

Efficient sampling from the Bingham distribution

Rong Ge¹, Holden Lee², Jianfeng Lu², and Andrej Risteski³

¹Duke University, Computer Science Department
rongge@cs.duke.edu

²Duke University, Mathematics Department
{holee, jianfeng}@math.duke.edu

³Carnegie Mellon University, Machine Learning Department
aristeski@andrew.cmu.edu

September 30, 2020

Abstract

We give a algorithm for exact sampling from the Bingham distribution $p(x) \propto \exp(x^\top Ax)$ on the sphere \mathcal{S}^{d-1} with expected runtime of $\text{poly}(d, \lambda_{\max}(A) - \lambda_{\min}(A))$. The algorithm is based on rejection sampling, where the proposal distribution is a polynomial approximation of the pdf, and can be sampled from by explicitly evaluating integrals of polynomials over the sphere. Our algorithm gives exact samples, assuming exact computation of an inverse function of a polynomial. This is in contrast with Markov Chain Monte Carlo algorithms, which are not known to enjoy rapid mixing on this problem, and only give approximate samples.

As a direct application, we use this to sample from the posterior distribution of a rank-1 matrix inference problem in polynomial time.

Keywords: Sampling, Bingham distribution, posterior inference, non-log-concave

1 Introduction

Sampling from a probability distribution p given up to a constant of proportionality is a fundamental problem in Bayesian statistics and machine learning. A common instance of this in statistics and machine learning is posterior inference (sampling the parameters of a model θ , given data x), where the unknown constant of proportionality comes from an application of Bayes rule: $p(\theta|x) \propto p(x|\theta)p(\theta)$.

However, for standard approaches to sampling such as the Langevin Monte Carlo algorithm, provable results on efficient (polynomial-time) sampling often require that p be log-concave or close to log-concave. In this work, we consider the problem of sampling from a specific non-log-concave probability distribution on the sphere \mathcal{S}^{d-1} in d dimensions: the *Bingham distribution*. In addition to having applications in statistics, the Bingham distribution is of particular interest as it models the local behavior of any smooth distribution around a stationary point.

We give a polynomial-time algorithm based on approximating the probability density function by a polynomial and explicitly evaluating its integral over the sphere. Our algorithm is of Las

Vegas type: It has the advantage of giving *exact* samples, assuming exact computation of an inverse function of a polynomial. Our approach contrasts with the usual Markov Chain Monte Carlo algorithms, which are not known to enjoy rapid mixing on this problem, and only give approximate samples.

The Bingham distribution [Bin74] defined by a matrix $A \in \mathbb{R}^{d \times d}$ is the distribution on the sphere $\mathcal{S}^{d-1} \subseteq \mathbb{R}^d$ whose density function with respect to the uniform (surface) measure is given by

$$p(x) := \frac{dP}{d\mu_{\mathcal{S}^{d-1}}}(x) \propto \exp(x^\top Ax).$$

Note that due to the symmetric form, without loss of generality, we can assume A is symmetric. This distribution finds frequent use in *directional statistics*, which studies distributions over the unit sphere. In particular, the Bingham distribution is widely used in paleomagnetic data analysis [Ons80] and has applications to computer vision [AT00; HW08; GP13] and even differential privacy [CSS13; WWS15]. As shown in Section 1.2, it also naturally appears in the posterior distribution for a rank-1 matrix inference problem, a special case of matrix factorization.

Our main theorem is given below. In the following, we will identify a probability distribution over \mathcal{S}^{d-1} with its density function with respect to the uniform measure on \mathcal{S}^{d-1} .

Theorem 1.1. *Let A be a symmetric matrix with maximum and minimum eigenvalue λ_{\max} and λ_{\min} , respectively. Let $p(x) \propto \exp(x^\top Ax)$ be a probability distribution over \mathcal{S}^{d-1} . Then, given an oracle for solving a univariate polynomial equation, Algorithm 1 produces a sample from $p(x)$ and runs in expected time $\text{poly}(\lambda_{\max} - \lambda_{\min}, d)$.*

We can consider the Bingham distribution as a “model” non-log-concave distribution, because any smooth probability distribution looks like a Bingham distribution in a sphere of small radius around a stationary point. More precisely, suppose $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is 3-times differentiable, $p(x) = e^{-f(x)}$ on \mathbb{R}^d , and $\nabla f(x_0) = 0$. Then we have that as $x \rightarrow x_0$,

$$p(x) = \exp \left\{ -[f(x_0) + (x - x_0)^\top (\nabla^2 f(x_0))(x - x_0) + O(\|x - x_0\|^3)] \right\}.$$

Note that if we can sample from small spheres around a point, we can also sample from a small ball around the point by first estimating and sampling from the marginal distribution of the radius.

Moreover, the Bingham distribution already illustrates the challenges associated with sampling non-log-concave distributions. First, it can be arbitrarily non-log-concave, as the minimum eigenvalue of the Hessian can be arbitrarily negative. Second, when A has distinct eigenvalues, the function $f(x) = x^\top Ax$ on \mathcal{S}^{d-1} has $2(d-1)$ saddle points and 2 minima which are antipodal. Hence, understanding how to sample from the Bingham distribution may give insight into sampling from more general non-log-concave distributions.

1.1 Related work

We first discuss general work on sampling, and then sampling algorithms specific to the Bingham distribution.

¹The more general *Fisher-Bingham distribution* includes a linear term, and so can locally model any smooth probability distribution.

Langevin Monte Carlo [RDF78; RT96] is a generic algorithm for sampling from a probability distribution $p(x) \propto e^{-f(x)}$ on \mathbb{R}^d given gradient access to its negative log-pdf f . It is based on discretizing Langevin diffusion, a continuous Markov process. In the case where p is log-concave, Langevin diffusion is known to mix rapidly [BE85], and Langevin Monte Carlo is an efficient algorithm [Dal16; DM16]. More generally, for Langevin diffusion over a compact manifold (such as \mathcal{S}^{d-1}), positive Ricci curvature can offset non-log-concavity of p , and rapid mixing continues to hold if the sum of the Hessian of f and Ricci curvature at any point is lower bounded by a positive constant [BE85; Hsu02]. In our setting, this is only the case when the maximum and minimum eigenvalues of A differ by less than $\frac{d-1}{2}$: $\lambda_{\max}(A) - \lambda_{\min}(A) < \frac{d-1}{2}$. We note there are related algorithms such as Hamiltonian Monte Carlo [Dua+87] that are more efficient in the log-concave case, but still suffer from torpid mixing in general.

Next, we consider algorithms tailored for the Bingham distribution. An important observation is that the normalizing constant of the Bingham distribution is given by the hypergeometric function of a matrix argument [MJ09, p.182],

$$\int_{\mathcal{S}^{d-1}} \exp(x^\top Ax) d\mathcal{S}^{d-1}(x) = {}_1F_1\left(\frac{1}{2}; \frac{n}{2}; D\right)^{-1}$$

where D is the diagonal matrix of eigenvalues of A . Methods to approximate the hypergeometric function are given in [KE06], however, with super-polynomial dependence on the degree of the term where it is truncated, and hence on the accuracy required.

The previous work [KGM13] gives a rejection sampling based algorithm where the proposal distribution is an angular central gaussian envelope, that is, the distribution of a normalized gaussian random variable. This distribution has density function $p(x) \propto (x^\top \Omega x)^{-d/2}$ for Ω chosen appropriately depending on A . The efficiency of rejection sampling is determined by the maximum ratio between the desired ratio and the proposal distribution. Their bound for this ratio depends on the normalizing constant for the Bingham distribution [KGM13, (3.5)], and they only give a polynomial-in-dimension bound when the temperature approaches zero (that is, for the distribution $\exp(\beta x^\top Ax)$ as $\beta \rightarrow \infty$). Our algorithm is also based on rejection sampling; however, we use a more elaborate proposal distribution, for which we are able to show that the ratio is bounded at all temperatures.

1.2 Application to rank-1 matrix inference

The algorithm we give has an important application to a particularly natural statistical inference problem: that of recovering a rank-1 matrix perturbed by Gaussian noise.

More precisely, suppose that an observation Y is produced as follows: we sample $x \sim \mathcal{D}$ for a prior distribution \mathcal{D} and $N \sim \mathcal{N}(0, \gamma^2 I)$, then output $Y = xx^\top + N$. By Bayes Rule, the posterior distribution over x has the form

$$p(x|Y) \propto \exp\left(-\frac{1}{2\gamma^2} \|Y - xx^\top\|_F^2\right) p(x). \tag{1}$$

In the particularly simple case where \mathcal{D} is uniform over the unit sphere, this posterior has the form we study in our paper:

$$p(x|Y) \propto \exp\left(\frac{1}{2\gamma^2} x^\top Y x\right)$$

for $x \in \mathcal{S}^{d-1}$. Thus, we are able to do posterior sampling. More generally, for radially symmetric $p(x)$, we can approximately sample from the radial distribution of the marginal, after which the problem reduces to a problem on \mathcal{S}^{d-1} . Note that our algorithm does not require the model to be well-specified, i.e., it does not require Y to be generated from the hypothesized distribution.

In existing literature, the statistics community has focused more on questions of *recovery* (can we achieve a non-trivial “correlation” with the planted vector x under suitable definitions of correlation) and *detection* (can we decide with probability $1 - o(1)$ as $d \rightarrow \infty$ whether the matrix presented is from the above distribution with a “planted” vector x , or is sampled from a Gaussian) under varying choices for the prior \mathcal{D} . In particular, they study the threshold for γ at which each of the respective tasks is possible. The two most commonly studied priors \mathcal{D} are uniform over the unit sphere (*spiked Wishart model*), and the coordinates of x being $\pm \frac{1}{\sqrt{d}}$ uniformly at random (*spiked Wigner*). For a recent treatment of these topics, see e.g., [Péc06; Per+18].

However, the statistical tests involve calculating integrals over the posterior distribution (□) (for instance, the MMSE $\hat{x}\hat{x}^\top = \frac{\int xx^\top \exp(-\frac{1}{2\gamma^2}\|Y-xx^\top\|_F^2)p(x) dx}{\int \exp(-\frac{1}{2\gamma^2}\|Y-xx^\top\|_F^2)p(x) dx}$), and the question of algorithmic efficiency of this calculation is not considered. Our work makes these statistical tests algorithmic (for spherically symmetric priors), because integrals over the posterior distribution can be approximated through sampling.

On the algorithmic side, the closest relative to our work is the paper by [MR20], which considers the low-rank analogue of the problem we are interested in: namely, sampling from the distribution

$$p(X) \propto \exp\left(-\frac{1}{2\gamma^2}\|XX^\top - Y\|_F^2\right)$$

supported over matrices $X \in \mathbb{R}^{d \times k}$, s.t. $Y = X_0 X_0^\top + \gamma N$, for some matrix $X_0 \in \mathbb{R}^{d \times k}$ and $N \sim \mathcal{N}(0, I)$. It proves that a slight modification of Langevin Monte Carlo can be used to sample from this distribution efficiently in the *low-temperature* limit, namely when $\gamma = \Omega(d)$.

For comparison, in this paper, we can handle an *arbitrary* temperature, but only the rank-1 case (i.e. $k = 1$). Moreover, the algorithm here is substantially different, based on a polynomial approximation of the pdf, rather than MCMC. Extending either approach to the full regime (arbitrary k and arbitrary temperature) is an important and challenging problem.

2 Algorithm based on polynomial approximation

We present our rejection sampling algorithm as Algorithm □. Our main theorem is the following.

Theorem 1.1. *Let A be a symmetric matrix with maximum and minimum eigenvalue λ_{\max} and λ_{\min} , respectively. Let $p(x) \propto \exp(x^\top Ax)$ be a probability distribution over \mathcal{S}^{d-1} . Then, given an oracle for solving a univariate polynomial equation, Algorithm □ produces a sample from $p(x)$ and runs in expected time $\text{poly}(\lambda_{\max} - \lambda_{\min}, d)$.*

Before proceeding to the proof of Theorem □.1, we make a few remarks about the statement. Firstly, we work in the real model of computation. Solving a polynomial equation can be done to machine precision using binary search, so the only errors present when actually running the algorithm are roundoff errors.

Algorithm 1 Sampling algorithm for Bingham distribution

Input: Symmetric matrix A
Output: A random sample $x \sim p(x) \propto \exp(x^\top Ax)$ on \mathcal{S}^{d-1}

- 1: Diagonalize $[V, \Lambda] = \text{diag}(A)$ such that $A = V\Lambda V^\top$; let λ_{\min} and λ_{\max} denote the smallest and largest eigenvalues respectively;
 - 2: Set $D = \Lambda - \lambda_{\min}I_d$;
 - 3: Set $n = (\lambda_{\max} - \lambda_{\min})^2$;
 - 4: **repeat** ▷ Rejection sampling for $z \sim \tilde{p}(z) \propto \exp(z^\top Dz)$ on \mathcal{S}^{d-1}
 - 5: **for** $i = 1 \rightarrow d$ **do** ▷ Sample proposal $z \sim q(z) \propto (z^\top (I + D/n)z)^n$ on \mathcal{S}^{d-1}
one coordinate at a time
 - 6: **if** $i = 1$ **then**
 - 7: Let $D_1 = D$;
 - 8: Determine the marginal distribution $q(z_1)$ whose pdf is given as follows, where $(D_1)_{-1}$ represents the submatrix of D_1 obtained from deleting the first row and column (see Theorem 2.4, (5) for details)
 - $$\frac{1}{Z} \int_{y \in \mathcal{S}^{d-2}} \left((1 + (D_1)_{11}/n)z_1^2 + (1 - z_1^2)y^\top (I_{d-1} + (D_1)_{-1}/n)y \right)^n d\mathcal{S}^{d-2}(y);$$
 - 9: Sample $z_1 \sim q(z_1)$ via inverse transform sampling (Lemma 2.3);
 - 10: Let $y_1 = z_1$;
 - 11: **else**
 - 12: Let $D_i = y_{i-1}^2(D_{i-1})_{11} + (1 - y_{i-1}^2)(D_{i-1})_{-1} \in \mathbb{R}^{(d-i+1) \times (d-i+1)}$;
▷ We will sample from the distribution $\propto (y^\top (I + D_i/n)y)^n$.
 - 13: Determine the conditional marginal distribution $q(y_i|z_1, \dots, z_{i-1})$ where $z_i = y_i \sqrt{1 - \sum_{j=1}^{i-1} z_j^2}$, whose pdf is given by (see Theorem 2.4, (5) for details)
 - $$\frac{1}{Z} \int_{(y_{i+1}, \dots, y_d) \in \mathcal{S}^{d-i-1}} \left((1 + (D_i)_{11}/n)y_i^2 + (1 - y_i^2)y^\top (I_{d-i} + (D_i)_{-1}/n)y \right)^n d\mathcal{S}^{d-i-1}(y);$$
 - 14: Sample $y_i \sim q(y_i|z_1, \dots, z_{i-1})$ via inverse transform sampling (Lemma 2.3);
 - 15: Let $z_i = y_i \sqrt{1 - \sum_{j=1}^{i-1} z_j^2}$;
 - 16: **end if**
 - 17: **end for**
 - 18: Accept z with probability $e^{-1} \frac{\exp(z^\top Dz)}{(z^\top (I+D/n)z)^n}$; ▷ Rejection sampling (see proof of Theorem 1.1 for explanation of the e^{-1} factor)
 - 19: **until** the sample z is accepted;
 - 20: **return** $x = Vz$;
-

The algorithm is based on rejection sampling: we calculate a proposal sample in time $\text{poly}(\lambda_{\max} - \lambda_{\min}, d)$, accept it with some probability, and otherwise repeat the process. In the parlance of algorithms, this means that it is a Las Vegas algorithm: it produces an exact sample, but has a randomized runtime. For the analysis, we lower bound the acceptance probability by an absolute constant. The number of proposals until acceptance follows a geometric distribution with success probability equal to the acceptance probability. Hence, the total time is polynomial with high

probability.

The analysis of our algorithm proceeds in the following steps:

1. By diagonalization and change-of-coordinates, we show that it suffices to provide an algorithm for sampling from distributions over the unit sphere $p : \mathcal{S}^{d-1} \rightarrow \mathbb{R}^+$ in the form

$$p(x) \propto \exp(x^\top Dx),$$

where $D \in \mathbb{R}^{d \times d}$ is diagonal and PSD.

2. We show that if we use $q(x) \propto (x^\top (I + D/n)x)^n$ as a proposal distribution, when $n \geq D_{\max}^2$ the ratio $\max\{\frac{p(x)}{q(x)}, \frac{q(x)}{p(x)}\}$ is bounded by an absolute constant.
3. We then show that CDF for the marginal distributions of $q(x)$ can be computed explicitly in polynomial time (in n, d), therefore using inverse transform sampling, one can sample from q in polynomial time.

Change-of-coordinates We first argue that it suffices to provide an algorithm for sampling from distributions over the unit sphere $p : \mathcal{S}^{d-1} \rightarrow \mathbb{R}^+$ in the form

$$p(x) \propto \exp(x^\top Dx)$$

where $D \in \mathbb{R}^{d \times d}$. To see this, note that if $A = VDV^\top$ with D diagonal and V orthogonal, then given a sample x from the distribution $\propto \exp(x^\top Dx)$, Vx is a sample from the distribution $\propto \exp(x^\top VDV^\top x)$. Moreover, we can assume that D is a PSD diagonal matrix, with smallest eigenvalue $D_{\min} = 0$ and largest eigenvalue D_{\max} . This is because replacing D by $D - cI_d$ simply multiplies $\exp(x^\top Dx)$ by a constant on \mathcal{S}^{d-1} , and we can take $c = D_{\min}$.

Proposal distribution Next we give a proposal distribution for rejection sampling based on polynomial approximation of p :

Lemma 2.1. *Let $D \in \mathbb{R}^{d \times d}$ be diagonal with maximum eigenvalue $D_{\min} \geq 0$ and largest eigenvalue D_{\max} . Let the distribution $q : \mathcal{S}^{d-1} \rightarrow \mathbb{R}^+$ be defined as $q(x) \propto (x^\top (I + D/n)x)^n$ for $n \geq 1$. Then,*

$$\max \left\{ \frac{p(x)}{q(x)}, \frac{q(x)}{p(x)} \right\} \leq \exp \left(\frac{D_{\max}^2}{2n} \right).$$

Moreover, if $D_{\min} = 0$, letting v be a unit eigenvector with eigenvalue 0, $1 \leq \frac{q(v)}{p(v)} \leq \exp(\frac{D_{\max}^2}{2n})$.

Note that only an upper bound on $\frac{p(x)}{q(x)}$ is necessary for rejection sampling; however, the lower bound comes for free with our approach.

Proof. First, we show that

$$-\frac{D_{\max}^2}{2n} \leq n \log(x^\top (I + D/n)x) - x^\top Dx \leq 0. \quad (2)$$

By Taylor's theorem with remainder, we have for $x \in \mathcal{S}^{d-1}$ that

$$\begin{aligned} \log(x^\top (I + D/n)x) &= \log(1 + x^\top Dx/n) \\ &= \frac{x^\top Dx}{n} - \frac{1}{2} \frac{1}{(1 + \xi)^2} \left(\frac{x^\top Dx}{n} \right)^2 \quad \text{for some } \xi \in [0, x^\top Dx/n]. \end{aligned}$$

Because $\|x\| = 1$, we have $x^\top Dx/n \leq D_{\max}/n$, so

$$\log(x^\top (I + D/n)x) \in \left[\frac{x^\top Dx}{n} - \frac{D_{\max}^2}{2n^2}, \frac{x^\top Dx}{n} \right]$$

Multiplying by n , (2) follows. Now (2) implies by exponentiation that

$$\exp\left(-\frac{D_{\max}^2}{2n}\right) \leq \frac{(x^\top (I + D/n)x)^n}{\exp(x^\top Dx)} \leq 1$$

and hence

$$\begin{aligned} \exp\left(-\frac{D_{\max}^2}{2n}\right) &\leq \frac{(x^\top (I + D/n)x)^n}{\int_{\mathcal{S}^{d-1}} (x^\top (I + D/n)x)^n d\mathcal{S}^{d-1}(x)} \bigg/ \frac{\exp(x^\top Dx)}{\int_{\mathcal{S}^{d-1}} \exp(x^\top Dx) d\mathcal{S}^{d-1}(x)} \\ &\leq \exp\left(\frac{D_{\max}^2}{2n}\right) \end{aligned}$$

from which the lemma immediately follows.

For the last statement, note that for $x = v$, the numerators $(x^\top (I + D/n)x)^n$ and $\exp(x^\top Dx)$ in the above expression both equal 1. \square

Sampling from proposal q Finally, we show that it is possible to sample from $q(x)$ efficiently in time polynomial in n, d . First we show that the high order moments for quadratic forms can be computed efficiently.

Lemma 2.2 (Calculating integrals of quadratic forms). *The integral*

$$\int_{\mathcal{S}^{d-1}} (x^\top Dx)^n d\mathcal{S}^{d-1}(x)$$

can be calculated in time $\text{poly}(n, d)$.

Proof. The result follows essentially from known formulas about moments of quadratic functions under a Gaussian distribution.

First, we show the task reduces to calculating

$$\mathbb{E}_{x \sim N(0, I_d)}[(x^\top Dx)^n].$$

A Gaussian can be sampled by sampling the norm of x and the direction of x independently. Hence,

$$\mathbb{E}_{x \sim N(0, I_d)}[(x^\top Dx)^n] = \mathbb{E}_{x \sim N(0, I_d)}[\|x\|^{2n}] \cdot \mathbb{E}_{x \sim N(0, I_d)} \left[\left(\left(\frac{x}{\|x\|} \right)^\top D \left(\frac{x}{\|x\|} \right) \right)^n \right]. \quad (3)$$

The second factor is (up to a constant) the integral of interest as $\frac{x}{\|x\|}$ is uniformly distributed over the sphere:

$$\mathbb{E}_{x \sim \mathcal{S}^{d-1}}[(x^\top Dx)^n] = \frac{\int_{\mathcal{S}^{d-1}} (x^\top Dx)^n d\mathcal{S}^{d-1}(x)}{\text{Vol}(\mathcal{S}^{d-1})} = \frac{\int_{\mathcal{S}^{d-1}} (x^\top Dx)^n d\mathcal{S}^{d-1}(x)}{2\pi^d/\Gamma(d/2)}.$$

The first factor in (3) has a simple closed-form expression given by Corollary A.3.

Thus it remains to calculate the LHS of (3), the expectation under the Gaussian. We use the recurrence from [Kan08], reprinted here as Corollary A.2: denoting $S(n) = \frac{1}{n!2^n} \mathbb{E}_{x \sim N(0, I_d)}[(x^\top Dx)^n]$, we have $S(0) = 1$ and for $n \geq 1$,

$$S(n) = \frac{1}{2n} \sum_{i=1}^n \text{Tr}(D^i) S(n-i) \tag{4}$$

which can be calculated in time $\text{poly}(n, d)$ by dynamic programming. □

Using this integral, we can compute the unnormalized cdf for the marginals of distribution q . This can then be combined with the technique of *inverse transform sampling*.

Lemma 2.3 (Inverse transform sampling). *Suppose that we know that the probability distribution on $[a, b]$ has pdf $p(x) \propto f(x)$, and we can calculate the (unnormalized) cdf $F(x) = \int_a^x f(t) dt$. Then given an oracle for computing the inverse of $G(x) = F(x)/F(b)$, one can sample from the distribution.*

Proof. The algorithm simply generates a uniformly random number $r \in [0, 1]$ and computes $G^{-1}(r)$. Since $G(x)$ is the cdf of the probability distribution we know $G^{-1}(r)$ is exactly a random variable from this probability distribution $p(x)$. □

Note that when the cdf $F(x)$ is a polynomial, it is possible to compute G^{-1} with accuracy ε in $\text{poly} \log(1/\varepsilon)$ time by binary search.

Combining Lemma 2.2 and 2.3 we are ready to show that one can sample from $q(x)$ efficiently.

Theorem 2.4. *Let D be a diagonal PSD matrix and let $q(x) \propto (x^\top (I + D/n)x)^n$. Given an oracle for solving a univariate polynomial equation, we can sample from $q(x)$ in time $\text{poly}(n, d)$.*

As suggested above, we can solve the polynomial equation using binary search, obtaining an ε -accurate solution using $\text{poly} \log(\frac{1}{\varepsilon})$ evaluations of the polynomial.

Proof. Note the theorem is trivial for $d = 1$, as $q(x)$ is the uniform distribution on $\mathcal{S}^0 = \{-1, 1\}$. Hence we assume $d > 1$.

We will sample the coordinates one at a time (see Algorithm 1). For notational convenience, let us denote by x_{-i} the set of coordinates of a vector x excluding the i -th.

Namely, we will show that:

1. We can efficiently sample from the marginal distribution of x_1 , denoted by $q(x_1)$, via inverse transform sampling. To do this, we exhibit a $\text{poly}(n, d)$ algorithm for calculating the CDF of $q(x_1)$.

²This is a slight abuse of notation, and it denotes the marginal probability of the first coordinate. We do this to reduce clutter in the notation by subscripting the appropriate coordinate.

2. For any x_1 , the conditional distribution $q(x_{-1}|x_1)$ also has the form $q(x_{-1}|x_1) \propto (x_{-1}^\top(I + \tilde{D}/n)x_{-1})^n$, for some diagonal PSD matrix $\tilde{D} \in \mathbb{R}^{(d-1) \times (d-1)}$.

Applying this recursively gives our theorem.

Towards proving part 1, the marginal can be written using the co-area formula as

$$\begin{aligned} q(x_1) &= \frac{(1 - x_1^2)^{-(d-1)/2} \int_{\mathcal{S}^{d-2}} q(x_1, x_{-1}) d\mathcal{S}^{d-2}(x_{-1})}{Z} \\ &= \frac{(1 - x^2)^{-(d-1)/2} \int_{\mathcal{S}^{d-2}} \left((1 + D_{11}/n)x_1^2 + \sum_{i=2}^d (1 + D_{ii}/n)x_i^2 \right)^n d\mathcal{S}^{d-2}(x_{-1})}{Z}, \end{aligned}$$

where $Z = \int_{\mathcal{S}^{d-1}} (x^\top(I + D/n)x)^n$.

Introducing the change of variables $x_{-1} = y\sqrt{1 - x_1^2}$ where $y = (y_2, \dots, y_d) \in \mathcal{S}^{d-2}$, we can rewrite the numerator as

$$\int_{\mathcal{S}^{d-2}} \left((1 + D_{11}/n)x_1^2 + (1 - x_1^2) \sum_{i=2}^d (1 + D_{ii}/n)y_i^2 \right)^n d\mathcal{S}^{d-2}(y).$$

Hence, the CDF for $q(x_1)$ has the form

$$\frac{1}{Z} \int_{x=-1}^{x_1} \int_{y \in \mathcal{S}^{d-2}} \left((1 + D_{11}/n)x^2 + (1 - x^2) \sum_{i=2}^d (1 + D_{ii}/n)y_i^2 \right)^n d\mathcal{S}^{d-2}(y) dx \quad (5)$$

If we can evaluate this integral in time $\text{poly}(n, d)$, we can sample from $q(x_1)$ by using inverse transform sampling.

Expanding the term inside the inner integral, (5) can be rewritten as

$$\frac{1}{Z} \sum_{k=0}^n \binom{n}{k} \int_{x=-1}^{x_1} \left((1 + D_{11}/n)x^2 \right)^{n-k} (1 - x^2)^k \int_{y \in \mathcal{S}^{d-2}} (y^\top(I_{d-1} + D_{-1}/n)y)^k d\mathcal{S}^{d-2}(y) dx$$

where D_{-1} is obtained from D by deleting the first row and column. By Lemma 2.2, we can calculate each of the integrals $\int_{y \in \mathcal{S}^{d-2}} (y^\top(I_{d-1} + D_{-1}/n)y)^k$ in time $\text{poly}(n, d)$. Also by Lemma 2.2, Z can be calculated in $\text{poly}(n, d)$.

Hence, it remains to show we can approximate in time $\text{poly}(n, d)$ an integral of the type

$$\int_{x=-1}^{x_1} x^{2(n-k)}(1 - x^2)^k dx. \quad (6)$$

We can do this in polynomial time by expanding this as a polynomial and explicitly computing the integral.

Towards showing part 2, we compute the marginal distribution by using Bayes's theorem and making the change of variables $x_{-1} = y\sqrt{1 - x_1^2}$, $y = (y_2, \dots, y_d) \in \mathcal{S}^{d-2}$,

$$\begin{aligned} q(x_{-1}|x_1) &\propto q(x_1, x_{-1}) \\ &= \left((1 + D_{11}/n)x_1^2 + \sum_{i=2}^d (1 + D_{ii}/n)x_i^2 \right)^n \\ &= \left((1 + D_{11}/n)x_1^2 + \sum_{i=2}^d (1 - x_1^2)(1 + D_{ii}/n)y_i^2 \right)^n \\ &= (y^\top \left((x_1^2(1 + D_{11}/n) + (1 - x_1^2)) I_{d-1} + (1 - x_1^2)D_{-1}/n \right) y)^n. \end{aligned}$$

The last expression has the form $(y^\top(1 + \tilde{D}/n)y)^n$, for

$$\tilde{D} = x_1^2 D_{11} I_{d-1} + (1 - x_1^2) D_{-1}, \quad (7)$$

which is diagonal. Thus, we can apply the same sampling procedure recursively to \tilde{D} . \square

Proof of Theorem 1.1. As noted, we have reduced to the case of diagonal D with minimum eigenvalue $D_{\min} = 0$. Let $n = D_{\max}^2$. From Theorem 2.4 we can sample from the distribution $q(x) \propto (x^\top D x)^n$ in time $\text{poly}(D_{\max}, d)$. By Lemma 2.1, we have

$$\exp(-1/2) \leq \max\{p(x)/q(x), q(x)/p(x)\} \leq \exp(1/2).$$

We would like to do rejection sampling: accept the sample with probability $Cp(x)/q(x)$, where C is a constant $C \leq e^{-1/2}$ to ensure this is always ≤ 1 ; otherwise generate another sample. Averaged over x drawn from $q(x)$, the probability of acceptance is then C .

However, we don't have access to the normalized distribution $q(x)$. Instead, we have the unnormalized distributions $q^*(x) = (x^\top(I + D/n)x)^n$ and $p^*(x) = \exp(x^\top D x)$. We use the ratio at a particular point v to normalize them. Let v be the eigenvector with eigenvalue 0. We accept a proposal with probability

$$e^{-1} \frac{p^*(x)}{q^*(x)} = e^{-1} \frac{q^*(v)}{p^*(v)} \cdot \frac{p^*(x)}{q^*(x)} = e^{-1} \frac{q(v)}{p(v)} \cdot \frac{p(x)}{q(x)}$$

Using the inequality for v in Lemma 2.1, this fits the above framework with $C = e^{-1} \frac{q(v)}{p(v)} \in [e^{-1}, e^{-1/2}]$. This ensures the probability of acceptance is at least e^{-1} . \square

3 Conclusion

We presented a Las Vegas polynomial time algorithm for sampling from the Bingham distribution $p(x) \propto \exp(x^\top A x)$ on the unit sphere \mathcal{S}^{d-1} . The techniques are based on a novel polynomial approximation of the pdf which we believe is of independent interest, and should find other applications.

There are several natural open problems to pursue—perhaps the most natural one is how to generalize our techniques for the rank- k case. Can these polynomial expansion techniques be used to sample other probabilities of interest Bayesian machine learning, e.g., posterior distributions in latent-variable models such as Gaussian mixture models? More generally, for what other non-log-concave distributions of practical interest can we design provably efficient algorithms?

References

- [AT00] M Antone and Seth Teller. “Automatic recovery of camera positions in urban scenes”. In: *Technical Report, MIT LCS TR-814* (2000).
- [BÉ85] Dominique Bakry and Michel Émery. “Diffusions hypercontractives”. In: *Séminaire de Probabilités XIX 1983/84*. Springer, 1985, pp. 177–206.
- [Bin74] Christopher Bingham. “An antipodally symmetric distribution on the sphere”. In: *The Annals of Statistics* (1974), pp. 1201–1225.

- [CSS13] Kamalika Chaudhuri, Anand D Sarwate, and Kaushik Sinha. “A near-optimal algorithm for differentially-private principal components”. In: *The Journal of Machine Learning Research* 14.1 (2013), pp. 2905–2943.
- [Dal16] Arnak S Dalalyan. “Theoretical guarantees for approximate sampling from smooth and log-concave densities”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* (2016).
- [DM16] Alain Durmus and Eric Moulines. “High-dimensional Bayesian inference via the Unadjusted Langevin Algorithm”. In: (2016).
- [Dua+87] Simon Duane et al. “Hybrid monte carlo”. In: *Physics letters B* 195.2 (1987), pp. 216–222.
- [GP13] Jared Glover and Sanja Popovic. “Bingham procrustean alignment for object detection in clutter”. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2013, pp. 2158–2165.
- [Hsu02] Elton P Hsu. *Stochastic analysis on manifolds*. Vol. 38. American Mathematical Soc., 2002.
- [HW08] Tom SF Haines and Richard C Wilson. “Belief propagation with directional statistics for solving the shape-from-shading problem”. In: *European Conference on Computer Vision*. Springer. 2008, pp. 780–791.
- [Kan08] Raymond Kan. “From moments of sum to moments of product”. In: *Journal of Multivariate Analysis* 99.3 (2008), pp. 542–554.
- [KE06] Plamen Koev and Alan Edelman. “The efficient evaluation of the hypergeometric function of a matrix argument”. In: *Mathematics of Computation* 75.254 (2006), pp. 833–846.
- [KGM13] John T Kent, Asaad M Ganeiber, and Kanti V Mardia. “A new method to simulate the Bingham and related distributions in directional data analysis with applications”. In: *arXiv preprint arXiv:1310.8110* (2013).
- [MJ09] Kanti V Mardia and Peter E Jupp. *Directional statistics*. Vol. 494. John Wiley & Sons, 2009.
- [MR20] Ankur Moitra and Andrej Risteski. “Fast Convergence for Langevin Diffusion with Matrix Manifold Structure”. In: *arXiv preprint arXiv:2002.05576* (2020).
- [Ons80] Tullis C Onstott. “Application of the Bingham distribution function in paleomagnetic studies”. In: *Journal of Geophysical Research: Solid Earth* 85.B3 (1980), pp. 1500–1510.
- [Péc06] Sandrine Péché. “The largest eigenvalue of small rank perturbations of Hermitian random matrices”. In: *Probability Theory and Related Fields* 134.1 (2006), pp. 127–173.
- [Per+18] Amelia Perry et al. “Optimality and sub-optimality of PCA I: Spiked random matrix models”. In: *The Annals of Statistics* 46.5 (2018), pp. 2416–2451.
- [RDF78] Peter J Rossky, JD Doll, and HL Friedman. “Brownian dynamics as smart Monte Carlo simulation”. In: *The Journal of Chemical Physics* 69.10 (1978), pp. 4628–4633.
- [RT96] Gareth O Roberts and Richard L Tweedie. “Exponential convergence of Langevin distributions and their discrete approximations”. In: *Bernoulli* (1996), pp. 341–363.

[WWS15] Yining Wang, Yu-Xiang Wang, and Aarti Singh. “Differentially private subspace clustering”. In: *Advances in Neural Information Processing Systems*. 2015, pp. 1000–1008.

A Moment calculations

For completeness, we present the moment calculations that are used in the proof of our main theorem.

Lemma A.1. *Let A be symmetric PSD, and let $\lambda_1, \dots, \lambda_d$ be its eigenvalues. Then the moment generating function of $z^\top Az$, $z \sim N(0, I_d)$, is*

$$f(x) = \left(\prod_{i=1}^d \frac{1}{1 - 2\lambda_i x} \right)^{\frac{1}{2}}$$

Proof. Without loss of generality A is diagonal. Then $z^\top Az = \sum_{i=1}^d \lambda_i z_i^2$. The mgf of $r \sim \chi_d^2$ is $\left(\frac{1}{1-2x}\right)^{\frac{1}{2}}$. Now use the following two facts:

1. If the mgf of X is M_X , then the mgf of aX is $M_X(at)$: $M_{aX}(t) = M_X(at)$.
2. The mgf of a sum of random variables is the product of the mgfs: $M_{X+Y}(t) = M_X(t)M_Y(t)$.

□

Corollary A.2 ([Kan08]). *Let A be symmetric PSD. Let $S(n) = \frac{1}{n!2^n} \mathbb{E}_{x \sim N(0, I_d)} (x^\top Ax)^n$. Then $S(0) = 1$ and for $n \geq 1$,*

$$S(n) = \frac{1}{2n} \sum_{i=1}^n \text{Tr}(A^i) S(n-i). \quad (8)$$

This can be calculated in polynomial time by dynamic programming.

Proof. Note $\text{Tr}(A^k) = \sum_{i=1}^d \lambda_i^k$. The moment generating function in Lemma A.1 satisfies the differential equation

$$f'(x) = \sum_{i=1}^d \frac{\lambda_i}{1 - 2\lambda_i x} f(x).$$

Matching the coefficient of x^{n-1} gives the equation. □

Corollary A.3. *For $n \geq 0$,*

$$\mathbb{E}_{x \sim N(0, I_d)} [\|x\|^{2n}] = \prod_{i=0}^{n-1} (d + 2i).$$

For $d = 1$, this agrees with the formula $\mathbb{E}_{x \sim N(0, 1)} [x^{2n}] = (2n - 1)!!$.

Proof. By Lemma A.1, the moment generating function of $\|x\|^2$ is $(1 - 2x)^{-\frac{d}{2}}$. Use the binomial series expansion. □