Approximating and Optimizing Large-scale Spectral-sums via Stochastic Chebyshev Expansion

Insu Han[1]

Joint work with Dmitry Malioutov[2], Haim Avron[3] and Jinwoo Shin[1]

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

PIML 2018, Santa Fe
Jan 22, 2018

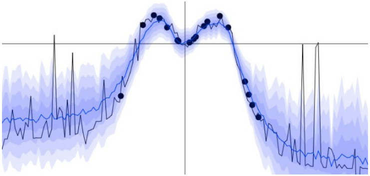
Outline

1. Summary of Spectral-sums
   - Formal Definition
   - Problems and Contributions
2. Approximating Spectral-sums
   - Algorithm and Analysis
   - Polynomial Approximation
   - Trace Estimator
3. Optimizing Spectral-sums
   - Gradient Descent for Spectral-sums
   - Unbiased Gradient Estimation
4. Experimantal Results
   - Approximating Spectral-sums
   - Optimizing Spectral-sums

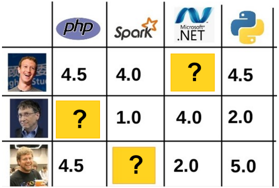
Outline

Matrix Functions in Machine Learning

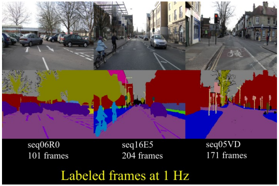
Matrix functions have been utilized in many machine learning problems:



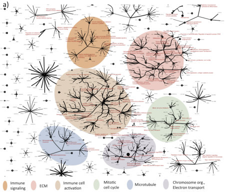
(a) Regression with Gaussian process



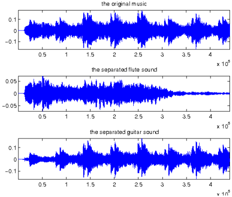
(b) Collaborative filtering for recommendation



(c) Image processing



(d) Gene expression



(e) Speech recognition

Definition of Spectral-sums

Given a symmetric matrix $A \in \mathbb{R}^{d \times d}$ and a scalar function $f : \mathbb{R} \to \mathbb{R}$, **spectral-sums** is defined as

$$\sum_{i=1}^{d} f(\lambda_i) = \mathtt{tr}\left(f(A)\right),$$

where $\lambda_1, \lambda_2, \ldots, \lambda_d$ are eigen (or singular) values of A.

Definition of Spectral-sums

Given a symmetric matrix $A \in \mathbb{R}^{d \times d}$ and a scalar function $f : \mathbb{R} \to \mathbb{R}$, **spectral-sums** is defined as

$$\sum_{i=1}^{d} f(\lambda_i) = \mathtt{tr}\,(f(A)),$$

where $\lambda_1, \lambda_2, \ldots, \lambda_d$ are eigen (or singular) values of A.

Examples

- If $f(x) = \log x$, it is the log-determinant
- If $f(x) = x^{-1}$, it is the trace of inverse
- If $f(x) = x^p$, it is the Schatten norm (the nuclear norm is the case $p = 1$)
- if $f(x) = x \log x$, it is the Von-Neumann entropy
- If $f(x) = \exp(x)$, it is the Estrada index
- If $f(x) = \begin{cases} 1 & \text{if } x \leq 0 \\ 0 & \text{otherwise} \end{cases}$, it is testing positive definiteness

Problems

Approximating spectral-sums

$$\mathtt{tr}\left(f(A)\right) := \sum_i f(\lambda_i) \approx ?$$

Optimizing spectral-sums

$$\min_A \mathtt{tr}\left(f(A)\right)$$

Problems

Approximating spectral-sums

$$\mathtt{tr}\left(f(A)\right) := \sum_i f(\lambda_i) \approx ?$$

Optimizing spectral-sums

$$\min_A \mathtt{tr}\left(f(A)\right)$$

Computational issue

- **Approximation:** The exact computation requires matrix decomposition methods with $O(d^3)$ operations for a $d \times d$ matrix.
- **Optimization:** Gradient descent methods can be used. Computing gradient of spectral-sums also requires decomposition methods with $O(d^3)$ operations.

Contributions

Approximating spectral-sums

$$\mathtt{tr}\,(f(A)) := \sum_i f(\lambda_i) \approx ?$$

Optimizing spectral-sums

$$\min_A \mathtt{tr}\,(f(A))$$

Computational issue

- **Approximation:** The exact computation requires matrix decomposition methods with $O(d^3)$ operations for a $d \times d$ matrix.
- **Optimization:** Gradient descent methods can be used. Computing gradient of spectral-sums also requires decomposition methods with $O(d^3)$ operations.

Our contributions

- We develop a fast algorithm for approximating spectral-sums of large-scale matrices with rigorous provable guarantee.
- We propose a fast (quadratic-time) unbiased gradient estimator for optimizing spectral-sums that guarantees to converge to the optimal.

Outline

Spectral-sums Approximation

Approximating spectral-sums

$$\texttt{tr}\left(f(A)\right) = \sum_i f(\lambda_i) \approx ?$$

Optimizing spectral-sums

$$\min_A \texttt{tr}\left(f(A)\right)$$

Key ideas of approximation

Spectral-sums Approximation

Approximating spectral-sums

$$\mathtt{tr}\left(f(A)\right) = \sum_i f(\lambda_i) \approx ?$$

Optimizing spectral-sums

$$\min_A \mathtt{tr}\left(f(A)\right)$$

Key ideas of approximation

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

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

Spectral-sums Approximation

Approximating spectral-sums

$$\mathtt{tr}\left(f(A)\right) = \sum_i f(\lambda_i) \approx ?$$

Optimizing spectral-sums

$$\min_A \mathtt{tr}\left(f(A)\right)$$

Key ideas of approximation

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

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

The bottleneck is A^n, i.e., n times of matrix-matrix multiplications $O(d^3)$.

Spectral-sums Approximation

Approximating spectral-sums

$\texttt{tr}\,(f(A)) = \sum_i f(\lambda_i) \approx ?$

Optimizing spectral-sums

$\min_A \texttt{tr}\,(f(A))$

Key ideas of approximation

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

$$\begin{aligned}\texttt{tr}\,(f(A)) &\approx \texttt{tr}\left(a_0 I + a_1 A + a_2 A^2 + \cdots + a_n A^n\right) \\ &= a_0 \cdot \texttt{tr}\,(I) + a_1 \cdot \texttt{tr}\,(A) + a_2 \cdot \texttt{tr}\left(A^2\right) + \cdots + a_n \cdot \texttt{tr}\,(A^n)\,.\end{aligned}$$

The bottleneck is A^n, i.e., n times of matrix-matrix multiplications $O(d^3)$.

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

Spectral-sums Approximation

Approximating spectral-sums

$$\mathtt{tr}\left(f(A)\right) = \sum_i f(\lambda_i) \approx ?$$

Optimizing spectral-sums

$$\min_A \mathtt{tr}\left(f(A)\right)$$

Algorithm description

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

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

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

Spectral-sums Approximation

Approximating spectral-sums

$$\texttt{tr}\left(f(A)\right) = \sum_i f(\lambda_i) \approx ?$$

Optimizing spectral-sums

$$\min_A \texttt{tr}\left(f(A)\right)$$

Algorithm description

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

$$\begin{aligned} \texttt{tr}\left(f(A)\right) &\approx \texttt{tr}\left(a_0 I + a_1 A + a_2 A^2 + \cdots + a_n A^n\right) \\ &= a_0 \cdot \texttt{tr}\left(I\right) + a_1 \cdot \texttt{tr}\left(A\right) + a_2 \cdot \texttt{tr}\left(A^2\right) + \cdots + a_n \cdot \texttt{tr}\left(A^n\right). \end{aligned}$$

We choose a_i as the i-th coefficient of the Chebyshev expansion to $f(x)$

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

Spectral-sums Approximation

Approximating spectral-sums

$$\mathtt{tr}\left(f(A)\right) = \sum_i f(\lambda_i) \approx ?$$

Optimizing spectral-sums

$$\min_A \mathtt{tr}\left(f(A)\right)$$

Algorithm description

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

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

We choose a_i as the i-th coefficient of the Chebyshev expansion to $f(x)$

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

We choose m Rademacher random vectors $\mathbf{v}_1, \dots, \mathbf{v}_m \in \{-1, 1\}^d$ and estimate the trace by

$$\mathtt{tr}(A^k) \approx \frac{1}{m} \sum_{i=1}^{m} \mathbf{v}_i^\top A^k \mathbf{v}_i.$$

Complexity and Error Bound

Complexity

The overall running time is

$$O\left(m \times n \times \|A\|_{\mathtt{mv}}\right),$$

where m is the number of samples for trace, n is the degree of Chebyshev expansion and $\|A\|_{\mathtt{mv}}$ is the complexity for multiplications A with a vector.

Complexity and Error Bound

Complexity

The overall running time is

$$O\left(m \times n \times \|A\|_{\texttt{mv}}\right),$$

where m is the number of samples for trace, n is the degree of Chebyshev expansion and $\|A\|_{\texttt{mv}}$ is the complexity for multiplications A with a vector.

Theorem (Han, Malioutov, Avron and Shin, 2016)

For symmetric matrix $A \in \mathbb{R}^{d \times d}$ having eigenvalues in $[\lambda_{\min}, \lambda_{\max}]$, the algorithm returns

$$\text{output} \in \left[(1-\varepsilon)\texttt{tr}\left(f(A)\right), (1+\varepsilon)\texttt{tr}\left(f(A)\right)\right], \qquad \textit{with probability } 1-\zeta,$$

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 A !

Polynomial Approximation

The most popular approach is Taylor series expansion. For example,

Taylor series expansion

$$\log x \approx -\sum_{j=0}^{n} \frac{(1-x)^j}{j}$$

Polynomial Approximation

The most popular approach is Taylor series expansion. For example,

Taylor series expansion

$$\log x \approx -\sum_{j=0}^{n} \frac{(1-x)^j}{j}$$

Chebyshev series expansion

$$\log x \approx \sum_{j=0}^{n} b_j T_j(x)$$

Here, $T_i(x)$ is i-th Chebyshev polynomial with $T_0(x) = 1, T_1(x) = x$ and $T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x)$ and

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

Why Chebyshev expansion?

The most popular approach is Taylor series expansion. For example,

Taylor series expansion

$$\log x \approx -\sum_{j=0}^{n} \frac{(1-x)^j}{j}$$

Chebyshev series expansion

$$\log x \approx \sum_{j=0}^{n} b_j T_j(x)$$

Advantage of Chebyshev series expansion

Chebyshev approximation has better convergence rate. For example,

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

for some constant $R > 1$.

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

Trace Estimator

Theorem (Hutchinson (1989))

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

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

Then, for any matrix $A \in \mathbb{R}^{d\times d}$*, it holds that* $\mathbb{E}\left[\mathbf{z}^\top A \mathbf{z}\right] = \mathtt{tr}(A)$.

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 (Hutchinson (1989))

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

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

Then, for any matrix $A \in \mathbb{R}^{d\times d}$*, it holds that* $\mathbb{E}\left[\mathbf{z}^\top A\mathbf{z}\right] = \mathtt{tr}(A)$.

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 (Hutchinson (1989))

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

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

Then, for any matrix $A \in \mathbb{R}^{d \times d}$*, it holds that* $\mathbb{E}\left[\mathbf{z}^\top A \mathbf{z}\right] = \mathtt{tr}(A)$.

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. $\mathbf{z} \in \{e_1, e_2, \ldots, e_d\}$

Bound on samples (Roosta et al., 2015)

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

Distribution	Bound on samples
Gaussian	$8\varepsilon^{-2}\log\left(\frac{2}{\zeta}\right)$
Rademacher	$6\varepsilon^{-2}\log\left(\frac{2}{\zeta}\right)$
Unit vector	$2\left(\frac{d \max\|A_{ii}\|}{\mathtt{tr}(A)}\right)^2 \varepsilon^{-2}\log\left(\frac{2}{\zeta}\right)$

Outline

Optimizing Spectral-sums

Approximating spectral-sums

$$\texttt{tr}\left(f(A)\right) \approx \mathbf{v}^\top p_n(A)\mathbf{v}$$

Optimizing spectral-sums

$$\min_A \texttt{tr}\left(f(A)\right)$$

Optimizing Spectral-sums

Approximating spectral-sums

$$\mathtt{tr}\left(f(A)\right) \approx \mathbf{v}^\top p_n(A)\mathbf{v}$$

Optimizing spectral-sums

$$\min_A \mathtt{tr}\left(f(A)\right)$$

Gradient descent methods

$$A \leftarrow A - \eta \nabla \mathtt{tr}\left(f(A)\right) \qquad (\eta : \text{step-size})$$

- Computing $\nabla \mathtt{tr}\left(f(A)\right) = f'(A)$ needs matrix decompositions with $O(d^3)$.

Optimizing Spectral-sums

Approximating spectral-sums

$$\mathtt{tr}\,(f(A)) \approx \mathbf{v}^\top p_n(A)\mathbf{v}$$

Optimizing spectral-sums

$$\min_A \mathtt{tr}\,(f(A))$$

Gradient descent methods

$$A \leftarrow A - \eta \nabla \mathbf{v}^\top p_n(A)\mathbf{v} \qquad (\eta : \text{step-size})$$

- Computing $\nabla \mathtt{tr}\,(f(A)) = f'(A)$ needs matrix decompositions with $O(d^3)$.
- One can use spectral-sums approximation by replacing the gradient with derivative of $\nabla \mathbf{v}^\top p_n(A)\mathbf{v}$.

Optimizing Spectral-sums

Approximating spectral-sums

$$\mathtt{tr}\,(f(A)) \approx \mathbf{v}^\top p_n(A)\mathbf{v}$$

Optimizing spectral-sums

$$\min_A \mathtt{tr}\,(f(A))$$

Gradient descent methods

$$A \leftarrow A - \eta \nabla \mathbf{v}^\top p_n(A)\mathbf{v} \qquad (\eta : \text{step-size})$$

- Computing $\nabla \mathtt{tr}\,(f(A)) = f'(A)$ needs matrix decompositions with $O(d^3)$.
- One can use spectral-sums approximation by replacing the gradient with derivative of $\nabla \mathbf{v}^\top p_n(A)\mathbf{v}$.
- It is required matrix-vector multiplications and vector outer products:

$$\mathbf{v}^\top p_n(A)\mathbf{v} = \mathbf{v}^\top \left(\sum_{j=0}^n b_j \mathbf{w}_j\right)$$

$$\nabla \mathbf{v}^\top p_n(A)\mathbf{v} = \sum_{j=1}^n \left(\sum_{i=j}^n b_i \mathbf{y}_{i-j}\right) \mathbf{w}_{j-1}^\top$$

where $\mathbf{w}_j := T_j(A)\mathbf{v}$ and $\mathbf{y}_{j+1} = 2\mathbf{w}_{j+1} - \mathbf{w}_{j-1}$.

Optimizing Spectral-sums

Approximating spectral-sums

$$\mathtt{tr}\,(f(A)) \approx \mathbf{v}^\top p_n(A)\mathbf{v}$$

Optimizing spectral-sums

$$\min_A \mathtt{tr}\,(f(A))$$

Gradient descent methods

$$A \leftarrow A - \eta \nabla \mathbf{v}^\top p_n(A)\mathbf{v} \qquad (\eta : \text{step-size})$$

- Computing $\nabla \mathtt{tr}\,(f(A)) = f'(A)$ needs matrix decompositions with $O(d^3)$.
- One can use spectral-sums approximation by replacing the gradient with derivative of $\nabla \mathbf{v}^\top p_n(A)\mathbf{v}$.
- It is required matrix-vector multiplications and vector outer products:

$$\mathbf{v}^\top p_n(A)\mathbf{v} = \mathbf{v}^\top \left(\sum_{j=0}^n b_j \mathbf{w}_j\right)$$

$$\nabla \mathbf{v}^\top p_n(A)\mathbf{v} = \sum_{j=1}^n \left(\sum_{i=j}^n b_i \mathbf{y}_{i-j}\right) \mathbf{w}_{j-1}^\top$$

  where $\mathbf{w}_j := T_j(A)\mathbf{v}$ and $\mathbf{y}_{j+1} = 2\mathbf{w}_{j+1} - \mathbf{w}_{j-1}$.
- Both spectral-sums and its derivative can be approximated with $O(d^2)$.

Optimizing Spectral-sums

Approximating spectral-sums

$$\mathtt{tr}\,(f(A)) \approx \mathbf{v}^\top p_n(A)\mathbf{v}$$

Optimizing spectral-sums

$$\min_A \mathtt{tr}\,(f(A))$$

Biased gradient estimator

$$A \leftarrow A - \eta \nabla \mathbf{v}^\top p_n(A)\mathbf{v} \qquad (\eta : \text{step-size})$$

Optimizing Spectral-sums

Approximating spectral-sums

$$\mathtt{tr}\left(f(A)\right) \approx \mathbf{v}^\top p_n(A)\mathbf{v}$$

Optimizing spectral-sums

$$\min_A \mathtt{tr}\left(f(A)\right)$$

Biased gradient estimator

$$A \leftarrow A - \eta \nabla \mathbf{v}^\top p_n(A)\mathbf{v} \qquad (\eta : \text{step-size})$$

- Even if the gradient estimate,i.e., $\nabla \mathbf{v}^\top p_n(A)\mathbf{v}$, is fast and accurate itself, there always exists a biased error:

$$\mathrm{E}\left[\nabla \mathbf{v}^\top p_n(A)\mathbf{v}\right] = \nabla \mathtt{tr}\left(p_n(A)\right) \neq \nabla \mathtt{tr}\left(f(A)\right)$$

$$f(x) - p_n(x) = \sum_{j=n+1}^{\infty} b_j T_j(x) \neq 0.$$

Optimizing Spectral-sums

Approximating spectral-sums

$$\texttt{tr}\,(f(A)) \approx \mathbf{v}^\top p_n(A)\mathbf{v}$$

Optimizing spectral-sums

$$\min_A \texttt{tr}\,(f(A))$$

Biased gradient estimator

$$A \leftarrow A - \eta \nabla \mathbf{v}^\top p_n(A)\mathbf{v} \qquad (\eta : \text{step-size})$$

- Even if the gradient estimate,i.e., $\nabla \mathbf{v}^\top p_n(A)\mathbf{v}$, is fast and accurate itself, there always exists a biased error:

$$\mathrm{E}\left[\nabla \mathbf{v}^\top p_n(A)\mathbf{v}\right] = \nabla \texttt{tr}\,(p_n(A)) \neq \nabla \texttt{tr}\,(f(A))$$

$$f(x) - p_n(x) = \textstyle\sum_{j=n+1}^{\infty} b_j T_j(x) \neq 0.$$

- The biased error might be accumulated over iterations of the gradient descent scheme (it is not an issue for approximating spectral-sums).

Optimizing Spectral-sums

Approximating spectral-sums

$$\mathtt{tr}\left(f(A)\right) \approx \mathbf{v}^\top p_n(A)\mathbf{v}$$

Optimizing spectral-sums

$$\min_A \mathtt{tr}\left(f(A)\right)$$

Biased gradient estimator

$$A \leftarrow A - \eta \nabla \mathbf{v}^\top p_n(A)\mathbf{v} \qquad (\eta : \text{step-size})$$

- Even if the gradient estimate,i.e., $\nabla \mathbf{v}^\top p_n(A)\mathbf{v}$, is fast and accurate itself, there always exists a biased error:

$$\mathrm{E}\left[\nabla \mathbf{v}^\top p_n(A)\mathbf{v}\right] = \nabla \mathtt{tr}\left(p_n(A)\right) \neq \nabla \mathtt{tr}\left(f(A)\right)$$

$$f(x) - p_n(x) = \textstyle\sum_{j=n+1}^{\infty} b_j T_j(x) \neq 0.$$

- The biased error might be accumulated over iterations of the gradient descent scheme (it is not an issue for approximating spectral-sums).
- How can we design an unbiased estimator?

Randomized Chebyshev Expansion for Unbiasedness

The original Chebyshev expansion uses deterministic polynomial degree:

$$f(x) = \sum_{j=0}^{\infty} b_j T_j(x), \qquad p_n(x) := \sum_{j=0}^{n} b_j T_j(x).$$

Randomized Chebyshev Expansion for Unbiasedness

The original Chebyshev expansion uses deterministic polynomial degree:

$$f(x) = \sum_{j=0}^{\infty} b_j T_j(x), \qquad p_n(x) := \sum_{j=0}^{n} b_j T_j(x).$$

Unbiased polynomial approximation

We now randomly sample degree n with probability q_n and define

$$\widehat{p}_n(x) := \sum_{j=0}^{n} \frac{b_j}{1 - \sum_{i=0}^{j-1} q_i} T_j(x).$$

Randomized Chebyshev Expansion for Unbiasedness

The original Chebyshev expansion uses deterministic polynomial degree:

$$f(x) = \sum_{j=0}^{\infty} b_j T_j(x), \qquad p_n(x) := \sum_{j=0}^{n} b_j T_j(x).$$

Unbiased polynomial approximation

We now randomly sample degree n with probability q_n and define

$$\widehat{p}_n(x) := \sum_{j=0}^{n} \frac{b_j}{1 - \sum_{i=0}^{j-1} q_i} T_j(x).$$

$$\mathrm{E}\left[\widehat{p}_n(x)\right] = \sum_{n=0}^{\infty} q_n \widehat{p}_n(x) = \sum_{j=0}^{\infty} \left(\sum_{n=j}^{\infty} q_n \right) \frac{b_j}{1 - \sum_{i=0}^{j-1} q_i} T_j(x) = f(x)$$

$$\mathrm{E}\left[\nabla \widehat{p}_n(x)\right] = \nabla \mathrm{E}\left[\widehat{p}_n(x)\right] = \nabla f(x)$$

Randomized Chebyshev Expansion for Unbiasedness

The original Chebyshev expansion uses deterministic polynomial degree:

$$f(x) = \sum_{j=0}^{\infty} b_j T_j(x), \qquad p_n(x) := \sum_{j=0}^{n} b_j T_j(x).$$

Unbiased polynomial approximation

We now randomly sample degree n with probability q_n and define

$$\widehat{p}_n(x) := \sum_{j=0}^{n} \frac{b_j}{1 - \sum_{i=0}^{j-1} q_i} T_j(x).$$

- This becomes an unbiased estimator: $\mathrm{E}\left[\widehat{p}_n(x)\right] = f(x), \mathrm{E}\left[\nabla \widehat{p}_n(x)\right] = f'(x)$.

Randomized Chebyshev Expansion for Unbiasedness

The original Chebyshev expansion uses deterministic polynomial degree:

$$f(x) = \sum_{j=0}^{\infty} b_j T_j(x), \qquad p_n(x) := \sum_{j=0}^{n} b_j T_j(x).$$

Unbiased polynomial approximation

We now randomly sample degree n with probability q_n and define

$$\widehat{p}_n(x) := \sum_{j=0}^{n} \frac{b_j}{1 - \sum_{i=0}^{j-1} q_i} T_j(x).$$

- This becomes an unbiased estimator: $\mathrm{E}\left[\widehat{p}_n(x)\right] = f(x), \mathrm{E}\left[\nabla \widehat{p}_n(x)\right] = f'(x)$.
- For optimizing spectral-sums, we can use $A \leftarrow A - \eta \, \nabla \mathbf{v}^\top \widehat{p}_n(A)\mathbf{v}$.

Randomized Chebyshev Expansion for Unbiasedness

The original Chebyshev expansion uses deterministic polynomial degree:

$$f(x) = \sum_{j=0}^{\infty} b_j T_j(x), \qquad p_n(x) := \sum_{j=0}^{n} b_j T_j(x).$$

Unbiased polynomial approximation

We now randomly sample degree n with probability q_n and define

$$\widehat{p}_n(x) := \sum_{j=0}^{n} \frac{b_j}{1 - \sum_{i=0}^{j-1} q_i} T_j(x).$$

- This becomes an unbiased estimator: $\mathrm{E}\left[\widehat{p}_n(x)\right] = f(x), \mathrm{E}\left[\nabla \widehat{p}_n(x)\right] = f'(x)$.
- For optimizing spectral-sums, we can use $A \leftarrow A - \eta \, \nabla \mathbf{v}^\top \widehat{p}_n(A) \mathbf{v}$.
- The unbiasedness holds for any distribution, but for optimization, an estimator with small variance guarantees fast convergence to the optimal.

Randomized Chebyshev Expansion for Unbiasedness

The original Chebyshev expansion uses deterministic polynomial degree:

$$f(x) = \sum_{j=0}^{\infty} b_j T_j(x), \qquad p_n(x) := \sum_{j=0}^{n} b_j T_j(x).$$

Unbiased polynomial approximation

We now randomly sample degree n with probability q_n and define

$$\widehat{p}_n(x) := \sum_{j=0}^{n} \frac{b_j}{1 - \sum_{i=0}^{j-1} q_i} T_j(x).$$

- This becomes an unbiased estimator: $\mathrm{E}\left[\widehat{p}_n(x)\right] = f(x), \mathrm{E}\left[\nabla \widehat{p}_n(x)\right] = f'(x)$.
- For optimizing spectral-sums, we can use $A \leftarrow A - \eta\, \nabla \mathbf{v}^\top \widehat{p}_n(A) \mathbf{v}$.
- The unbiasedness holds for any distribution, but for optimization, an estimator with small variance guarantees fast convergence to the optimal.
- How can we obtain a distribution with small variance?

Optimal Degree Distribution for Unbiasedness

We define the Chebyshev weighted variance of our estimator as

$$\mathrm{Var}\left[\widehat{p}_n\right] := \mathrm{E}\left[\int_{-1}^{1} \frac{\left(\widehat{p}_n(x) - f(x)\right)^2}{\sqrt{1-x^2}} dx\right]. \qquad (1)$$

Optimal Degree Distribution for Unbiasedness

We define the Chebyshev weighted variance of our estimator as

$$\mathrm{Var}\left[\widehat{p}_n\right] := \mathrm{E}\left[\int_{-1}^{1} \frac{\left(\widehat{p}_n(x) - f(x)\right)^2}{\sqrt{1-x^2}} dx\right]. \tag{1}$$

Problem

For optimizing spectral-sums, we aim to minimize the variance of unbiased gradient estimator when the expected degree is given by N:

$$\min_{\{q_n : n \geq 0\}} \mathrm{Var}\left[\widehat{p}_n\right] \qquad \text{s.t.} \qquad \mathrm{E}\left[n\right] = N$$

Optimal Degree Distribution for Unbiasedness

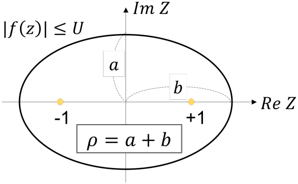
We define the Chebyshev weighted variance of our estimator as

$$\mathrm{Var}\left[\widehat{p}_n\right] := \mathrm{E}\left[\int_{-1}^{1} \frac{\left(\widehat{p}_n(x) - f(x)\right)^2}{\sqrt{1-x^2}} dx\right]. \tag{1}$$

Theorem (Han, Avron and Shin, 2018)

Suppose analytic function f is $|f(z)| \le U$ and bounded by ellipse with foci $+1, -1$ and sum of major and minor semi-axis lengths equals to $\rho > 1$. Let $k = \min\{N, \left\lfloor \frac{\rho}{\rho-1} \right\rfloor\}$, then the distribution that minimizes the variance (1) *is:*

$$q_n^* = \begin{cases} 0 & \text{for } n < N-k \\ 1 - \dfrac{k(\rho-1)}{\rho} & \text{for } n = N-k \\ \dfrac{k(\rho-1)^2}{\rho^{n+1}} & \text{for } n > N-k \end{cases}$$

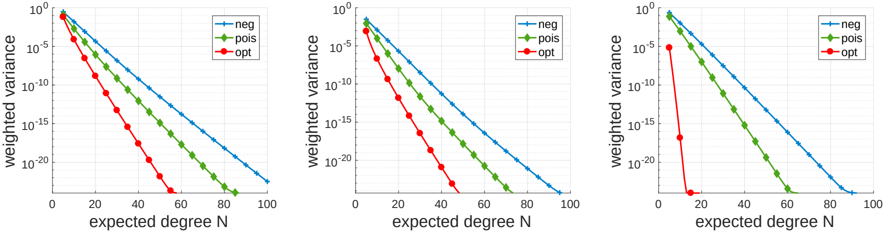


Optimal Degree Distribution for Unbiasedness

We define the Chebyshev weighted variance of our estimator as

$$\mathrm{Var}\left[\widehat{p}_n\right] := \mathrm{E}\left[\int_{-1}^{1} \frac{\left(\widehat{p}_n(x) - f(x)\right)^2}{\sqrt{1-x^2}} dx\right]. \quad (1)$$

Synthetic evaluation



(a) $f(x) = \log x, x \in [0.05, 0.95]$ (b) $f(x) = x^{0.5}, x \in [0.05, 0.95]$ (c) $f(x) = \exp(x), x \in [-1, 1]$

Figure: Chebyshev weighted variance with negative binomial (neg), Poisson (pois) and our distribution (opt) with the mean 10

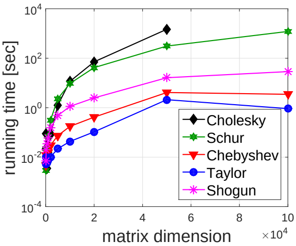
The optimal distribution has the smallest variance among all distributions.

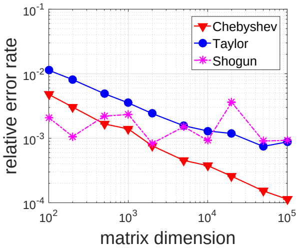
Outline

Experiments for Approximation

Approximation of log-determinant for random sparse matrices

- **Methods:** Cholesky decomposition, Schur complement, Shogun machine learning library [1], Taylor expansion and Chebyshev expansion (our method)
- Cholesky and Schur methods compute log-determinant exactly.
- Our proposed method runs much faster than other methods except Taylor's one. E.g., It takes about 130 seconds for matrix with dimension 10^7.
- Chebyshev is superior in accuracy compared to both Taylor and Shogun. E.g., Approximation error is less than 0.1% for $m = 50$ and $n = 25$.





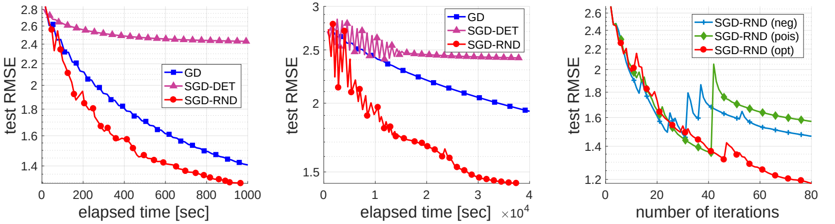
[1] Shogun (`http://shogun-toolbox.org`) provides highly optimized log-determinant.

Experiments for Optimization: Matrix Completion

Schatten norm minimization for matrix completion under MovieLens 1M/10M dataset

$$\min_{L \in \mathbb{R}^{d_1 \times d_2}} \mathtt{tr}\left(\sqrt{L^\top L}\right) + \lambda \left\| \mathcal{P}(L) - \mathcal{P}(B) \right\|_F^2$$

- **Methods**: exact gradient descent (GD), deterministic Chebyshev expansion (SGD-DET), randomized Chebyshev expansion (SGD-RND, our method)
- SGD-RND has even less biased error than that with SGD-DET.
- SGD-RND shows the best peformance with up to 5 times of speedup.
- Comparing with other distributions, the optimal one shows stable convergence.



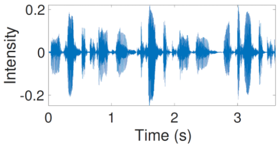
(a) MovieLens 1M (b) MovieLens 10M (c) Varying degree distributions

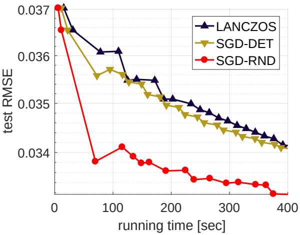
Experiments for Optimization: Learning Gaussian Process

Log-determinant optimization for Gaussian process under natural sound modeling

$$\boxed{\min_\theta -\log\det K(X,\theta) + \mathtt{tr}\left(\mathbf{y}^\top K(X,\theta)^{-1}\mathbf{y}\right)}$$

- The goal is to find hyperparameter θ given training data $(X,\mathbf{y})$ contains $d=35,000$ and test 391 points. $K\in\mathbb{R}^{d\times d}$ is RBF kernel matrix of θ and X.
- **Methods**: deterministic Chebyshev expansion (SGD-DET), randomized approximation (SGD-RND) and Lanczos method (LANCZOS, Dong et al. (2017))
- SGD-RND converges even faster than LANCZOS up to 8 times because LANCZ is also biased estimator.





Conclusion

1. We develop a fast algorithm for approximating spectral-sums with Chebyshev expansion and trace estimator via matrix-vector multiplication.

2. We develop an unbiased gradient estimator for optimizing spectral-sums which is applicable to stochastic gradient descent. We find the optimal degree distribution whose variance achieves the minimum.

3. Our algorithm takes 130 seconds with $< 0.1\%$ error for approximating spectral-sums of matrices with dimension 10^7. For optimization, ours runs up-to 8 times faster than the state-of-the-art method in Gaussian process.

4. Our method for approximating and optimizing spectral-sums can be used in many scientific and practical applications.

Conclusion

1. We develop a fast algorithm for approximating spectral-sums with Chebyshev expansion and trace estimator via matrix-vector multiplication.
2. We develop an unbiased gradient estimator for optimizing spectral-sums which is applicable to stochastic gradient descent. We find the optimal degree distribution whose variance achieves the minimum.
3. Our algorithm takes 130 seconds with $< 0.1\%$ error for approximating spectral-sums of matrices with dimension 10^7. For optimization, ours runs up-to 8 times faster than the state-of-the-art method in Gaussian process.
4. Our method for approximating and optimizing spectral-sums can be used in many scientific and practical applications.

Thank you for your attention !

Outline

Dong, K., Eriksson, D., Nickisch, H., Bindel, D., and Wilson, A. G. (2017). Scalable log determinants for gaussian process kernel learning. In *Advances in Neural Information Processing Systems*, pages 6330–6340.