Large-scale Log-determinant Computation through Stochastic Chebyshev Expansions

Insu Han HAWKI17@KAIST.AC.KR
Department of Electrical Engineering, Korea Advanced Institute of Science and Technology, Korea

Dmitry Malioutov DMALIOUTOV@US.IBM.COM
Business Analytics and Mathematical Sciences, IBM Research, Yorktown Heights, NY, USA

Jinwoo Shin JINWOOS@KAIST.AC.KR
Department of Electrical Engineering, Korea Advanced Institute of Science and Technology, Korea

Abstract

Logarithms of determinants of large positive definite matrices appear ubiquitously in machine learning applications including Gaussian graphical and Gaussian process models, partition functions of discrete graphical models, minimum-volume ellipsoids, metric learning and kernel learning. Log-determinant computation involves the Cholesky decomposition at the cost cubic in the number of variables, i.e., the matrix dimension, which makes it prohibitive for large-scale applications. We propose a linear-time randomized algorithm to approximate log-determinants for very large-scale positive definite and general non-singular matrices using a stochastic trace approximation, called the Hutchinson method, coupled with Chebyshev polynomial expansions that both rely on efficient matrix-vector multiplications. We establish rigorous additive and multiplicative approximation error bounds depending on the condition number of the input matrix. In our experiments, the proposed algorithm can provide very high accuracy solutions at orders of magnitude faster time than the Cholesky decomposition and Schur completion, and enables us to compute log-determinants of matrices involving tens of millions of variables.

1. Introduction

Scalability of machine learning algorithms for extremely large data-sets and models has been increasingly the focus of attention for the machine learning community, with prominent examples such as first-order stochastic optimization methods and randomized linear algebraic computations. One of the important tasks from linear algebra that appears in a variety of machine learning problems is computing the log-determinant of a large positive definite matrix. For example, serving as the normalization constant for multivariate Gaussian models, log-determinants of covariance (and precision) matrices play an important role in inference, model selection and learning both the structure and the parameters for Gaussian graphical models and Gaussian processes (Rue & Held, 2005; Rasmussen & Williams, 2005; Dempster, 1972). Log-determinants also play an important role in a variety of Bayesian machine learning problems, including sampling and variational inference (MacKay, 2003). In addition, metric and kernel learning problems attempt to learn quadratic forms adapted to the data, and formulations involving Bregman divergences of log-determinants have become very popular (Davis et al., 2007; Van Aelst & Rousseeuw, 2009). Finally, log-determinant computation also appears in some discrete probabilistic models, e.g., tree mixture models (Meila & Jordan, 2001; Anandkumar et al., 2012) and Markov random fields (Wainwright & Jordan, 2006). In planar Markov random fields (Schraudolph & Kamenetsky, 2009; Johnson et al., 2010) inference and learning involve log-determinants of general non-singular matrices.

For a positive semi-definite matrix $B \in \mathbb{R}^{d\times d}$, numerical linear algebra experts recommend to compute log-determinant using the Cholesky decomposition. Suppose the Cholesky decomposition is $B = LL^T$, then $\log\det(B) = 2\sum_i \log L_{ii}$. The computational complexity of Cholesky decomposition is cubic with respect to the number of variables, i.e., $O(d^3)$, in general. For large-scale applications involving more than tens of thousands of variables, this operation is not feasible. Our aim is to com-

Proceedings of the 32^{nd} International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37.

pute accurate approximate log-determinants for matrices of much larger size involving *tens of millions* of variables.

Contribution. Our approach to compute accurate approximations of log-determinant for a positive definite matrix uses a combination of stochastic trace-estimators and Chebyshev polynomial expansions. Using the Chebyshev polynomials, we first approximate the log-determinant by the trace of power series of the input matrix. We then use a stochastic trace-estimator, called the *Hutchison method* (Hutchinson, 1989), to estimate the trace using multiplications between the input matrix and random vectors. The main assumption for our method is that the matrix-vector product can be computed efficiently. For example, the time-complexity of the proposed algorithm grows linearly with respect to the number of non-zero entries in the input matrix. We also extend our approach to general non-singular matrices to compute the absolute values of their log-determinants. We establish rigorous additive and multiplicative approximation error bounds for approximating the log-determinant under the proposed algorithm. Our theoretical results provide an analytic understanding on our Chebyshev-Hutchison method depending on sampling number, polynomial degree and the condition number (i.e., the ratio between the largest and smallest singular values) of the input matrix. In particular, they imply that if the condition number is $O(1)$, then the algorithm provides ε-approximation guarantee (in multiplicative or additive) in linear time for any constant $\varepsilon > 0$.

We first apply our algorithm to obtain a randomized linear-time approximation scheme for counting the number of spanning trees in a certain class of graphs where it could be used for efficient inference in tree mixture models (Meila & Jordan, 2001; Anandkumar et al., 2012). We also apply our algorithm for finding maximum likelihood parameter estimates of Gaussian Markov random fields of size 5000×5000 (involving 25 million variables!), which is infeasible for the Cholesky decomposition. Our experiments show that our proposed algorithm is orders of magnitude faster than the Cholesky decomposition and Schur completion for sparse matrices and provides solutions with 99.9% accuracy in approximation. It can also solve problems of dimension tens of millions in a few minutes on our single commodity computer. Furthermore, the proposed algorithm is very easy to parallelize and hence has a potential to handle even a bigger size. In particular, the Schur method was used as a part of QUIC algorithm (Hsieh et al., 2013) for sparse inverse covariance estimation with over million variables, hence our algorithm could be used to further improve its speed and scale.

Related work. Stochastic trace estimators have been studied in the literature in a number of applications. (Bekas et al., 2007; Malioutov et al., 2006) have used a stochastic trace estimator to compute the diagonal of a matrix or of matrix inverse. Polynomial approximations to band-pass filters have been used to count the number of eigenvalues in certain intervals (Di Napoli et al., 2013). Stochastic approximations of score equations have been applied in (Stein et al., 2013) to learn large-scale Gaussian processes. The works closest to ours which have used stochastic trace estimators for Gaussian process parameter learning are (Zhang & Leithead, 2007) and (Aune et al., 2014) which instead use Taylor expansions and Cauchy integral formula, respectively. A recent improved analysis using Taylor expansions has also appeared in (Boutsidis et al., 2015). However, as reported in Section 5, our method using Chebyshev expansions provides much better accuracy in experiments than that using Taylor expansions, and (Aune et al., 2014) need Krylov-subspace linear system solver that is computationally expensive in general. (Pace & LeSage, 2004) also use Chebyshev polynomials for log-determinant computation, but the method is deterministic and only applicable to polynomials of small degree. The novelty of our work is combining the Chebyshev approximation with Hutchison trace estimators, which allows to design a linear-time algorithm with approximation guarantees.

2. Background

In this section, we describe the preliminaries for our approach to approximate the log-determinant of a *positive definite* matrix. Our approach combines the following two techniques: (a) designing a trace-estimator for the log-determinant of positive definite matrix via Chebyshev approximation (Mason & Handscomb, 2002) and (b) approximating the trace of positive definite matrix via Monte Carlo methods, e.g., Hutchison method (Hutchinson, 1989).

2.1. Chebyshev Approximation

The Chebyshev approximation technique is used to approximate analytic function with certain orthonormal polynomials. We use $p_n(x)$ to denote the Chebyshev approximation of degree n for a given function $f : [-1, 1] \to \mathbb{R}$:

$$f(x) \approx p_n(x) = \sum_{j=0}^{n} c_j T_j(x),$$

where the coefficient c_i and the i-th Chebyshev polynomial $T_i(x)$ are defined as

$$c_i = \begin{cases} \dfrac{1}{n+1} \displaystyle\sum_{k=0}^{n} f(x_k)\, T_0(x_k) & \text{if } i = 0 \\ \dfrac{2}{n+1} \displaystyle\sum_{k=0}^{n} f(x_k)\, T_i(x_k) & \text{otherwise} \end{cases} \tag{1}$$

$$T_{i+1}(x) = 2xT_i(x) - T_{i-1}(x) \qquad \text{for } i \geq 1 \tag{2}$$

where $x_k = \cos\left(\frac{\pi(k+1/2)}{n+1}\right)$ for $k = 0, 1, 2, \ldots n$ and $T_0(x) = 1, T_1(x) = x$ (Mason & Handscomb, 2002).

Chebyshev approximation for scalar functions can be naturally generalized to matrix functions. Using the Chebyshev approximation $p_n(x)$ for function $f(x) = \log(1-x)$ we obtain the following approximation to the log-determinant of a positive definite matrix $B \in \mathbb{R}^{d\times d}$:

$$\begin{aligned}\log\det B = \log\det(I-A) &= \sum_{i=1}^{d}\log(1-\lambda_i)\\ &\approx \sum_{i=1}^{d} p_n(\lambda_i) = \sum_{i=1}^{d}\sum_{j=0}^{n} c_j T_j(\lambda_i)\\ &= \sum_{j=0}^{n} c_j \sum_{i=1}^{d} T_j(\lambda_i) = \sum_{j=0}^{n} c_j \mathtt{tr}\left(T_j(A)\right),\end{aligned}$$

where $A = I - B$ has eigenvalues $0 \le \lambda_1, \ldots, \lambda_d \le 1$ and the last equality is from the fact that $\sum_{i=1}^{d} p(\lambda_i) = \mathtt{tr}(p(A))$ for any polynomial $p(\cdot)$.[1] We remark that other polynomial approximations, e.g., Taylor, can also be used to approximate log-determinants. We focus on the Chebyshev approximation, where Chevyshev approximation is known to be an optimal polynomial interpolation that minimize the ℓ_∞-error (de De Villiers, 2012).

2.2. Trace Approximation via Monte-Carlo Method

The main challenge to compute the log-determinant of a positive definite matrix in the previous section is calculating the trace of $T_j(A)$ efficiently without evaluating the entire matrix A^k. We consider a Monte-Carlo approach for estimating the trace of a matrix. First, a random vector $\mathbf{z}$ is drawn from some fixed distribution, such that the expectation of $\mathbf{z}^\top A\mathbf{z}$ is equal to the trace of A. By sampling m such i.i.d. random vectors, and averaging we obtain an estimate of $\mathtt{tr}(A)$.

It is known that the Hutchinson method, where components of the random vectors Z are i.i.d. Rademacher random variables, i.e., $\Pr(+1) = \Pr(-1) = \frac{1}{2}$, has the smallest variance among such Monte-Carlo methods (Hutchinson, 1989; Avron & Toledo, 2011). It has been used extensively in many applications (Avron, 2010; Hutchinson, 1989; Aravkin et al., 2012). Formally, the Hutchinson trace estimator $\mathtt{tr}_m(A)$ is known to satisfy the following:

$$\mathbf{E}\left[\mathtt{tr}_m(A) := \frac{1}{m}\sum_{i=1}^{m}\mathbf{z}_i^\top A\mathbf{z}_i\right] = \mathtt{tr}(A)$$

$$\mathbf{Var}\left[\mathtt{tr}_m(A)\right] = 2\left(\|A\|_F^2 - \sum_{i=1}^{n} A_{ii}^2\right).$$

[1] $\mathtt{tr}(\cdot)$ denotes the trace of a matrix.

Note that computing $\mathbf{z}^\top A\mathbf{z}$ requires only multiplications between a matrix and a vector, which is particularly appealing when evaluating A itself is expensive, e.g., $A = B^k$ for some matrix B and large k. Furthermore, given any matrix X, one can compute $\mathbf{z}^\top T_j(X)\,\mathbf{z}$ more efficiently using the following recursion on the vector $w_j = T_j(X)\mathbf{z}$:

$$w_{j+1} = 2Xw_j - w_{j-1},$$

which follows directly from (2).

3. Log-determinant Approximation Scheme

Now we are ready to present algorithms to approximate the absolute value of log-determinant of an arbitrary non-singular square matrix C. Without loss of generality, we assume that singular values of C are in the interval $[\sigma_{\min}, \sigma_{\max}]$ for some $\sigma_{\min}, \sigma_{\max} > 0$, i.e., the condition number $\kappa(C)$ is at most $\kappa_{\max} := \sigma_{\max}/\sigma_{\min}$. The proposed algorithms are not sensitive to tight knowledge of $\sigma_{\min}, \sigma_{\max}$, but some loose lower and upper bounds on them, respectively, suffice.

3.1. Algorithm for Positive Definite Matrices

In this section, we describe our proposed algorithm for estimating the log-determinant of a positive definite matrix whose eigenvalues are less than one, i.e., $\sigma_{\max} < 1$. It is used as a subroutine for estimating the log-determinant of a general non-singular matrix in the next section.

Algorithm 1 Log-determinant approximation for positive definite matrices with $\sigma_{\max} < 1$

Input: positive definite matrix $B \in \mathbb{R}^{d\times d}$ with eigenvalues in $[\delta\ ,\ 1-\delta]$ for some $\delta > 0$, sampling number m and polynomial degree n
Initialize: $A \leftarrow I - B$, $\Gamma \leftarrow 0$
for $i = 0$ **to** n **do**
  $c_i \leftarrow i$-th coefficient of Chebyshev approximation for $\log(1 - \frac{(1-2\delta)x+1}{2})$
end for
for $i = 1$ **to** m **do**
  Draw a Rademacher random vector $\mathbf{v}$ and $\mathbf{u} \leftarrow c_0\,\mathbf{v}$
  if $n > 1$ **then**
    $\mathbf{w}_0 \leftarrow \mathbf{v}$ and $\mathbf{w}_1 \leftarrow A\mathbf{v}$
    $\mathbf{u} \leftarrow \mathbf{u} + c_1 A\mathbf{v}$
    for $j = 2$ **to** n **do**
      $\mathbf{w}_2 \leftarrow 2A\mathbf{w}_1 - \mathbf{w}_0$
      $\mathbf{u} \leftarrow \mathbf{u} + c_j\,\mathbf{w}_2$
      $\mathbf{w}_0 \leftarrow \mathbf{w}_1$ and $\mathbf{w}_1 \leftarrow \mathbf{w}_2$
    end for
  end if
  $\Gamma \leftarrow \Gamma + \mathbf{v}^\top\mathbf{u}/m$
end for
Output: Γ

We establish the following theoretical guarantee of the above algorithm, where its proof is given in Section 4.3.

Theorem 1 *Given $\varepsilon, \zeta \in (0,1)$, consider the following inputs for* **Algorithm 1***:*

- $B \in \mathbb{R}^{d\times d}$ *be a positive definite matrix with eigenvalues in* $[\delta, 1-\delta]$ *for some* $\delta \in (0, 1/2)$
- $m \geq 54\varepsilon^{-2}\log\left(\frac{2}{\zeta}\right)$
- $n \geq \frac{\log\left(\frac{20}{\varepsilon}\left(\sqrt{\frac{2}{\delta}-1}-1\right)\frac{\log(2/\delta)}{\log(1/(1-\delta))}\right)}{\log\left(\frac{\sqrt{2-\delta}+\sqrt{\delta}}{\sqrt{2-\delta}-\sqrt{\delta}}\right)}$
$= O\left(\sqrt{\frac{1}{\delta}}\log\left(\frac{1}{\varepsilon\delta}\right)\right)$

Then, it follows that

$$\Pr\left[\ |\log\det B - \Gamma| \leq \varepsilon\,|\log\det B|\ \right] \geq 1-\zeta$$

where Γ *is the output of* **Algorithm 1**.

The bound on polynomial degree n in the above theorem is relatively tight, e.g., it implies to choose $n = 14$ for $\delta = 0.1$ and $\varepsilon = 0.01$. While our bound on sampling number m is not tight, we observe that $m \approx 30$ is sufficient for high accuracy in our experiments. We also remark that the time-complexity of **Algorithm 1** is $O(mn\|B\|_0)$, where $\|B\|_0$ is the number of non-zero entries of B. This is because the algorithm requires only multiplications of matrices and vectors. In particular, if $m, n = O(1)$, the complexity is linear with respect to the input size. Therefore, Theorem 1 implies that one can choose $m, n = O(1)$ for ε-multiplicative approximation with probability $1-\zeta$ given constants $\varepsilon, \zeta > 0$.

3.2. Algorithm for General Non-Singular Matrices

Now, we are ready to present our linear-time approximation scheme for the log-determinant of general non-singular matrix C, through generalizing the algorithm in the previous section. The idea is simple: run **Algorithm 1** with normalization of positive definite matrix C^TC. This is formally described in what follows.

Algorithm 2 Log-determinant approximation for general non-singular matrices

Input: matrix $C \in \mathbb{R}^{d\times d}$ with singular values are in the interval $[\sigma_{\min}, \sigma_{\max}]$ for some $\sigma_{\min}, \sigma_{\max} > 0$, sampling number m and polynomial degree n

Initialize: $B \leftarrow \frac{1}{\sigma_{\min}^2+\sigma_{\max}^2}C^TC$, $\delta \leftarrow \frac{\sigma_{\min}^2}{\sigma_{\min}^2+\sigma_{\max}^2}$

$\Gamma \leftarrow$ Output of **Algorithm 1** for inputs B, m, n, δ

Output: $\Gamma \leftarrow \left(\Gamma + d\log\left(\sigma_{\min}^2+\sigma_{\max}^2\right)\right)/2$

Algorithm 2 is motivated to design from the equality $\log|\det C| = \frac{1}{2}\log\det C^TC$. Given non-singular matrix C, one need to choose appropriate $\sigma_{\max}, \sigma_{\min}$. In most applications, $\sigma_{\max}$ is easy to choose, e.g., one can choose

$$\sigma_{\max} = \sqrt{\|C\|_1\|C\|_\infty},$$

or one can run the power iteration (Ipsen, 1997) to estimate a better bound. On the other hand, $\sigma_{\min}$ is generally not easy to obtain, except for special cases. It is easy to obtain in the problem of counting spanning trees we studied in Section 3.3, and it is explicitly given as a parameter in many machine learning log-determinant applications (Wainwright & Jordan, 2006). In general, one can use the inverse power iteration (Ipsen, 1997) to estimate it. Furthermore, the smallest singular value is easy to compute for random matrices (Tao & Vu, 2009; 2010) and diagonal-dominant matrices (Gershgorin, 1931; Morača, 2008).

The time-complexity of **Algorithm 2** is still $O(mn\|C\|_0)$ instead of $O(mn\|C^TC\|_0)$ since **Algorithm 1** requires multiplication of matrix C^TC and vectors. We state the following additive error bound of the above algorithm.

Theorem 2 *Given $\varepsilon, \zeta \in (0,1)$, consider the following inputs for* **Algorithm 2***:*

- $C \in \mathbb{R}^{d\times d}$ *be a matrix having singular values in the interval* $[\sigma_{\min}, \sigma_{\max}]$ *for some* $\sigma_{\min}, \sigma_{\max} > 0$
- $m \geq \mathcal{M}\left(\varepsilon, \frac{\sigma_{\max}}{\sigma_{\min}}, \zeta\right)$ *and* $n \geq \mathcal{N}\left(\varepsilon, \frac{\sigma_{\max}}{\sigma_{\min}}\right)$*, where*

$$\mathcal{M}(\varepsilon,\kappa,\zeta) := \frac{14}{\varepsilon^2}\left(\log\left(1+\kappa^2\right)\right)^2\log\left(\frac{2}{\zeta}\right)$$

$$\mathcal{N}(\varepsilon,\kappa) := \frac{\log\left(\frac{20}{\varepsilon}\left(\sqrt{2\kappa^2+1}-1\right)\frac{\log(2+2\kappa^2)}{\log(1+\kappa^{-2})}\right)}{\log\left(\frac{\sqrt{2\kappa^2+1}+1}{\sqrt{2\kappa^2+1}-1}\right)} = O\left(\kappa\log\frac{\kappa}{\varepsilon}\right)$$

Then, it follows that

$$\Pr\left[\ |\log\left(|\det C|\right) - \Gamma| \leq \varepsilon d\ \right] \geq 1-\zeta$$

where Γ *is the output of* **Algorithm 2**.

Proof. The proof of Theorem 2 is quite straightforward using Theorem 1 for B with the facts that

$$2\log|\det C| = \log\det B + d\log\left(\sigma_{\min}^2+\sigma_{\max}^2\right)$$

and $|\log\det B| \leq d\log\left(1+\frac{\sigma_{\max}^2}{\sigma_{\min}^2}\right)$. ∎

We remark that the condition number $\sigma_{\max}/\sigma_{\min}$ decides the complexity of **Algorithm 2**. As one can expect, the approximation quality and algorithm complexity become

worse for matrices with very large condition numbers, as the Chebyshev approximation for the function $\log x$ near the point 0 is more challenging and requires higher degree approximations.

When $\sigma_{\max} \geq 1$ and $\sigma_{\min} \leq 1$, i.e., we have mixed signs for logs of the singular values, a multiplicative error bound (as stated in Theorem 1) can not be obtained since the log-determinant can be zero in the worst case. On the other hand, when $\sigma_{\max} < 1$ or $\sigma_{\min} > 1$, we further show that the above algorithm achieves an ε-multiplicative approximation guarantee, as stated in the following corollaries.

Corollary 3 *Given $\varepsilon, \zeta \in (0,1)$, consider the following inputs for* **Algorithm 2***:*

- $C \in \mathbb{R}^{d \times d}$ *be a matrix having singular values in the interval* $[\sigma_{\min}, \sigma_{\max}]$ *for some* $\sigma_{\max} < 1$
- $m \geq \mathcal{M}\left(\varepsilon \log \frac{1}{\sigma_{\max}}, \frac{\sigma_{\max}}{\sigma_{\min}}, \zeta\right)$
- $n \geq \mathcal{N}\left(\varepsilon \log \frac{1}{\sigma_{\max}}, \frac{\sigma_{\max}}{\sigma_{\min}}\right)$

Then, it follows that

$$\Pr\left[\, |\log|\det C| - \Gamma| \leq \varepsilon \, |\log|\det C|| \right] \geq 1 - \zeta$$

where Γ *is the output of* **Algorithm 2***.*

Corollary 4 *Given $\varepsilon, \zeta \in (0,1)$, consider the following inputs for* **Algorithm 2***:*

- $C \in \mathbb{R}^{d \times d}$ *be a matrix having singular values in the interval* $[\sigma_{\min}, \sigma_{\max}]$ *for some* $\sigma_{\min} > 1$
- $m \geq \mathcal{M}\left(\varepsilon \log \sigma_{\min}, \frac{\sigma_{\max}}{\sigma_{\min}}, \zeta\right)$
- $n \geq \mathcal{N}\left(\varepsilon \log \sigma_{\min}, \frac{\sigma_{\max}}{\sigma_{\min}}\right)$

Then, it follows that

$$\Pr\left[\, |\log\det C - \Gamma| \leq \varepsilon \log\det C \right] \geq 1 - \zeta$$

where Γ *is the output of* **Algorithm 2***.*

The proofs of the above corollaries are given in the supplementary material due to the space limitation.

3.3. Application to Counting Spanning Trees

We apply **Algorithm 2** to a concrete problem, where we study counting the number of spanning trees in a simple undirected graph $G = (V, E)$ where there exists a vertex i^* such that $(i^*, j) \in E$ for all $j \in V \setminus \{i^*\}$. Counting spanning trees is one of classical well-studied counting problems, and also necessary in machine learning applications, e.g., tree mixture models (Meila & Jordan, 2001; Anandkumar et al., 2012). We denote the maximum and average degrees of vertices in $V \setminus \{i^*\}$ by $\Delta_{\mathtt{max}}$ and $\Delta_{\mathtt{avg}} > 1$, respectively. In addition, we let $L(G)$ denote the Laplacian matrix of G. Then, from Kirchhoff's matrix-tree theorem, the number of spanning tree $\tau(G)$ is equal to $\tau(G) = \det L(i^*)$, where $L(i^*)$ is the $(|V|-1) \times (|V|-1)$ submatrix of $L(G)$ that is obtained by eliminating the row and column corresponding to i^* (Kocay & Kreher, 2004). Now, it is easy to check that eigenvalues of $L(i^*)$ are in $[1, 2\Delta_{\mathtt{max}} - 1]$. Under these observations, we derive the following corollary.

Corollary 5 *Given $0 < \varepsilon < \frac{2}{\Delta_{\mathtt{avg}} - 1}, \zeta \in (0,1)$, consider the following inputs for* **Algorithm 2***:*

- $C = L(i^*)$
- $m \geq \mathcal{M}\left(\frac{\varepsilon(\Delta_{\mathtt{avg}}-1)}{4}, 2\Delta_{\mathtt{max}} - 1, \zeta\right)$
- $n \geq \mathcal{N}\left(\frac{\varepsilon(\Delta_{\mathtt{avg}}-1)}{4}, 2\Delta_{\mathtt{max}} - 1\right)$

Then, it follows that

$$\Pr\left[|\log\tau(G) - \Gamma| \leq \varepsilon \log\tau(G)\right] \geq 1 - \zeta$$

where Γ *is the output of* **Algorithm 2***.*

The proof of the above corollary is given in the supplementary material due to the space limitation. We remark that the running time of **Algorithm 2** with inputs in the above theorem is $O(nm\Delta_{\mathtt{avg}}|V|)$. Therefore, for $\varepsilon, \zeta = \Omega(1)$ and $\Delta_{\mathtt{avg}} = O(1)$, i.e., G is sparse, one can choose $n, m = O(1)$ so that the running time of **Algorithm 2** is $O(|V|)$.

4. Proof of Theorem 1

In order to prove Theorem 1, we first introduce some necessary background and lemmas on error bounds of Chebyshev approximation and Hutchinson method we introduced in Section 2.1 and Section 2.2, respectively.

4.1. Convergence Rate for Chebyshev Approximation

Intuitively, one can expect that the approximated Chebyshev polynomial converges to its original function as degree n goes to ∞. Formally, the following error bound is known (Berrut & Trefethen, 2004; Xiang et al., 2010).

Theorem 6 *Suppose f is analytic with $|f(z)| \leq M$ in the region bounded by the ellipse with foci ± 1 and major and minor semiaxis lengths summing to $K > 1$. Let p_n denote the interpolant of f of degree n in th Chebyshev points as*

defined in section 2.1, then for each $n \geq 0$,

$$\max_{x\in[-1,1]} |f(x) - p_n(x)| \leq \frac{4M}{(K-1)\,K^n}$$

To prove Theorem 1 and Theorem 2, we are in particular interested in $f(x) = \log(1-x)$, for $x \in [\delta, 1-\delta]$. Since Chebyshev approximation is defined in the interval $[-1,1]$, e.g., see Section 2.1, one can use the following linear mapping $g : [-1,1] \to [\delta, 1-\delta]$ so that

$$\max_{x\in[-1,1]} |(f\circ g)(x) - p_n(x)| \;=\; \max_{x\in[\delta,1-\delta]} |f(x) - \widetilde{p}_n(x)|\,,$$

where $\widetilde{p}_n(x) = \left(p_n \circ g^{-1}\right)(x)$.

We choose the ellipse region, denoted by $\mathcal{E}_K$, in the complex plane with foci at ± 1 and its semimajor axis length is $1/(1-\delta)$ where $f \circ g$ is analytic on and inside. The length of semimajor axis of the ellipse is equal to $\sqrt{(1/(1-\delta))^2 - 1}$. Hence, the convergence rate K can be set to

$$K = \frac{1}{1-\delta} + \sqrt{\left(\frac{1}{1-\delta}\right)^2 - 1} = \frac{\sqrt{2-\delta}+\sqrt{\delta}}{\sqrt{2-\delta}-\sqrt{\delta}} > 1$$

The constant M can be also obtained as follows:

$$\begin{aligned}
\max_{z\in\mathcal{E}_K} |(f\circ g)(z)| &= \max_{z\in\mathcal{E}_K} |\log(1-g(z))| \\
&\leq \max_{z\in\mathcal{E}_K} \sqrt{(\log|1-g(z)|)^2 + \pi^2} \\
&= \sqrt{\left(\log\left|1 - g\left(-\frac{1}{1-\delta}\right)\right|\right)^2 + \pi^2} \\
&\leq 5\log\left(\frac{2}{\delta}\right) := M.
\end{aligned}$$

where the inequality in the second line holds because $|\log z| = |\log|z| + i\arg(z)| \leq \sqrt{(\log|z|)^2 + \pi^2}$ for any $z \in \mathbb{C}$ and equality in the third line holds by the maximum-modulus theorem.

Hence, for $x \in [\delta, 1-\delta]$,

$$|\log(1-x) - \widetilde{p}_n(x)| \leq \frac{20\log(2/\delta)}{(K-1)\,K^n}.$$

Under these observations, we establish the following lemma that is a "matrix version" of Theorem 6.

Lemma 7 *Let $B \in \mathbb{R}^{d\times d}$ be a positive definite matrix whose eigenvalues are in $[\delta, 1-\delta]$ for $\delta \in (0, 1/2)$. Then, it holds that*

$$\left|\log\det B - \mathtt{tr}\big(\widetilde{p}_n(I-B)\big)\right| \leq \frac{20d\log(2/\delta)}{(K-1)\,K^n}$$

where $K = \frac{\sqrt{2-\delta}+\sqrt{\delta}}{\sqrt{2-\delta}-\sqrt{\delta}}$.

Proof. Let $\lambda_1, \lambda_2, \cdots, \lambda_d \in [\delta, 1-\delta]$ be eigenvalues of matrix $A = I - B$. Then, we have

$$\begin{aligned}
&|\log\det(I-A) - \mathtt{tr}\left(\widetilde{p}_n(A)\right)| \\
&= \left|\mathtt{tr}\left(\log(I-A)\right) - \mathtt{tr}\left(\widetilde{p}_n(A)\right)\right| \\
&= \left|\sum_{i=1}^d \log(1-\lambda_i) - \sum_{i=1}^d \widetilde{p}_n(\lambda_i)\right| \\
&\leq \sum_{i=1}^d |\log(1-\lambda_i) - \widetilde{p}_n(\lambda_i)| \\
&\leq \sum_{i=1}^d \frac{20\log(2/\delta)}{(K-1)\,K^n} = \frac{20d\log(2/\delta)}{(K-1)\,K^n}
\end{aligned}$$

where we use Theorem 6. This completes the proof of Lemma 7. ∎

4.2. Approximation Error of Hutchinson Method

In this section, we use the same notation, e.g., f, p_n, used in the previous section and we analyze the Hutchinson's trace estimator $\mathtt{tr}_m(\cdot)$ defined in Section 2.2. To begin with, we state the following theorem that is proven in (Roosta-Khorasani & Ascher, 2013).

Theorem 8 *Let $A \in \mathbb{R}^{d\times d}$ be a positive definite or negative definite matrix. Given $\varepsilon_0, \zeta_0 \in (0,1)$,*

$$\Pr\left[|\mathtt{tr}_m(A) - \mathtt{tr}(A)| \leq \varepsilon_0\,\mathtt{tr}(A)\right] \geq 1 - \zeta_0$$

holds if sampling number m is larger than $6\varepsilon_0^{-2}\log\left(\frac{2}{\zeta_0}\right)$.

The theorem above provides a lower-bound on the sampling complexity of Hutchinson method, which is independent of a given matrix A. To prove Theorem 1, we need an error bound on $\mathtt{tr}_m(\widetilde{p}_n(A))$. However, in general we may not know whether or not $\widetilde{p}_n(A)$ is positive definite or negative definite. We can guarantee that the eigenvalues of $\widetilde{p}_n(A)$ will be negative using the following lemma.

Lemma 9 *$\widetilde{p}_n(x)$ is a negative-valued polynomial in the interval $[\delta, 1-\delta]$ if*

$$\frac{20\log(2/\delta)}{(K-1)\,K^n} \leq \log\left(\frac{1}{1-\delta}\right)$$

where we recall that $K = \frac{\sqrt{2-\delta}+\sqrt{\delta}}{\sqrt{2-\delta}-\sqrt{\delta}}$.

Proof. From Theorem 6, we have

$$\begin{aligned}
\max_{[\delta,1-\delta]} \widetilde{p}_n(x) &= \max_{[\delta,1-\delta]} f(x) + (\widetilde{p}_n(x) - f(x)) \\
&\leq \max_{[\delta,1-\delta]} f(x) + \max_{[\delta,1-\delta]} |\widetilde{p}_n(x) - f(x)| \\
&\leq \log(1-\delta) + \frac{20\log(2/\delta)}{(K-1)\,K^n} \leq 0,
\end{aligned}$$

where we use $\frac{20\log(2/\delta)}{(K-1)K^n} \leq -\log(1-\delta)$. This completes the proof of Lemma 9. ■

4.3. Proof of the Theorem 1

Now we are ready to prove Theorem 1. First, one can check that sampling number n in the condition of Theorem 1 satisfies

$$\frac{20\log(2/\delta)}{(K-1)K^n} \leq \frac{\varepsilon}{2}\log\left(\frac{1}{1-\delta}\right). \quad (3)$$

Hence, from Lemma 9, it follows that $\widetilde{p}_n(A)$ is negative definite where $A = I - B$ and eigenvalues of B are in $[\delta, 1-\delta]$. Hence, we can apply Theorem 8 as

$$\Pr\left[\left|\mathtt{tr}\left(\widetilde{p}_n(A)\right) - \mathtt{tr}_m\left(\widetilde{p}_n(A)\right)\right| \leq \frac{\varepsilon}{3}\left|\mathtt{tr}\left(\widetilde{p}_n(A)\right)\right|\right] \geq 1-\zeta, \quad (4)$$

for $m \geq 54\varepsilon^{-2}\log\left(\frac{2}{\zeta}\right)$. In addition, from Theorem 7, we have

$$\left|\mathtt{tr}\left(\widetilde{p}_n(A)\right)\right| - \left|\log\det B\right| \leq \left|\log\det B - \mathtt{tr}\left(\widetilde{p}_n(A)\right)\right| \leq \frac{20d\log(2/\delta)}{(K-1)K^n} \leq \frac{\varepsilon}{2}d\log\left(\frac{1}{1-\delta}\right) \leq \frac{\varepsilon}{2}\left|\log\det B\right|,$$

which implies that

$$\left|\mathtt{tr}\left(\widetilde{p}_n(A)\right)\right| \leq \left(\frac{\varepsilon}{2}+1\right)\left|\log\det B\right| \leq \frac{3}{2}\left|\log\det B\right|.$$

Combining the above inequality with (3) and (4) leads to the conclusion of Theorem 1 as follows:

$$\begin{aligned}
&1-\zeta \\
&\leq \Pr\left[\left|\mathtt{tr}\left(\widetilde{p}_n(A)\right) - \mathtt{tr}_m\left(\widetilde{p}_n(A)\right)\right| \leq \frac{\varepsilon}{3}\left|\mathtt{tr}\left(\widetilde{p}_n(A)\right)\right|\right] \\
&\leq \Pr\left[\left|\mathtt{tr}\left(\widetilde{p}_n(A)\right) - \mathtt{tr}_m\left(\widetilde{p}_n(A)\right)\right| \leq \frac{\varepsilon}{2}\left|\log\det B\right|\right] \\
&\leq \Pr[\left|\mathtt{tr}\left(\widetilde{p}_n(A)\right) - \mathtt{tr}_m\left(\widetilde{p}_n(A)\right)\right| \\
&\qquad + \left|\log\det B - \mathtt{tr}\left(\widetilde{p}_n(A)\right)\right| \\
&\qquad \leq \frac{\varepsilon}{2}\left|\log\det B\right| + \frac{\varepsilon}{2}\left|\log\det B\right|] \\
&\leq \Pr\left[\left|\log\det B - \mathtt{tr}_m\left(\widetilde{p}_n(A)\right)\right| \leq \varepsilon\left|\log\det B\right|\right] \\
&= \Pr\left[\left|\log\det B - \Gamma\right| \leq \varepsilon\left|\log\det B\right|\right],
\end{aligned}$$

where $\Gamma = \mathtt{tr}_m\left(\widetilde{p}_n(A)\right)$.

5. Experiments

5.1. Performance Evaluation and Comparison

We first investigate the empirical performance of our proposed algorithm on large sparse random matrices.[2] We generate a random matrix $C \in \mathbb{R}^{d\times d}$, where the number of non-zero entries per each row is around 10. We first select five non-zero off-diagonal entries in each row with values uniformly distributed in $[-1, 1]$. To make the matrix symmetric, we set the entries in transposed positions to the same values. Finally, to guarantee positive definiteness, we set its diagonal entries to absolute row-sums and add a small weight, 10^{-3}.

Figure 1 (a) shows the running time of **Algorithm 2** from $d = 10^3$ to 3×10^7, where we choose $m = 10$, $n = 15$, $\sigma_{\min} = 10^{-3}$ and $\sigma_{\max} = \|C\|_1$. It scales roughly linearly over a large range of sizes. We use a machine with 3.40 Ghz Intel I7 processor with 24 GB RAM. It takes only 500 seconds for a matrix of dimension 3×10^7 with 3×10^8 non-zero entries. In Figure 1 (b), we study the relative accuracy compared to the exact log-determinant computation up to size 3×10^4. Relative errors are very small, below 0.1%, and appear to only improve for higher dimensions.

Under the same setup, we also compare the running time of our algorithm with other algorithm for computing determinants: Cholesky decomposition and Schur complement. The latter was used for sparse inverse covariance estimation with over a million variables (Hsieh et al., 2013) and we run the code implemented by the authors. The running time of the algorithms are reported in Figure 1 (c). The proposed algorithm is dramatically faster than both exact algorithms. We also compare the accuracy of our algorithm to a related stochastic algorithm that uses Taylor expansions (Zhang & Leithead, 2007). For a fair comparison we use a large number of samples, $n = 1000$, for both algorithms to focus on the polynomial approximation errors. The results are reported in Figure 1 (d), showing that our algorithm using Chebyshev expansions is superior in accuracy compared to the one based on Taylor series.

5.2. Maximum Likelihood Estimation for GMRF

GMRF with 25 million variables for synthetic data. We now apply our proposed algorithm for maximum likelihood (ML) estimation in Gaussian Markov Random Fields (GMRF) (Rue & Held, 2005). GMRF is a multivariate joint Gaussian distribution defined with respect to a graph. Each node of the graph corresponds to a random variable in the Gaussian distribution, where the graph captures the conditional independence relationships (Markov properties) among the random variables. The model has been extensively used in many applications in computer vision, spatial statistics, and other fields. The inverse covariance matrix J (also called information or precision matrix) is positive definite and sparse: J_{ij} is non-zero only if the edge $\{i, j\}$ is contained in the graph.

We first consider a GMRF on a square grid of size 5000×5000 (with $d = 25$ million variables) with precision matrix $J \in \mathbb{R}^{d\times d}$ parameterized by ρ, i.e., each node has four

[2] Our code is at http://sites.google.com/site/mijirim/logdet_code.zip

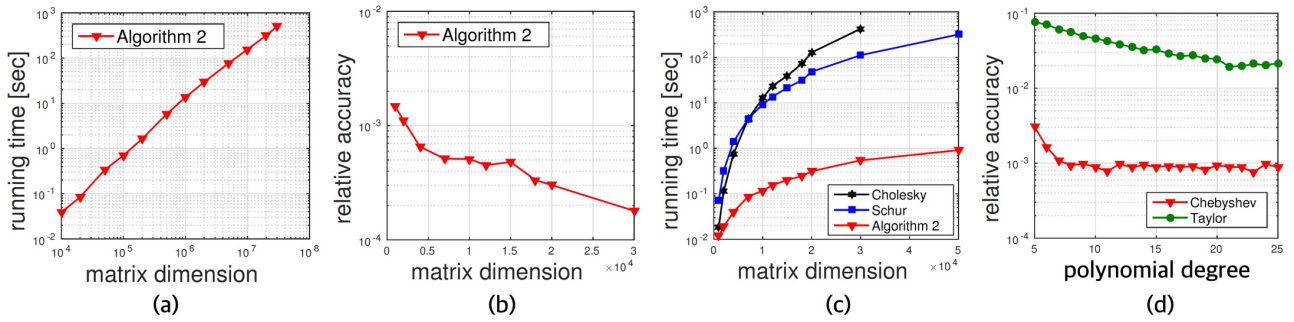

Figure 1. Performance evaluations of **Algorithm 2** and comparisons with other ones: (a) running time vs. dimension, (b) relative accuracy, (c) comparison in running time with Cholesky decomposition and Schur complement and (d) comparison in accuracy with Taylor approximation in (Zhang & Leithead, 2007). The relative accuracy means a ratio between the absolute error of the output of an approximation algorithm and the actual value of log-determinant.

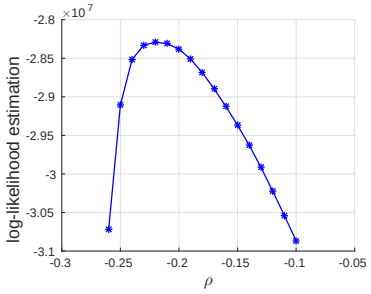

Figure 2. Log-likelihood estimation for hidden parameter ρ for square GMRF model of size 5000×5000.

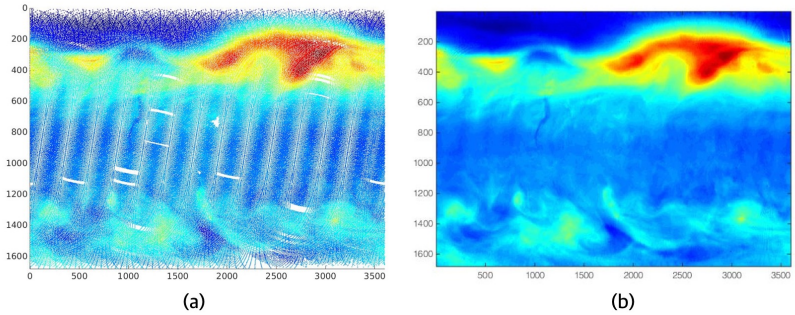

Figure 3. GMRF interpolation of ozone measurements: (a) original sparse measurements and (b) interpolated values using a GMRF with parameters fitted using **Algorithm 2**.

neighbors with partial correlation ρ. We generate a sample $\mathbf{x}$ from the GMRF model (using Gibbs sampler) for parameter $\rho = -0.22$. The log-likelihood of the sample is: $\log p(\mathbf{x}|\rho) = \log\det J(\rho) - \mathbf{x}^\top J(\rho)\mathbf{x} + G$, where $J(\rho)$ is a matrix of dimension 25×10^6 and 10^8 non-zero entries, and G is a constant independent of ρ. We use **Algorithm 2** to estimate the log-likelihood as a function of ρ, as reported in Figure 2. The estimated log-likelihood is maximized at the correct (hidden) value $\rho = -0.22$.

GMRF with 6 million variables for ozone data. We also consider GMRF parameter estimation from real spatial data with missing values. We use the data-set from (Aune et al., 2014) that provides satellite measurements of ozone levels over the entire earth following the satellite tracks. We use a resolution of 0.1 degrees in lattitude and longitude, giving a spatial field of size 1681×3601, with over 6 million variables. The data-set includes 172,000 measurements. To estimate the log-likelihood in presence of missing values, we use the Schur-complement formula for determinants. Let the precision matrix for the entire field be $J = \begin{pmatrix} J_o & J_{o,z} \\ J_{z,o} & J_z \end{pmatrix}$, where subsets $\mathbf{x}_o$ and $\mathbf{x}_z$ denote the observed and unobserved components of $\mathbf{x}$. The marginal precision matrix of $\mathbf{x}_o$ is $\bar{J}_o = J_o - J_{o,z}J_z^{-1}J_{z,o}$. Its log-determinant is computed as $\log(\det(\bar{J}_o)) = \log\det(J) - \log\det(J_z)$ via Schur complements. To evaluate the quadratic term $x_o'\bar{J}_o x_o$ of the log-likelihood we need a single linear solve using an iterative solver. We use a linear combination of the thin-plate model and the thin-membrane models (Rue & Held, 2005), with two parameters α and β: $J = \alpha I + (\beta)J_{tp} + (1-\beta)J_{tm}$ and obtain ML estimates using **Algorithm 2**. Note that $\sigma_{min}(J) = \alpha$. We show the sparse measurements in Figure 3 (a) and the GMRF interpolation using fitted values of parameters in Figure 3 (b).

6. Conclusion

Tools from numerical linear algebra, e.g. determinants, matrix inversion and linear solvers, eigenvalue computation and other matrix decompositions, have been playing an important theoretical and computational role for machine learning applications. In this paper, we designed and analyzed a high accuracy linear-time approximation algorithm for the logarithm of matrix determinants, where its exact computation requires cubic-time. We believe that the proposed algorithm will find numerous applications in machine learning problems.

Acknowledgement

We would like to thank Haim Avron and Jie Chen for fruitful comments on Chebyshev approximations, and Cho-Jui Hsieh for providing the code for Shur complement-based log-det computation.

References

Anandkumar, A., Huang, F., Hsu, D. J., and Kakade, S.M. Learning mixtures of tree graphical models. In *Advances in Neural Information Processing Systems*, pp. 1052–1060, 2012.

Aravkin, A., Friedlander, M. P., Herrmann, F. J., and Van Leeuwen, T. Robust inversion, dimensionality reduction, and randomized sampling. *Mathematical Programming*, 134(1):101–125, 2012.

Aune, E., Simpson, D.P., and Eidsvik, J. Parameter estimation in high dimensional Gaussian distributions. *Statistics and Computing*, 24(2):247–263, 2014.

Avron, H. Counting triangles in large graphs using randomized matrix trace estimation. In *Workshop on Large-scale Data Mining: Theory and Applications*, 2010.

Avron, H. and Toledo, S. Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix. *Journal of the ACM*, 58(2):8, 2011.

Bekas, C, Kokiopoulou, E, and Saad, Y. An estimator for the diagonal of a matrix. *Applied numerical mathematics*, 57(11):1214–1229, 2007.

Berrut, J. P. and Trefethen, L. N. Barycentric Lagrange Interpolation. *SIAM Review*, 46(3):501–517, 2004.

Boutsidis, Christos, Drineas, Petros, Kambadur, Prabhanjan, and Zouzias, Anastasios. A Randomized Algorithm for Approximating the Log Determinant of a Symmetric Positive Definite Matrix. *arXiv preprint arXiv:1503.00374*, 2015.

Davis, J.V., Kulis, B., Jain, P., Sra, S., and Dhillon, I.S. Information-theoretic metric learning. In *ICML*, 2007.

de De Villiers, J. *Mathematics of Approximation*. Mathematics Textbooks for Science and Engineering. Atlantis Press, 2012. ISBN 9789491216503. URL https://books.google.co.kr/books?id=l5mIro_6RlUC.

Dempster, A. P. Covariance selection. *Biometrics*, pp. 157–175, 1972.

Di Napoli, E., Polizzi, E., and Saad, Y. Efficient estimation of eigenvalue counts in an interval. *arXiv preprint arXiv:1308.4275*, 2013.

Gershgorin, S. Abramovich. Uber die abgrenzung der eigenwerte einer matrix. *Izvestiya or Russian Academy of Sciences*, (6):749–754, 1931.

Hsieh, C.J., Sustik, M. A., Dhillon, I. S., Ravikumar, P. K., and Poldrack, R. BIG & QUIC: Sparse inverse covariance estimation for a million variables. In *Adv. in Neural Information Processing Systems*, pp. 3165–3173, 2013.

Hutchinson, M.F. A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. *Communications in Statistics-Simulation and Computation*, 18(3):1059–1076, 1989.

Ipsen, Ilse CF. Computing an eigenvector with inverse iteration. *SIAM review*, 39(2):254–291, 1997.

Johnson, J. K., Netrapalli, P., and Chertkov, M. Learning planar ising models. *preprint arXiv:1011.3494*, 2010.

Kocay, W. and Kreher, D.L. *Graphs, Algorithms, and Optimization*. Discrete Mathematics and Its Applications. CRC Press, 2004. ISBN 9780203489055.

MacKay, D.J.C. *Information theory, inference, and learning algorithms*. Cambridge University Press, 2003.

Malioutov, D. M., Johnson, J. K., and Willsky, A.S. Low-rank variance estimation in large-scale GMRF models. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing, 2006.*, volume 3, pp. III–III. IEEE, 2006.

Mason, J. C. and Handscomb, D. C. *Chebyshev polynomials*. CRC Press, 2002.

Meila, M. and Jordan, M.I. Learning with mixtures of trees. *Journal of Machine Learning Research*, 1:1–48, 2001.

Morača, N. Bounds for norms of the matrix inverse and the smallest singular value. *Linear Algebra and its Applications*, 429(10):2589–2601, 2008.

Pace, R. K. and LeSage, J. P. Chebyshev approximation of log-determinants of spatial weight matrices. *Computational Statistics & Data Analysis*, 45(2):179–196, 2004.

Rasmussen, C. E. and Williams, C.K. *Gaussian processes for machine learning*. MIT press, 2005.

Roosta-Khorasani, F. and Ascher, U. Improved bounds on sample size for implicit matrix trace estimators. *arXiv preprint arXiv:1308.2475*, 2013.

Rue, H. and Held, L. *Gaussian Markov random fields: theory and applications*. CRC Press, 2005.

Schraudolph, N. N. and Kamenetsky, D. Efficient exact inference in planar ising models. In *Advances in Neural Information Processing Systems*, pp. 1417–1424, 2009.

Stein, M. L., Chen, J., and Anitescu, M. Stochastic approximation of score functions for Gaussian processes. *The Annals of Applied Statistics*, 7(2):1162–1191, 2013.

Tao, T. and Vu, V. Random matrices: The distribution of the smallest singular values. *Geometric And Functional Analysis*, 20(1):260–297, 2010.

Tao, Terence and Vu, Van H. Inverse Littlewood-Offord theorems and the condition number of random discrete matrices. *Annals of Mathematics*, pp. 595–632, 2009.

Van Aelst, S. and Rousseeuw, P. Minimum volume ellipsoid. *Wiley Interdisciplinary Reviews: Computational Statistics*, 1(1):71–82, 2009.

Wainwright, M. J. and Jordan, M. I. Log-determinant relaxation for approximate inference in discrete Markov random fields. *Signal Processing, IEEE Trans. on*, 54 (6):2099–2109, 2006.

Xiang, Shuhuang, Chen, Xiaojun, and Wang, Haiyong. Error bounds for approximation in Chebyshev points. *Numerische Mathematik*, 116(3):463–491, 2010.

Zhang, Y. and Leithead, W. E. Approximate implementation of the logarithm of the matrix determinant in Gaussian process regression. *Journal of Statistical Computation and Simulation*, 77(4):329–348, 2007.