Large-scale Log-determinant Computation through Stochastic Chebyshev Expansions

Insu Han[1], Dmitry Malioutov[2], Jinwoo Shin[1]

[1]Korea Advanced Institute of Science and Technology (KAIST)
[2]IBM Research

ICML 2015, Lille
July 8, 2015

Outline

Outline

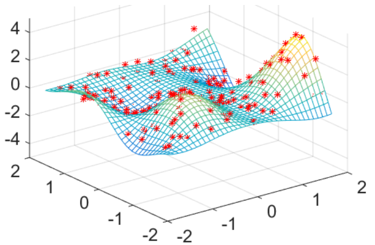
1. Summary
   - Problem
   - Algorithm and Error Bound
   - Related Work
2. Proof
   - Why Chebyshev approximation?
   - Why Rademacher random vector?
   - Proof Strategy
3. Extension and Experiment
   - Log-determinant for general non-singular matrices
   - Experiment for Large-scale Data
   - Application: GMRF Interpolation of Ozone Measure

Problem

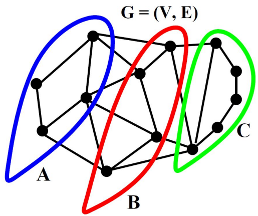
Problem: Computing matrix determinant

Determinant of positive semi-definite matrix plays an important role in many machine learning tasks including

- ML estimation for Gaussian graphical and Gaussian process model
- Discrete probabilistic models, e.g., tree mixture models and Markov random fields
- Minimum-volume ellipsoids
- Metric learning and kernel learning



(a) MAP estimate for Gaussian process model



(b) Gaussian graphical model

Our Contribution

Computational issue

The exact computation requires $O(d^3)$ operations for a $d \times d$ matrix.

- The cubic growth in the running time makes the computation infeasible (i.e., too slow) for large-scale problems.
- The popular matrix decomposition methods (such as Cholesky) can cause **memory overflow** even for sparse matrices of $d = 10^5$ on the single commodity machine having 32 GB memory.

Our Contribution

Computational issue

The exact computation requires $O(d^3)$ operations for a $d \times d$ matrix.

- The cubic growth in the running time makes the computation infeasible (i.e., too slow) for large-scale problems.
- The popular matrix decomposition methods (such as Cholesky) can cause **memory overflow** even for sparse matrices of $d = 10^5$ on the single commodity machine having 32 GB memory.

Contribution at a high level

We develop a fast algorithm for approximating the log-determinant of a large-scale sparse positive semi-definite matrix with rigorous provable guarantee.

- Our algorithm computes the log-determinants of matrices involving tens of millions of variables (i.e., $d \approx 10^7$) with 99.9% accuracy in a few minutes.

Our Algorithm

Key ideas of our algorithm

- The log-determinant is equal to trace of the matrix logarithm

$$\log\det B = \mathtt{tr}(\log B).$$

Our Algorithm

Key ideas of our algorithm

- The log-determinant is equal to trace of the matrix logarithm

$$\log\det B = \mathtt{tr}(\log B).$$

- The log function can be approximated to n-th degree polynomial i.e., $\log x \approx a_0 + a_1 x + \cdots + a_n x^n$

$$\begin{aligned}\mathtt{tr}(\log B) &\approx \mathtt{tr}(a_0 I + a_1 B + a_2 B^2 + \cdots + a_n B^n)\\ &= a_0 \cdot \mathtt{tr}(I) + a_1 \cdot \mathtt{tr}(B) + a_2 \cdot \mathtt{tr}(B^2) + \cdots + a_n \cdot \mathtt{tr}(B^n).\end{aligned}$$

Our Algorithm

Key ideas of our algorithm

- The log-determinant is equal to trace of the matrix logarithm

$$\log \det B = \mathtt{tr}(\log B).$$

- The log function can be approximated to n-th degree polynomial i.e., $\log x \approx a_0 + a_1 x + \cdots + a_n x^n$

$$\begin{aligned}\mathtt{tr}(\log B) &\approx \mathtt{tr}(a_0 I + a_1 B + a_2 B^2 + \cdots + a_n B^n) \\ &= a_0 \cdot \mathtt{tr}(I) + a_1 \cdot \mathtt{tr}(B) + a_2 \cdot \mathtt{tr}(B^2) + \cdots + a_n \cdot \mathtt{tr}(B^n).\end{aligned}$$

- For some random vector $\mathbf{z} \in \mathbb{R}^d$, it is known $\mathtt{tr}(B^k) = \mathbb{E}\left[\mathbf{z}^\top B^k \mathbf{z}\right]$.

Our Algorithm

Algorithm description

- The log-determinant is equal to trace of the matrix logarithm

$$\log\det B = \mathtt{tr}(\log B).$$

- The log function can be approximated to n-th degree polynomial i.e., $\log x \approx a_0 + a_1 x + \cdots + a_n x^n$

$$\begin{aligned}\mathtt{tr}(\log B) &\approx \mathtt{tr}(a_0 I + a_1 B + a_2 B^2 + \cdots + a_n B^n) \\ &= a_0 \cdot \mathtt{tr}(I) + a_1 \cdot \mathtt{tr}(B) + a_2 \cdot \mathtt{tr}(B^2) + \cdots + a_n \cdot \mathtt{tr}(B^n).\end{aligned}$$

- For some random vector $\mathbf{z} \in \mathbb{R}^d$, it is known $\mathtt{tr}(B^k) = \mathbb{E}\left[\mathbf{z}^\top B^k \mathbf{z}\right]$.

Our Algorithm

Algorithm description

- The log-determinant is equal to trace of the matrix logarithm

$$\log\det B = \mathtt{tr}(\log B).$$

- The log function can be approximated to n-th degree polynomial i.e., $\log x \approx a_0 + a_1 x + \cdots + a_n x^n$

$$\begin{aligned}\mathtt{tr}(\log B) &\approx \mathtt{tr}(a_0 I + a_1 B + a_2 B^2 + \cdots + a_n B^n)\\ &= a_0 \cdot \mathtt{tr}(I) + a_1 \cdot \mathtt{tr}(B) + a_2 \cdot \mathtt{tr}(B^2) + \cdots + a_n \cdot \mathtt{tr}(B^n).\end{aligned}$$

  - We choose a_i as the i-th coefficient of the Chebyshev expansion to $\log x$
- For some random vector $\mathbf{z} \in \mathbb{R}^d$, it is known $\mathtt{tr}(B^k) = \mathbb{E}\left[\mathbf{z}^\top B^k \mathbf{z}\right]$.

Our Algorithm

Algorithm description

- The log-determinant is equal to trace of the matrix logarithm
$$\log\det B = \mathtt{tr}(\log B).$$
- The log function can be approximated to n-th degree polynomial i.e., $\log x \approx a_0 + a_1 x + \cdots + a_n x^n$
$$\begin{aligned}\mathtt{tr}(\log B) &\approx \mathtt{tr}(a_0 I + a_1 B + a_2 B^2 + \cdots + a_n B^n)\\ &= a_0 \cdot \mathtt{tr}(I) + a_1 \cdot \mathtt{tr}(B) + a_2 \cdot \mathtt{tr}(B^2) + \cdots + a_n \cdot \mathtt{tr}(B^n).\end{aligned}$$
  - We choose a_i as the i-th coefficient of the Chebyshev expansion to $\log x$
- For some random vector $\mathbf{z} \in \mathbb{R}^d$, it is known $\mathtt{tr}(B^k) = \mathbb{E}\left[\mathbf{z}^\top B^k \mathbf{z}\right]$.
  - We choose m Rademacher random vectors $\mathbf{z}_1, \ldots, \mathbf{z}_m \in \{-1, 1\}^d$ and estimate the trace by
$$\mathtt{tr}(B^k) \approx \frac{1}{m} \sum_{i=1}^{m} \mathbf{z}_i^\top B^k \mathbf{z}_i.$$

Complexity and Error Bound

Complexity

The overall running time is

$$O\left(m \times n \times \# \text{ of non-zero entries in } B\right),$$

where m is the number of samples for trace estimate and n is the degree of the Chebyshev polynomial.

Complexity and Error Bound

Complexity

The overall running time is

$$O\left(m \times n \times \#\text{ of non-zero entries in } B\right),$$

where m is the number of samples for trace estimate and n is the degree of the Chebyshev polynomial.

Theorem (Han, Malioutov and Shin, 2015)

For positive semi-definite matrix $B \in \mathbb{R}^{d \times d}$ having eigenvalues in $[\lambda_{min}, \lambda_{max}]$, the algorithm returns

$$\text{output} \in [(1-\varepsilon)\log\det B, (1+\varepsilon)\log\det B], \qquad \textit{with probability } 1-\zeta,$$

if we choose $m \geq \varepsilon^{-2} \log\left(\frac{1}{\zeta}\right)$ *and* $n \geq \sqrt{\frac{\lambda_{max}}{\lambda_{min}}} \log\left(\frac{1}{\varepsilon}\frac{\lambda_{max}}{\lambda_{min}}\right)$.

Therefore, the algorithm runs in $O^*(\sqrt{\frac{\lambda_{max}}{\lambda_{min}}} d)$ time for sparse matrix B !

Related Work

Approximation methods for log-determinant

- Taylor series expansion & trace estimator [Barry and Pace, 1999]
- Taylor series & trace estimator and error compensation schemes for accuracy [Zhang and Leithead, 2007]
- Taylor series & trace estimator and approximate largest eigenvalue by power method [Boutsidis et al., 2015]
- Chebyshev expansion & exact trace calculation [Pace and LeSage, 2004]
- Cauchy integral for matrix logarithm & trace estimator [Aune et al., 2014]
- Split a matrix into diagonal and non-diagonal part and stochastic approach for non-diagonal part [Dorn and Enßlin, 2015]

We first use Chebyshev approximation and Trace estimator for log-determinant !

Outline

1. Summary
   - Problem
   - Algorithm and Error Bound
   - Related Work
2. Proof
   - Why Chebyshev approximation?
   - Why Rademacher random vector?
   - Proof Strategy
3. Extension and Experiment
   - Log-determinant for general non-singular matrices
   - Experiment for Large-scale Data
   - Application: GMRF Interpolation of Ozone Measure

Polynomial Approximation

The most popular approach is the Taylor series approximation.

[0] The prime sum $\sum'$ denotes a sum whose first term is halved

Polynomial Approximation

The most popular approach is the Taylor series approximation.

Taylor series approximation

$$\log(1-x) \approx -x - \frac{x^2}{2} - \frac{x^3}{3} - \cdots - \frac{x^n}{n} \quad \text{for } x \in [-1, 1]$$

[0] The prime sum $\sum'$ denotes a sum whose first term is halved

Polynomial Approximation

The most popular approach is the Taylor series approximation.

Taylor series approximation

$$\log(1-x) \approx -x - \frac{x^2}{2} - \frac{x^3}{3} - \cdots - \frac{x^n}{n} \quad \text{for } x \in [-1,1]$$

Chebyshev series approximation

$$\log(1-x) \approx {\sum_{i=0}^{n}}' c_i T_i(x) \quad \text{for } x \in [-1,1]$$

- $T_i(x)$ is i-th degree Chebyshev polynomial
  e.g., $T_0(x) = 1$, $T_1(x) = x$, $T_2(x) = 2x^2 - 1$ and $T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x)$.
- For $0 \le i \le n$,

$$c_i = \frac{2}{n+1} \sum_{k=0}^{n} \log\left(1 - \cos\left(\frac{\pi(k+1/2)}{n+1}\right)\right) T_i\left(\cos\left(\frac{\pi(k+1/2)}{n+1}\right)\right)$$

[0] The prime sum $\sum'$ denotes a sum whose first term is halved

Why Chebyshev Approximation?

Advantage of Chebyshev approximation

1. Taylor series approximation only gives local approximation while Chebyshev's one approximates over the entire closed interval.
2. Chebyshev approximation has better convergence rate. For example, approximation error of $\log x$ in $[\delta, 1-\delta]$ is bounded

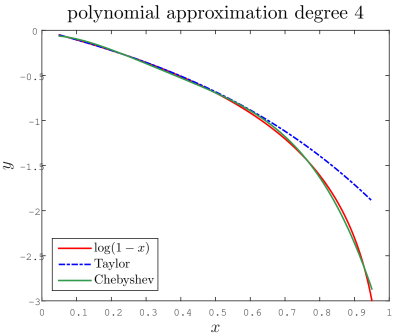
$$\max_{x\in[\delta,1-\delta]} |\log x - p_n(x)| \leq O\left(R^{-n}\right)$$

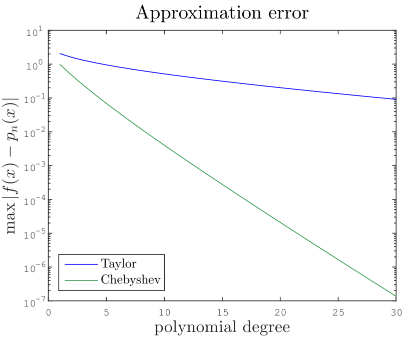
for some constant $R > 1$.

	Taylor approximation	Chebyshev approximation
Convergence rate R	$1 + O(\delta)$	$1 + O\left(\sqrt{\delta}\right)$

Why Chebyshev Approximation?

Comparision Taylor series with Chebyshev approximation





Chebyshev provides a much tighter approximation with more uniform errors.

Trace Estimator

Theorem

Let $\mathbf{z} = [z_1, z_2, \ldots, z_d]^\top \in \mathbb{R}^d$ *be a random vector such that*

$$\mathbb{E}\left[z_i z_j\right] = 0 \text{ for } i \neq j \text{ and } \mathbb{E}\left[z_i^2\right] = 1 \text{ for } 1 \leq i \leq d.$$

Then,

$$\mathbb{E}\left[\mathbf{z}^\top B \mathbf{z}\right] = \mathtt{tr}\left(B\right)$$

for positive semi-definite matrix $B \in \mathbb{R}^{d \times d}$.

Examples of random vector

- Gaussian distribution, i.e. $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$
- Rademacher distribution, i.e. $\Pr(+1) = \Pr(-1) = \frac{1}{2}$
- Unit vector i.e. $\mathbf{z} \in \{e_1, e_2, \ldots, e_d\}$

Trace Estimator

Theorem

Let $\mathbf{z} = [z_1, z_2, \dots, z_d]^\top \in \mathbb{R}^d$ *be a random vector such that*

$$\mathbb{E}[z_i z_j] = 0 \text{ for } i \neq j \text{ and } \mathbb{E}\left[z_i^2\right] = 1 \text{ for } 1 \leq i \leq d.$$

Then,

$$\mathbb{E}\left[\mathbf{z}^\top B \mathbf{z}\right] = \mathtt{tr}(B)$$

for positive semi-definite matrix $B \in \mathbb{R}^{d \times d}$.

Examples of random vector

- Gaussian distribution, i.e. $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$
- Rademacher distribution, i.e. $\Pr(+1) = \Pr(-1) = \frac{1}{2}$
- Unit vector i.e. $\mathbf{z} \in \{e_1, e_2, \dots, e_d\}$

Next: Why we choose Rademacher?

Why Rademacher Distribution?

Rademacher is the best for the number of samples

A lower bound on the number of samples m for trace estimator is provided [Roosta-Khorasani and Ascher, 2014]

$$\left| \mathtt{tr}\,(B) - \frac{1}{m} \sum_{i=0}^{m} \mathbf{z}^\top B \mathbf{z} \right| \leq \varepsilon \cdot |\mathtt{tr}\,(B)|$$

with probability at least $1 - \zeta$.

Distribution	Bound on samples	Variance of estimator
Gaussian	$8\varepsilon^{-2} \log(2/\zeta)$	$2\|B\|_F$
Rademacher	$6\varepsilon^{-2} \log(2/\zeta)$	$2\left(\|B\|_F - \sum_{i=1}^{d} B_{ii}^2\right)$
Unit vector	$2\left(\frac{d \max \|B_{ii}\|}{\mathtt{tr}(B)}\right)^2 \varepsilon^{-2} \log(2/\zeta)$	$d \sum_{i=1}^{d} B_{ii}^2 - \mathtt{tr}^2(B)$

Rademacher estimators achieves the smallest lower bound and variance!

Proof Strategy on the Error Bound

Proof Strategy

- Without loss of generality, we assume all eigenvalues are in the interval $[\delta, 1-\delta]$.
- We use [1]Chebyshev polynomial p_n and [2]Rademacher trace estimator:

$$\log\det B = \mathtt{tr}\left(\log B\right) \overset{1}{\approx} \mathtt{tr}\left(p_n(B)\right) \overset{2}{\approx} \frac{1}{m}\sum_{i=0}^{m} \mathbf{z}_i^\top p_n(B)\mathbf{z}_i$$

Proof Strategy on the Error Bound

Proof Strategy

- Without loss of generality, we assume all eigenvalues are in the interval $[\delta, 1-\delta]$.
- We use [1]Chebyshev polynomial p_n and [2]Rademacher trace estimator:

$$\log\det B = \mathtt{tr}\left(\log B\right) \overset{1}{\approx} \mathtt{tr}\left(p_n(B)\right) \overset{2}{\approx} \frac{1}{m}\sum_{i=0}^{m} \mathbf{z}_i^\top p_n(B)\mathbf{z}_i$$

1. We first prove the following using [Xiang et al., 2010]

$$\left|\mathtt{tr}\left(\log B\right) - \mathtt{tr}\left(p_n(B)\right)\right| \leq O\left(d \cdot R^{-n}\right)$$

for some constant $R = \frac{1}{1-\delta} + \sqrt{\left(\frac{1}{1-\delta}\right)^2 - 1}$.

Proof Strategy on the Error Bound

Proof Strategy

- Without loss of generality, we assume all eigenvalues are in the interval $[\delta, 1-\delta]$.
- We use [1]Chebyshev polynomial p_n and [2]Rademacher trace estimator:

$$\log\det B = \mathtt{tr}\left(\log B\right) \overset{1}{\approx} \mathtt{tr}\left(p_n(B)\right) \overset{2}{\approx} \frac{1}{m}\sum_{i=0}^{m} \mathbf{z}_i^\top p_n(B)\mathbf{z}_i$$

1. We first prove the following using [Xiang et al., 2010]

$$\left|\mathtt{tr}\left(\log B\right) - \mathtt{tr}\left(p_n(B)\right)\right| \le O\left(d \cdot R^{-n}\right)$$

   for some constant $R = \frac{1}{1-\delta} + \sqrt{\left(\frac{1}{1-\delta}\right)^2 - 1}$.

2. [Roosta-Khorasani and Ascher, 2014] proves that for $m \ge \varepsilon^{-2}\log\left(2/\zeta\right)$

$$\left|\mathtt{tr}\left(p_n(B)\right) - \frac{1}{m}\sum_{i=0}^{m} \mathbf{z}_i^\top p_n(B)\mathbf{z}_i\right| \le \varepsilon \cdot \left|\mathtt{tr}\left(p_n(B)\right)\right|$$

   with probability at least $1-\zeta$.

Outline

Extension to General Non-Singular Matrices

Consider general non-singular matrix $C \in \mathbb{R}^{d \times d}$.

Extension to General Non-Singular Matrices

Consider general non-singular matrix $C \in \mathbb{R}^{d \times d}$.

The idea for generalization

$$\log(|\det C|) = \frac{1}{2} \log\det\left(C^T C\right)$$

Extension to General Non-Singular Matrices

Consider general non-singular matrix $C \in \mathbb{R}^{d \times d}$.

The idea for generalization

$$\log(|\det C|) = \frac{1}{2} \log \det \left(C^T C \right)$$

Obviously, matrix $C^T C$ is a positive semi-definite matrix.

Extension to General Non-Singular Matrices

Consider general non-singular matrix $C \in \mathbb{R}^{d \times d}$.

The idea for generalization

$$\log(|\det C|) = \frac{1}{2} \log \det \left(C^T C \right)$$

Obviously, matrix $C^T C$ is a positive semi-definite matrix.

Algorithm for general non-singular matrix C

- Use our algorithm for $C^T C$ and halve the output value.

Extension to General Non-Singular Matrices

Consider general non-singular matrix $C \in \mathbb{R}^{d \times d}$.

The idea for generalization

$$\log(|\det C|) = \frac{1}{2} \log \det \left(C^T C \right)$$

Obviously, matrix $C^T C$ is a positive semi-definite matrix.

Algorithm for general non-singular matrix C

- Use our algorithm for $C^T C$ and halve the output value.

Complexity

The overall running time is still

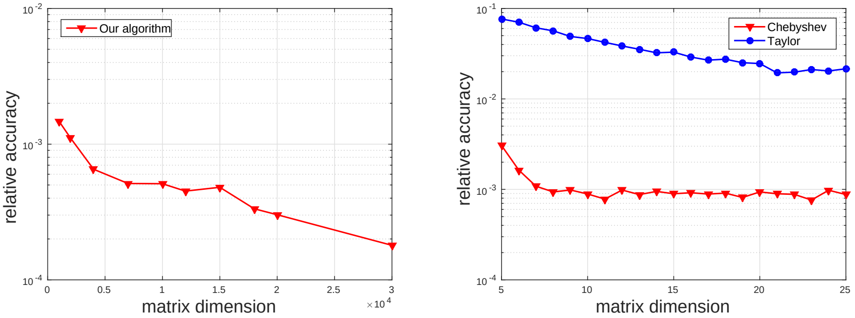
$$O\left(m \times n \times \# \text{ of non-zero entries in } C\right),$$

where m is the number of samples for trace estimate and n is the degree of the Chebyshev polynomial.

Experiment for Large-scale Data

Accuracy

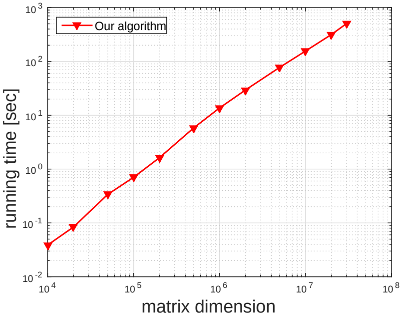
- Approximation error is less than 0.1% for $m = 10$ and $n = 15$.
- Chebyshev is superior in accuracy compared to Taylor.

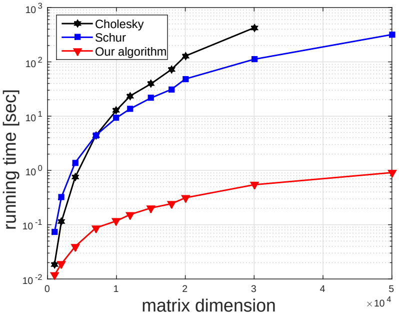


Experiment for Large-scale Data

Running time

- We choose $m = 10$ and $n = 15$.
- Compare to exact method such as Cholesky decomposition and Schur complement, our algorithm is dramatically faster!
- For example, it takes about 130 sec for $10^7 \times 10^7$ matrix.





Application: GMRF Interpolation of Ozone Measure

Problem

- Interpolate sparse irregular satellite measurements of ozone levels.
- Given 172,000 data, we can interpolate large-scale ozone measurements with over 6 million variables (1681×3601 grid points).
- ML estimate for Gaussian Markov random field interpolation using proposed algorithm.
- Determinant computation is necessery for precision matrix of GMRF.

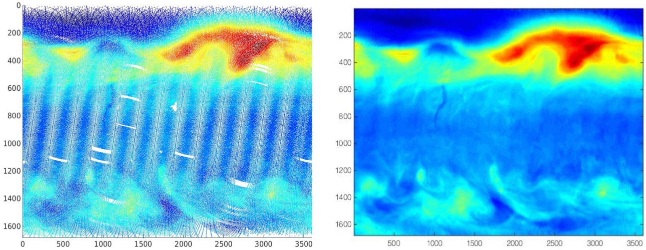


Figure: (a) original sparse measurements (b) interpolated values using a GMRF

Conclusion

Conclusion

- We analyzed a high accuracy linear-time approximation algorithm for log-determinant.
- We hope the proposed algorithm will find many applications in large-scale machine learning problems.

Conclusion

Conclusion

- We analyzed a high accuracy linear-time approximation algorithm for log-determinant.
- We hope the proposed algorithm will find many applications in large-scale machine learning problems.

Thank you for your attention !

References I

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

Barry, R. P. and Pace, R. K. (1999).
Monte carlo estimates of the log determinant of large sparse matrices.
Linear Algebra and its applications, 289(1):41–54.

Boutsidis, C., Drineas, P., Kambadur, P., and Zouzias, A. (2015).
A randomized algorithm for approximating the log determinant of a symmetric positive definite matrix.
arXiv preprint arXiv:1503.00374.

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

Dorn, S. and Enßlin, T. A. (2015).
Stochastic determination of matrix determinants.
arXiv preprint arXiv:1504.02661.

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

Roosta-Khorasani, F. and Ascher, U. (2014).
Improved bounds on sample size for implicit matrix trace estimators.
Foundations of Computational Mathematics, pages 1–26.

References II

Xiang, S., Chen, X., and Wang, H. (2010).
Error bounds for approximation in chebyshev points.
Numerische Mathematik, 116(3):463–491.

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