious given this description. One first builds a random recursive forest on the groups. This gives the partition Π_1. Given Π_1, build a random recursive forest on the points within each block of the partition. Figure 3.5 illustrates the representation.

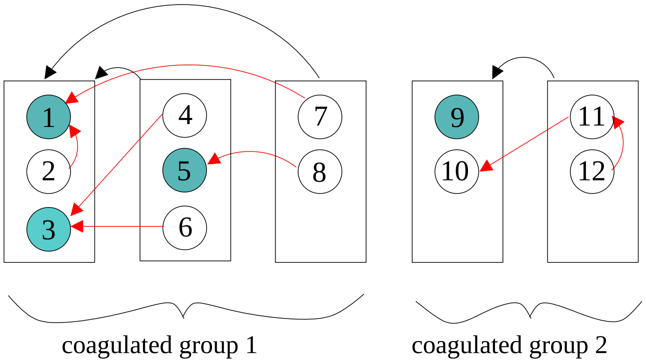


Figure 3.5: Forest representation of a nested Dirichlet Process: Groups are first coagulated via a CRP. The points within each coagulated group are then fragmented by CRPs. The resulting partition is $\{1,2,7\},\{3,4,6\},\{5,8\},\{9\},\{10,11,12\}$. No edges may be formed between points in separate coagulated groups.

We may also provide a sequential procedure for drawing forests for the nDP. Order the points according to some base. For the i^{th} point, if it is the first point from its group $J(i)$, then connect that group to one of the previously encountered groups with probability $\propto 1$ or mark it a root with probability $\propto \theta_1$. The assignment $A_{J(i)}$ then is determined. Connect point i to one of the previously encountered points which are also assigned to $A_{J(i)}$ with probability $\propto 1$ or label i as a root with probability $\propto \theta_0$. We note, however, that the sequential procedure for the nDP may not yield a good sequentially allocated split-merge sampler. The reason is that the outgoing edge for a given group constrains the edges of all the points in that group. Thus, choosing another group to coagulate with before inspecting the values of all the points in the given group may lead to bad proposals.

Nested Chinese Restaurant Process

The nested Chinese Restaurant Process (nCRP) introduced by Blei et al. (2010) gives a potentially infinite hierarchical model for topics. The nCRP induces a nested partition. We may describe the process as the following pure fragmentation process. Start with a partition Π_0 with all the documents in a single cluster. From partition Π_{i-1}, generate a refinement by independently fragmenting each block of Π_{i-1} using a CRP.

One combinatorial representation of the nested partitions induced by the nCRP is obvious: letting each block of each partition be a node in a graph, connect each block with its "parent" block in the fragmentation process. We can also give a more compact representation where the vertices of the graph are the data points or documents themselves, rather than blocks of a partition.

Start by drawing a random recursive tree with the documents as vertices. At the first step, cut the outgoing edge of the i^{th} document with probability $\frac{\theta}{\theta+i-1}$. Label the cut edges with 1 to denote the step at which they were cut. This gives a $CRP(\theta)$ partition. At step t, repeat the process for each of the connected components of the partition obtained at step $t-1$. In this case, the i^{th} document refers to the i^{th} document within a connected component. Each repetition of the cutting process induces a partitioning of a component into smaller components which is $CRP(\theta)$ distributed.

We note that there is only a single recursive tree drawn as the start of the process. One does not need to draw a new recursive tree at each step, even though the nCRP arises from fragmenting each block independently. This makes representing the nCRP simple since, like the HDP and CRP, the underlying graph describing the partition may be represented using only a single, colored outgoing edge for each vertex where the color denotes the level at which the edge is cut..

This representation yields a natural Gibbs sampler in the recursive forest representation for the nCRP. At each step of the sampler, a new outgoing edge and the level at which it is cut are drawn. The given construction for drawing a random recursive forest describing a nCRP distributed nested partition is a sequential procedure which may be used to form a split-merge sampler as well.

Pitman-Yor as a new hierarchical model

Thus far, we have described hierarchical models formed by fragmentation-coagulations, coagulation-fragmentations, and pure fragmentations. None of these exploit the relationship between fragmentation and coagulation. It is, in fact, not a surprise that they are not able to since currently there is no known clear and simple way to calculate probabilities for the time-reversed partitioning processes for the given models. This leads us to examine one case where the fragmentation-coagulation duality is understood, which is the family $\{CRP(\theta, \alpha) : 0 \leq \alpha < 1\}$ where $CRP(\theta, 0)$ denotes the one-parameter CRP. In this case, one may construct a single process $\{\Pi_\alpha^\theta\}_\alpha$ on partitions of n, such that Π_α^θ is $CRP(n, \theta, \alpha)$ distributed.

The following fragmentation-coagulation relationship holds (Pitman and Picard (2006)). For

$0 < \alpha < 1, 0 \le \beta < 1, \theta > -\alpha\beta$,

$$(\theta, \alpha) \underset{(-\alpha\beta,\alpha)-frag}{\overset{(\theta/\alpha,\beta)-coag}{\leftrightarrows}} (\theta, \alpha\beta). \tag{3.4}$$

In other words, given a $CRP(\theta, \alpha\beta)$ distributed partition Π and applying a $CRP(-\alpha\beta, \alpha)$ independently on each of the blocks gives a refinement Π' of Π that is $CRP(\theta, \alpha\beta)$ distributed. Likewise, given a $CRP(\theta, \alpha)$ distributed partition and coalescing its blocks according to a $CRP(\theta/\alpha, \beta)$ gives coarsening which is $CRP(\theta, \alpha\beta)$ distributed.

A special case of interest where $\beta = 0$ is

$$(\theta, 0) \underset{(0,\alpha)-frag}{\overset{(\theta/\alpha,0)-coag}{\leftrightarrows}} (\theta, \alpha). \tag{3.5}$$

Forests and fragmentation-coagulation

We now give a simple combinatorial argument using random recursive forests that proves the special case in equation 3.5 as well as gives a representation that may be used in Markov chain samplers.

We generate a colored forest via a fragmentation process and show that it results in the same sequential probabilities as a $CRP(\theta, \alpha)$ processes. First generate an uncolored random recursive forest that induces a $CRP(n, \theta, 0)$ distribution and then color the edges as follows. Proceed in the order given by the base for the random recursive forest. If the outgoing edge of the i^{th} node links to a red root or a node with a red outgoing edge, then color it red with probability α; otherwise, color the edge black. The probability that i^{th} outgoing edge connects to a previous component of size m is $\frac{m}{\theta+i-1}\left(1 - \frac{1}{m}\right)\alpha = \frac{m-\alpha}{\theta+i-1}$ which is precisely the sequential probability from the usual construction for a $CRP(n, \theta, \alpha)$. By removing the red edges and considering only the black edges, we fragment the $CRP(n, \theta, 0)$ tree to obtain a $CRP(n, \theta, \alpha)$ tree. This proves the direction that a $CRP(\theta, \alpha)$ process can be obtained from a $CRP(\theta, 0)$ process via a $CRP(0, \alpha)$ fragmentation.

Consider the following forest generating process. Let n points arrive in a given order. The j^{th} point may (1) attach via a black edge to an earlier point $i < j$ with probability proportional to 1 if i is not a root and $1 - \alpha$ if it is, (2) attach to itself with probability $\propto \theta$, (3) draw a red edge to an earlier root r with probability $\propto \alpha$. The sequential probabilities for drawing edges matches the fragmentation process above, so we have a sequential procedure for drawing "Pitman-Yor forests."

The proof that a $CRP(\theta/\alpha, 0)$ coagulation of a $CRP(\theta, \alpha)$ distributed partition yields a $CRP(\theta, 0)$ distributed partition is similar. Start with a black colored forest corresponding to a $CRP(\theta, \alpha)$ partition. Proceeding in the order dictated by the given base, consider the roots of the black forest. At the $(i+1)^{th}$ root draw a self-loop with probability $\frac{\theta/\alpha}{i+\theta/\alpha} = \frac{\theta}{i\alpha+\theta}$. We find that the resulting sequential probabilities for generating the partition defined by the connected components of the forest with both red and black edges matches a $CRP(\theta, 0)$.

This construction, like previously discussed constructions, gives a convenient representation of a nested partition in terms of a recursive tree with colored edges. However, it only covers a special case of the coagulation-fragmentation duality relationship. We can calculate the general sequential probabilities obtained by composing a CRP fragmentation and CRP coagulation process

Suppose the outgoing edges for the first i vertices have already been drawn. Let k_i be the number of black roots among the first i points, and let k'_i be the number of red roots among the first i points. The vertex $i+1$ is designated a new root with probability

$$\frac{\theta_0 + \alpha_0 k'_i}{k_i + \theta_0} \frac{\theta + \alpha k_i}{i + \theta}$$

In the case where $\theta_0 = \theta/\alpha, \alpha_0 = \beta$, i.e. the coagulation process is $CRP(\theta/\alpha, \beta)$, then the incremental probability of creating a new root is $\frac{\theta_0+\alpha_0 k'_i}{i+\theta} \frac{\theta+\alpha k_i}{\theta_0+k_i} = \frac{\theta+\alpha\beta k'_i}{i+\theta}$ which is exactly the same as for a $CRP(\theta, \alpha\beta)$. One may similarly calculate that the probability of creating attaching vertex $i+1$ to a given component matches the corresponding probability for a $CRP(\theta, \alpha\beta)$ as well. This shows the direction $CRP(\theta, \alpha) \xleftarrow{(\theta/\alpha,\beta)-coag} CRP(\theta, \alpha\beta)$. Simple algebra also shows that these parameters are the only ones such that a CRP distributed coagulation of a CRP generated partition yields another CRP.

To prove the opposite direction of equation 3.4, we calculate the sequential probability of the $(i+1)^{th}$ point starts a new block given that the process is a $CRP(\theta_0, \alpha_0)$ fragmentation of a $CRP(\theta, \alpha\beta)$. Let n_{ic} be the number of nodes in component c of the $CRP(\theta, \alpha\beta)$ distributed partition when restricted to the first i data points. Let k_i be the number of components of the $CRP(\theta, \alpha\beta)$ distributed partition and k'_{ic} be the number of fragments in component c when restricted to the first i data points and $k'_{i\cdot} = \sum_c k_{ic}$ be the total number of fragments. The sequential probability for labelling the $(i+1)^{th}$ point a root is

$$\frac{\theta + \alpha\beta k_i}{i + \theta} + \sum_{c=1}^{k_i} \frac{n_{ic} - \alpha\beta}{i + \theta} \frac{-\theta_0 + \alpha_0 k'_{ic}}{n_{ic} + \theta_0}$$

When $\theta_0 = -\alpha\beta, \alpha_0 = \alpha$, then the formula simplifies to the incremental probability of creating a new block at point $i+1$ for a $CRP(\theta, \alpha)$.

Modeling and sampling

We now consider the implications to proposing new hierarchical models, the relationships to existing models, and sampling for new and existing models. The general idea we apply is that one starts with a random recursive tree and introduce a fragmentation process to cut edges. This gives a prior for a top-down hierarchical clustering approach where one understands the prior over partitions at every level of the hierarchy. One may do the same in reverse by starting with no edges and adding them in via a coagulation. These processes include the nCRP. Furthermore, using fragmentation-coagulation duality, if the nCRP is extended by allowing for negative θ values in the parameters, then one can obtain a hierarchy of Pitman-Yor distributed partitions. Representations for all these models may be described by a random recursive tree plus a sequence of cuts. Thus, there is always a natural Gibbs sampler for all these models which samples the edge and at what level it is cut.

Consider the following process on partitions. Let T be a random recursive tree. Attach to each node i a pair (U_i, V_i) of $Uniform(0,1)$ random variables. For fixed θ, α, induce a set of cuts as follows. Proceeding according to the ordering of nodes in the random recursive tree,

$$U_i \text{ cuts the outgoing edge of node } i \text{ if } U_i \leq \tfrac{\theta}{i-1+\theta} \tag{3.6}$$

$$V_i \text{ cuts the edge of } i \text{ if the edge connects to a node whose outgoing edge is cut and } V_i \leq \alpha$$

Let the connected components of the resulting forest define the partition $\Pi_{\theta,\alpha}$. From the previous discussion, this partition clearly has a $CRP(\theta, \alpha)$ distribution. Thus, we have defined a stochastic process on partitions $\{\Pi_{\theta,\alpha}\}_{\theta,\alpha}$ indexed by parameters θ, α. Furthermore, any sequence $(\theta(t), \alpha(t))_t$ such that $\theta(t), \alpha(t)$ are both monotonically increasing induces a sequence of nested partitions.

This class of models contains none of the previously described existing models. The reason is that the process is only Markov when θ is held fixed, and it is in the regime where the fragmentation-coagulation duality is understood. If the nCRP is extended to allow for negative parameter values, then the nCRP falls into this regime with appropriately constrained parameter values. Of the models that start with a partition Π_0 and apply Markov fragmentation and coagulation kernels, there are three classes in which there is a representation as a single random recursive tree with cuts. The first is a sequence of fragmentations, of which the nCRP is an example. The second is a sequence of coagulations. When using the CRP as the coagulation kernel, one may regard this as a "reversed" nCRP. Since the family of distributions from the nCRP and "reversed" nCRP models only coincide when fragmentation-coagulation duality holds, the "reversed" nCRP is a genuinely different model. The third class is a single fragmentation followed by a sequence of coagulations, where the HDP is a canonical example of. We note that the nDP does not fall into one of these classes as it is a coagulation followed by a fragmentation.

The relationship between the random recursive tree with cuts representation and Gibbs samplers is obvious. We have also used this relationship in developing an HDP sampler. This representation is also useful for developing auxiliary variable samplers when the α parameter of a CRP is treated a hyperparameter in a model. From the rule defined by 3.6 for cutting edges based on the α parameter, one sees that the coloring of the forest is obtained via a sequence of $Bernoulli(\alpha)$ draws on edges connected to a black root or pointing to a node with a red outgoing edge. This branching process is analogous to the sequential process that defines the negative binomial distribution. The likelihood for α is given by

$$\mathcal{L}(\alpha; F) \propto \alpha^r (1-\alpha)^b$$

where r is the number of red edges and b is the number of black edges pointing to nodes with red outgoing edges. If a conjugate $Beta$ prior is placed on α, then the posterior of α given the forest is also $Beta$. This gives a combinatorial explanation for the auxiliary variable method for sampling α described by Teh (2006).

Tree-structured stick breaking

Adams et al. (2010) introduced the general class of tree-structured stick breaking processes. These processes may clearly be regarded as fragmentation processes on sticks. Thus, they bear strong similarity to the nCRP which is also a pure fragmentation process. Indeed, we can show that both tree-structured stick breaking and the nCRP will generate the same latent distribution over a hierarchy of nested partitions. As a distribution over partitions, the main difference between the two is in a stopping rule which decides the level of the hierarchy that a point belongs to. The tree-structured stick breaking process introduces an explicit stick at each node designating the probability that a point (or document) stops at a node and is not allocated to one of the children. This means that the tree-structured stick breaking process will define a partition, not nested partition, of the points. For the nCRP, points (or documents) always belong to a leaf node and a node stops splitting when it contains a singleton or a maximum depth is reached. For both models, sharing of information comes from the underlying hierarchy. For an nCRP, the word probabilities for a document depend on the entire path from the root to the document. Thus, in addition to the nested partition of the documents, one has a non-nested partitioning of words into nodes in the hierarchy. The parameters associated with each node are independent in the prior distribution. Soft-sharing amongst documents is achieved by the hard coupling of words from different documents in the same node. In the tree-structured stick breaking process, the parameters associated with each node are drawn conditional on the parent node's parameters. Thus, sharing occurs both from the hard coupling of documents in the same node and from soft-sharing induced by the hierarchy of parameters.

We now prove that the nCRP and tree-structured stick breaking generate the same latent nested partition. Simply factor the stick probabilities associated with the tree-structured stick breaking process into a component designating the stopping probabilities and a component for the infinite latent hierarchy. The latent hierarchy will have the same distribution associated with the nCRP. Let ν_ϵ be the stopping probability at node ϵ and β_ϵ be the stick at the parent node of ϵ that denotes the probability that a point would go to epsilon if it were not stopped. The probability of a point belonging to node ϵ is

$$\begin{aligned}\pi_\epsilon &= \nu_\epsilon \beta_\epsilon \prod_{\epsilon' \prec \epsilon} \beta_{\epsilon'}(1-\nu_{\epsilon'}) \\ &= \left(\nu_\epsilon \prod_{\epsilon' \prec \epsilon}(1-\nu_{\epsilon'})\right)\left(\prod_{\epsilon' \preceq \epsilon} \beta_{\epsilon'}\right).\end{aligned}$$

The probability on the right consisting of the β sticks is simply the continuous version of choosing a path in the nCRP. When the sticks are chosen to follow a GEM distribution, then after integrating out the sticks, then one would have exactly the nCRP. The probability consisting of only ν sticks, may be regarded as the probability of stopping at a level of the hierarchy. We illustrate the process and compare it to the nested CRP in Figure 3.6.

Sequential constructions and fragmentation-coagulation duality

So far, our descriptions of sequential constructions have focused on the application of sequential methods for drawing from a prior distribution. From a sequential method for drawing from the

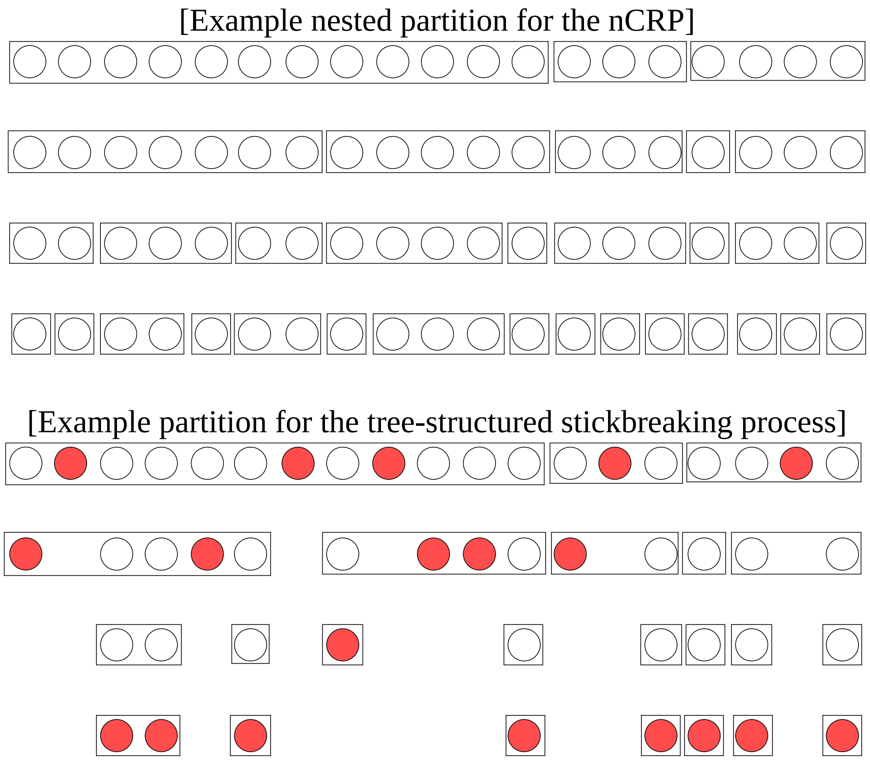

[Example nested partition for the nCRP]
[Example partition for the tree-structured stickbreaking process]

Figure 3.6: One obtains the combinatorial picture of the tree-structured stick breaking process by first drawing nested partitions representing the nCRP (top figure). Each row contains the same points, but a different partitioning of the points. The multiple levels form a hierarchy of nested partitions. To obtain the combinatorial equivalent of the tree-structured stick breaking process (bottom figure), at each node, randomly select a set of points to be colored red. Delete all points under the colored points. The assignment of colored points to boxes gives the partition defined by the tree structured stick breaking process.

prior, one may add a likelihood term and obtain a sequential construction for a proposal distribution in an MH algorithm.

We can also use sequential constructions to study the interplay between parameters for an HDP. First consider the HDP with a single group and with a $CRP(\theta, \alpha)$ fragmentation process and $CRP(\theta_0, \alpha_0)$ coagulation process.

Suppose the outgoing edges for the first i vertices have already been drawn. Let k_i be the number of black roots among the first i points, and let k_i' be the number of red roots among the first i points. The vertex $i+1$ is designated a new root with probability

$$\frac{\theta_0 + \alpha_0 k_i'}{k_i + \theta_0} \frac{\theta + \alpha k_i}{i + \theta}$$

In the case where $\theta_0 = \theta/\alpha$, i.e. the coagulation process is $CRP(\theta/\alpha, 0)$, then the incremental probability of creating a new root is $\frac{\theta+\alpha k_i}{i+\theta}\frac{\theta_0+\alpha_0 k_i'}{k_i+\theta_0}$ exactly the same as a $CRP(\theta_0, \alpha_0)$. One may similarly calculate that the probability of creating attaching vertex $i+1$ to a component matches the $CRP(\theta_0, \alpha_0)$ probabilities. Since the incremental probabilities for a fragmentation followed by a coagulation is the same as that of a coagulation followed by a fragmentation, this proves the fragmentation-coagulation duality.

When there are multiple groups for the HDP, the incremental probability that vertex $i+1$ is a root is

$$\frac{\theta_0+\alpha_0 k_i'}{k_{i\cdot}+\theta_0}\frac{\theta+\alpha k_{ig}}{n_{ig}+\theta}$$

where k_{ig} is the number of black roots among the first i vertices in group g, n_{ig} is the number of vertices among the first i vertices in group g, and $k_{i\cdot} = \sum_g k_{ig}$ is the total number of black roots among the first i vertices.

3.8 Applications to non-Partition Problems

So far, we have only described the use of combinatorial structures for tackling the problem of sampling distributions on partitions and hierarchies. However, distributions on the combinatorial structures themselves are also of interest.

Reservoir Sampling

As an example, consider the problem of drawing a random sample of size k from multiple streams of data. The solution to this problem for a single stream, called the reservoir sampling problem, was given by Vitter (1985). In modern large-scale data analysis in the map-reduce framework, one typically does not have a single stream but many streams which need to processed and combined in a distributed fashion. Previous approaches have been based on weighted version of reservoir sampling such as those given by Efraimidis and Spirakis (2006) and Kolonko and Wäsch (2006). These approaches require maintaining a weight for each element of the sample and placing the top k weights in a heap. Using the ideas we have developed thus far, we give an elegant solution that solves the problem for an unweighted random sample without replacement in a single map-reduce pass and does not require storing weights for each element. Given a random k sample from each stream in the map phase, the only additional summary statistic needed to perform the reduce phase is a count for the length of each stream. Furthermore, we show that without maintaining any additional information we can combine the output of multiple k samples to obtain a random sample of size $> k$.

We first describe the single stream solution and relate it to random permutations and our multiple stream solution. For the single stream solution, start by including the first k elements from the stream in the sample. On the i^{th} element of the stream, replace a random element of the current sample with the newly encountered element with probability $1/i$. If we cast the problem of finding the first k elements of a random permutation, then we obtain the solution given in section

3.4 which, when restricted to storing only the first k elements of the permutation, is identical the reservoir sampling solution except that the explicit permutation is maintained.

For the multiple stream case, suppose there are two streams s_1, s_2 for simplicity. Generalizing to more than two streams is trivial. Choose a base which contains a dummy element in the first position and puts all the elements of s_1 before s_2. An arborescence tree respecting that base gives a cyclic permutation of the dummy element and all the elements of s_1, s_2, and removing the dummy element gives a permutation of the elements of the two streams. We first describe how to extract the first k elements after the dummy element in a cyclic permutation from the arborescence tree representation. Recall that the edges of the arborescence tree denote the “sit to the right” process in a Chinese restaurant process. For a cyclic permutation with a dummy element, the analogous goal in the corresponding Chinese restaurant process with a single round table is to find the k persons that sit immediately to the right of the head of the table. The first person to the right of the table head is the *last* customer to sit to the right of the head. We denote the this customer as node x_{i_1} in the arborescence tree. The second to last customer that tried to sit next to the head of the table but was pushed aside by x_{i_1} we will denote as x_{i_2}. All customers corresponding to the subtree rooted at x_{i_1} will have inserted themselves between x_{i_2} and the head of the table. Thus, to find the k customers that sit to the right of the head, we look for the last customer to the sit to head and all the customers in the corresponding subtree. If that subtree is of size $\geq k$, then stop. Otherwise, find the next to last customer that tried to sit next to the head, and repeat.

For the problem of merging two streams, the goal is to find how many customers in stream s_2 inserted themselves between the head of the table and the last customer from the first stream s_1 who tried to sit next to the head. This is exactly the problem of finding the size of a subtree restricted to the subset of points in s_2 which we described in section 3.4. Thus we have the algorithm $MergeReservoirs$ which calculates the number of samples to pull from each stream to obtain a final random sample. We note that the resulting sample will typically be bigger than k because the algorithm can take all customers in the union of the two samples who sit immediately to the right of the head.

Other problems of interest

For example, consider the matching problem. Permutations encode matchings. If the matching is between sources $s_1, ..., s_n$ and targets $t_1, ..., t_n$, and one believes that the matching $s_i \to t_i$ for all i is approximately correct, then one may put a prior on permutations which strongly prefers singletons.

Another example is the ranking problem on n items $v_1, ..., v_n$. Both arborescence trees and permutations are combinatorial structures of interest since they respectively represent partial and total orderings. When each observation consists of a set of items that are presented in a particular order and the resulting ranking, then random recursive forests give one probability model that takes into account the the presentation of the items via a base.

Algorithm 4 $MergeReservoirs$

```
Require: Parameters $B_1, B_2$ containing random $k$ samples without replacement from streams
    $s_1, s_2$ and counts $n_1, n_2$ denoting the length of the streams
Ensure: $k_1 + k_2 \geq k$ and $0 \geq k_1, k_2 \leq k$ where $k_i$ is the number of elements to be taken from
    sample $B_i$
Set $k_1 = k_2 = 0$.
for $k_1 = 0 \rightarrow k$ do
    Draw $z \sim Beta - Binomial(n_2 - k_2, 1, n_1 - k_1)$
    if $z > 0$ and $k_1 + k_2 + z > k$ then
        $k_2 = min(k_2 + z, k)$
        return $k_1, k_2$
    end if
end for
return $k_1, k_2$
```

Chapter 4

Markov Chains on Graphs and Split-Merge Samplers

4.1 Introduction

Developing Markov Chain Monte Carlo (MCMC) methods is somewhat of a black art in which one draws from a rich toolbox of ideas that others have developed. The ideas presented here both add to and are inspired by ideas from this toolbox, in particular, the idea of variable augmentation.

In this chapter, the primary idea on which others are built is to augment the state space so that, instead of a single variable $X(t)$ at a time t, one has a tuple or a stopped Markov chain $X_1(t), X_2(t), ..., X_\tau(t)$. This augmented representation may be used to both make better proposals as well as reduce computational costs when one constructs the Markov chains to contain symmetry that can be exploited.

The main ideas that build off this Markov chain on Markov chain representation is a generalizaton of the notion of launch states that was presented by Jain and Neal (2004) and the use of stopping times to perform early rejection of a bad proposal. As an application, we demonstrate the use of early rejection for split-merge samplers.

In this chapter, we describe Markov Chain Monte Carlo methods in which the state space is described by a graphical model. In particular, we consider when the states themselves are Markov chains, so that the MCMC method is a Markov chain on Markov chains. We use this idea to propose modifications to the split-merge HDP sampler described in the previous chapter, and we provide experimental results for the new HDP samplers.

4.2 Markov kernels on branch processes

To tackle the problem of improving split-merge samplers, we first introduce a few ideas that are generally applicable for developing MCMC algorithms. The common thread among these ideas is that we build Markov chains $X(1), X(2), ...$ where the states of the Markov chain are stochastic processes $X(t) = (X_1(t), X_2(t), ...)$. To distinguish the two, we call $\{X(t)\}_{t\in\mathbb{N}}$ the trunk chain and any $X(t) = (X_1(t), X_2(t), ...)$ a branch process. Given a desired multivariate stationary distribution p on k variables, the goal is to construct Markov chains such that the stationary distribution on the first k variables $(X_1(t), ..., X_k(t))$ of a branch process is p. For ease of exposition, we will assume that all finite-dimensional distributions under consideration are discrete.

We use branch processes to exploit three ideas. First, each branch process is a sequential procedure that allows one to incremental build viable proposals. Second, underlying each branch process is a graphical model encoded by a DAG. When the graphical model contains symmetric structures, then one can exploit the symmetries as a way to save on computational cost as well as a means of dealing with nuisance parameters. Third, the branch processes allow the introduction of stopping times. The stopping times may also be used to save computation by performing early rejection when a partially computed proposal is sufficient for determining that the complete proposal is unlikely to be accepted. They may also be used to continue the branch process to make a second, improved proposal.

Many of these ideas have existed in the current literature in some form. For example, the split-merge sampler of Jain and Neal (2004) implicitly uses a branch process where the restricted Gibbs sweeps are a sequential method used to generate viable proposals and the launch states are nuisance parameters that exploit symmetries in the underlying graphical model so that the launch state probabilities never need to be computed. Stopping times may be used to describe Wolff's algorithm for the Ising model as well as the Delayed Rejection technique of Tierney and Mira (1999). In both cases, the branch process continues until a good proposal is reached. While we are not aware of other cases that use stopping times to perform early rejection, the idea of using multiple stages for the acceptance rule has been described in Christen and Fox (2005), Murray (2007), Liu (2008), and Dostert et al. (2006).

Markov chains on graphical models

We describe a Markov chain on branch processes by first considering a more general formulation where the state of the Markov chain at time t consists of a random distribution $G(t)$ represented by a graphical model on $S(t)$ vertices, and a tuple $\mathbf{X}(t) = (X_1(t), X_2(t), ..., X_{S(t)})$ assigning values to vertices of the graphical model. This is an instance of a random proposal distribution as described by Besag et al. (1995). For this chapter, a we will use the term graphical model to refer to a graph describing the set of conditional independence assumptions along with a set of clique potentials or conditional distributions encoding a joint distribution over the vertices of the graph.

Let f be the desired stationary distribution on k variables. Place a distribution q over graphical models such that the marginal distribution $p(x_1, ..., x_k|G) = f(x_1, ...x_k)$ of the the first k variables is equal to the desired stationary distribution for any G in the support of q. If $\{(G(t), \mathbf{X}(t))\}_t$ is an Markov chain with a unique stationary distribution given by $q(g)p(\mathbf{x}|g)$, then the chain restricted to $\{(X_1(t), ..., X_k(t))\}_t$ has the desired stationary distribution f. This is easily proved by simply marginalizing all extraneous variables.

Stopping times and MH on branch processes

To construct a MH algorithm on using a Markov chain on graphical models, consider the following manipulations on the state of the chain to generate a proposal:

1. generate a new set of augmentation variables $X_{k+1}(t), X_{k+2}(t), ...X_{S(t)}(t)|G(t), X_1(t), ..., X_k(t)$ given the first k elements and the graphical model $G(t)$,

2. draw a new graphical model $G'(t)|G(t), \mathbf{X}_{1,...,S(t)}(t)$ of size $S'(t)$ from some proposal distribution.

3. randomly permute the tuple to obtain $\pi\mathbf{X}_{1,...,S(t)}(t) = (X_{\pi(1)}(t), ..., X_{\pi_{S(t)}})$ where π is drawn from some distribution conditional on $G'(t), \mathbf{X}(t)$ and then extending the tuple by drawing $X'_{S(t)+1}, X'_{S(t)+2}, ..., X'_{S'(t)}|\pi\mathbf{X}(t), G'(t)$ if $S'(t) > S(t)$ and projecting onto the first $S'(t)$ variables otherwise.

The first two manipulations play the role of generating a new set of states which the chain can move to but do not modify any of the variables $X_1(t), X_2(t), ..., X_k(t)$ of interest, while the last manipulation uses the new set of states to proposes a change to these variables of interest.

Of special interest are MH algorithms on graphical models with a form amenable for describing sequentially sampled proposals. Fix a distribution G_0 for a discrete time stochastic process and a distribution for a stopping time with respect to the natural filtration of the process. A proposal consists of the following steps:

1. generate augmentation variables $X_{k+1}(t), X_{k+2}(t), ...X_{S(t)}|G_0, X_1(t), ..., X_k(t)$ and a stopping time $S(t)$

2. draw a random permutation on $S(t)$ elements conditional on $\mathbf{X}(t)$

3. construct a stopped chain $X'_1(t), X'_2(t), ..., X'_{S'(t)}(t)$ such that $X'_i(t) = X_{\pi(i)}(t)$ if $i \leq S(t)$ and $X'_i(t)$ is drawn from the conditional distribution $X'_i(t)|X'_1(t), X'_2(t), ..., X'_{i-1}(t), S'(t) \geq i$. randomly permute the tuple to $\pi\mathbf{X}(t) = (X_{\pi(1)}(t), ..., X_{\pi_{S(t)}})$.

In this case, the random graphical model is identified by the stopping time $S(t)$ and the augmentation variables $X_{k+1}(t), X_{k+2}(t), ..., X_{S(t)}$ are sampled simultaneously with the stopping time. Since a random distribution may be selected as the augmented variable $X_{k+1}(t)$, this formulation is not less restrictive. However, each formulation provides a different perspective since the graphs associated with each formulation are different.

Example: Metropolis-Hastings as a chain on pairs

To illustrate the idea of a Markov chain on branch processes, consider the traditional Metropolis-Hasting (MH) algorithm as a Markov chain on branch processes. In this case the stopping time is a constant $S(t) = 2$ so the stopped chain is always of the form $X_1(t), X_2(t)$. Since the graphical model is fixed, each step of the Markov chain consists of two parts. The first is a Gibbs step drawing the second element of the pair, in other words, a draw $X_2(t)|X_1(t)$. The second part is a choice to accept or reject the transposition $(X_1(t), X_2(t)) \to (X_2(t), X_1(t))$ according to the MH ratio.

Useful properties of MH algorithms on branch processes

The motivation for developing MH algorithms on branch processes are two-fold. First, the use of stopping times allow the design of chains which adjust the amount of computation spent on one proposal. The branch process may be stopped at the first instance a viable proposal has been generated or when it is clear that no future proposals are likely to be accepted. Second, the algorithms can eliminate certain computations by exploiting symmetries in the graphical models and in the encodings of a distribution as potentials on the graph. This is due to the same potentials appearing in both the numerator and denominator of the MH ratio. The simplest example of this is for a MH algorithm with a symmetric proposal distribution. While one must be able to sample from the symmetric proposal, the proposal probability never needs be to computed.

We first give existing applications of stopping times as well as applying them to the HDP sampler. We then describe the role of symmetry in saving computation.

Example: Stopping times and delayed rejection

Delayed rejection (Tierney and Mira (1999)) is a MCMC technique that has a simple formulation in terms of stopping times. Given a $X_1(t)$ from the desired stationary distribution, draw a sequence of proposals $X_2(t), X_3(t), ...$. Let $S(t)$ be a stopping time with stopping probabilities that are recursively defined as follows. $p(S(t) = k|X_1(t) = x_1, ..., X_k(t) = x_k, S(t) \geq k) = \min\left\{1, \frac{p(X_1(t)=x_1,...,X_k(t)=x_k,S(t)\geq k)}{p(X_1(t)=x_k,...,X_k(t)=x_1,S(t)\geq k)}\right\}$. In other words, $S(t)$ is chosen such that if the branch chain stops at $k+1$, then the MH ratio for the move $(X_1(t), ..., X_{S(t)}(t), S(t)) \to (X_{S(t)}(t), ..., X_1(t), S(t))$ will be accepted with probability 1. To ensure that the sequence of proposals stop, $S(t)$ may be capped at some maximum time K though the acceptance probability may not be 1 if $S(t) = K$. We note that the permutation associated with this proposal is the one which reverses the tuple $(X_1(t), ..., X_{S(t)}(t))$.

Example: Stopping times and the Wolff sampler

The Wolff sampler for the Ising model is another example of a sampler employing a stopping time. The Ising model is a binary Markov random field with probability given by

$$P(\vec{x}) \propto \exp\left(\sum_{i,j} K_{ij} 1(x_i = x_j)\right).$$

The Wolff sampler proceeds as follows:

1. Choose a starting cluster C consisting of a single node at random.

2. At each step, choose an unchecked node x_i in C, marked it checked, and visit all the unvisited links adjacent to x_i. For each link $x_i \leftrightarrow x_j$, activate the bond with probability $1 - \exp(-2K_{ij})$ and add x_j to C.

3. Repeat until there are no unchecked nodes in C, and flip all the spins in C.

Recast in terms of stopping times, the Wolff sampler builds a branch process on spin-states where the stopping time is the time the MH ratio hits 1.

Early rejection

The previous examples give cases where the branch chain stops when a good enough proposal is reached. This prevents computation from being wasted when rejecting prematurely. For the HDP sampler, we use stopping times to reject when the proposal is unlikely to be accepted but before the entire proposal is drawn. This skips computations that are likely to be thrown away. For split-merge samplers these computations are costly since each split or merge proposal has computation time proportional to the size of the merged block. For well-separated clusters in a mixture model, the sampler already has information that a merge is highly unlikely to be accepted even before calculating the merge-to-split proposal probability.

Two-stage acceptance probabilities

Suppose we have a Metropolis-Hastings algorithm where the MH ratio for a move from $x \to x'$ can be decomposed into a product $R(x \to x') = \gamma(x, x')\psi(x, x')$, and ψ is expensive to calculate.

We can construct a chain which breaks down the acceptance rule into two stages, the first is based only on γ, and the second requires the expensive computation of ψ but may be skipped if the move is rejected in the first stage. This idea of using multiple stages for the acceptance rule has been described in Christen and Fox (2005), Murray (2007), Liu (2008), and Dostert et al. (2006).

Consider the chain that accepts $x \to x'$ and $x' \to x$ with the respective two-stage probabilities

$$\begin{aligned} \alpha(x \to x') &= \min\left\{\gamma(x, x'), 1\right\} \min\left\{\psi(x, x'), 1\right\} \\ \alpha(x' \to x) &= \min\left\{\gamma(x, x')^{-1}, 1\right\} \min\left\{\psi(x, x')^{-1}, 1\right\}. \end{aligned}$$

One can easily verify this chain satisfies detailed balance and has the same invariant distribution as the original chain using the MH algorithm.

The acceptance probability of this two-stage rule is always less than or equal to the usual MH acceptance probability. To obtain, a better acceptance rate, note that

$$\begin{aligned} \gamma(x, x')\psi(x, x') &= \min\{\gamma(x, x'), 1\}(\max\{\gamma(x, x'), 1\}\psi(x, x')) \\ &= \tilde{\gamma}(x, x')\tilde{\psi}(x, x') \end{aligned}$$

where $\tilde{\gamma}(x, x') = \min\{\gamma(x, x'), 1\}$. In other words, if the first stage is accepted with probability 1, the excess in the first stage can always added to the second stage to boost the acceptance probability. This leads us to choose the following as the first stage acceptance probability of a split-to-merge move:

$$\gamma(\mathcal{B}_{split}, \mathcal{B}_{merge}) = \min\left\{\frac{f(\mathcal{X}(B_1 \cup B_2))}{f(\mathcal{X}(B_1))f(\mathcal{X}(B_2))}, 1\right\}. \tag{4.1}$$

Since $f(\mathcal{X}(B_1)), f(\mathcal{X}(B_2))$ are likely to already have been computed, rejecting a merge can take a single likelihood calculation. For a merge-to-split move, the first state acceptance probability is always 1 since $\gamma(\mathcal{B}_{split}, \mathcal{B}_{merge})^{-1} \geq 1$.

Early stopping and a random two-stage acceptance rule

For merge-to-split moves, the final split state is unknown before calculating the proposal probabilities, so the two-stage acceptance rule for split-to-merge moves does not apply. Instead we propose a rule to reject once there is enough information to determine a split is unlikely. We do this by extending the two-stage acceptance rule and using a random decomposition of the MH ratio based on a stopping time.

Suppose the proposal $X_1(t) \to X_1'(t)$ is drawn via a sequential procedure and let $X_2(t), ..., X_k(t)$ denote the intermediate states of the procedure. For notational convenience, we will drop the index t in the rest of the discussion.

At any time s, the MH ratio may be decomposed into a product $R(X_1 \to X_1') = \gamma_s(X_1, X_2, ..., X_s)\psi_s(X_1, ..., X_k, X_1')$, so that $\gamma_s > 0$ depends only on events up to time s and $\psi_s > 0$ can depend on any event. This decomposition gives a two-stage acceptance rule for any fixed choice s.

Rather than a fixed time, consider the case where the decomposition depends on a random stopping time S with respect to the natural filtration of $X_1, ..., X_k$. This can be beneficial since often one does not know how many steps are required before an informed decision to reject can be made, and the number of steps may depend on the random choices taken by the sequential procedure. It is easy to prove that this modification preserves detailed balance.

Theorem 11. *Given the assumptions in the previous paragraphs, the Markov chain which accepts the proposal $X_1 \to X_1'$ according on the two-stage acceptance probabilities*

$$\begin{aligned}\alpha_S(X_1 \to X_1') &= \min\{\gamma_S(X_1, X_2, ..., X_S), 1\} \min\{\psi_S(X_1, ..., X_k, X_1'), 1\}\\ \alpha_S(X_1' \to X_1) &= \min\left\{\gamma_S(X_1, X_2, ..., X_S)^{-1}, 1\right\} \min\left\{\psi_S(X_1, ..., X_k, X_1')^{-1}, 1\right\}\end{aligned}$$

gives a Markov chain with the same invariant distribution as a chain generated using the MH algorithm.

Proof. One simply verifies the detailed balance condition on the augmented chain (X, X', S).

$$\begin{aligned}\alpha_s(X_1 \to X_1')p(X_1, X_1', s) &= \alpha_s(X_1 \to X_1')p(X_1, X_1')p(S = s|X_1, X_1')\\ &= \alpha_s(X_1' \to X_1)p(X_1', X_1)p(S = s|X_1', X_1)\\ &= \alpha_s(X_1' \to X_1)p(X_1', X_1, t)\end{aligned}$$

□

where the second step follows from the non-random two-stage acceptance rule and the assumption that the proposals in both directions share the same sequential procedure.

We note that the decomposition is asymmetrical since γ_S does not depend on X_1'. This is necessary since X_1' is unknown during the sequential procedure for proposing a new state. If one wishes to perform some form of early stopping on the reverse proposal $X_1' \to X_1$, then the stopping times for both proposal directions must share a common filtration to obtain computational benefits.

Application to the HDP split-merge sampler

For split-merge samplers, we use the usual two-stage acceptance rule for split-to-merge moves and combine it with the two-stage early stopping rule for merge-to-split moves. We illustrate the application of the two-stage early stopping rule using our split-merge HDP sampler. Modifying the method for other split-merge samplers is straightforward.

Consider a split proposal $Y_m \to Y_s$ for the HDP split-merge sampler, and denote the two blocks in the split by B_0, B_1. Given a base, the proposal draws a branch chain $Z_k', Y_m, Z_1, Z_2, ..., Z_{k-1}, Z_k = Y_s, Z_1', Z_2', ..., Z_{k-1}'$ where k is the size of the merged block, Z_i denotes the subforest of Y_s restricted to the points in the base up to the i^{th} point of the merged block and Z_i' denotes the same but for Y_m instead of Y_s. Note that the Z_i, Z_i' are deterministic functions of Y_s, Y_m respectively. Thus, the natural filtration of $Y_m, (Z_2, Z_2'), (Z_3, Z_3'), ..., (Z_{k-1}, Z_{k-1}'), Y_s$ is also the natural filtration of just the sequential splitting procedure which draws $Y_m, Z_2, Z_3, ..., Z_{k-1}, Y_s$. The corresponding merge proposal has branch chain $Y_s, Z_2, ..., Z_{k-1}, Y_m, Z_2, Z_3, ..., Z_{k-1}$.

All the split-merge samplers with sequential procedures have a MH ratio of the form

$$\begin{aligned} R(Y_m \to Y_s) = \prod_{i=1}^{k} \Bigg(& \frac{f(x_i|x_1, ..., x_{i-1}, Z_{i-1})^{1(x_i \in B_0)} f(x_i|x_1, ..., x_{i-1}, Z_{i-1})^{1(x_i \in B_1)}}{f(x_i|x_1, ..., x_{i-1}, Z_{i-1}')} \times \\ & \times \frac{q_m(Z_{i-1}' \to Z_i')}{q_s(Z_{i-1} \to Z_i)} \frac{p(Z_i|Z_{i-1})}{p(Z_i'|Z_{i-1}')} \Bigg) \end{aligned}$$

where q_m, q_s are the sequential proposal probabilities for the MCMC sampler, and p gives the corresponding sequential allocation probabilities for the prior. Taking the product from 1 to a fixed time s rather than k, one obtains a term $\gamma_t(Y_m, Z_1, ..., Z_s)$ that does not depend on any of the terms occurring after Z_s. This gives a decomposition suitable for early rejection. We choose the stopping time S to be the hitting time $S = \min\{s : \gamma_t(x, Z_s) < c\}$ for some constant c. Other stopping times may be sensible, but we choose this one for simplicity.

For a split-to-merge proposal, we may simply use the likelihood term

$$\prod_{i=1}^{k} \frac{f(x_i|x_1, ..., x_{i-1}, Z_{i-1}')}{f(x_i|x_1, ..., x_{i-1}, B_0)^{1(x_i \in B_0)} f(x_i|x_1, ..., x_{i-1}, B_1)^{1(x_i \in B_1)}}$$

to construct the first stage acceptance probability.

Symmetries in graphical models

Another advantage to describing MH algorithms in terms of branch chains and graphical models is that it lends itself to identifying symmetries that may be exploited for computational benefits. Let $G(t)$ denote a graph with associated potentials $\{\psi_C^{(t)}\}_{C \subset V(G(t))}\}$ and the tuple $\mathbf{X}(t) = (X_1(t), ..., X_{S(t)}(t))$ be a tuple generated from this graphical model. Let $\pi(t)$ be the proposed permutation. For simplicity, assume that the proposed graphical model $G'(t)$ is the same as $G(t)$. Given a potential ψ_C where $C \subset V(G(t))$, define the action of $\pi(t)$ on the potential by $\pi(t)\psi_C = \psi_{\pi(t)^{-1}C}$.

The distribution of the permuted variables may be expressed as either a permutation of the tuple or of the vertices of the graph.

$$p(\pi(t)\mathbf{X}(t)|G(t)) = \prod_C \psi_C(\pi(t)\mathbf{X}) = \prod_C (\pi(t)\psi_C)(\mathbf{X}).$$

Thus, the set of potentials such that $\psi_C = \pi(t)\psi_C$, in other words the potentials invariant under $\pi(t)$, will appear in both the numerator and denominator of the MH ratio and cancel each other out.

Launch States and Reversible Jump MCMC

We now demonstrate one useful application of the feature of not needing to calculate the full proposal probability. Jain and Neal (2004) introduced the notion of a launch state in the context of samplers for Dirichlet Process Mixture Models with conjugate priors. Rather than restricting moves of a chain to local moves to states with similar posterior probability, launch states offer a means for a chain to propose jumping to a distant local mode of the posterior. We give a modest, but powerful, generalization the idea of launch states presented in Jain and Neal (2007) that makes it generally applicable to MCMC methods but particularly applicable for reversible jump MCMC methods.

Consider a Markov chain on 4-tuples $(X_1(t), X_\ell(t), Y_1(t), Y_\ell(t))$ which has stationary distribution $p(x)q_\ell(x \to x_\ell)q(x_\ell \to y)q_\ell(y \to y_\ell)$. $X_\ell(t), Y_\ell(t)$ are designated launch states.

Step $t + 1$ of the chain regenerates the variables except $X_1(t)$ and proposes the move $(X_1(t), X_\ell(t), Y_1(t), Y_\ell(t)) \to (Y_1(t), Y_\ell(t), X_1(t), X_\ell(t))$. In other words, the chain proposes a cyclic permutation of the variables of offset 2. The MH ratio for such a proposal is

$$\frac{p(Y_1(t))q(Y_\ell(t)) \to X_1(t)}{p(X_1(t))q(X_\ell(t) \to Y_1(t))}. \tag{4.2}$$

The launch proposal probabilities conveniently cancel out. This is illustrated by the following diagrams that describe the proposed move as a cyclic permutation of offset 2.

The MH ratio is the ratio of the two joint probabilities described by the diagrams. The links $X_1(t) \to X_\ell(t), Y_1(t) \to Y_\ell(t)$ appear in both diagrams and if the conditional probabilities describing the edge potentials for those links are the same, then the conditional probabilities associated with the links cancel in the MH ratio.

$$\begin{tikzcd}
X(t) \arrow[r] & X_\ell(t) \arrow[d] & X(t) \arrow[r] & X_\ell(t) \\
Y_\ell(t) & Y(t) \arrow[l] & Y_\ell(t) \arrow[u] & Y(t) \arrow[l]
\end{tikzcd}$$

Dirichlet Process example and a general recipe for sampling with launch states

We describe the DP mixture model samplers of Jain and Neal (2007) and Jain and Neal (2004) which introduce the idea of launch states and show how to obtain a more general recipe for sampling with launch states without using restricted Gibbs sweeps.

In both of Jain and Neal's papers, the proposal distribution consists of 1) sampling a transposition uniformly at random to determine what blocks to split or merge and 2) generating a new proposal state via a launch state and a final Gibbs sweep. Since step 1 is a uniform draw and does not depend on the current state of the Markov chain, it is inconsequential in computing the MH ratio, but for completeness in describing the samplers, we give the deterministic rule for choosing whether to split or merge given a transposition. If the elements of the transposition are both in the same block B_{merge}, the sampler proposes to split block B_{merge} into randomly generated blocks B_1 and B_2 via a series of Gibbs sweeps. Otherwise, the elements of the transposition are in separate blocks which we denote as B_1 and B_2, and the sampler proposes to merge B_1 and B_2.

For step 2, we may apply equation 4.2 to calculate the MH ratio. However, we give a slightly more general description that clearly suggests methods which do not rely on restricted Gibbs sweeps. To simplify the exposition, we will assume that the chain proposes a split from a merged state. We may write the two branch chains corresponding to the split-to-merge and merge-to-split proposals as the following graphical model:

$$\begin{aligned}
Y_{sm} &= X^{split} \to X_\ell^{split} \to Z_1^{sm} \to \cdots \to Z_{k_{sm}}^{sm} \to X^{merge} \to X_\ell^{merge} \\
Y_{ms} &= X^{merge} \to X_\ell^{merge} \to Z_1^{ms} \to \cdots \to Z_{k_{ms}}^{ms} \to X^{split} \to X_\ell^{split}
\end{aligned}$$

where the Z's denote the intermediate states of the final Gibbs sweeps and the k_{sm}, k_{ms} are deterministic functions of the blocks being modified. The MH ratio for the proposed move is

$$R = \frac{p(Y_{ms})}{p(Y_{sm})} = \frac{p(X^{merge})q(X_\ell^{merge} \to Z_1^{ms} \to \cdots \to Z_{k_{ms}}^{ms} \to X^{split})}{p(X^{split})q(X_\ell^{split} \to Z_1^{sm} \to \cdots \to Z_{k_{sm}}^{sm} \to X^{merge})}. \tag{4.3}$$

In the restricted Gibbs sweep case, there is a unique sequence of intermediate states that takes $X_\ell^{split} \to X^{merge}$. Thus, we have $q(X_\ell^{split} \to X^{merge}) = q(X_\ell^{split} \to Z_1^{sm} \to \cdots \to Z_{k_{sm}}^{sm} \to X^{merge})$. The same property holds for the merge launch state to split state probabilities, and one recovers the usual launch state MH ratio equation 4.2. This uniqueness condition is desirable since it does not introduce variables that may be Rao-Blackwellized away, but it is not necessary. This leads to the following general MCMC launch state procedure:

Algorithm 5 $LaunchStateMCMC$

```
Require: Current state $X_1(t)$, black box launch state generator $launch(x)$, calculable transition probabilities $\{q_i\}_i$ for a (not necessarily homogeneous) Markov process, calculable stopping time probabilities for the Markov process.
  Draw $Z_1(t) = launch(X_1(t))$
  Draw a realization $Z_1(t), Z_2(t), ..., Z_K(t) = Y_1(t)$ of the stopped process defined by $\{q_i\}_i$
  Draw $Z'_1(t) = launch(Y_1(t))$
  Draw a realization $Z'_1(t), Z'_2(t), ..., Z'_{K'}(t) = Y_1(t)$ of the stopped process defined by $\{q_i\}_i$
  Accept $X_1(t) \to Y_1(t)$ with probability
      $\frac{p(Y_1(t))}{p(X_1(t))} \frac{q(K'|Z'_1(t), ..., Z'_{K'}(t))}{q(K|Z_1(t), ..., Z_K(t))} \frac{\prod_{i=2} K' q_i(Z'_{i-1}(t) \to Z'_i(t))}{\prod_{i=2} K q_i(Z_{i-1}(t) \to Z_i(t))}.$
```

Reversible jump MCMC

Reversible jump MCMC (RJ-MCMC) methods are a natural area in which the idea of launch states may be applied. In RJ-MCMC methods, such as samplers for non-conjugate Dirichlet process mixture models (DPMMs), must deal with the issue of proposing new parameters or deleting parameters. Two main challenges exist for this. One is to handle the change in dimension of the parameters and maintain the detailed balance condition when the parameters are continuous. The other is to propose a good set of parameters.

To handle the change of number parameters, RJ-MCMC methods typically rely on augmenting the states with fewer parameters with additional random components to make the dimensions match. For example, consider the case where a model has either 1 or 2 parameters. The 1-parameter state has the form (θ_1, U) where U is a random component to make the dimensions of the Markov chain's state space consistent. However, the joint distribution (θ_1, U) of the parameters for the 1-parameter model is different from the joint distribution (θ'_1, θ'_2) for the 2-parameter model. The seminal paper of Green (1995) on RJ-MCMC handles this by only making proposals $(\theta_1, U) \to g(\theta_1, U) = (\theta'_1, \theta'_2)$ where g is an invertible, differentiable function. This is essentially a change of variables, and correspondingly, the MH ratio involves the Jacobian of g. The MH ratio for a move $(\theta_1, U) \to (\theta'_1, \theta'_2)$ is

$$\frac{p_2 f_2(\theta'_1, \theta'_2)}{p_1 f_1(\theta_1) p(U|\theta_1)} \frac{q_{21}}{q_{12}} \left| \frac{\partial(\theta'_1, \theta'_2)}{\partial(\theta_1, U)} \right|$$

where p_i is the prior probability of being in model i, q_{ij} is the probability of proposing to move to model j given the current state is in model i, and f_i is the posterior probability of the parameters given they are for model i.

Rather than padding models with fewer parameters with a few random components, we can also concatenate the parameters into a longer vector. For the 1- and 2-parameter models, this means that the underlying graphical mode is on the 3-tuple $(\theta_1, \theta'_1, \theta'_2)$. This is the approach taken in Carlin and Chib (1995) and later related to Green's approach by Besag (1997), Dellaportas et al.

(2002) and Godsill (2001). This eschews the problem of having both models share the same 2-dimensional measurable space. Each model has an independent space. There is no Jacobian to deal with, and no restriction that the mapping g be diffeomorphic since it is always the identity.

Our approach of using stopped branch chains bears similarities to both approaches. Like Green's approach, the variables share a common measurable space. Like Carlin and Chib's approach, the parameters of interest at any step are projections of a tuple, so there is no complication of having to specifying a diffeomorphic change of variables

The use of launch states provides a simple, flexible means to produce good proposals. The tradeoff is that, if one could collapse the sampler and remove the launch state, the collapsed sampler potentially mixes better than the uncollapsed sampler. However, launch states are most appropriate for cases where the probability of obtaining a launch state cannot even be calculated much less integrated out, and improved proposals can potentially improve the sampler more than augmentation hurts.

4.3 Comparison of HDP mixture model samplers

We evaluate the performance of the samplers on two examples. We will refer to the HDP Gibbs sampler based on the forest representation as simply the Forest Gibbs Sampler. Each sweep consists of a forward and backward pass. For the split-merge (SM) sampler, we interleave 1 Gibbs sweeps using the forest representation for every 15 split-merge proposals. We consider both the split-merge sampler with early rejection and without early rejection. We did not consider early stopping in this experiment since it is unlikely to help for such a small dataset. Each split-merge proposal is regarded as a sweep. The Gibbs sampler in the CRF representation is referred to as the CRF Gibbs Sampler. We calculate the effective sample size using the code package in R (Plummer et al. (2006)).

Beta-Bernoulli Example

The first is the synthetic Beta-Bernoulli example used in Jain and Neal (2004) and Dahl (2003). The synthetic datasets are small with each consisting of 100 points on 5 clusters. The dimension of the data is varied from 6 to 18 dimensions as well as the concentration parameter in the $CRP(\theta)$ process at each level of the hierarchy. We use the same parameter θ on both levels. We refer the reader to Jain and Neal (2004) for details about the exact parameters that generate the clusters. To adapt the dataset for the HDP setting, we assign a point in cluster i to group i with probability 0.6. Otherwise one of the 4 other groups are chosen at random. Each sampler is initialized to a partition which contains a single cluster. Each is then run for 200,000 to 300,000 sweeps of which the first 100,000 are treated as burnin. One sample was taken per sweep. For each dimension we generated 10 datasets and ran all the samplers on each of the datasets. Due the the different characterstics of each dataset, the effective sample sizes for the different datasets can greatly vary . Due to this, we compare the ratio of each sampler's effective sample size per likelihood evaluation to the Gibbs sampler in the CRF representation. For each sampler, this gives the number of effective samples for the same computational cost as drawing an effective sample size of 1 using the CRF Gibbs sampler.

Discussion of Beta-Bernoulli results

Tables 4.1,4.2,4.3, and 4.4 summarize the results of the experiment. The results suggest there are two regimes in which the samplers exhibit dramatically different performance characteristics. In the first regime, the cluster sizes are not well-defined and the chain transitions between partitions with differing numbers of clusters. In this case, the forest-based Gibbs sampler and the split-merge samplers perform dramatically better than the CRF Gibbs sampler. In the second regime, the cluster sizes are well-defined and virtually all split-merge steps are rejected. We note that the mixing properties of the Markov chain are arguably less interesting when the clusters are well defined since the chain is performing Bayesian averaging over parameters that change very little. In this regime, the CRF Gibbs sampler achieves better effective sample sizes per computational unit than our proposed samplers when it appeared to mix. However, the CRF Gibbs sampler was unreliable and often got stuck in a local mode as illustrated in figure 4.6. The degradation of performance of the forest-based Gibbs sampler was similar to the expected worst case performance derived in section 3.6 where the forest-based Gibbs sampler is roughly three times worse than the CRF Gibbs sampler. for a CRP. Figure 4.1 and subsequent figures show these two regimes.

With regards to mixing, in all cases, the split-merge samplers appeared to reach a good mode faster than the forest-based Gibbs sampler and CRF Gibbs sampler. In most cases, the forest-based Gibbs sampler reached a mode faster than the CRF Gibbs sampler. Since these experiments were all performed with all points initialized to a single block, the CRF Gibbs sampler often became stuck at smaller number of clusters than the rest of the samplers. As expected, the split-merge samplers were the best at escaping from local modes.

The experimental results also showed that the introduction of an early rejection step to the split-merge sampler substantially improves the plain split-merge sampler in most cases. Since the implementation of the split-merge sampler interleaves split-merge moves with the forest-based Gibbs sampler, the early rejection step essentially made split-merge moves nearly free when they are not beneficial.

Ramachandran Density Estimation

We also consider a dataset for estimating the Ramachandran probability distributions given a residue and its neighbors used in Ting et al. (2010). For these experiments only the central residue Arginine and its right neighboring residue were used. There are 20 possible neighboring residues yielding 20 groups in the HDP. There were a total of 2155 measurements of ϕ, ψ angles. We ran each sampler for 30,000 iterations and treated the first 15,000 as burnin. For these chains, the assignment ot points to an initial cluster was made by fitting a finite Gaussian mixture model. In this experiment, the forest-based Gibbs sampler and split-merge samplers substantially outperform the CRF Gibbs sampler. The split-merge sampler with early rejection but no early stopping performed the best out of the algorithms tried. The results are summarized in Table 4.5 and figures 4.7, 4.8,

kernel	theta	dim	ratio	sd	25 pct	50 pct	75 pct
Forest	1	6	5.53	1.13	3.15	5.41	7.63
SM(3,1) w/ rej	1	6	6.84	1.18	4.01	6.55	10.26
SM(5,1) w/ rej	1	6	12.92	1.15	8.86	11.99	22.29
SM(3,1)	1	6	5.24	1.17	3.49	5.58	6.78
SM(5,1)	1	6	5.94	1.15	3.73	5.88	11.36
Forest	1	9	13.30	1.28	4.19	11.05	40.49
SM(3,1) w/ rej	1	9	32.77	1.41	13.72	44.51	69.34
SM(5,1) w/ rej	1	9	51.60	1.30	13.57	86.49	245.84
SM(3,1)	1	9	26.17	1.42	10.64	39.64	56.57
SM(5,1)	1	9	23.09	1.31	5.63	40.52	107.74
Forest	1	12	3.52	1.45	0.35	6.03	27.08
SM(3,1) w/ rej	1	12	12.44	1.97	0.60	30.00	142.90
SM(5,1) w/ rej	1	12	22.48	1.61	0.32	62.47	404.64
SM(3,1)	1	12	9.58	1.98	0.53	16.37	112.58
SM(5,1)	1	12	9.47	1.62	0.15	26.94	189.60
Forest	1	15	0.89	1.65	0.24	0.33	4.58
SM(3,1) w/ rej	1	15	2.38	2.12	0.27	0.36	79.71
SM(5,1) w/ rej	1	15	2.61	1.68	0.21	0.33	15.78
SM(3,1)	1	15	1.60	2.12	0.17	0.27	56.32
SM(5,1)	1	15	1.01	1.68	0.08	0.13	5.89
Forest	1	18	1.39	2.19	0.18	0.30	426.29
SM(3,1) w/ rej	1	18	4.71	2.58	0.22	0.37	344.14
SM(5,1) w/ rej	1	18	5.81	1.90	0.16	0.29	1117.01
SM(3,1)	1	18	3.12	2.60	0.16	0.25	251.38
SM(5,1)	1	18	2.37	1.90	0.07	0.13	412.06

Table 4.1: Ratio of effective size of sampler per 1M likelihood evaluations vs. CRF Gibbs sampler. $\theta = 1$

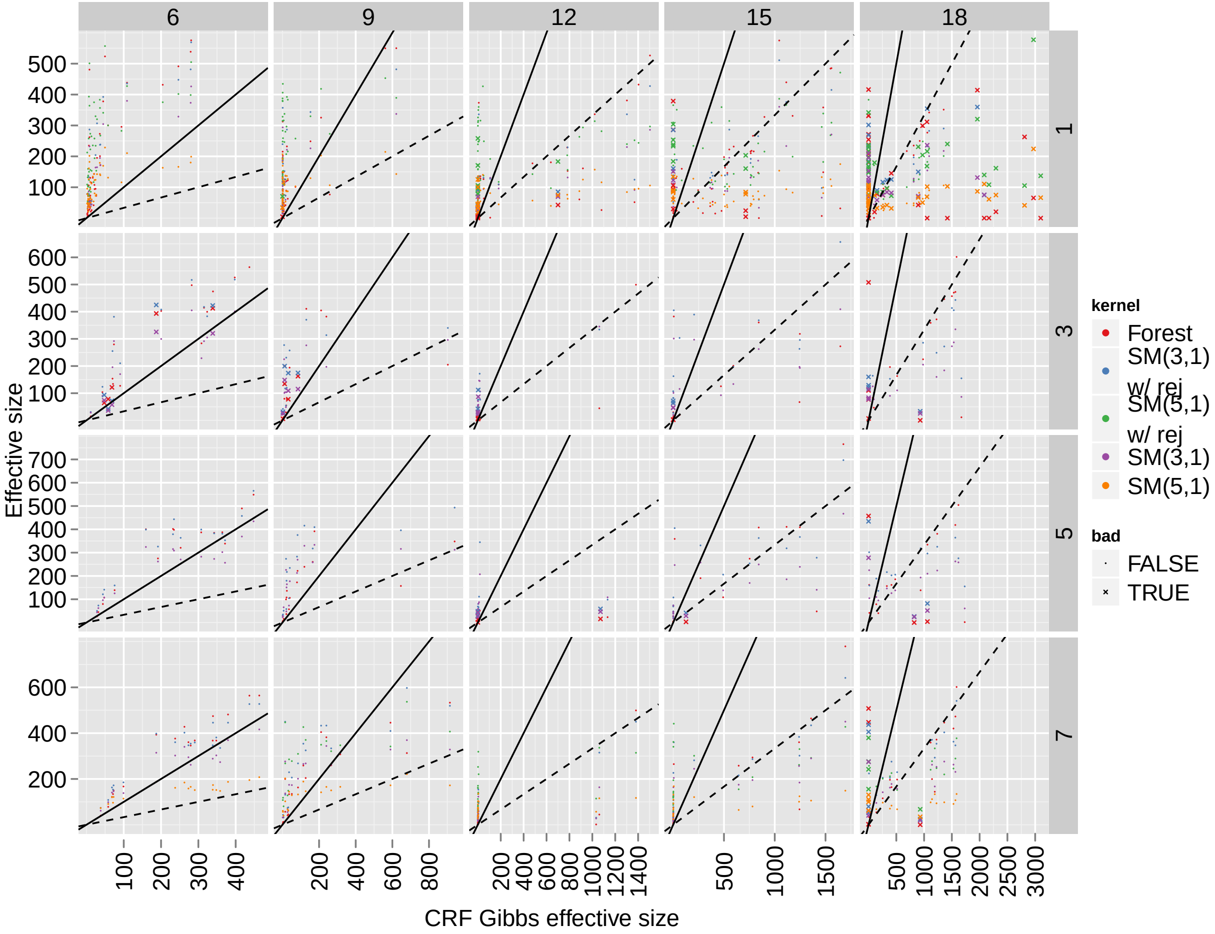


Figure 4.1: These figures show the effective sample size per 1 million likelihood evaluations for each of the proposed samplers in relation to the CRF Gibbs sampler. The x shaped points represent runs in which one of the chains clearly failed to mix since the different samplers did not agree on the average number of clusters. The solid line has slope 1 and denotes the region where a sampler exhibited equal performance compared to the CRF Gibbs sampler. The dotted line represents the expected worst case performance of a Forest Gibbs sampler compared to the CRF Gibbs sampler. The heuristic derivation is in section 3.6. In general, the points fall into three regimes: one regime in which the newly proposed samplers substantially outperform the CRF Gibbs sampler, and a second regime in which the number of clusters is basically fixed and all split-merge moves are rejected. In this regime, the performance of the forest-based Gibbs sampler is often close to the expected worst case performance when compared to a CRF Gibbs sampler. For cases where the calculated effective size was worse than the expected worst case, many are clear cases where the CRF Gibbs sampler failed to mix.

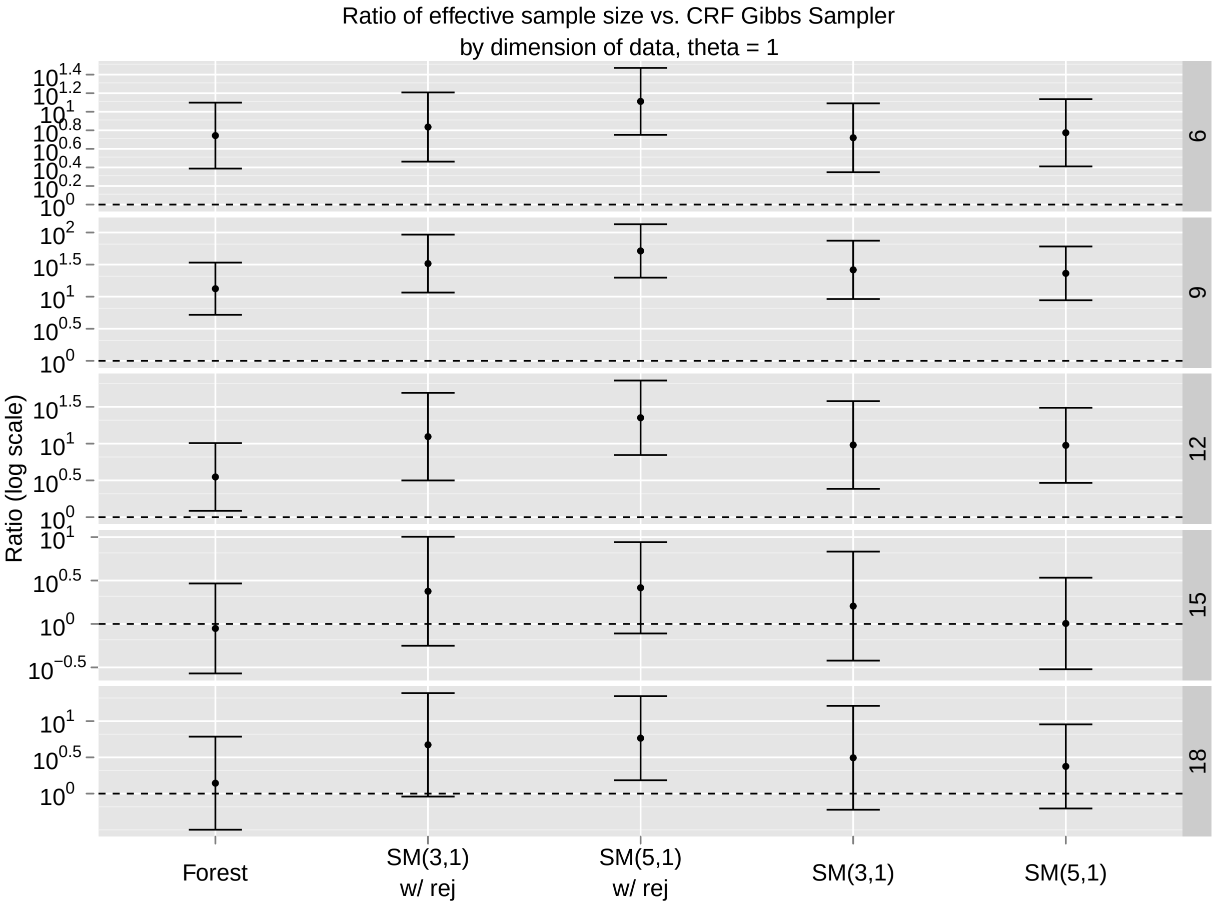


Figure 4.2: These figures show the harmonic mean of the ratio of effective size of each sampler relative to the CRF Gibbs sampler. Ninety-five percent confidence intervals are included as well. In the cases where the posterior has significant mass on both clusterings with four and with five clusters, the newly proposed samplers significantly outperform the CRF Gibbs sampler. However, in higher dimensions where the posterior tends to concentrate on only clusterings with 4 clusters, the new samplers do no better than the CRF Gibbs sampler on average. The early rejection step for the split-merge sampler typically improves the performance.

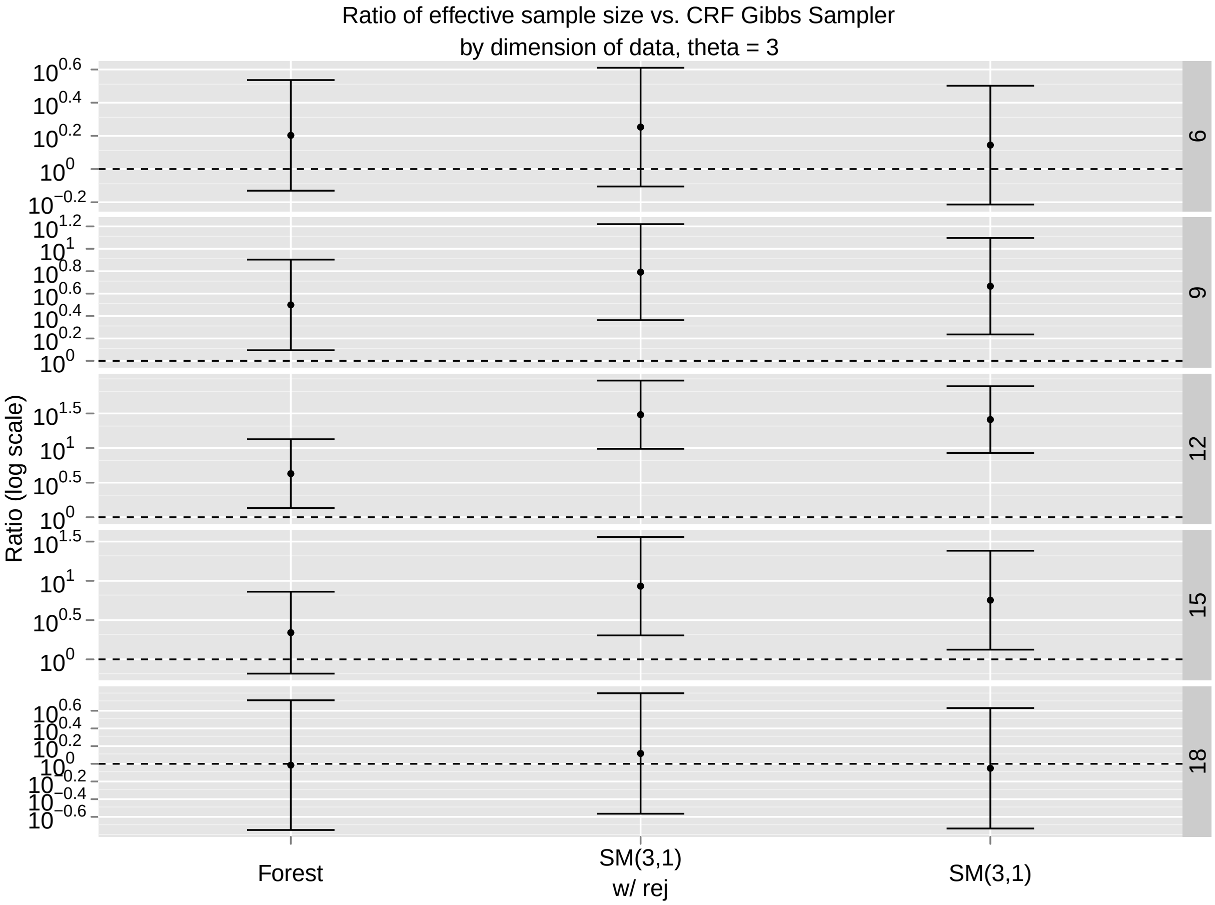


Figure 4.3: These figures show the harmonic mean of the ratio of effective size of each sampler relative to the CRF Gibbs sampler when the CRP parameter at both levels of the HDP hierarchy is $\theta = 3$.

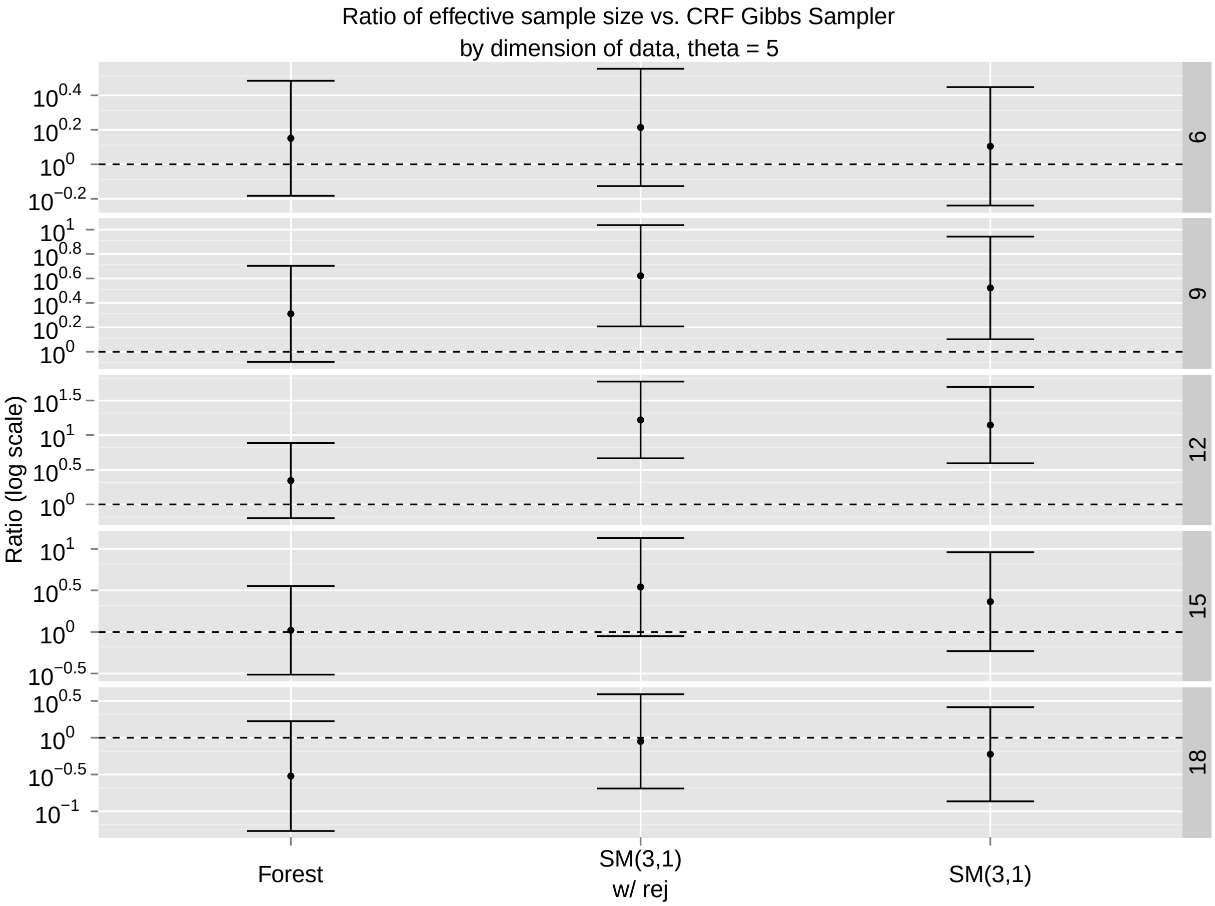


Figure 4.4: These figures show the harmonic mean of the ratio of effective size of each sampler relative to the CRF Gibbs sampler when the CRP parameter at both levels of the HDP hierarchy is $\theta = 3$.

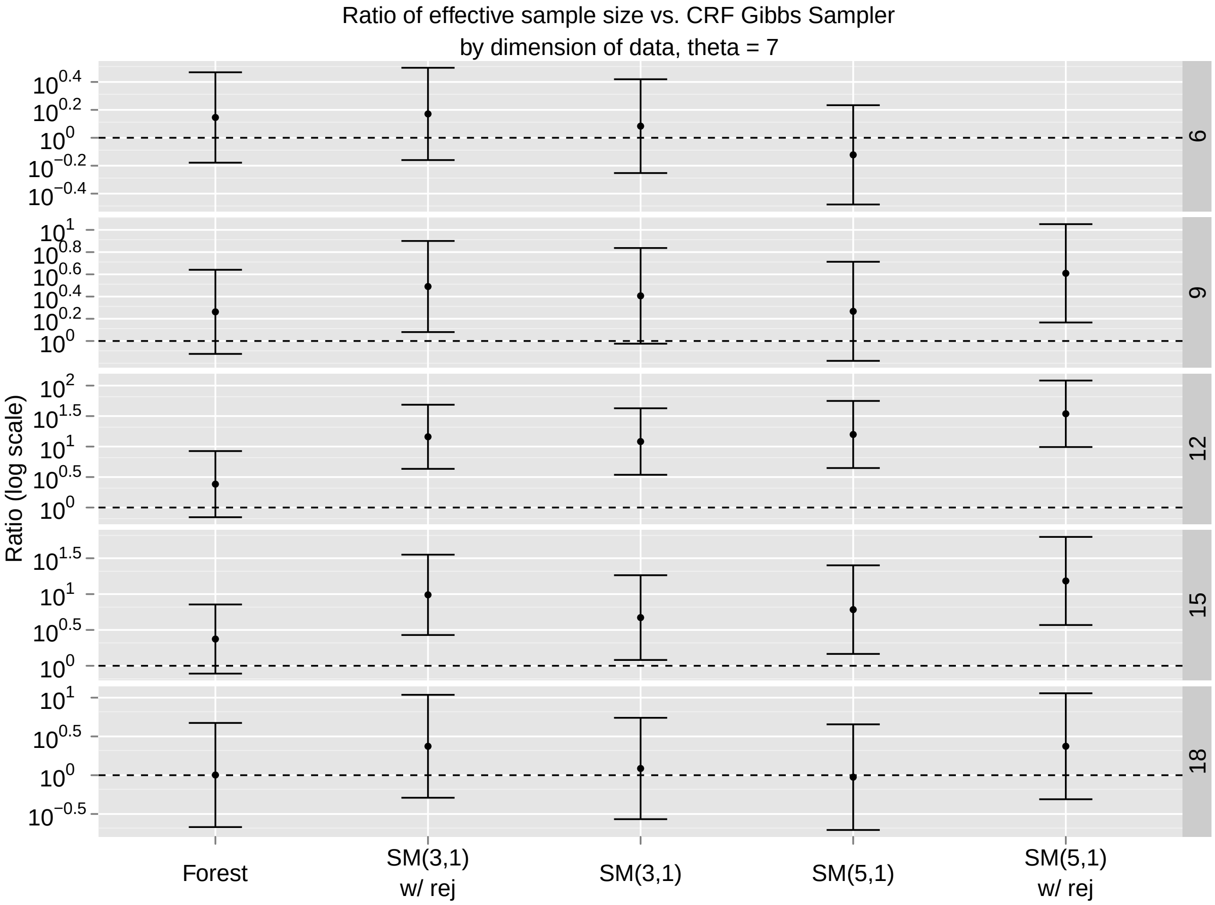


Figure 4.5: These figures show the harmonic mean of the ratio of effective size of each sampler relative to the CRF Gibbs sampler when the CRP parameter at both levels of the HDP hierarchy is $\theta = 3$.

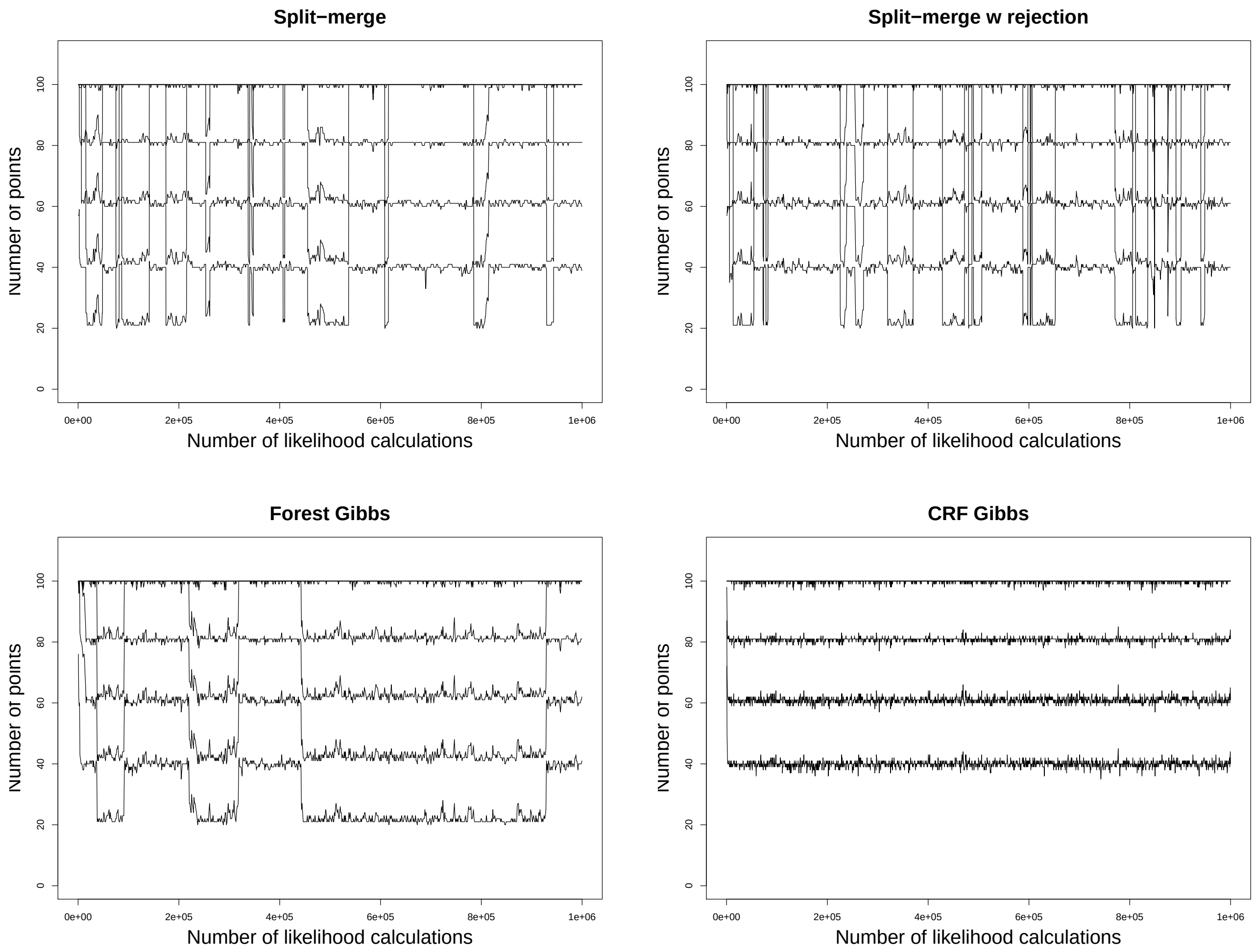


Figure 4.6: These figures show traces for the cluster sizes over time for a chain where the CRF Gibbs sampler becomes stuck and fails to switch between different cluster sizes. The n^{th} line from the bottom represents the total size of of the n largest blocks.

kernel	theta	dim	ratio	sd	25 pct	50 pct	75 pct
Forest	3	6	1.60	1.08	1.30	1.40	1.95
SM(3,1) w/ rej	3	6	1.79	1.14	1.23	1.90	2.41
SM(3,1)	3	6	1.39	1.14	0.94	1.46	1.94
Forest	3	9	3.16	1.27	1.96	2.80	4.67
SM(3,1) w/ rej	3	9	6.19	1.34	2.87	7.30	13.78
SM(3,1)	3	9	4.63	1.35	2.45	5.06	10.35
Forest	3	12	4.26	1.57	2.10	5.53	10.85
SM(3,1) w/ rej	3	12	30.38	1.56	16.40	32.87	58.25
SM(3,1)	3	12	25.81	1.52	13.83	24.29	45.07
Forest	3	15	2.19	1.66	0.32	3.47	14.92
SM(3,1) w/ rej	3	15	8.55	2.12	0.38	21.82	82.58
SM(3,1)	3	15	5.67	2.13	0.24	15.16	62.91
Forest	3	18	0.97	2.71	0.30	0.32	0.95
SM(3,1) w/ rej	3	18	1.31	2.40	0.23	0.29	1.62
SM(3,1)	3	18	0.89	2.40	0.16	0.21	1.24

Table 4.2: Ratio of effective size of sampler per 1M likelihood evaluations vs. CRF Gibbs sampler. $\theta = 3$

and 4.9. Figure 4.3 shows estimated Ramachandran probabilities given alanine and alanine or alanine and glycine as the central and right residue respectively.

In this dataset, the forest-based Gibbs sampler consistently outperformed the CRF Gibbs sampler, and split-merge steps further improved the effective sample sizes per unit of computation. Both the early rejection and early stopping rules substantially improved the autocorrelation per unit computation over the plain split-merge proposals. However, combining both the early rejection and early stopping rules did no better than the early rejection rule by itself.

We also examined the burn-in period in this example and found that the forest-based Gibbs sampler finds a good local mode significantly faster than the CRF Gibbs sampler. The split-merge sampler did not find modes significantly faster than the forest-based Gibbs sampler unless early rejection was used. With early rejection, a good mode was found faster. Early stopping did not appear to help reach a good mode faster. All samplers found similar values for the complete data log-likelihood after the burn in period.

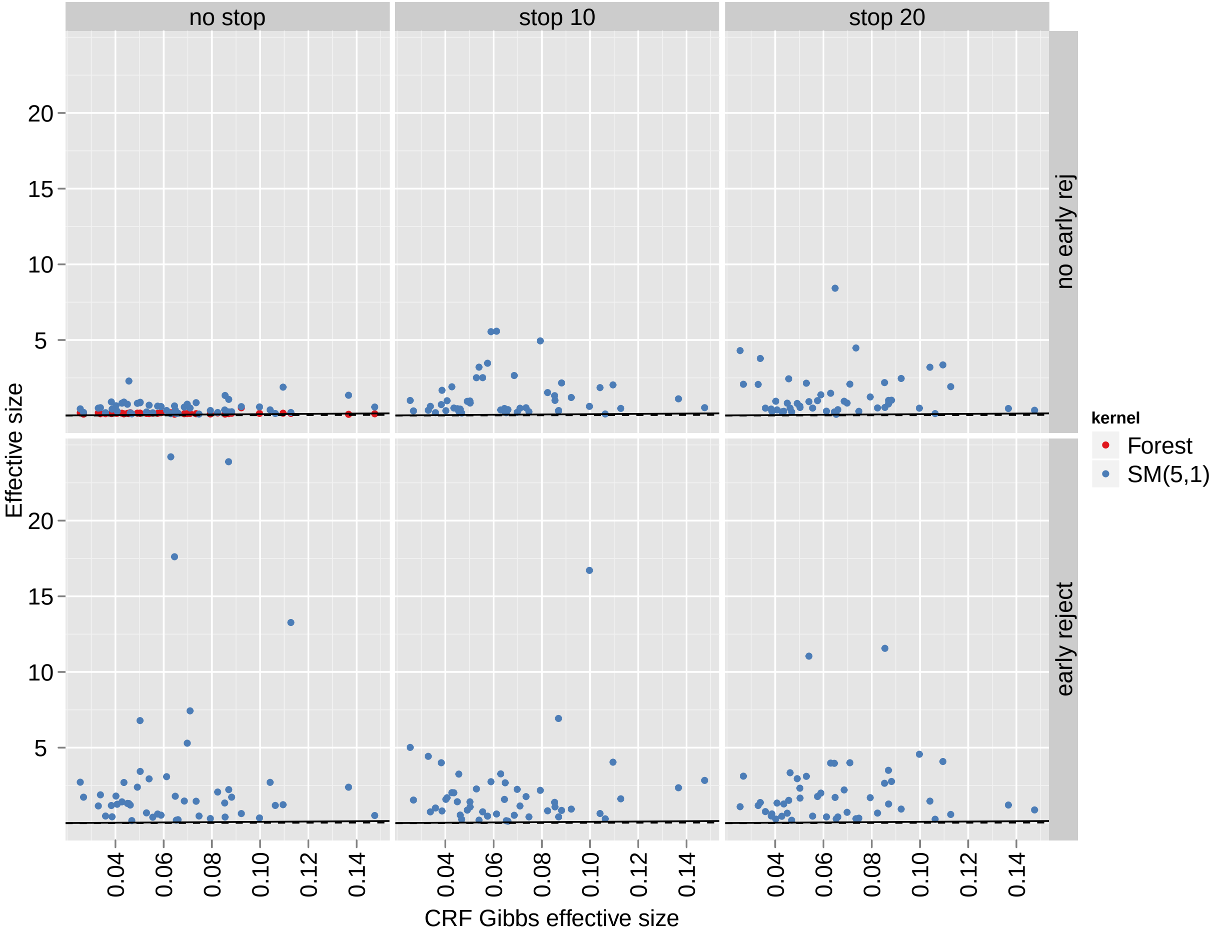


Figure 4.7: These figures show the effective sample size per 1 million likelihood evaluations for each of the proposed samplers in relation to the CRF Gibbs sampler. Each point represents a pair of chains with the same initialization. We see the split-merge samplers are the most effective, in particular the split-merge samplers with the optimizations of early stopping or early rejection have the best effective sample sizes.

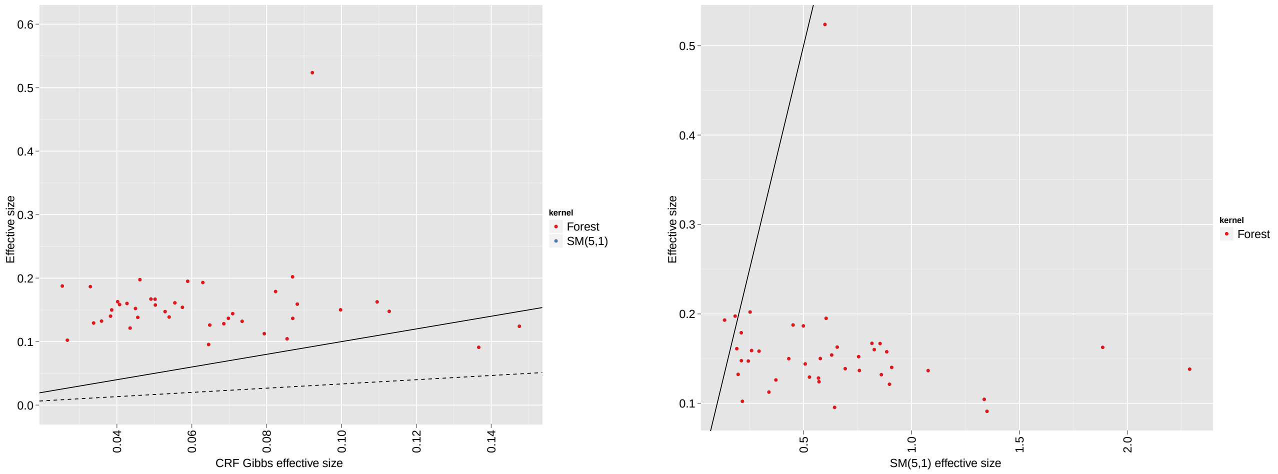


Figure 4.8: These figures show the effective sample size per 1 million likelihood evaluations for the Forest-based Gibbs samplers in relation to the CRF Gibbs sampler (left) and Split-Merge(5,1) sampler (right). As with the Beta-Binomial data sets, the solid line represents equal performance, and the dotted line represents the expected worse case performance. In this case, the forest-based Gibbs sampler consistently outperforms the CRF Gibbs sampler but is also consistently worse than the split-merge sampler.

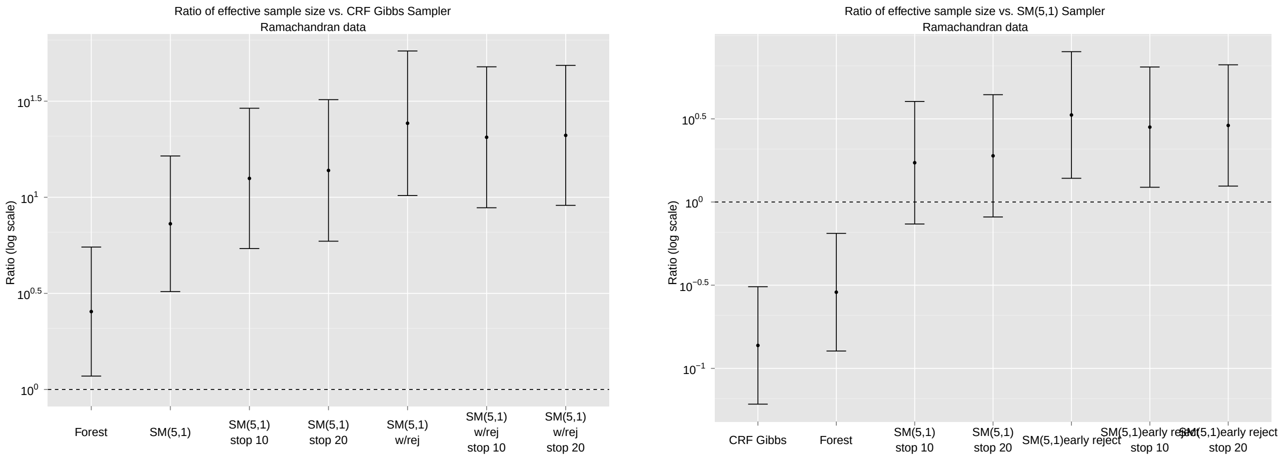


Figure 4.9: This figure shows the harmonic mean of the ratio of effective size of each sampler relative to the CRF Gibbs sampler (left) and SM(5,1) sampler(right). The bars represent 95% confidence intervals. All the newly proposed sampler significantly outperform the CRF Gibbs sampler. The split-merge samplers with early rejection also significantly outperform the plain split-merge sampler.

kernel	theta	dim	ratio	sd	25 pct	50 pct	75 pct
Forest	5	6	1.41	1.08	1.16	1.34	1.75
SM(3,1) w/ rej	5	6	1.64	1.09	1.23	1.67	2.15
SM(3,1)	5	6	1.27	1.10	0.91	1.34	1.73
Forest	5	9	2.04	1.24	1.69	2.35	2.97
SM(3,1) w/ rej	5	9	4.19	1.30	2.64	4.06	8.30
SM(3,1)	5	9	3.33	1.32	2.01	3.01	7.64
Forest	5	12	2.21	1.75	1.29	3.36	7.23
SM(3,1) w/ rej	5	12	16.59	1.79	16.74	25.60	51.46
SM(3,1)	5	12	13.96	1.78	12.25	19.09	42.30
Forest	5	15	1.05	1.71	0.35	0.60	4.78
SM(3,1) w/ rej	5	15	3.47	1.95	0.34	0.83	49.10
SM(3,1)	5	15	2.31	1.97	0.23	0.63	35.01
Forest	5	18	0.30	2.79	0.21	0.31	0.50
SM(3,1) w/ rej	5	18	0.89	2.18	0.22	0.38	0.92
SM(3,1)	5	18	0.60	2.18	0.17	0.25	0.60

Table 4.3: Ratio of effective size of sampler per 1M likelihood evaluations vs. CRF Gibbs sampler. $\theta = 5$

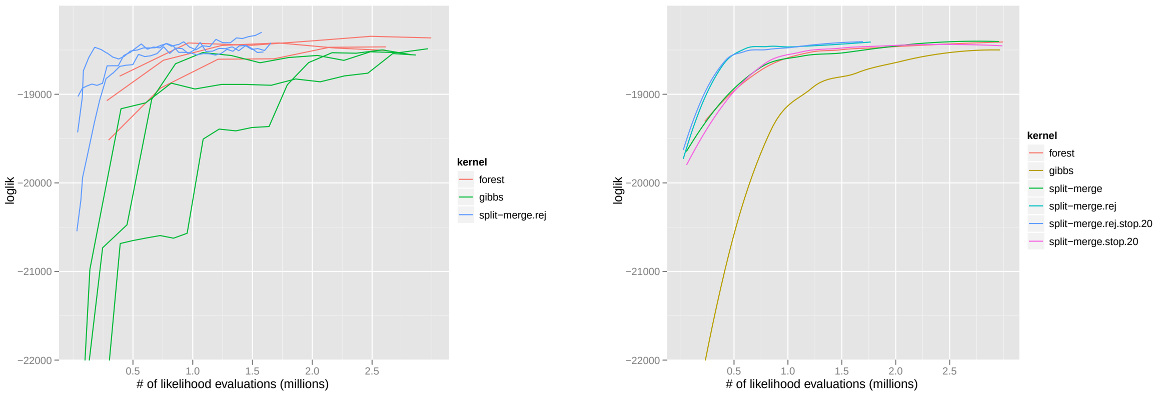


Figure 4.10: The left plot is a trace plot of the complete data log likelihood as a function of number of likelihood evaluations for the Arginine data set. The right plot is a smoothed version taken over 5 runs. Clearly, the split-merge sampler with early rejection and mixed with Gibbs steps in the forest representation performs the best, and the CRF Gibbs sampler does worst.

kernel	theta	dim	ratio	sd	25 pct	50 pct	75 pct
Forest	7	6	1.40	1.05	1.23	1.32	1.66
SM(3,1) w/ rej	7	6	1.48	1.07	1.19	1.31	1.90
SM(3,1)	7	6	1.21	1.08	0.93	1.17	1.62
SM(5,1)	7	6	0.75	1.13	0.47	0.59	1.25
Forest	7	9	1.83	1.20	1.20	1.92	2.76
SM(3,1) w/ rej	7	9	3.09	1.29	1.53	2.63	5.89
SM(5,1) w/ rej	7	9	4.07	1.39	1.37	3.74	10.73
SM(3,1)	7	9	2.55	1.35	1.32	2.22	5.89
Forest	7	12	2.42	1.74	1.79	4.24	8.88
SM(3,1) w/ rej	7	12	14.47	1.68	13.68	24.59	43.10
SM(5,1) w/ rej	7	12	34.50	1.75	35.66	71.46	114.14
SM(3,1)	7	12	12.09	1.76	10.03	20.01	30.66
Forest	7	15	2.36	1.52	0.43	3.00	7.82
SM(3,1) w/ rej	7	15	9.77	1.82	0.37	32.42	61.96
SM(5,1) w/ rej	7	15	15.25	2.06	0.27	70.25	134.39
SM(3,1)	7	15	4.70	1.95	0.26	19.32	29.80
Forest	7	18	1.01	2.35	0.30	0.38	2.50
SM(3,1) w/ rej	7	18	2.36	2.30	0.29	0.45	15.40
SM(5,1) w/ rej	7	18	2.36	2.41	0.21	0.30	17.77
SM(3,1)	7	18	1.22	2.25	0.19	0.36	3.82

Table 4.4: Ratio of effective size of sampler per 1M likelihood evaluations vs. CRF Gibbs sampler. $\theta = 7$

kernel	stop	rej	ratio	sd	25 pct	50 pct	75 pct
Forest	no stop	no early rej	2.54	1.08	1.80	2.79	3.69
SM(5,1)	no stop	no early rej	7.28	1.13	3.87	8.09	15.61
SM(5,1)	stop 10	no early rej	12.54	1.16	6.06	11.79	24.24
SM(5,1)	stop 20	no early rej	13.80	1.17	6.37	12.54	25.62
SM(5,1)	no stop	early reject	24.28	1.19	11.12	25.98	54.46
SM(5,1)	stop 10	early reject	20.52	1.16	10.08	21.52	41.53
SM(5,1)	stop 20	early reject	21.00	1.16	8.84	29.39	40.66

Table 4.5: Ramachandran data: Ratio of effective size of sampler per 1M likelihood evaluations vs. CRF Gibbs sampler.

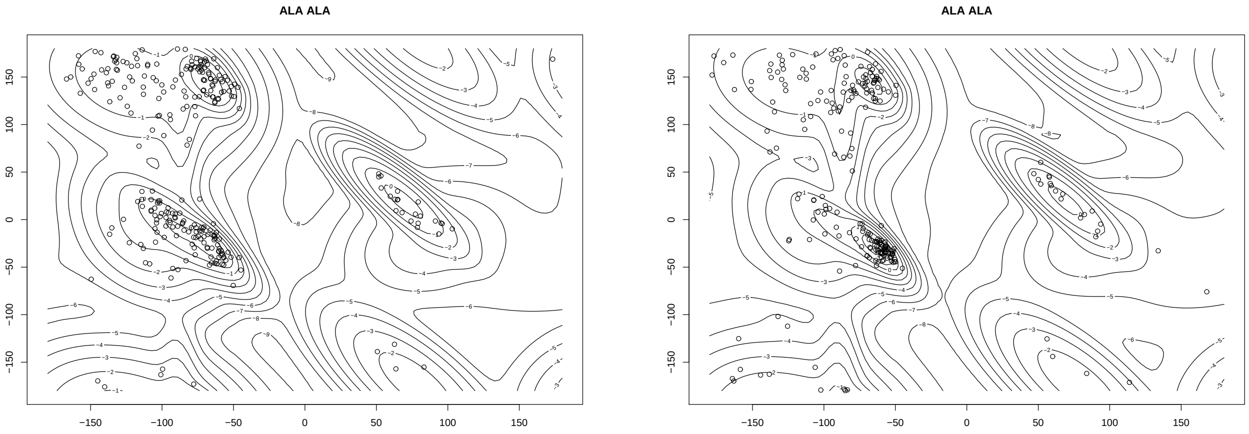


Figure 4.11: The left plot shows an estimated Ramachandran density given alanine and alanine as the central and right residues respectively. The right plot is the same but conditional on alanine and glycine instead. The contour plots show that while components are shared in the HDP, the different density estimates still how some variation between the different groups in the HDP.

4.4 Discussion of empirical results

Empirically, the forest-based Gibbs sampler was generally more effective than the standard CRF Gibbs sampler. This was true both for small synthetic datasets as well as a larger real world dataset. The addition of split-merge moves further improved the forest-based Gibbs sampler. However, for both the forest-based Gibbs split-merge samplers, if the clusters are well-defined, then the CRF Gibbs sampler performs significantly better. However, we regard this case as less interesting since only a few samples are needed to accurately approximate the posterior.

Of the two optimizations of early rejection and early stopping that we added to split-merge samplers, early rejection of bad merge proposals was most beneficial. While early stopping is an improvement over the plain split-merge sampler, it is unclear if it is beneficial when combined with early rejection.

We note that our version of the split-merge sampler is analogous to Dahl's sequentially allocated procedure for CRP mixture models. The split proposal is formed by taking the sequential procedure for drawing from the prior and adding a likelihood term. At the start of procedure to draw a split proposal, two blocks are initialized to contain nothing. One may also take Jain and Neal's approach of running a Gibbs sweep. For a split proposal in that approach, two blocks are initialized to a random split of the initial block. In the forest representation where the sufficient statistics on subtrees are memoized, performing a random split into two subtrees and calculating a single backward sweep does not take more computation than the sequential allocation approach. Such an approach may be useful when the prior places most of its mass in areas of low posterior probability.

Chapter 5

Conclusion

This thesis examines two distinct topics related to graphs. In each, our goals are to examine an existing problem under a new perspective and to demonstrate how the insights gained may be turned into practical tools.

In the first topic, we analyze how the graph construction method affects the limit operator of the graph Laplacian and analyze the relationship between graph Laplacians and LLE. This analysis covers most graph constructions of interest and can be easily extended to cover most graph constructions used in practice. Furthermore, it introduces the idea of pilot estimates and variable bandwidths to graph Laplacians and suggests which graph constructions have good theoretical and computational properties. A natural extension to this topic is to analyze the broader class of manifold learning methods. Though not included in this thesis, we have analyzed methods that lead to second-order smoothness functionals, namely Hessian LLE and local tangent space analysis (LTSA).

In the second topic, we give useful paradigms for viewing nonparametric Bayesian priors as combinatorial stochastic processes. In particular, we give representations of the underlying infinite stick-breaking processes as random graphs, and we introduce the discrete fragmentation and coagulation processes as a means to characterize priors for different hierarchical Bayesian models. These representations are also useful for developing MCMC algorithms. We give two new samplers for the hierarchical Dirichlet process and show their, sometimes dramatic, empirical improvement over existing samplers, and we sketch new algorithms for other nonparametric Bayesian models based on their combinatorial representations. These ideas are developed in the more general context of studying distributions on combinatorial objects, and we give one application of the link between random permutations and random trees to devise an algorithm for generating a random sample without replacement from distributed streams. We leave several topics only partially explored in this topic including a more formal treatment of the MCMC samplers which we only give a sketch of as well as exploring a particularly attractive hierarchical model with levels distributed as a Pitman-Yor process.

Bibliography

I. S. Abramson. On bandwidth variation in kernel estimates-a square root law. *The Annals of Statistics*, 10:1217–1223, 1982.

R. P. Adams, Z. Ghahramani, and M. I. Jordan. Tree-structured stick breaking for hierarchical data. In *Advances in Neural Information Processing Systems (NIPS)*, 2010.

M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

M. Belkin and P. Niyogi. Semi-supervised learning on Riemannian manifolds. *Machine Learning*, 56:209–239, 2004.

M. Belkin and P. Niyogi. Towards a theoretical foundation for Laplacian-based manifold methods. *Conference on Learning Theory (COLT)*, 2005.

M. Belkin and P. Niyogi. Convergence of Laplacian eigenmaps. In *Advances in Neural Information Processing Systems (NIPS)*, 2006.

J. Besag. Discussion of 'On Bayesian analysis of mixtures with an unknown number of components,' by S. Richardson and P. J. Green. *Journal of the Royal Statistical Society Series B*, 59:774, 1997.

J. Besag, P. Green, D. Higdon, and K. Mengersen. Bayesian computation and stochastic systems. *Statistical Science*, 10(1):3–41, 1995.

D. M. Blei and P. Frazier. Distance dependent Chinese restaurant processes. In *International Conference on Machine Learning*, 2010.

D. M. Blei, T. L. Griffiths, and M. I. Jordan. The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies. *Journal of the ACM*, 57(2):7, 2010.

W. M. Boothby. *An Introduction to Differentiable Manifolds and Riemannian Geometry*. Academic Press, 1986.

O. Bousquet, O. Chapelle, and M. Hein. Measure based regularization. In *Advances in Neural Information Processing Systems (NIPS)*, 2003.

B. P. Carlin and S. Chib. Bayesian model choice via Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society Series B*, pages 473–484, 1995.

S. Chaiken. A combinatorial proof of the all minors matrix tree theorem. *SIAM Journal on Algebraic and Discrete Methods*, 3(3):319–329, 1982.

J. A. Christen and C. Fox. MCMC using an approximation. *Journal of Computational and Graphical statistics*, 14(4): 795–810, 2005.

D. B. Dahl. An improved merge-split sampler for conjugate Dirichlet process mixture models. *Department of Statistics, University of Wisconsin Technical Report*, 1086, 2003.

P. Dellaportas, J. J. Forster, and I. Ntzoufras. On Bayesian model and variable selection using MCMC. *Statistics and Computing*, 12(1):27–36, 2002.

L. P. Devroye and T. J. Wagner. The strong uniform consistency of nearest neighbor density estimates. *The Annals of Statistics*, pages 536–540, 1977.

P. Dostert, Y. Efendiev, T. Y. Hou, and W. Luo. Coarse-gradient Langevin algorithms for dynamic data integration and uncertainty quantification. *Journal of Computational Physics*, 217(1):123–142, 2006.

P. S. Efraimidis and P. G. Spirakis. Weighted random sampling with a reservoir. *Information Processing Letters*, 97 (5):181–185, 2006.

S. Ethier and T. Kurtz. *Markov Processes: Characterization and Convergence*. Wiley, 1986.

E. Giné and V. Koltchinskii. Empirical graph Laplacian approximation of Laplace-Beltrami operators: large sample results. In *4th International Conference on High Dimensional Probability*, 2005.

S.J. Godsill. On the relationship between Markov chain Monte Carlo methods for model uncertainty. *Journal of Computational and Graphical Statistics*, 10(2):230–248, 2001.

P. J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711, 1995.

A. Grigor'yan. Heat kernels on weighted manifolds and applications. *Cont. Math*, 398:93–191, 2006.

M. Hein, J. Audibert, and U. Von Luxburg. From Graphs to Manifolds - Weak and Strong Pointwise Consistency of Graph Laplacians. In *Conference on Learning Theory (COLT)*, 2005.

M. Hein, J.-Y. Audibert, and U. von Luxburg. Graph Laplacians and their convergence on random neighborhood graphs. *Journal of Machine Learning Research*, 8:1325–1370, 2007.

J. Jacod and A. N. Širjaev. *Limit Theorems for Stochastic Processes*. Springer, 2003.

S. Jain and R. M. Neal. A split-merge Markov chain Monte Carlo procedure for the Dirichlet process mixture model. *Journal of Computational and Graphical Statistics*, 13(1):158–182, 2004.

S. Jain and R. M. Neal. Splitting and merging components of a nonconjugate Dirichlet process mixture model. *Bayesian Analysis*, 2(3):445–472, 2007.

O. Kallenberg. *Foundations of Modern Probability*. Springer Verlag, 2002.

R. Kannan, S. Vempala, and A. Vetta. On clusterings: Good, bad and spectral. *Journal of the ACM*, 51(3):497–515, 2004.

J. F. Kingman. The coalescent. *Stochastic processes and their applications*, 13(3):235–248, 1982.

M. Kolonko and D. Wäsch. Sequential reservoir sampling with a nonuniform distribution. *ACM Transactions on Mathematical Software*, 32(2):257–273, June 2006.

S. Lafon. *Diffusion Maps and Geometric Harmonics*. PhD thesis, Yale University, CT, 2004.

J. S. Liu. *Monte Carlo strategies in scientific computing*. Springer Verlag, 2008.

D. O. Loftsgaarden and C. P. Quesenberry. A nonparametric estimate of a multivariate density function. *Annals of Mathematical Statistics*, 36(3):1049–1051, 1965.

M. Maier, U. von Luxburg, and M. Hein. Influence of graph construction on graph-based clustering measures. In *Advances in Neural Information Processing Systems (NIPS)*, 2008.

I. Murray. *Advances in Markov chain Monte Carlo methods*. PhD thesis, University College London, 2007.

B. Nadler, S. Lafon, R.R. Coifman, and I.G. Kevrekidis. Diffusion maps, spectral clustering and reaction coordinates of dynamical systems. *Applied and Computational Harmonic Analysis*, 21(1):113–127, 2006.

A. Ozakin. Submanifold Density Estimation. In *Advances in Neural Information Processing Systems 22 (NIPS)*, 2009.

B. Pelletier. Kernel density estimation on Riemannian manifolds. *Statistics and Probability Letters*, 73(3):297 – 304, 2005.

J. Pitman and J. Picard. *Combinatorial Stochastic Processes*. Springer, 2006.

M. Plummer, N. Best, K. Cowles, and K. Vines. CODA: Convergence Diagnosis and Output Analysis for MCMC. *R News*, 6(1):7–11, 2006.

A. Rodriguez, D. B. Dunson, and A. E. Gelfand. The nested Dirichlet process. *Journal of the American Statistical Association*, 103(483):1131–1154, 2008.

S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500): 2323, 2000.

B. W. Silverman. *Density estimation for statistics and data analysis*. Chapman and Hall/CRC, 1998.

A. Singer. From graph to manifold Laplacian: the convergence rate. *Applied and Computational Harmonic Analysis*, 21(1):128–134, 2006.

Y. W. Teh. A Bayesian interpretation of interpolated Kneser-Ney. Technical report, 2006.

Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

Y. W. Teh, H. Daumé III, and D. M. Roy. Bayesian agglomerative clustering with coalescents. In *Advances in Neural Information Processing Systems*, 2008.

G. R. Terrell and D. W. Scott. Variable kernel density estimation. *The Annals of Statistics*, 20(3):1236–1265, 1992.

R. J. Thibaux. *Nonparametric Bayesian Models for Machine Learning*. PhD thesis, UC Berkeley, 2008.

L. Tierney and A. Mira. Some adaptive Monte Carlo methods for Bayesian inference. *Statistics in Medicine*, 18(1718): 2507–2515, 1999.

D. Ting, G. Wang, M. Shapovalov, R. Mitra, M. I. Jordan, and R. L. Dunbrack. Neighbor-dependent Ramachandran probability distributions of amino acids developed from a hierarchical Dirichlet process model. *PLoS Computational Biology*, 6(4):e1000763, 2010.

J. S. Vitter. Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)*, 11(1):37–57, 1985.

U. von Luxburg, M. Belkin, and O. Bousquet. Consistency of spectral clustering. *Annals of Statistics*, 36(2):555–586, 2008.

C. Wang and D. M. Blei. A Split-Merge MCMC Algorithm for the Hierarchical Dirichlet Process. *Arxiv preprint arXiv:1201.1657*, 2012.

H. S. Wilf and A. Nijenhuis. *Combinatorial algorithms*. Society for Industrial Mathematics, 1989.

L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In *Advances in Neural Information Processing Systems (NIPS)*, 2004.

X. Zhu. *Semi-Supervised Learning with Graphs*. PhD thesis, Carnegie Mellon University, 2005. CMU-LTI-05-192.

X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. In *International Conference on Machine Learning (ICML)*, 2003.