Stochastic Spectral and Conjugate Descent Methods

Dmitry Kovalev[1,2] Eduard Gorbunov[2] Elnur Gasanov[1,2] Peter Richtárik[1,2,3]

[1]King Abdullah University of Science and Technology (KAUST), Kingdom of Saudi Arabia [2]Moscow Institute of Physics and Technology (MIPT), Russia [3] University of Edinburgh, United Kingdom

1. Introduction

Consider the optimization problem

$$\min_{x\in\mathbb{R}^n} f(x) := \tfrac{1}{2}x^\top \mathbf{A}x - b^\top x,$$

where $\mathbf{A}$ is an $n \times n$ symmetric positive definite matrix. The problem has a unique solution: $x_* = \mathbf{A}^{-1}b$. We are interested in the case when n is huge (millions, billions). Note that f is (strongly) convex and quadratic.

2. Algorithm: Stochastic Descent

The state-of-the-art methods for convex optimization in huge dimensions are randomized coordinate descent (RCD) methods. We now describe a method which includes RCD as a special case: **stochastic descent (SD).** SD is a special case of the **sketch-and-project** method developed in [1].

Algorithm 1 [1, 3] (Stochastic Descent).

Parameter: *some distribution $\mathcal{D}$ over vectors in $\mathbb{R}^n$*
Initialization: *Choose $x_0 \in \mathbb{R}^n$*
for $t = 0, 1, 2 \ldots$ ***do***
  Draw a fresh sample s_t from $\mathcal{D}$
  $x_{t+1} \leftarrow x_t - \frac{s_t^\top(\mathbf{A}x_t - b)}{s_t^\top \mathbf{A} s_t} s_t$
end for

RCD is obtained as a special case by letting $\mathcal{D}$ be a distribution over unit coordinate (i.e., basis) vectors in $\mathbb{R}^n$: $\{e_1, e_2, \cdots, e_n\}$:

$$s_t \sim \mathcal{D} \quad\Leftrightarrow\quad s_t = e_i \quad \text{with probability} \quad p_i > 0.$$

Theorem 1 [1, 3]. *Algorithm 1 converges linearly in expectation as*

$$(1-\rho_{\max})^t\|x_0 - x_*\|_{\mathbf{A}}^2 \le \mathbb{E}_{s\sim\mathcal{D}}[\|x_t - x_*\|_{\mathbf{A}}^2] \le (1-\rho_{\min})^t\|x_0 - x_*\|_{\mathbf{A}}^2,$$

where $\|x\|_{\mathbf{A}} = (x^\top \mathbf{A} x)^{1/2}$, $\mathbf{W} := \mathbb{E}_{s\sim\mathcal{D}}\left[\frac{\mathbf{A}^{1/2}ss^\top\mathbf{A}^{1/2}}{s^\top \mathbf{A} s}\right]$, $\rho_{\max} = \lambda_{\max}(\mathbf{W})$, $\rho_{\min} = \lambda_{\min}(\mathbf{W})$. *Moreover,* $0 < \rho_{\min} \le 1/n$ *and* $\rho_{\max} \le 1$.

3. Research Question

RCD with probabilities $p_i = \mathbf{A}_{ii}/\mathrm{Tr}(\mathbf{A})$ satisfies: $\rho_{\min} = \lambda_1/\mathrm{Tr}(\mathbf{A})$, where λ_1 is the smallest eigenvalue of $\mathbf{A}$. When $\rho_{\min}$ is small, RCD is slow. **Can we modify RCD by utilizing some spectral information, if known, so that the rate gets improved?**

4. New Algorithm

Let $\mathbf{A} = \sum_{i=1}^n \lambda_i u_i u_i^\top$ be the eigenvalue decomposition of $\mathbf{A}$, with $0 < \lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$ being the eigenvalues, and $u_1, \ldots, u_n$ the eigenvectors.

Algorithm 2 [2] (Stochastic Spectral Coordinate Descent).

Parameter: *Choose $k \in \{0, \ldots, n-1\}$; set $C_k = k\lambda_{k+1} + \sum_{i=k+1}^n \lambda_i$*
Run Algorithm 1 with the following distribution $\mathcal{D}$:

$$s_t = \begin{cases} e_i & \text{with probability } p_i = \frac{\mathbf{A}_{ii}}{C_k}, \quad i = 1, 2, \ldots, n \\ u_i & \text{with probability } p_{n+i} = \frac{\lambda_{k+1} - \lambda_i}{C_k}, \quad i = 1, 2, \ldots, k. \end{cases}$$

Note that for $k = 0$, Algorithm 2 reduces to RCD.

Theorem 2. *For every $n \ge 2$, Algorithm 2 has the rate*

$$\rho_{\min} = \frac{\lambda_{k+1}}{C_k}.$$

Moreover, the rate improves as k grows, and interpolates between the RCD rate $\lambda_1/\mathrm{Tr}(\mathbf{A})$ for $k = 0$, and the optimal rate $1/n$ for $k = n-1$:

$$\frac{\lambda_1}{\mathrm{Tr}(\mathbf{A})} = \frac{\lambda_1}{C_0} \le \cdots \le \frac{\lambda_{k+1}}{C_k} \le \cdots \le \frac{\lambda_{n-1}}{C_{n-2}} \le \frac{\lambda_n}{C_{n-1}} = \frac{1}{n}.$$

The total work of Algorithm 2 depends on k:

$$Work(\mathcal{D}) := \underbrace{P(\mathcal{D})}_{\text{preprocessing cost}} + \underbrace{C(\mathcal{D})}_{\text{cost of 1 iteration}} \times \underbrace{I(\mathcal{D})}_{\text{number of iterations till } \epsilon\text{-solution}}$$

k	$P(\mathcal{D})$	$C(\mathcal{D})$	$I(\mathcal{D})$
0	$O(n)$	$O(n)$	$\frac{\mathrm{Tr}(\mathbf{A})}{\lambda_1}\ln(1/\epsilon)$
$0 < k < n-1$	computation of λ_i for $i = 1, 2, \ldots, k+1$ computation of u_i for $i = 1, 2, \ldots, k$	$O(n)$	$\frac{C_k}{\lambda_{k+1}}\ln(1/\epsilon)$
$n-1$	computation of λ_i for $i = 1, 2, \ldots, n$ computation of u_i for $i = 1, 2, \ldots, n-1$	$O(n)$	$n\ln(1/\epsilon)$

5. Numerical Experiments

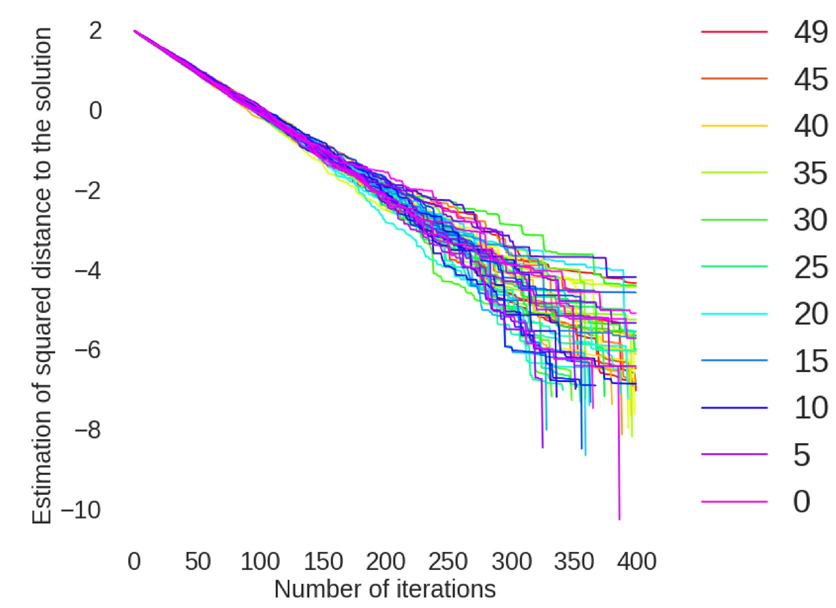

Figure: Eigenvalues were sampled from uniform distribution on $[10; 11]$; $n = 50$

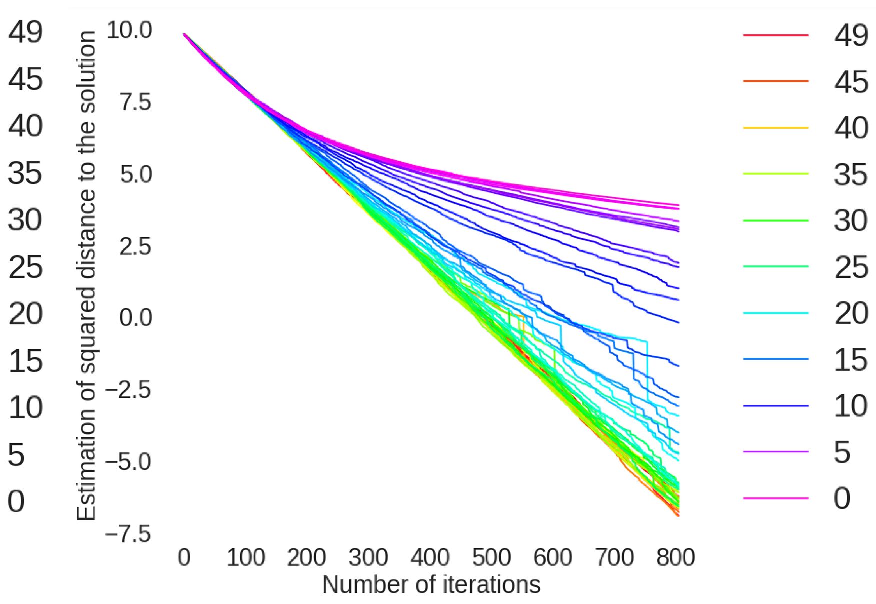

Figure: Eigenvalues were sampled from uniform distribution on $[0; 100,000]$; $n = 50$

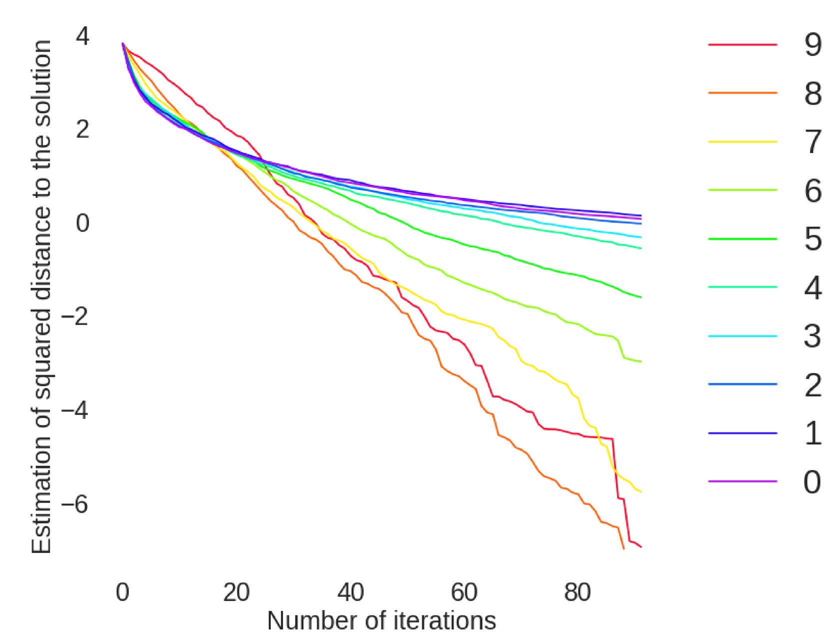

Figure: Eigenvalues decay exponentially; $n = 10$

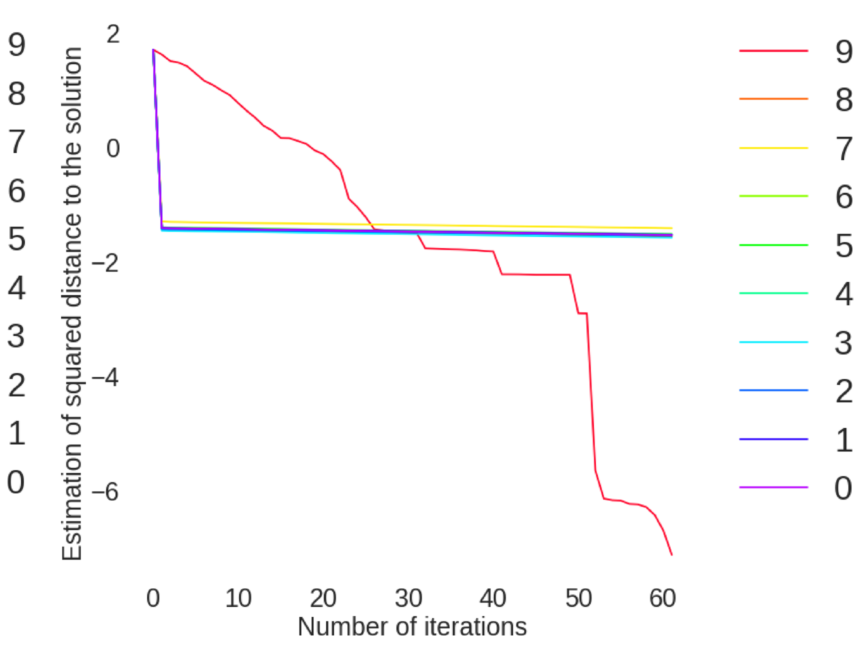

Figure: All eigenvalues equal to 1, except for the largest, which is equal to 1,000; $n = 10$

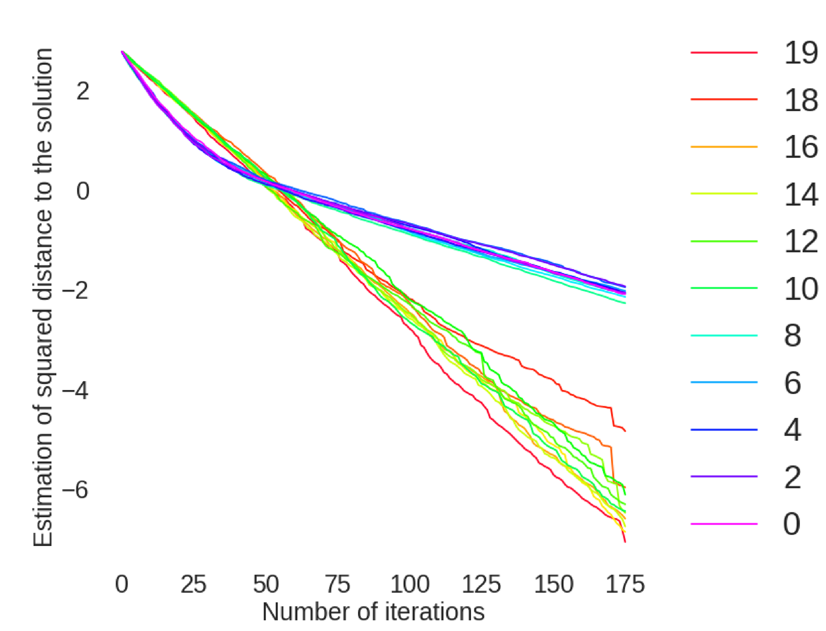

Figure: Half of eigenvalues were sampled from uniform distribution on $[10, 11]$ and half from uniform distribution on $[100; 101]$; $n = 20$

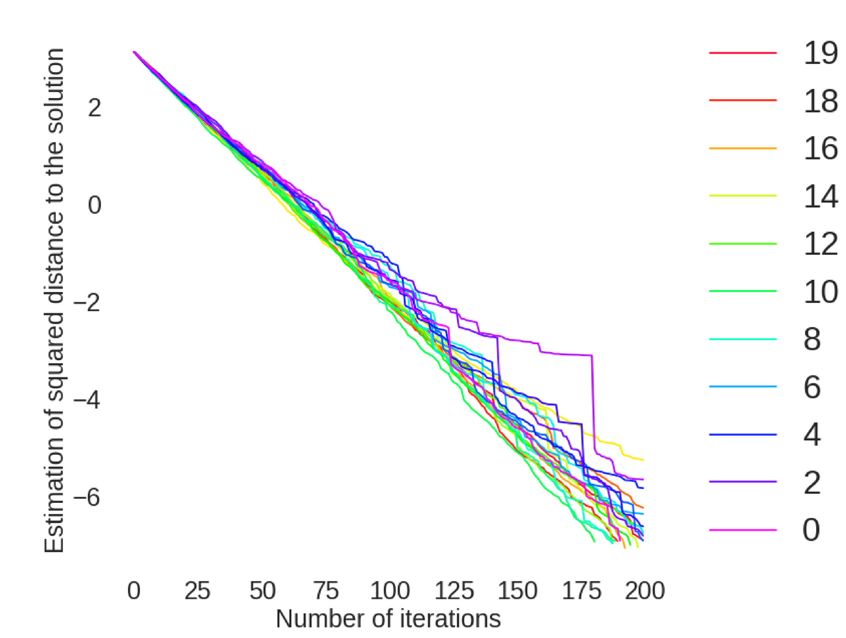

Figure: Half of eigenvalues were sampled from uniform distribution on $[50, 51]$ and half from uniform distribution on $[100, 101]$; $n = 20$

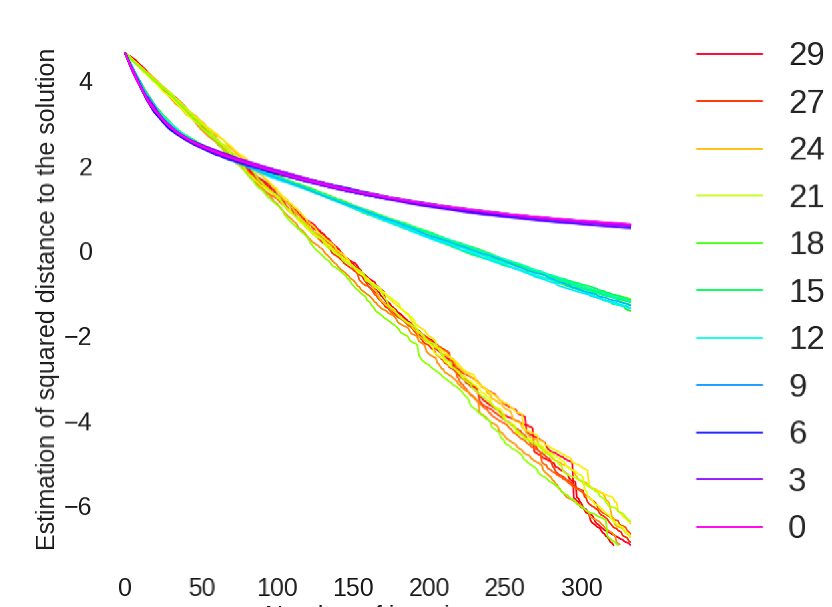

Figure: One third of eigenvalues were sampled from uniform distribution on $[10; 11]$, one third from uniform distribution on $[100; 101]$ and one third from uniform distribution on $[1,000; 1,001]$; $n = 30$

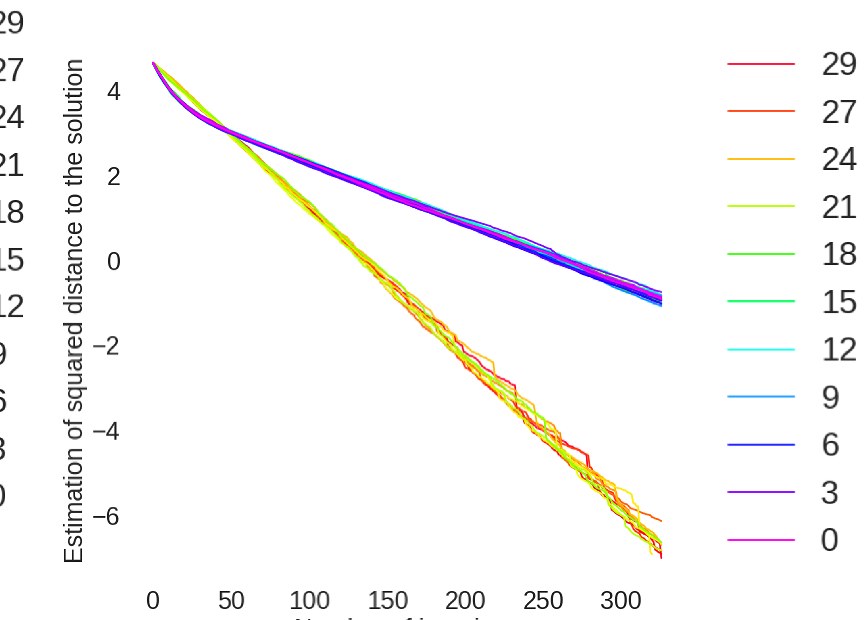

Figure: Two thirds of eigenvalues were sampled from uniform distribution on $[100; 101]$ and one third from uniform distribution on $[1000, 1001]$; $n = 30$

6. Bibliography

[1] R. M. Gower and P. Richtárik. Randomized iterative methods for linear systems. *SIAM Journal on Matrix Analysis and Applications*, 36(4):1660–1690, 2015.

[2] D. Kovalev, E. Gorbunov, E. Gasanov, and P. Richtárik. Stochastic spectral and conjugate descent methods. *NeurIPS 2018.*

[3] P. Richtárik and M. Takáč. Stochastic reformulations of linear systems: Algorithms and convergence theory. *arXiv:1706.01108*, 2017.