Optimal mini-batch size for stochastic variance-reduced methods

Nidham Gazagnadou[a], Robert M. Gower

Télécom ParisTech, Paris, France

PGMO days 2018 - November 21, 2018

[a]This work was supported by grants from Région Ile-de-France

Outline

1. Variance-reduced methods
2. Upper bounding the expected smoothness constant
3. Optimal mini-batch size
4. Numerical experiments
5. Conclusion

Talk Overview

Variance-reduced methods

Upper bounding the expected smoothness constant

Optimal mini-batch size

Numerical experiments

Conclusion

Variance-reduced methods

Usual training problem

- **"Big" Data**
  - **n**: number of observations
  - **d**: dimension of each observation

Usual training problem

- **"Big" Data**
  - **n**: number of observations
  - **d**: dimension of each observation

- **Empirical Risk Minimization (ERM)**

$$\text{find } w^* \in \arg\min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} f_i(w) = \frac{1}{n} \sum_{i=1}^{n} \phi_i(a_i^\top w) + \frac{\lambda}{2} \|w\|^2$$

where

- $a_i \in \mathbb{R}^d$: feature vectors (input)
- $\lambda > 0$: regularization parameter
- $w \in \mathbb{R}^d$: parameter/model

Usual training problem

- **"Big" Data**
  - **n**: number of observations
  - **d**: dimension of each observation

- **Empirical Risk Minimization (ERM)**

$$\text{find } w^* \in \arg\min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} f_i(w) = \frac{1}{n} \sum_{i=1}^{n} \phi_i(a_i^\top w) + \frac{\lambda}{2} \|w\|^2$$

  where
  - $a_i \in \mathbb{R}^d$: feature vectors (input)
  - $\lambda > 0$: regularization parameter
  - $w \in \mathbb{R}^d$: parameter/model

- **Covered problems**
  - Ridge regression: $\phi_i(z) = \frac{1}{2}(z - y_i)^2$
  - Logistic regression: $\phi_i(z) = \log(1 + e^{-y_i z})$

  with $y_i \in \mathbb{R}$ or $\{-1, 1\}$ the labels (output)

Solving the ERM problem

- **Goal**

$$\text{find } w^* \in \arg\min_{w \in \mathbb{R}^d} f(w) = \frac{1}{n} \sum_{i=1}^{n} f_i(w)$$

[1] Robbins and Monro, 1951b

Solving the ERM problem

- **Goal**

$$\text{find } w^* \in \arg\min_{w \in \mathbb{R}^d} f(w) = \frac{1}{n} \sum_{i=1}^{n} f_i(w)$$

- **Gradient descent (GD)**

$$w^{k+1} = w^k - \frac{\gamma}{n} \left(\sum_{i=1}^{n} \nabla f_i(w^k) \right)$$

with γ the step size.

[1] Robbins and Monro, 1951b

Solving the ERM problem

- **Goal**

$$\text{find } w^* \in \arg\min_{w \in \mathbb{R}^d} f(w) = \frac{1}{n} \sum_{i=1}^{n} f_i(w)$$

- **Gradient descent (GD)**

$$w^{k+1} = w^k - \frac{\gamma}{n} \left(\sum_{i=1}^{n} \nabla f_i(w^k) \right)$$

with γ the step size.

Problem: $\mathcal{O}(nd)$ computations / iteration

[1] Robbins and Monro, 1951b

Solving the ERM problem

- **Goal**

$$\text{find } w^* \in \arg\min_{w \in \mathbb{R}^d} f(w) = \frac{1}{n} \sum_{i=1}^{n} f_i(w)$$

- **Gradient descent (GD)**

$$w^{k+1} = w^k - \frac{\gamma}{n} \left(\sum_{i=1}^{n} \nabla f_i(w^k) \right)$$

with γ the step size.

Problem: $\mathcal{O}(nd)$ computations / iteration

- **Stochastic Gradient Descent[1] (SGD)**

$$w^{k+1} = w^k - \gamma_k \nabla f_i(w^k)$$

with $(\gamma_k)_k$ a step size sequence.

[1] Robbins and Monro, 1951b

Solving the ERM problem

- **Goal**

$$\text{find } w^* \in \arg\min_{w \in \mathbb{R}^d} f(w) = \frac{1}{n} \sum_{i=1}^{n} f_i(w)$$

- **Gradient descent (GD)**

$$w^{k+1} = w^k - \frac{\gamma}{n} \left(\sum_{i=1}^{n} \nabla f_i(w^k) \right)$$

with γ the step size.

Problem: $\mathcal{O}(nd)$ computations / iteration

- **Stochastic Gradient Descent[1] (SGD)**

$$w^{k+1} = w^k - \gamma_k \nabla f_i(w^k)$$

with $(\gamma_k)_k$ a step size sequence.

Problem: need to tune the step size sequence

[1] Robbins and Monro, 1951b

Solving the ERM problem

- **Goal**

$$\text{find } w^* \in \arg\min_{w \in \mathbb{R}^d} f(w) = \frac{1}{n} \sum_{i=1}^{n} f_i(w)$$

- **Gradient descent (GD)**

$$w^{k+1} = w^k - \frac{\gamma}{n} \left(\sum_{i=1}^{n} \nabla f_i(w^k) \right)$$

with γ the step size.

Problem: $\mathcal{O}(nd)$ computations / iteration

- **Stochastic Gradient Descent**[1] **(SGD)**

$$w^{k+1} = w^k - \gamma_k \nabla f_i(w^k)$$

with $(\gamma_k)_k$ a step size sequence.

Problem: need to tune the step size sequence

$\rightarrow$ **Build better estimates of the gradient**

[1] Robbins and Monro, 1951b

Reformulation of the ERM

- **Sampling vector**

  Let $v \in \mathbb{R}^n$, with distribution $\mathcal{D}$ s.t.

$$\mathbb{E}_{\mathcal{D}}[v] = \frac{1}{n}\mathbb{1}$$

Reformulation of the ERM

- **Sampling vector**

  Let $v \in \mathbb{R}^n$, with distribution $\mathcal{D}$ s.t.

$$\mathbb{E}_{\mathcal{D}}[v] = \frac{1}{n}\mathbb{1}$$

- **Subsampled function**

$$f_v(w) \stackrel{\text{def}}{=} \sum_{i=1}^{n} f_i(w).v_i = \langle F(w), v \rangle$$

with $F(w) \stackrel{\text{def}}{=} (f_1(w), \ldots, f_n(w))^\top$.

Reformulation of the ERM

- **Sampling vector**

  Let $v \in \mathbb{R}^n$, with distribution $\mathcal{D}$ s.t.

$$\mathbb{E}_{\mathcal{D}}[v] = \frac{1}{n}\mathbb{1}$$

- **Subsampled function**

$$f_v(w) \stackrel{\text{def}}{=} \sum_{i=1}^{n} f_i(w).v_i = \langle F(w), v \rangle$$

  with $F(w) \stackrel{\text{def}}{=} (f_1(w), \ldots, f_n(w))^\top$.

- **Unbiased estimates**

$$\mathbb{E}_{\mathcal{D}}[f_v(w)] = \langle F(w), \mathbb{E}_{\mathcal{D}}[v] \rangle = f(w)$$

$$\mathbb{E}_{\mathcal{D}}[\nabla f_v(w)] = \langle \nabla F(w), \mathbb{E}_{\mathcal{D}}[v] \rangle = \nabla f(w)$$

Reformulation of the ERM

- **Sampling vector**

  Let $v \in \mathbb{R}^n$, with distribution $\mathcal{D}$ s.t.

$$\mathbb{E}_{\mathcal{D}}[v] = \frac{1}{n}\mathbb{1}$$

- **Subsampled function**

$$f_v(w) \stackrel{\text{def}}{=} \sum_{i=1}^{n} f_i(w).v_i = \langle F(w), v \rangle$$

  with $F(w) \stackrel{\text{def}}{=} (f_1(w), \ldots, f_n(w))^\top$.

- **Unbiased estimates**

$$\mathbb{E}_{\mathcal{D}}[f_v(w)] = \langle F(w), \mathbb{E}_{\mathcal{D}}[v] \rangle = f(w)$$

$$\mathbb{E}_{\mathcal{D}}[\nabla f_v(w)] = \langle \nabla F(w), \mathbb{E}_{\mathcal{D}}[v] \rangle = \nabla f(w)$$

$\rightarrow$ **Motivates using SGD**: $w^{k+1} = w^k - \gamma_k \nabla f_v(w^k)$

Controlled stochastic reformulation of the ERM

- **Adding a control**

$$\text{find } w^* \in \arg\min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} f_i(w) = \mathbb{E}_{\mathcal{D}}\Big[f_v(w) \underbrace{- z_v(w) + \mathbb{E}_{\mathcal{D}}\left[z_v(w)\right]}_{\text{unbiased correction term}}\Big]$$

with $z_v(\cdot)$ a random function.

Controlled stochastic reformulation of the ERM

- **Adding a control**

$$\text{find } w^* \in \arg\min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} f_i(w) = \mathbb{E}_{\mathcal{D}}\Big[f_v(w) \underbrace{- z_v(w) + \mathbb{E}_{\mathcal{D}}\left[z_v(w)\right]}_{\text{unbiased correction term}}\Big]$$

with $z_v(\cdot)$ a random function.

- **New gradient estimator**

$$\boldsymbol{g_v(w)} \stackrel{\text{def}}{=} \nabla f_v(w) - \nabla z_v(w) + \mathbb{E}_{\mathcal{D}}\left[\nabla z_v(w)\right]$$

Iteration: $w^{k+1} = w^k - \gamma \boldsymbol{g_v(w^k)}$.

Stochastic variance-reduced methods

- **Recovered algorithms**
  - **SGD**
  - **SVRG**[2]
  - **SAGA**[3] $\begin{cases} f_v(w) = \frac{1}{n}\langle F(w), v\rangle \\ z_v(w) = \frac{1}{n}\langle \underbrace{\boldsymbol{J}^\top w}_{\text{linear estimation of } F(w)}, v\rangle \end{cases}$

    with $\boldsymbol{J}$ a matrix of parameters in $R^{d\times n}$.

[2] Johnson and Zhang, 2013

[3] Defazio, Bach and Lacoste-Julien, 2014b

Stochastic variance-reduced methods

- **Recovered algorithms**
  - **SGD**
  - **SVRG**[2]
  - **SAGA**[3] $\begin{cases} f_v(w) = \frac{1}{n}\langle F(w), v\rangle \\ z_v(w) = \frac{1}{n}\langle \underbrace{\boldsymbol{J}^\top w}_{\text{linear estimation of } F(w)}, v\rangle \end{cases}$

    with $\boldsymbol{J}$ a matrix of parameters in $R^{d\times n}$.
- **Variance term**

  $\mathbb{E}_{\mathcal{D}}\left[\|g_v(w) - \nabla f(w)\|_2^2\right]$ low for $\boldsymbol{J} \approx \nabla F(w)$ (true Jacobian)

  We want simultaneously
  - $w^k \to w^*$
  - $J^k \to \nabla F(w^k)$

[2] Johnson and Zhang, 2013

[3] Defazio, Bach and Lacoste-Julien, 2014b

Advantages of stochastic variance-reduced methods

- **Summary**

$$w^{k+1} = w^k - \gamma \mathbf{g_v}(\mathbf{w^k})$$

Such that

  - Unbiased gradient: $\mathbb{E}\left[g_v(w^k)\right] = \nabla f(w^k)$
  - Decreasing variance: $\mathbb{E}\left[\left\|g_v(w^k) - \nabla f(w^k)\right\|_2^2\right] \xrightarrow[w^k \to w^*]{} 0$

Advantages of stochastic variance-reduced methods

- **Summary**

$$w^{k+1} = w^k - \gamma \mathbf{g_v}(\mathbf{w^k})$$

Such that

  - Unbiased gradient: $\mathbb{E}\left[g_v(w^k)\right] = \nabla f(w^k)$
  - Decreasing variance: $\mathbb{E}\left[\left\|g_v(w^k) - \nabla f(w^k)\right\|_2^2\right] \xrightarrow[w^k \to w^*]{} 0$

- **Advantages**
  - **No need to tune $(\gamma_k)_k$**

Advantages of stochastic variance-reduced methods

- **Summary**

$$w^{k+1} = w^k - \gamma \mathbf{g_v}(\mathbf{w^k})$$

Such that

  - Unbiased gradient: $\mathbb{E}\left[g_v(w^k)\right] = \nabla f(w^k)$
  - Decreasing variance: $\mathbb{E}\left[\left\|g_v(w^k) - \nabla f(w^k)\right\|_2^2\right] \xrightarrow[w^k \to w^*]{} 0$

- **Advantages**

  - **No need to tune $(\gamma_k)_k$**
  - No "bounded gradient" assumption such as $\mathbb{E}\left[\left\|\nabla f_v(w^k)\right\|_2^2\right] \leq B$

Advantages of stochastic variance-reduced methods

- **Summary**

$$w^{k+1} = w^k - \gamma \mathbf{g_v}(\mathbf{w^k})$$

Such that

  - Unbiased gradient: $\mathbb{E}\left[g_v(w^k)\right] = \nabla f(w^k)$
  - Decreasing variance: $\mathbb{E}\left[\left\|g_v(w^k) - \nabla f(w^k)\right\|_2^2\right] \xrightarrow[w^k \to w^*]{} 0$

- **Advantages**

  - **No need to tune $(\gamma_k)_k$**
  - No "bounded gradient" assumption such as $\mathbb{E}\left[\left\|\nabla f_v(w^k)\right\|_2^2\right] \leq B$
  - $f_v(w)$ determines the smoothness of the controlled stochastic reformulation ($z_v(w)$ linear in w)

Mini-batch SAGA

- **Mini-batching process**
  - Sample a mini-batch $B \subset [n] \stackrel{\text{def}}{=} \{1, \ldots, n\}$ s.t. $|B| = \tau$

$$v = \frac{1}{\tau} \sum_{i \in B} e_i$$

    where $(e_i)_{1 \leq i \leq n}$ being the basis vectors.
  - Compute τ individual stochastic gradients $\nabla f_i(w^k)$, $\forall i \in B$

$$\nabla f_v(w^k) = \frac{1}{\tau} \sum_{i \in B} \nabla f_i(w^k)$$

  - Update the Jacobian estimate

$$J_i^k = \nabla f_i(w^k), \quad \forall i \in B$$

Mini-batch SAGA

- **Mini-batching process**
  - Sample a mini-batch $B \subset [n] \stackrel{\text{def}}{=} \{1, \ldots, n\}$ s.t. $|B| = \tau$

$$v = \frac{1}{\tau} \sum_{i \in B} e_i$$

    where $(e_i)_{1 \leq i \leq n}$ being the basis vectors.
  - Compute τ individual stochastic gradients $\nabla f_i(w^k)$, $\forall i \in B$

$$\nabla f_v(w^k) = \frac{1}{\tau} \sum_{i \in B} \nabla f_i(w^k)$$

  - Update the Jacobian estimate

$$J_i^k = \nabla f_i(w^k), \quad \forall i \in B$$

- **Why is it useful?**
  - Works well in practice
  - Calculating several gradients per iteration is cheap

Mini-batch SAGA

- **Mini-batching process**
  - Sample a mini-batch $B \subset [n] \stackrel{\text{def}}{=} \{1, \ldots, n\}$ s.t. $|B| = \tau$

$$v = \frac{1}{\tau} \sum_{i \in B} e_i$$

    where $(e_i)_{1 \leq i \leq n}$ being the basis vectors.
  - Compute τ individual stochastic gradients $\nabla f_i(w^k)$, $\forall i \in B$

$$\nabla f_v(w^k) = \frac{1}{\tau} \sum_{i \in B} \nabla f_i(w^k)$$

  - Update the Jacobian estimate

$$J_i^k = \nabla f_i(w^k), \quad \forall i \in B$$

- **Why is it useful?**
  - Works well in practice
  - Calculating several gradients per iteration is cheap
- **What is the optimal mini-batch size?**

$$\rightarrow \textbf{find the "best" } \tau \textbf{ value}$$

Basis of our work

- **JacSketch convergence[4] theorem**

Definition (Stochastic Lyapunov function)

$$\Psi^k \stackrel{\text{def}}{=} \left\| w^k - w^* \right\|_2^2 + \frac{\gamma}{2\tau L_{\max}} \left\| \mathbf{J}^k - \nabla \mathbf{F}(w^*) \right\|_{\mathrm{F}}^2$$

- $\nabla \mathbf{F}(w) = [\nabla f_1(w), \ldots, \nabla f_n(w)] \in \mathbb{R}^{d \times n}$: Jacobian matrix
- $\left\{ w^k, \mathbf{J}^k \right\}_{k \geq 0}$ are the points and Jacobian estimated by *JacSketch*

[4] Gower, Richtárik and Bach, 2018

Basis of our work

- **JacSketch convergence[4] theorem**

Definition (Stochastic Lyapunov function)

$$\Psi^k \stackrel{\text{def}}{=} \left\|w^k - w^*\right\|_2^2 + \frac{\gamma}{2\tau L_{\max}} \left\|\mathbf{J}^k - \nabla \mathbf{F}(w^*)\right\|_{\mathrm{F}}^2$$

  - $\nabla \mathbf{F}(w) = [\nabla f_1(w), \ldots, \nabla f_n(w)] \in \mathbb{R}^{d \times n}$: Jacobian matrix
  - $\left\{w^k, \mathbf{J}^k\right\}_{k \geq 0}$ are the points and Jacobian estimated by *JacSketch*

- **Linear convergence in the iterates**

$$\mathbb{E}\left[\left\|w^k - w^*\right\|_2^2\right] \leq \rho^k . \Psi^0$$

with a convergence rate $0 < \rho < 1$.

[4] Gower, Richtárik and Bach, 2018

Methodology

- **Optimal mini-batch size**
  For a desired precision $\epsilon > 0$,

$$\text{find} \quad \tau^* \in \arg\min_{\tau \in [n]} K_{\text{total}}(\tau) = \tau K_{\text{iteration}}(\tau)$$

[5] Gower, Richtárik and Bach, 2018

Methodology

gradients per iteration

- **Optimal mini-batch size**

  For a desired precision $\epsilon > 0$,

$$\text{find} \quad \tau^* \in \arg\min_{\tau \in [n]} K_{\text{total}}(\tau) = \tau K_{\text{iteration}}(\tau)$$

[5] Gower, Richtárik and Bach, 2018

Methodology

- **Optimal mini-batch size**

  For a desired precision $\epsilon > 0$,

gradients per iteration

iterations to achieve $\mathbb{E}\left[\Psi^k\right] \leq \epsilon \Psi^0$

$$\text{find} \quad \tau^* \in \arg\min_{\tau \in [n]} K_{\text{total}}(\tau) = \tau K_{\text{iteration}}(\tau)$$

[5] Gower, Richtárik and Bach, 2018

Methodology

- **Optimal mini-batch size**

  For a desired precision $\epsilon > 0$,

$$\text{find} \quad \tau^* \in \underset{\tau \in [n]}{\arg\min}\, \underbrace{K_{\text{total}}(\tau)}_{\#\text{ iterations to achieve } \mathbb{E}\left[\Psi^k\right] \leq \epsilon \Psi^0} = \tau \underbrace{K_{\text{iteration}}(\tau)}_{\#\text{ gradients per iteration}}$$

- **Total complexity**[5]

$$K_{\text{total}}(\tau) = \max\left\{\frac{4\tau(\mathcal{L}_1 + \lambda)}{\mu}, n + \frac{n-\tau}{n-1}\frac{4(L_{\max} + \lambda)}{\mu}\right\}\log\left(\frac{1}{\epsilon}\right)$$

[5] Gower, Richtárik and Bach, 2018

Methodology

gradients per iteration

iterations to achieve $\mathbb{E}\left[\Psi^k\right] \leq \epsilon \Psi^0$

- **Optimal mini-batch size**

  For a desired precision $\epsilon > 0$,

$$\text{find} \quad \tau^* \in \arg\min_{\tau \in [n]} K_{\text{total}}(\tau) = \tau K_{\text{iteration}}(\tau)$$

- **Total complexity**[5]

$$K_{\text{total}}(\tau) = \max\left\{\frac{4\tau(\mathcal{L}_1 + \lambda)}{\mu}, n + \frac{n-\tau}{n-1}\frac{4(L_{\max} + \lambda)}{\mu}\right\} \log\left(\frac{1}{\epsilon}\right)$$

$\rightarrow$ **Here, what are $\mathcal{L}_1, L_{\max}, \mu$?**

[5] Gower, Richtárik and Bach, 2018

The most important object of the study

Assumption (Expected smoothness constant)

There is $\mathcal{L}_1 > 0$ *s.t.*

$$\mathbb{E}_{\mathcal{D}}\left[\left\|g_v(w^k) - g_v(w^*)\right\|_2^2\right] \leq 2\boldsymbol{\mathcal{L}_1}(f(w) - f(w^*)), \quad \forall w \in \mathbb{R}^d$$

The most important object of the study

Assumption (Expected smoothness constant)

There is $\mathcal{L}_1 > 0$ *s.t.*

$$\mathbb{E}_{\mathcal{D}}\left[\left\|g_v(w^k) - g_v(w^*)\right\|_2^2\right] \leq 2\boldsymbol{\mathcal{L}_1}(f(w) - f(w^*)), \quad \forall w \in \mathbb{R}^d$$

- $\boldsymbol{\mathcal{L}_1}$ **embodies the iteration complexity**, i.e. the rate of convergence

The most important object of the study

Assumption (Expected smoothness constant)

There is $\mathcal{L}_1 > 0$ *s.t.*

$$\mathbb{E}_{\mathcal{D}}\left[\left\|g_v(w^k) - g_v(w^*)\right\|_2^2\right] \leq 2\boldsymbol{\mathcal{L}_1}(f(w) - f(w^*)), \quad \forall w \in \mathbb{R}^d$$

- $\boldsymbol{\mathcal{L}_1}$ **embodies the iteration complexity**, i.e. the rate of convergence
- It also gives a **new bound for the step size** γ

The most important object of the study

Assumption (Expected smoothness constant)

There is $\mathcal{L}_1 > 0$ s.t.

$$\mathbb{E}_{\mathcal{D}}\left[\left\|g_v(w^k) - g_v(w^*)\right\|_2^2\right] \leq 2\boldsymbol{\mathcal{L}_1}(f(w) - f(w^*)), \quad \forall w \in \mathbb{R}^d$$

- **$\mathcal{L}_1$ embodies the iteration complexity**, i.e. the rate of convergence
- It also gives a **new bound for the step size γ**
- It is a common theoretical object ruling several algorithms (SGD, SAGA, SVRG...)

Basic assumptions

Assumption (i.i.d. data)

n **i.i.d.** *observations:* $(a_i, y_i) \in \mathbb{R}^d \times \mathbb{R}$ *or* $\mathbb{R}^d \times \{-1, 1\}$

Assumption (f is μ-strongly convex)

$$f(y) \geq f(x) + \nabla f(x)^\top (y - x) + \frac{\boldsymbol{\mu}}{2} \|y - x\|_2^2$$

Assumption (f is L-smooth)

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq \boldsymbol{L} \|x - y\|_2$$

Additional assumption and notations

Assumption (Bounded second derivatives)

$\exists U \in \mathbb{R}$ *s.t.* $\forall x \in \mathbb{R}, \forall i \in [n]$

$$\phi_i''(x) \leq U$$

Definition (Subsample/batch function)

$$f_B(w) \stackrel{\text{def}}{=} \frac{1}{\tau} \sum_{i \in B} \phi_i(a_i^\top w) + \frac{\lambda}{2} \|w\|_2^2 \quad \forall B \subset [n]$$

$$\implies \nabla^2 f_B(w) = \frac{1}{\tau} \sum_{i \in B} \phi_i''(a_i^\top w) a_i a_i^\top + \lambda \mathbf{I}_d \preceq \frac{U}{\tau} \mathbf{A}_B \mathbf{A}_B^\top + \lambda \mathbf{I}_d$$

$$\text{with } \mathbf{A}_B = \underbrace{\begin{bmatrix} \vdots & & \vdots \\ a_{i_1} & \dots & a_{i_\tau} \\ \vdots & & \vdots \end{bmatrix}}_{\tau} \Bigg\} d$$

The jungle of smoothness constants

Definition (Subsample smoothness constant)

$$L_B \stackrel{\text{def}}{=} \frac{U}{|B|}\lambda_{\max}\left(\sum_{i\in B} a_i a_i^\top\right) = \frac{U}{|B|}\lambda_{\max}\left(\mathbf{A}_B\mathbf{A}_B^\top\right)$$

f_B is L_B–smooth.

- $B = [n] \implies L$: smoothness constant of f
- $B = \{i\} \implies L_i$: smoothness constant of f_i

Definition (Maximum smoothness constant)

$$L_{\max} \stackrel{\text{def}}{=} \max_{i\in[n]} L_i$$

Definition (Average smoothness constant)

$$\bar{L} \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} L_i$$

Key concept

Definition (τ-nice sampling)

S (a random set-valued mapping) is a τ-nice sampling if

$$\mathbb{P}\left[S = B\right] = \frac{1}{\binom{n}{\tau}} \quad \forall B \in \mathrm{supp}(S)$$

where $\mathrm{supp}(S) \stackrel{\text{def}}{=} \{B \subset [n] \ : \ |B| = \tau\}$.

Definition (Expected smoothness constant)

For a given sampling S,

$$\mathcal{L}_1 \stackrel{\text{def}}{=} \frac{1}{c_1} \max_{i=1,\ldots,n} \left\{ \sum_{B \in \mathrm{supp}(S) \,|\, i \in B} L_B \right\}$$

with $c_1 \stackrel{\text{def}}{=} |\{B \in \mathrm{supp}(S) \ : \ i \in B\}|$.

Key concept

Definition (τ-nice sampling)

S (a random set-valued mapping) is a τ-nice sampling if

$$\mathbb{P}\left[S = B\right] = \frac{1}{\binom{n}{\tau}} \quad \forall B \in \mathrm{supp}(S)$$

where $\mathrm{supp}(S) \stackrel{\text{def}}{=} \{B \subset [n] \ : \ |B| = \tau\}$.

Definition (Expected smoothness constant)

For a given sampling S,

$$\mathcal{L}_1 \stackrel{\text{def}}{=} \frac{1}{c_1} \max_{i=1,\ldots,n} \left\{ \sum_{B \in \mathrm{supp}(S) \,|\, i \in B} L_B \right\}$$

with $c_1 \stackrel{\text{def}}{=} |\{B \in \mathrm{supp}(S) \ : \ i \in B\}|$.

Here: $c_1 = \binom{n-1}{\tau-1}$

Key concept

Definition (τ-nice sampling)

S (a random set-valued mapping) is a τ-nice sampling if

$$\mathbb{P}\left[S = B\right] = \frac{1}{\binom{n}{\tau}} \quad \forall B \in \mathrm{supp}(S)$$

where $\mathrm{supp}(S) \stackrel{\text{def}}{=} \{B \subset [n] \ : \ |B| = \tau\}$.

Definition (Expected smoothness constant)

For a given sampling S,

$$\mathcal{L}_1 \stackrel{\text{def}}{=} \frac{1}{c_1} \max_{i=1,\ldots,n} \left\{ \sum_{B \in \mathrm{supp}(S) \,|\, i \in B} L_B \right\}$$

with $c_1 \stackrel{\text{def}}{=} |\{B \in \mathrm{supp}(S) \ : \ i \in B\}|$.

Here: $c_1 = \binom{n-1}{\tau-1}$

$\rightarrow$ **Problem: Calculating $\mathcal{L}_1$ is intractable for large n**

Talk Overview

Variance-reduced methods

Upper bounding the expected smoothness constant

Optimal mini-batch size

Numerical experiments

Conclusion

Upper bounding the expected smoothness constant

Extreme values of $\mathcal{L}_1$

- **Recall**: $S = \tau$-nice sampling

$$\mathcal{L}_1 = \frac{1}{\binom{n-1}{\tau-1}} \max_{i=1,\ldots,n} \left\{ \sum_{B \in \mathrm{supp}(S) \mid i \in B} L_B \right\}$$

Extreme values of $\mathcal{L}_1$

- **Recall**: $S = \tau$-nice sampling

$$\mathcal{L}_1 = \frac{1}{\binom{n-1}{\tau-1}} \max_{i=1,\ldots,n} \left\{ \sum_{B \in \mathrm{supp}(S) \mid i \in B} L_B \right\}$$

- If $\boldsymbol{\tau = 1}$

Extreme values of $\mathcal{L}_1$

- **Recall**: $S = \tau$-nice sampling

$$\mathcal{L}_1 = \frac{1}{\binom{n-1}{0}} \max_{i=1,\ldots,n} \left\{ \sum_{B \in \mathrm{supp}(S) \mid i \in B} L_B \right\}$$

- If $\boldsymbol{\tau = 1}$
  - Recovered algorithm: SAGA

Extreme values of $\mathcal{L}_1$

- **Recall**: $S = \tau$-nice sampling

$$\mathcal{L}_1 = \max_{i=1,\ldots,n} \left\{ \sum_{B \in \{\{1\},\ldots,\{n\}\} \mid i \in B} L_B \right\}$$

- If $\boldsymbol{\tau = 1}$
  - Recovered algorithm: SAGA
  - $\mathrm{supp}(S) = \{\{1\}, \ldots, \{n\}\}$

Extreme values of $\mathcal{L}_1$

- **Recall**: $S = \tau$-nice sampling

$$\mathcal{L}_1 = \max_{i=1,\ldots,n} L_i$$

- If $\boldsymbol{\tau = 1}$
  - Recovered algorithm: SAGA
  - $\mathrm{supp}(S) = \{\{1\}, \ldots, \{n\}\}$

Extreme values of $\mathcal{L}_1$

- **Recall**: $S = \tau$-nice sampling

$$\mathcal{L}_1 = \frac{1}{\binom{n-1}{\tau-1}} \max_{i=1,\ldots,n} \left\{ \sum_{B \in \mathrm{supp}(S) \mid i \in B} L_B \right\}$$

- If $\boldsymbol{\tau = 1}$
  - Recovered algorithm: SAGA
  - $\mathrm{supp}(S) = \{\{1\}, \ldots, \{n\}\}$

$$\boxed{\mathcal{L}_1 = L_{\mathbf{max}}}$$

Extreme values of $\mathcal{L}_1$

- **Recall**: $S = \tau$-nice sampling

$$\mathcal{L}_1 = \frac{1}{\binom{n-1}{\tau-1}} \max_{i=1,\ldots,n} \left\{ \sum_{B \in \mathrm{supp}(S) \mid i \in B} L_B \right\}$$

- If $\boldsymbol{\tau = 1}$
  - Recovered algorithm: SAGA
  - $\mathrm{supp}(S) = \{\{1\}, \ldots, \{n\}\}$

$$\boxed{\mathcal{L}_1 = L_{\mathbf{max}}}$$

- If $\boldsymbol{\tau = n}$

Extreme values of $\mathcal{L}_1$

- **Recall**: $S = \tau$-nice sampling

$$\mathcal{L}_1 = \frac{1}{\binom{n-1}{n-1}} \max_{i=1,\ldots,n} \left\{ \sum_{B \in \mathrm{supp}(S) \mid i \in B} L_B \right\}$$

- If $\boldsymbol{\tau = 1}$
  - Recovered algorithm: SAGA
  - $\mathrm{supp}(S) = \{\{1\}, \ldots, \{n\}\}$

$$\boxed{\mathcal{L}_1 = L_{\mathbf{max}}}$$

- If $\boldsymbol{\tau = n}$
  - Recovered algorithm: Gradient descent

Extreme values of $\mathcal{L}_1$

- **Recall**: $S = \tau$-nice sampling

$$\mathcal{L}_1 = \max_{i=1,\ldots,n} \left\{ \sum_{B \in \{[n]\} \mid i \in B} L_B \right\}$$

- If $\boldsymbol{\tau = 1}$
  - Recovered algorithm: SAGA
  - $\mathrm{supp}(S) = \{\{1\}, \ldots, \{n\}\}$

$$\boxed{\mathcal{L}_1 = L_{\mathbf{max}}}$$

- If $\boldsymbol{\tau = n}$
  - Recovered algorithm: Gradient descent
  - $\mathrm{supp}(S) = \{[n]\}$

Extreme values of $\mathcal{L}_1$

- **Recall**: $S = \tau$-nice sampling

$$\mathcal{L}_1 = \max_{i=1,\ldots,n} L_{[n]}$$

- If $\boldsymbol{\tau = 1}$
  - Recovered algorithm: SAGA
  - $\mathrm{supp}(S) = \{\{1\}, \ldots, \{n\}\}$

$$\boxed{\mathcal{L}_1 = L_{\mathbf{max}}}$$

- If $\boldsymbol{\tau = n}$
  - Recovered algorithm: Gradient descent
  - $\mathrm{supp}(S) = \{[n]\}$

Extreme values of $\mathcal{L}_1$

- **Recall**: $S = \tau$-nice sampling

$$\mathcal{L}_1 = \max_{i=1,\ldots,n} L_{[n]}$$

- If $\boldsymbol{\tau = 1}$
  - Recovered algorithm: SAGA
  - $\mathrm{supp}(S) = \{\{1\}, \ldots, \{n\}\}$

$$\boxed{\mathcal{L}_1 = L_{\mathbf{max}}}$$

- If $\boldsymbol{\tau = n}$
  - Recovered algorithm: Gradient descent
  - $\mathrm{supp}(S) = \{[n]\}$

$$\boxed{\mathcal{L}_1 = L}$$

Extreme values of $\mathcal{L}_1$

- **Recall**: $S = \tau$-nice sampling

$$\mathcal{L}_1 = \max_{i=1,\ldots,n} L_{[n]}$$

- If $\boldsymbol{\tau = 1}$
  - Recovered algorithm: SAGA
  - $\mathrm{supp}(S) = \{\{1\}, \ldots, \{n\}\}$

$$\boxed{\mathcal{L}_1 = L_{\mathbf{max}}}$$

- If $\boldsymbol{\tau = n}$
  - Recovered algorithm: Gradient descent
  - $\mathrm{supp}(S) = \{[n]\}$

$$\boxed{\mathcal{L}_1 = L}$$

$\rightarrow$ **$\mathcal{L}_1$ interpolates between L_{max} and L**

Our upper bounds

Lemma (Simple combination bound)

If S is a τ-nice sampling, we have

$$\mathcal{L}_1 \leq \frac{1}{\tau}\frac{n-\tau}{n-1}L_{\max} + \frac{n}{\tau}\frac{\tau-1}{n-1}\bar{L}$$

Proof: Weyl's inequality + double counting argument

Lemma (Bernstein bound)

If S is a τ-nice sampling, we have

$$\mathcal{L}_1 \leq \frac{1}{\tau}\left(\frac{n-\tau}{n-1} + \frac{4}{3}\log d\right)L_{\max} + 2\frac{\tau-1}{\tau}\frac{n}{n-1}L$$

Proof: In the annex

Interpolation of our bounds

- **Heuristic**:

$$\frac{1}{\tau}\frac{n-\tau}{n-1}L_{\max} + \frac{n}{\tau}\frac{\tau-1}{n-1}L$$

Name	Value for $\tau = 1$	Value for $\tau = n$
Simple combination	$L_{\max}$	$\bar{L}$
Bernstein	$\left(1 + \frac{4}{3}\log d\right) L_{\max}$	$2L + \frac{1}{n}\frac{4}{3}\log d\, L_{\max}$
Heuristic	$L_{\max}$	L

Table 1: Upper bounds of $\mathcal{L}_1$ and heuristic with the values at extreme points.

Talk Overview

Variance-reduced methods

Upper bounding the expected smoothness constant

Optimal mini-batch size

Numerical experiments

Conclusion

Optimal mini-batch size

Simple combination optimal mini-batch

- **Total complexity**

$$K_{\text{total}}(\tau) = \max\left\{\frac{4\tau(\mathcal{L}_1+\lambda)}{\mu}, n + \frac{n-\tau}{n-1}\frac{4(L_{\max}+\lambda)}{\mu}\right\}\log\left(\frac{1}{\epsilon}\right)$$

- **Pessimistic total complexity**

$$K_{\text{total}}(\tau) \leq \hat{K}_{\text{total}}(\tau)$$

$$= \max\left\{\underbrace{n\frac{\tau-1}{n-1}\frac{4\bar{L}}{\mu} + \frac{n-\tau}{n-1}\frac{4L_{\max}}{\mu} + \frac{4\tau\lambda}{\mu}}_{LHS(\tau)}, \underbrace{n + \frac{n-\tau}{n-1}\frac{4(L_{\max}+\lambda)}{\mu}}_{RHS(\tau)}\right\}\log\left(\tfrac{1}{\epsilon}\right)$$

- **Optimal mini-batch**

$$\text{find } \tilde{\tau} \in \arg\min_{\tau\in[n]} \hat{K}_{\text{total}}(\tau) \implies \boxed{\tilde{\tau} = \left\lfloor 1 + \frac{\mu(n-1)}{4(\bar{L}+\lambda)}\right\rfloor}$$

Minimization of the pessimistic complexity

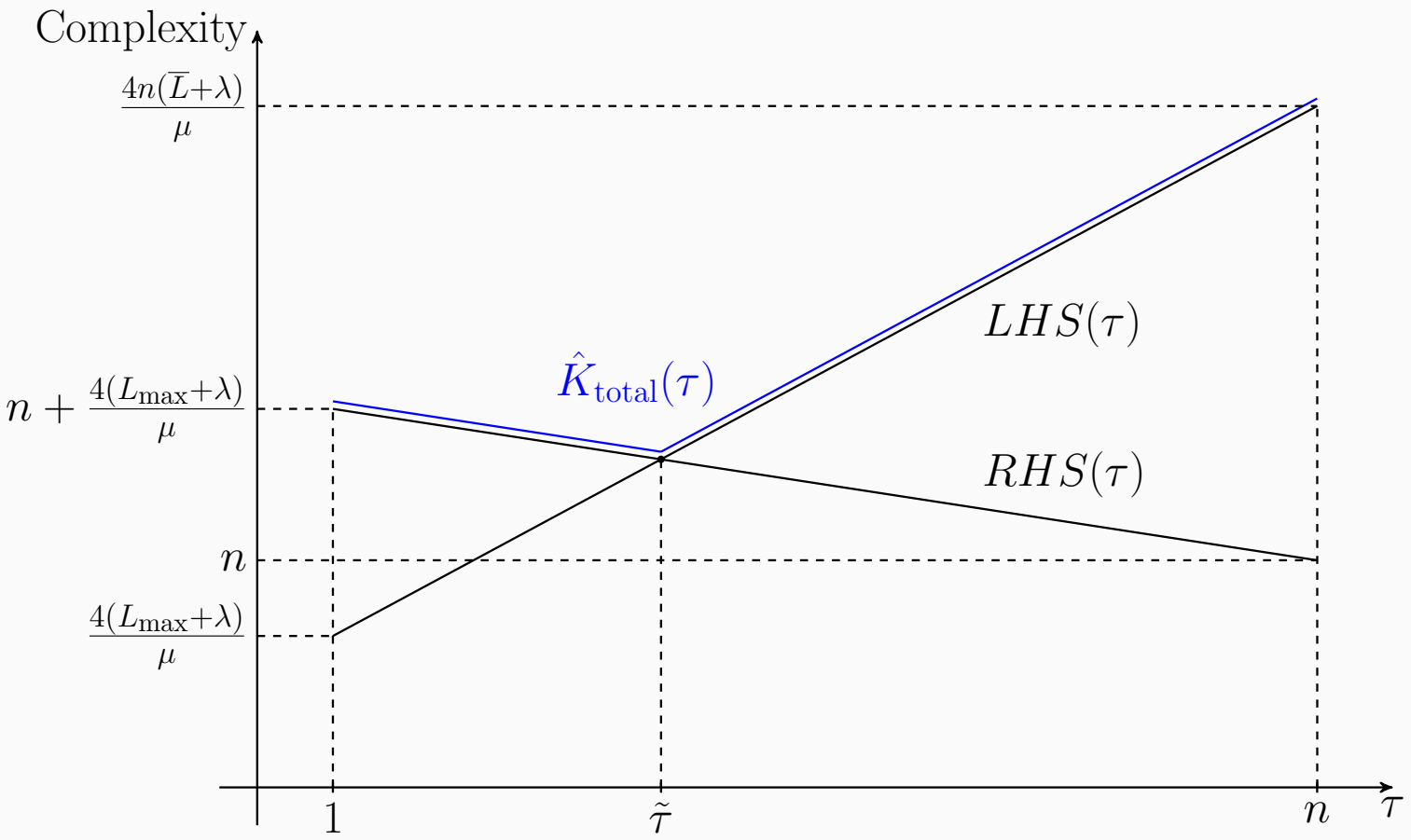


Figure 1: Optimal mini-batch size $\tilde{\tau}$ for the simple combination bound.

$$\text{where} \begin{cases} LHS(\tau) = & n\frac{\tau-1}{n-1}\frac{4(\bar{L}+\lambda)}{\mu} + \frac{n-\tau}{n-1}\frac{4(L_{\max}+\lambda)}{\mu} \\ RHS(\tau) = & n + \frac{n-\tau}{n-1}\frac{4(L_{\max}+\lambda)}{\mu} \end{cases}$$

Summary of the optimal theoretical values

- **Simple combination bound**

$$\tilde{\tau} = \left\lfloor 1 + \frac{\mu(n-1)}{4(\bar{L}+\lambda)} \right\rfloor$$

- **Bernstein bound**

$$\tilde{\tau} = \begin{cases} \left\lfloor 1 + \frac{\mu(n-1)}{4(2L+\lambda)} - \frac{4}{3}\log d \frac{n-1}{n} \frac{L_{\max}}{2L+\lambda} \right\rfloor & \text{if } \frac{4}{3}\log d \frac{4L_{\max}}{\mu} \leq n \ , \\ 1 & \text{otherwise} \ . \end{cases}$$

Talk Overview

Variance-reduced methods

Upper bounding the expected smoothness constant

Optimal mini-batch size

Numerical experiments

Conclusion

Numerical experiments

Experimental setup

- **Question**: How tight are our upper bounds of $\mathcal{L}_1$?
- **Data**: artificially generated $\mathbf{A}$ matrix
  - Uniformly random

$$\mathbf{A} \in \mathbb{R}^{d \times n} \mid [\mathbf{A}]_{ij} \sim \mathcal{U}([0,1))$$

  - Alone eigenvalue

$$\mathbf{A} = \operatorname{diag}(1, \ldots, 1, L_{\max}) \in \mathbb{R}^{n \times n}$$

  - Staircase-eigenvalues

$$\mathbf{A} = \operatorname{diag}\left(1, \frac{L_{\max}}{n}, \ldots, (n-2)\frac{L_{\max}}{n}, L_{\max}\right) \in \mathbb{R}^{n \times n}$$

Heuristic curve

Lemma (Simple combination bound)

If S is a τ-nice sampling, we have

$$\mathcal{L}_1 \leq \frac{n}{\tau}\frac{\tau-1}{n-1}\bar{L} + \frac{1}{\tau}\frac{n-\tau}{n-1}L_{\max}$$

Heuristic curve

Lemma (Simple combination bound)

If S is a τ-nice sampling, we have

$$\mathcal{L}_1 \leq \frac{n}{\tau}\frac{\tau-1}{n-1}\bar{L} + \frac{1}{\tau}\frac{n-\tau}{n-1}L_{\max}$$

- **Heuristic**:

$$\frac{n}{\tau}\frac{\tau-1}{n-1}L + \frac{1}{\tau}\frac{n-\tau}{n-1}L_{\max}$$

Heuristic curve

Lemma (Simple combination bound)

If S is a τ-nice sampling, we have

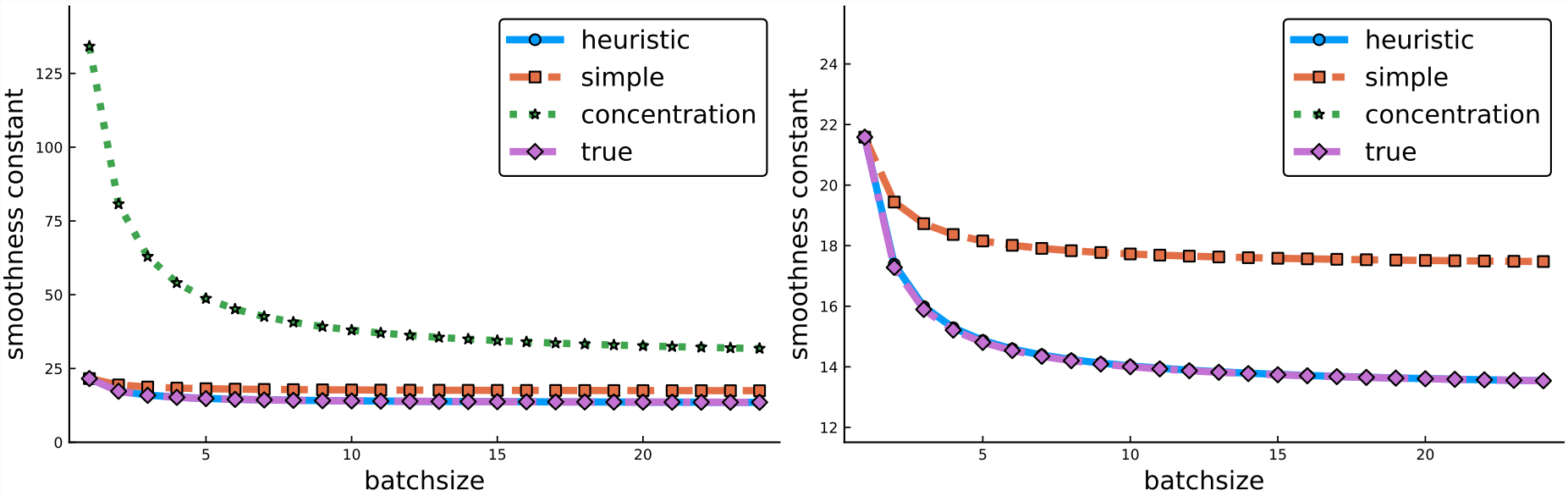
$$\mathcal{L}_1 \leq \frac{n}{\tau}\frac{\tau-1}{n-1}\bar{L} + \frac{1}{\tau}\frac{n-\tau}{n-1}L_{\max}$$

- **Heuristic**:

$$\frac{n}{\tau}\frac{\tau-1}{n-1}L + \frac{1}{\tau}\frac{n-\tau}{n-1}L_{\max}$$

$\rightarrow$ **Is our heuristic a good approximation of $\mathcal{L}_1$?**

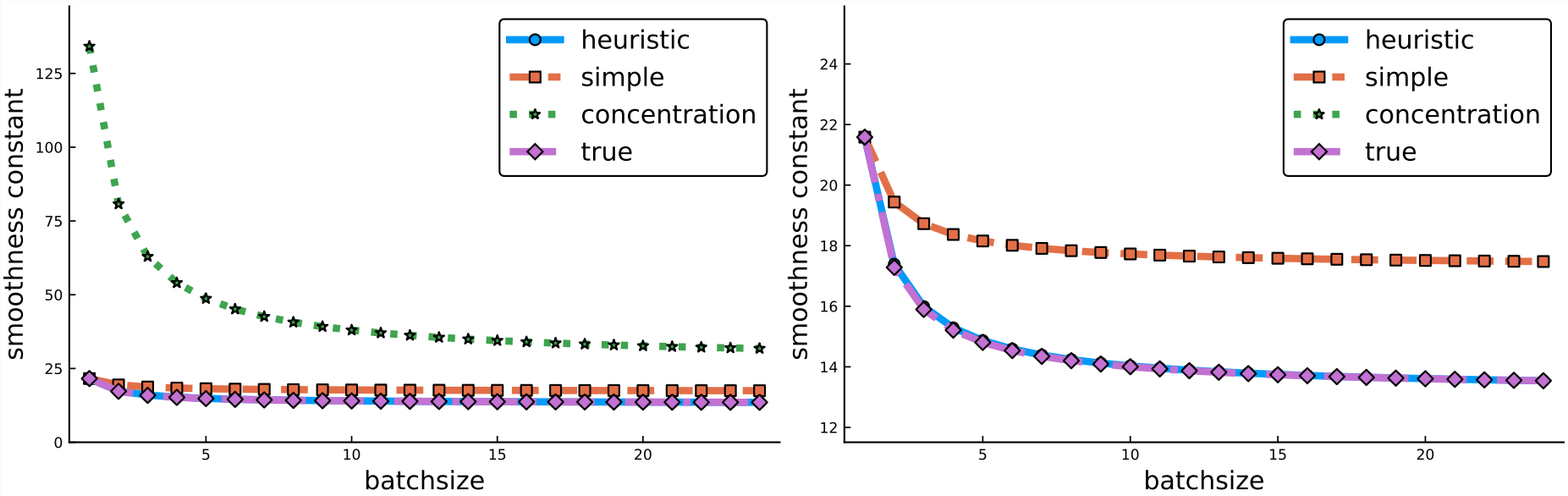
Uniformly random data



(a) Whole figure. **(b)** Zoom.

Figure 2: Upper bounds, heuristic and $\mathcal{L}_1$ computed on artificial random matrix, for $n = 24,\ d = 50$ ($L_{\max} \approx 22,\ \bar{L} \approx 17,\ L \approx 14$).

Uniformly random data



(a) Whole figure. **(b)** Zoom.

Figure 2: Upper bounds, heuristic and $\mathcal{L}_1$ computed on artificial random matrix, for $n = 24,\ d = 50$ ($L_{\max} \approx 22,\ \bar{L} \approx 17,\ L \approx 14$).

$\rightarrow$ **Heuristic close to $\mathcal{L}_1$**

Alone-eigenvalue data

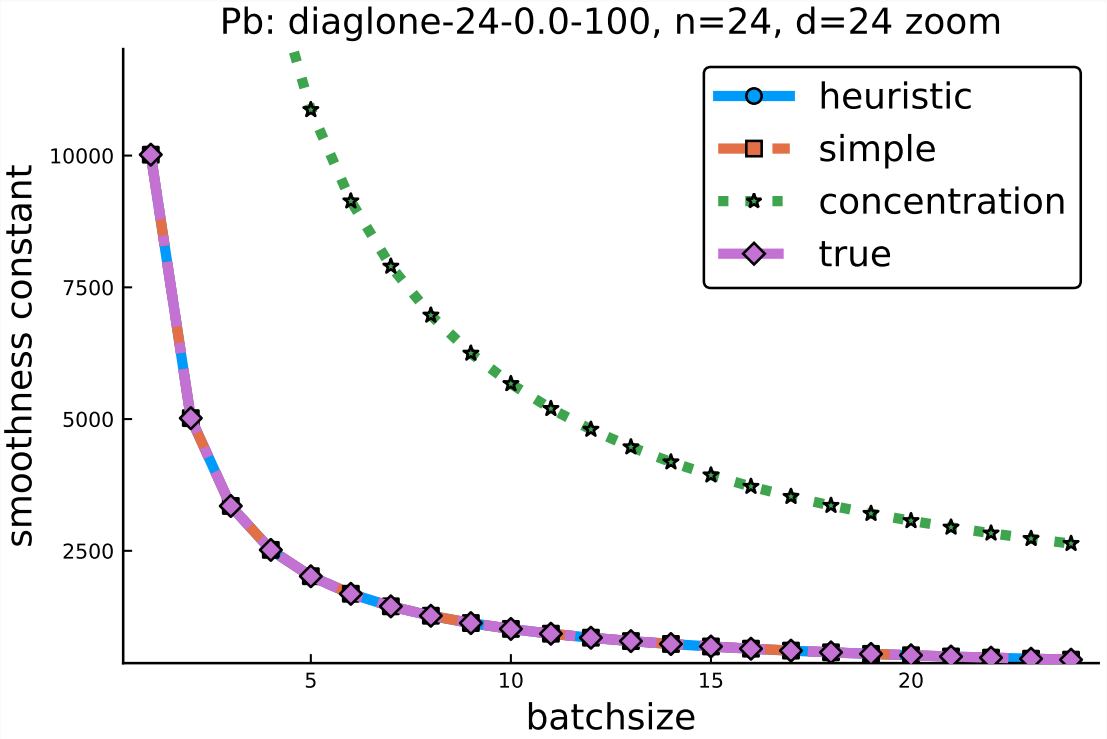


Figure 3: Upper bounds, heuristic and $\mathcal{L}_1$ computed on artificial alone-eigenvalue matrix for $n = d = 24$ and $L_{\max} = 10000$ ($L \approx 434$, $\bar{L} \approx 435$).

Alone-eigenvalue data

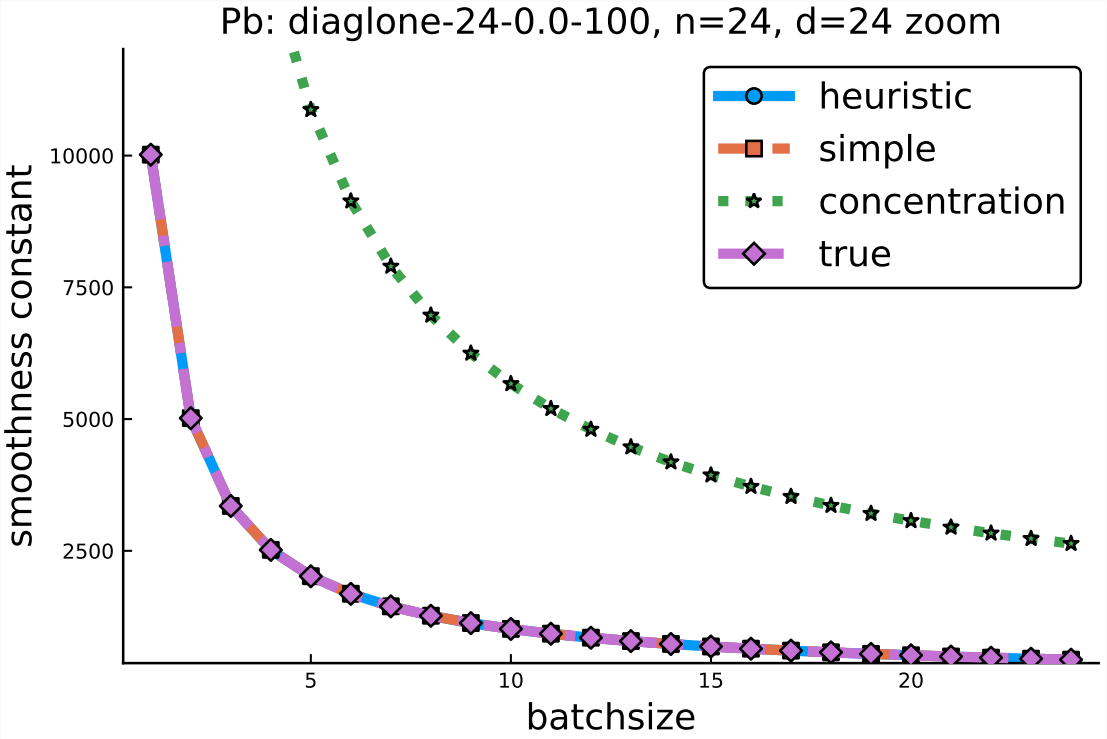


Figure 3: Upper bounds, heuristic and $\mathcal{L}_1$ computed on artificial alone-eigenvalue matrix for $n = d = 24$ and $L_{\max} = 10000$ ($L \approx 434$, $\bar{L} \approx 435$).

$\rightarrow$ **Simple bound good for upper bound for $L_j \gg L_i \; \forall i \neq j$**

Staircase-eigenvalues data

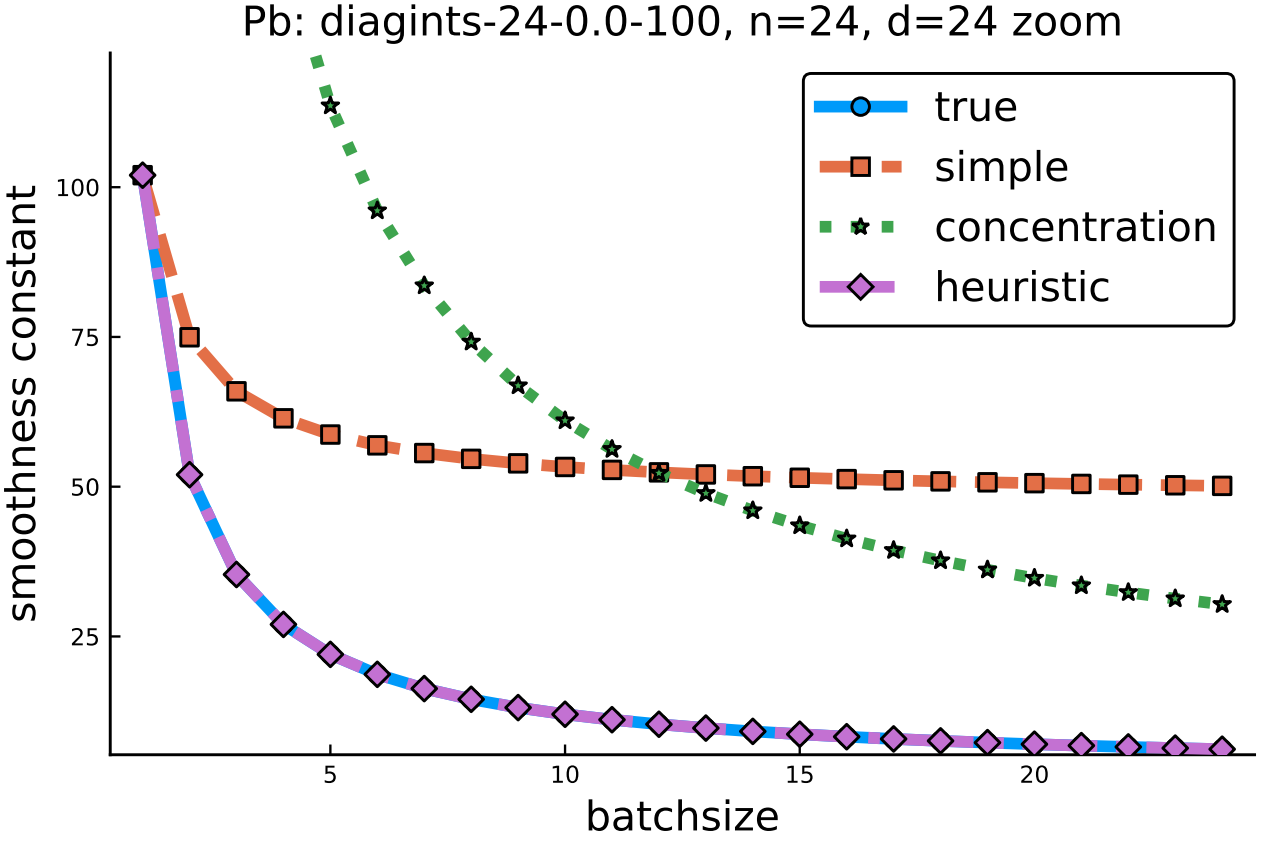


Figure 4: Upper bounds, heuristic and $\mathcal{L}_1$ computed on artificial staircase-eigenvalues matrix, $n = d = 24$ ($L_{\max} \approx 102$, $\bar{L} \approx 50$, $L \approx 6$).

Staircase-eigenvalues data

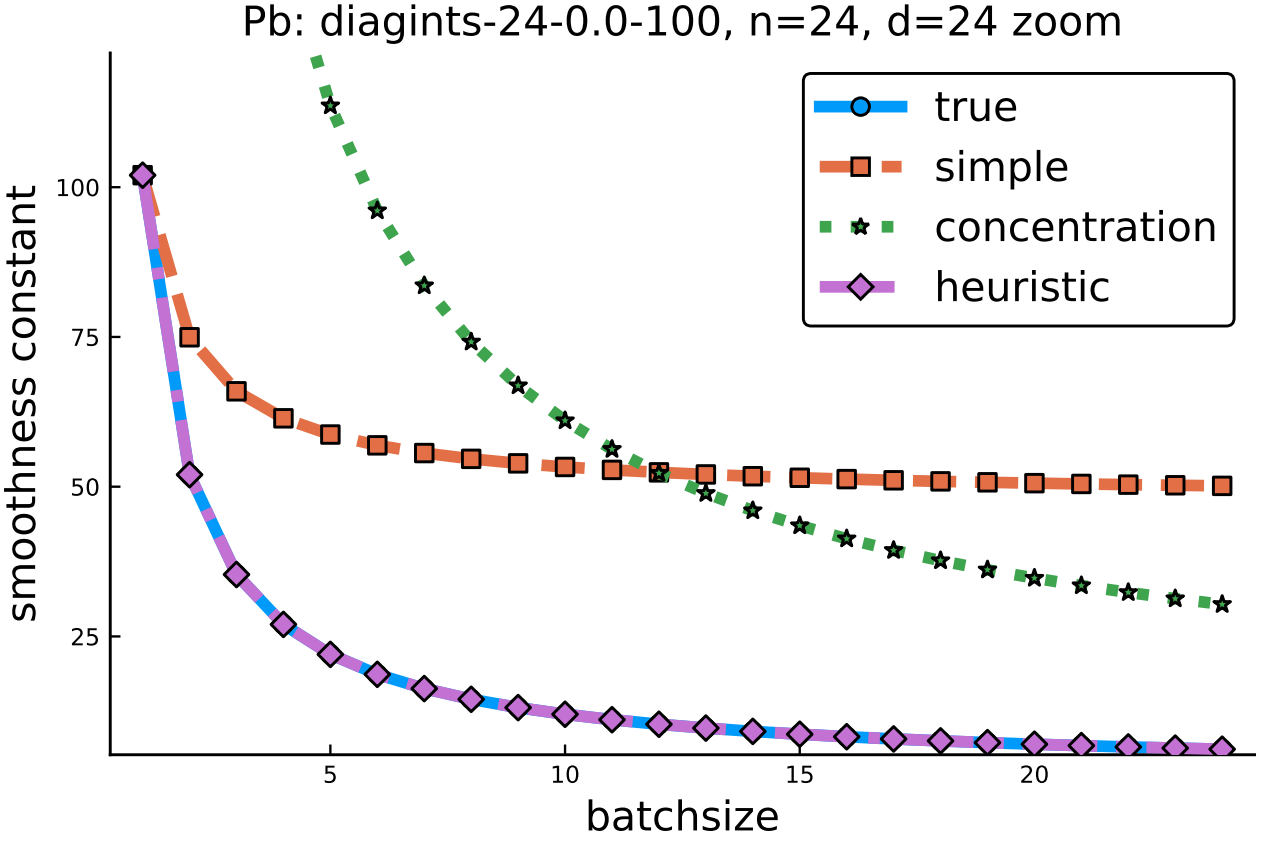


Figure 4: Upper bounds, heuristic and $\mathcal{L}_1$ computed on artificial staircase-eigenvalues matrix, $n = d = 24$ ($L_{\max} \approx 102$, $\bar{L} \approx 50$, $L \approx 6$).

$\rightarrow$ **Two regimes when the spectrum is equally distributed**

Talk Overview

Variance-reduced methods

Upper bounding the expected smoothness constant

Optimal mini-batch size

Numerical experiments

Conclusion

Conclusion

Summary

What was done

- Upper bounds of $\mathcal{L}_1$
- Theoretical optimal value for τ

What is to be done

- Test experimentally the optimality of $\tilde{\tau}$ on real and artificial data
- Extend the study to the optimal step size γ

References (1/2)

- F. Bach. "Sharp analysis of low-rank kernel matrix approximations". In:ArXiv e-prints (Aug. 2012). arXiv: 1208.2015 [cs.LG].
- C. C. Chang and C. J. Lin. "LIBSVM : A library for support vector machines". In: ACM Transactions on Intelligent Systems and Technology 2.3 (Apr. 2011), pp. 127.
- A. Defazio, F. Bach, and S. Lacoste-julien. "SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives". In: Advances in Neural Information Processing Systems 27. 2014, pp. 16461654.
- R. M. Gower, P. Richtrik, and F. Bach. "Stochastic Quasi-Gradient Methods: Variance Reduction via Jacobian Sketching". In: arXiv preprint arXiv:1805.02632 (2018).
- D. Gross and V. Nesme. "Note on sampling without replacing from a finite collection of matrices". In: arXiv preprint arXiv:1001.2738 (2010)
- W. Hoeffding. "Probability inequalities for sums of bounded random variables". In: Journal of the American statistical association 58.301 (1963), pp. 1330.

References (2/2)

- R. Johnson and T. Zhang. "Accelerating Stochastic Gradient Descent using Predictive Variance Reduction". In: Advances in Neural Information Processing Systems 26. Curran Associates, Inc., 2013, pp. 315323.
- H. Robbins and S. Monro. "A stochastic approximation method". In: Annals of Mathematical Statistics 22 (1951), pp. 400407.
- M. Schmidt, N. Le Roux, and F. Bach. "Minimizing finite sums with the stochastic average gradient". In: Mathematical Programming 162.1 (2017), pp. 83112.
- J. A. Tropp. "An Introduction to Matrix Concentration Inequalities". In: ArXiv e-prints (Jan. 2015). arXiv:1501.01571 [math.PR]
- J. A. Tropp. "Improved analysis of the subsampled randomized Hadamard transform". In: Advances in Adaptive Data Analysis 3.01n02 (2011), pp. 115126.
- J. A. Tropp. "User-Friendly Tail Bounds for Sums of Random Matrices". In: Foundations of Computational Mathematics 12.4 (2012), pp. 389434.

Questions?

JacSketch Lyapunov function

Definition (Stochastic Lyapunov function)

$$\Psi^k \stackrel{\text{def}}{=} \left\|w^k - w^*\right\|_2^2 + \frac{\gamma}{2\tau L_{\max}} \left\|\mathbf{J}^k - \nabla \mathbf{F}(w^*)\right\|_{\mathrm{F}}^2$$

- $\|\cdot\|_{\mathrm{F}}$: Frobenius norm
- $\nabla \mathbf{F}(w) = [\nabla f_1(w), \ldots, \nabla f_n(w)] \in \mathbb{R}^{d \times n}$: Jacobian matrix
- $\left\{w^k, \mathbf{J}^k\right\}_{k \geq 0}$ are the points and Jacobian estimate

If $\epsilon > 0$ denotes the desired precision, Theorem 3.6 ensures that, for a step size $\gamma = \min\left\{\frac{1}{4\mathcal{L}_1}, \frac{1}{\frac{1}{\tau}\frac{n-\tau}{n-1}L_{\max} + \frac{\mu}{4}\frac{n}{\tau}}\right\}$,

$$\mathbb{E}\left[\Psi^k\right] \leq \epsilon \Psi^0 \quad .$$

(Gower, Richtárik and Bach, 2018)

From sampling without to with replacement

Lemma (Domination of the trace of the mgf of a sample without replacement)

Consider two finite sequences, of same length, $\{\mathbf{X}_k\}$ and $\{\mathbf{M}_k\}$ of Hermitian random matrices of same size sampled respectively with and without replacement from a finite set $\mathcal{X}$. Let $\theta \in \mathbb{R}$, then

$$\mathbb{E}\,\mathrm{tr}\exp\left(\theta \sum\nolimits_k \mathbf{M}_k\right) \leq \mathbb{E}\,\mathrm{tr}\exp\left(\theta \sum\nolimits_k \mathbf{X}_k\right) \quad .$$

(Gross and Nesme, 2010)

Proof sketch of the matrix concentration bound (1/2)

(i) Write $\mathcal{L}_1$ as an **expectation**

$$
\begin{aligned}
\mathcal{L}_1 &= \max_{i=1,\dots,n} \mathbb{E}\left[L_{S^i \cup \{i\}}\right] \\
&= \max_{i=1,\dots,n} U\mathbb{E}\left[\lambda_{\max}\left(\frac{1}{\tau}\sum_{j\in S^n\cup\{i\}} a_j a_j^\top\right)\right] \\
&\leq \frac{1}{\tau}\frac{n-\tau}{n-1} L_{\max} + \frac{n}{\tau}\frac{\tau-1}{n-1} L \\
&\quad + \max_{i=1,\dots,n} U\mathbb{E}\left[\lambda_{\max}\left(\underbrace{\frac{1}{\tau}\sum_{j\in S^i} a_j a_j^\top - \frac{1}{\tau}\frac{\tau-1}{n-1}\sum_{j\in[n]\setminus\{i\}} a_j a_j^\top}_{\mathbf{N}=\sum_k \mathbf{M}_k}\right)\right]
\end{aligned}
$$

Proof sketch of the matrix concentration bound (1/2)

(i) Write $\mathcal{L}_1$ as an **expectation**

$$
\begin{aligned}
\mathcal{L}_1 &= \max_{i=1,\dots,n} \mathbb{E}\left[L_{S^i \cup \{i\}}\right] \\
&= \max_{i=1,\dots,n} U\mathbb{E}\left[\lambda_{\max}\left(\frac{1}{\tau}\sum_{j\in S^n\cup\{i\}} a_j a_j^\top\right)\right] \\
&\leq \frac{1}{\tau}\frac{n-\tau}{n-1} L_{\max} + \frac{n}{\tau}\frac{\tau-1}{n-1} L \\
&\quad + \max_{i=1,\dots,n} U\mathbb{E}\left[\lambda_{\max}\left(\underbrace{\frac{1}{\tau}\sum_{j\in S^i} a_j a_j^\top - \frac{1}{\tau}\frac{\tau-1}{n-1}\sum_{j\in[n]\setminus\{i\}} a_j a_j^\top}_{\mathbf{N}=\sum_k \mathbf{M}_k}\right)\right]
\end{aligned}
$$

Heuristic

Proof sketch of the matrix concentration bound (2/2)

(ii) Write $\mathbf{N}$ as a sum of random matrices and apply

Theorem (Matrix Bernstein Inequality Without Replacement)

Let $\mathcal{X}$ be a finite set of Hermitian matrices with dimension d s.t.

$$\lambda_{\max}(\mathbf{X}) \leq L \quad \forall \mathbf{X} \in \mathcal{X} \ .$$

Sample $\{\mathbf{X}_k\}$ and $\{\mathbf{M}_k\}$ uniformly at random from $\mathcal{X}$ resp. with and without replacement s.t.

$$\mathbb{E}\,\mathbf{X}_k = 0 \quad \forall k \ .$$

Let $\mathbf{Y} \stackrel{def}{=} \sum_k \mathbf{X}_k$ and $\mathbf{N} \stackrel{def}{=} \sum_k \mathbf{M}_k$. Then

$$\mathbb{E}\,\lambda_{\max}(\mathbf{N}) \leq \sqrt{2v(\mathbf{Y})\log d} + \frac{1}{3}L\log d \ .$$

where

$$v(\mathbf{Y}) \stackrel{def}{=} \left\|\mathbb{E}\,\mathbf{Y}^2\right\| = \left\|\sum_k \mathbb{E}\,\mathbf{X}_k^2\right\| = \lambda_{\max}\left(\sum_k \mathbb{E}\,\mathbf{X}_k^2\right).$$