Randomized Kaczmarz algorithms: Exact MSE analysis and optimal sampling probabilities

RANDOMIZED KACZMARZ ALGORITHMS: EXACT MSE ANALYSIS AND OPTIMAL SAMPLING PROBABILITIES

Ameya Agaskar[1,2]*, Chuang Wang*[3,4] *and Yue M. Lu*[1]

[1]Harvard University, Cambridge, MA 02138, USA
[2]MIT Lincoln Laboratory, Lexington, MA 02420, USA
[3] Institute of Theoretical Physics, Chinese Academy of Sciences, Beijing 100190, China
[4] University of the Chinese Academy of Sciences, Beijing 100049, China
Email: {aagaskar, yuelu}@seas.harvard.edu, wangchuang@itp.ac.cn

ABSTRACT

The Kaczmarz method, or the algebraic reconstruction technique (ART), is a popular method for solving large-scale overdetermined systems of equations. Recently, Strohmer *et al.* proposed the randomized Kaczmarz algorithm, an improvement that guarantees exponential convergence to the solution. This has spurred much interest in the algorithm and its extensions. We provide in this paper an exact formula for the mean squared error (MSE) in the value reconstructed by the algorithm. We also compute the exponential decay rate of the MSE, which we call the "annealed" error exponent. We show that the typical performance of the algorithm is far better than the average performance. We define the "quenched" error exponent to characterize the typical performance. This is far harder to compute than the annealed error exponent, but we provide an approximation that matches empirical results. We also explore optimizing the algorithm's row-selection probabilities to speed up the algorithm's convergence.

Index Terms— Overdetermined linear systems, Kaczmarz Algorithm, randomized Kaczmarz algorithm

1. INTRODUCTION

The Kaczmarz algorithm [1], also known under the name Algebraic Reconstruction Technique (ART) [2], is a popular method for solving a large-scale overdetermined system of linear equations. Let

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}, \tag{1}$$

where $\boldsymbol{A}$ is a full-rank $m \times n$ matrix with $m \geq n$. Given $\boldsymbol{y} \in \mathbb{R}^m$, the algorithm proceeds to solve for $\boldsymbol{x}$ as follows: An initial guess $\boldsymbol{x}^{(0)}$ is chosen arbitrarily. The iterations then start with the first row, proceed in succession to the last row, and then cycle back to the first row, and so on. When row r is chosen, the current estimate $\boldsymbol{x}^{(k)}$ is projected onto the hyperplane $\{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{a}_r^T \boldsymbol{x} = y_r\}$ to obtain $\boldsymbol{x}^{(k+1)}$. Here, $\boldsymbol{a}_r^T$ is the rth row of $\boldsymbol{A}$.

Due to its simplicity, the Kaczmarz algorithm has been widely used in signal and image processing. It is also a special case of the projection onto convex sets (POCS) algorithm [3] for finding an intersection of many convex sets: in our case, we are looking for the intersection of a set of $(n-1)$-dimensional hyperplanes in $\mathbb{R}^n$.

It is well-known that the rate of convergence of the original Kaczmarz algorithm depends heavily on the exact ordering of the rows in $\boldsymbol{A}$ [4]. Recognizing this issue, Strohmer and Vershynin proposed in [5] a randomized Kaczmarz algorithm (RKA) that, instead of cycling sequentially through the rows in a deterministic fashion, chooses a row *at random* at each step. In their paper, they analyzed a specific probability distribution: choosing row i with probability proportional to its squared norm $\|\boldsymbol{a}_i\|^2$. They then showed the following upper bound on the mean squared error (MSE) of the RKA:

$$\mathbb{E}\|\boldsymbol{x}^{(N)} - \boldsymbol{x}\|^2 \leq \left(1 - \kappa_{\boldsymbol{A}}^{-2}\right)^N \|\boldsymbol{x}^{(0)} - \boldsymbol{x}\|^2, \tag{2}$$

where $\kappa_{\boldsymbol{A}} \stackrel{\text{def}}{=} \|\boldsymbol{A}\|_F \|\boldsymbol{A}^{-1}\|_2$ is the scaled condition number of $\boldsymbol{A}$, and $\boldsymbol{A}^{-1}$ is its left-inverse. Since $\kappa_{\boldsymbol{A}} \geq \sqrt{n}$, the above bound guarantees that the MSE decays *exponentially* as the RKA iterations proceed.

The work of Strohmer and Vershynin spurred a great deal of interest in RKA and its various extensions (see, *e.g.*, [6]–[12]). The original analysis in [5] assumes that the linear inverse problem is consistent (*i.e.*, noise-free). The noisy case was studied in [7]. A more general algorithm, involving random projections onto blocks of rows, was analyzed in [10]. Recently, Zouzias and Freris [9] proposed a randomized extended Kaczmarz algorithm which converges to the least squares estimate of an inconsistent system of linear equations.

We provide three contributions in this paper:

1. *An exact MSE formula:* All previous works on analyzing the performance of RKA provide strict upper bounds on the MSE. In this paper, we present an *exact* closed-form formula for the MSE of RKA after N iterations, for any N. Due to space constraint, we only present the noise-free case in this paper. However, the technique we use can be extended to more general settings as studied in [7, 9, 10].

2. *Annealed and quenched error exponents:* We provide an exact formula for the annealed error exponent, which measures the asymptotic rate of decay of the MSE, and we provide a good approximation for the quenched error exponent, which measures the asymptotic rate of decay of the squared error during a typical realization of the algorithm.

3. *Optimal sampling probabilities:* Our exact MSE formula allows us to pose a simple semidefinite programming (SDP) problem, the solution of which leads to optimal row-selection probabilities to minimize the MSE of the RKA.

The Lincoln Laboratory portion of this work was sponsored by the Department of the Air Force under Air Force Contract #FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

This work was done while C. Wang was a visiting student at the Signals, Information, and Networks Group (SING) at the Harvard School of Engineering and Applied Sciences. Y. M. Lu was supported in part by the U.S. National Science Foundation under Grant CCF-1319140.

Randomized Kaczmarz Algorithm [5]

Input: $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ with rows $\boldsymbol{a}_1^T, \boldsymbol{a}_2^T, \ldots, \boldsymbol{a}_m^T$; $\boldsymbol{y} \in \mathbb{R}^m$; selection probabilities $p_1, \ldots, p_m$ with $\sum_i p_i = 1$; iteration count N.
Output: $\widehat{\boldsymbol{x}} \in \mathbb{R}^n$, an estimate for $\boldsymbol{x} \in \mathbb{R}^n$ solving $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$.
Initialize $\boldsymbol{x}^{(0)}$ arbitrarily.
for $k = 1$ to N **do**
  $r \leftarrow i$ with probability p_i.
  $\boldsymbol{x}^{(k)} \leftarrow \boldsymbol{x}^{(k-1)} + \frac{y_r - \boldsymbol{a}_r^T \boldsymbol{x}^{(k-1)}}{\|\boldsymbol{a}_r\|^2}\boldsymbol{a}_r$
end for
$\widehat{\boldsymbol{x}} \leftarrow \boldsymbol{x}^{(N)}$.

2. EXACT PERFORMANCE ANALYSIS

2.1. Overview of RKA

Given a matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ and vector $\boldsymbol{y} \in \mathbb{R}^m$, the randomized Kaczmarz algorithm attempts to find a solution $\boldsymbol{x} \in \mathbb{R}^n$ to (1) as follows[1] The iterand $\boldsymbol{x}^{(0)} \in \mathbb{R}^n$ is initialized arbitrarily. At each step k, a row r_k is chosen at random. The probability of choosing row i is p_i; the row-selection probabilities $p_1, \ldots, p_m$ are tunable parameters of the algorithm. The iterand is then updated according to the formula

$$\boldsymbol{x}^{(k)} = \boldsymbol{x}^{(k-1)} + \frac{y_{r_k} - \boldsymbol{a}_{r_k}^T \boldsymbol{x}^{(k-1)}}{\|\boldsymbol{a}_{r_k}\|^2}\boldsymbol{a}_{r_k}. \tag{3}$$

The algorithm is listed above. The intuition behind the algorithm is simple. Each row of $\boldsymbol{A}$ and its corresponding entry in $\boldsymbol{y}$ defines a hyperplane on which the solution $\boldsymbol{x}$ must lie; at each time step in the RKA algorithm we randomly select one of these hyperplanes and project the iterand onto it, getting closer to the true solution with each step.

2.2. An Exact MSE Formula

Originally, Strohmer *et al.* proposed a specific probability distribution: $p_i = \frac{\|\boldsymbol{a}_i\|^2}{\|\boldsymbol{A}\|_F^2}$, where $\|\cdot\|_F$ is the Frobenius norm, and analyzed the behavior of the algorithm in terms of the propeties of $\boldsymbol{A}$. However, the solution to (1) is invariant to arbitrary and independent scalings of the rows. Thus, by looking at the properties of a rescaled version of $\boldsymbol{A}$, their analysis can be applied to arbitrary row-selection probabilities. Indeed, their results show that

$$(1 - 2N/\kappa_{\boldsymbol{A}}(p)^2) \le \frac{\mathbb{E}\|\boldsymbol{x}_N - \boldsymbol{x}\|^2}{\|\boldsymbol{x}_0 - \boldsymbol{x}\|^2} \le (1 - \kappa_{\boldsymbol{A}}(p)^{-2})^N, \tag{4}$$

where $\kappa_{\boldsymbol{A}}(p) = \left\|\widetilde{\boldsymbol{A}}^{-1}\boldsymbol{D}_p^{-1/2}\right\|$, and we have defined $\widetilde{\boldsymbol{A}}$ as the row-normalized version of $\boldsymbol{A}$, and $\boldsymbol{D}_p$ as the diagonal matrix with $p_1, p_2, \ldots, p_m$ on the diagonal. $\widetilde{\boldsymbol{A}}^{-1}$ is the left-inverse, which is guaranteed to exist because $\boldsymbol{A}$ is a tall, full-rank matrix.

The upper bound in (2) is sufficient to show that the error decays exponentially as the RKA iterations proceed. However, we show that it is possible to compute the *exact* error after N iterations of RKA, for any $N \ge 1$, given the initial error. This will allow us to precisely characterize the rate of decay of the error.

Proposition 1. *After N iterations of the randomized Kaczmarz algorithm with initial iterand $\boldsymbol{x}^{(0)}$, the average error is given by*

$$\mathbb{E}\left\|\boldsymbol{x}^{(N)} - \boldsymbol{x}\right\|^2 = \operatorname{vec}(I_n)^T R_{\boldsymbol{A}}(\boldsymbol{p})^N \operatorname{vec}\left(\left(\boldsymbol{x}^{(0)} - \boldsymbol{x}\right)\left(\boldsymbol{x}^{(0)} - \boldsymbol{x}\right)^T\right), \tag{5}$$

where $\operatorname{vec}(\cdot)$ *is the vectorization operator stacking a matrix's columns into a vector, and we have defined*

$$R_{\boldsymbol{A}}(\boldsymbol{p}) = \sum_{i=1}^m p_i \left(\boldsymbol{P}_{\boldsymbol{a}_i}^{\perp} \otimes \boldsymbol{P}_{\boldsymbol{a}_i}^{\perp}\right). \tag{6}$$

Here, $\boldsymbol{P}_{\boldsymbol{a}_i}^{\perp} \overset{def}{=} \left(\boldsymbol{I}_n - \frac{\boldsymbol{a}_i\boldsymbol{a}_i^T}{\|\boldsymbol{a}_i\|^2}\right)$ *is the orthogonal projection onto the hyperplane orthogonal to* $\boldsymbol{a}_i$*, and* $\otimes$ *is the Kronecker matrix product, so* $R_{\boldsymbol{A}}(\boldsymbol{p})$ *is an* $n^2 \times n^2$ *matrix.*

Proof. To simplify the expressions, we define $\boldsymbol{z}^{(k)} = \boldsymbol{x}^{(k)} - \boldsymbol{x}$ as the error vector after k iterations of the algorithm, with $\boldsymbol{z}^{(0)}$ being the error in the initial guess. Then at each iteration, the error updates according to

$$\boldsymbol{z}^{(k)} = \boldsymbol{z}^{(k-1)} - \frac{\boldsymbol{a}_{r_k}^T \boldsymbol{z}^{(k-1)}}{\|\boldsymbol{a}_{r_k}\|^2}\boldsymbol{a}_{r_k} = \boldsymbol{P}_{\boldsymbol{a}_{r_k}}^{\perp}\boldsymbol{z}^{(k-1)}, \tag{7}$$

where the r_k are i.i.d. indices chosen according to the probabilities $p_1, \ldots, p_m$.

To simplify the notation, we define $\boldsymbol{Q}_k \overset{\text{def}}{=} \boldsymbol{P}_{\boldsymbol{a}_{r_k}}^{\perp}$. The error after N steps is then related to the initial error as

$$\boldsymbol{z}^{(N)} = \boldsymbol{Q}_N\boldsymbol{Q}_{N-1}\cdots\boldsymbol{Q}_1\boldsymbol{z}^{(0)}, \tag{8}$$

where $\boldsymbol{Q}_1, \ldots, \boldsymbol{Q}_N$ are i.i.d. random matrices. In particular, $\boldsymbol{Q}_k = \boldsymbol{P}_{\boldsymbol{a}_i}^{\perp} = \left(\boldsymbol{I} - \frac{\boldsymbol{a}_i\boldsymbol{a}_i^T}{\|\boldsymbol{a}_i\|^2}\right)$ with probability p_i. The MSE after N steps can be computed as follows

$$\begin{aligned}
\mathbb{E}\left\|\boldsymbol{z}^{(N)}\right\|^2 &= \mathbb{E}\left\|\boldsymbol{Q}_N\boldsymbol{Q}_{N-1}\cdots\boldsymbol{Q}_1\boldsymbol{z}^{(0)}\right\|^2 \\
&= \mathbb{E}\,\boldsymbol{z}^{(0)T}\boldsymbol{Q}_1\boldsymbol{Q}_2\cdots\boldsymbol{Q}_N\boldsymbol{I}\boldsymbol{Q}_N\cdots\boldsymbol{Q}_2\boldsymbol{Q}_1\boldsymbol{z}^{(0)} \\
&= \mathbb{E}\operatorname{trace}\left(\boldsymbol{Q}_1\boldsymbol{Q}_2\cdots\boldsymbol{Q}_N\boldsymbol{I}\boldsymbol{Q}_N\cdots\boldsymbol{Q}_2\boldsymbol{Q}_1\boldsymbol{z}^{(0)}\boldsymbol{z}^{(0)T}\right) \\
&= \left[\mathbb{E}\operatorname{vec}\left(\boldsymbol{Q}_1\boldsymbol{Q}_2\cdots\boldsymbol{Q}_N\boldsymbol{I}\boldsymbol{Q}_N\cdots\boldsymbol{Q}_2\boldsymbol{Q}_1\right)\right]^T \\
&\qquad \operatorname{vec}\left(\boldsymbol{z}^{(0)}\boldsymbol{z}^{(0)T}\right),
\end{aligned} \tag{9}$$

where we have used two elementary matrix identities: $\operatorname{trace}(\boldsymbol{A}\boldsymbol{B}) = \operatorname{trace}(\boldsymbol{B}\boldsymbol{A})$ and $\operatorname{trace}(\boldsymbol{A}^T\boldsymbol{B}) = \operatorname{vec}(\boldsymbol{A})^T\operatorname{vec}(\boldsymbol{B})$ for any matrices $\boldsymbol{A}$ and $\boldsymbol{B}$. We also used the fact that $\boldsymbol{Q}_k = \boldsymbol{Q}_k^T$. The expectation of the product of matrices can be explicitly computed as follows:

$$\begin{aligned}
&\mathbb{E}\operatorname{vec}\left(\boldsymbol{Q}_1\boldsymbol{Q}_2\cdots\boldsymbol{Q}_N\boldsymbol{I}\boldsymbol{Q}_N\cdots\boldsymbol{Q}_2\boldsymbol{Q}_1\right) \\
&= \mathbb{E}(\boldsymbol{Q}_1 \otimes \boldsymbol{Q}_1)\operatorname{vec}\left(\boldsymbol{Q}_2\boldsymbol{Q}_3\cdots\boldsymbol{Q}_N\boldsymbol{I}\boldsymbol{Q}_N\cdots\boldsymbol{Q}_3\boldsymbol{Q}_2\right) && (10) \\
&= \mathbb{E}(\boldsymbol{Q}_1 \otimes \boldsymbol{Q}_1)\mathbb{E}\operatorname{vec}\left(\boldsymbol{Q}_2\boldsymbol{Q}_3\cdots\boldsymbol{Q}_N\boldsymbol{I}\boldsymbol{Q}_N\cdots\boldsymbol{Q}_3\boldsymbol{Q}_2\right) && (11) \\
&= \left[\mathbb{E}\boldsymbol{Q}_1 \otimes \boldsymbol{Q}_1\right]^N \operatorname{vec}(\boldsymbol{I}), && (12)
\end{aligned}$$

where (10) is due to the identity $\operatorname{vec}(\boldsymbol{A}\boldsymbol{B}\boldsymbol{C}^T) = (\boldsymbol{C} \otimes \boldsymbol{A})\operatorname{vec}(\boldsymbol{B})$ and the fact that $\boldsymbol{Q}_1$ is symmetric, (11) is due to the independence of the random matrices, and (12) is the result of repeating the preceding two steps N times. Combining (9) and (12) completes the proof. □

Remark 1. $R_{\boldsymbol{A}}(\boldsymbol{p})$ *is an* $n^2 \times n^2$ *matrix; however, due to its structure, it can be multiplied by a vector in* $\mathbb{R}^{n^2}$ *using* $O(mn^2)$ *operations rather than the naive* $O(n^4)$.

[1] The extension of the analysis in this paper to the complex case is simple, but complicates the notation enough that we analyze only the real case here.

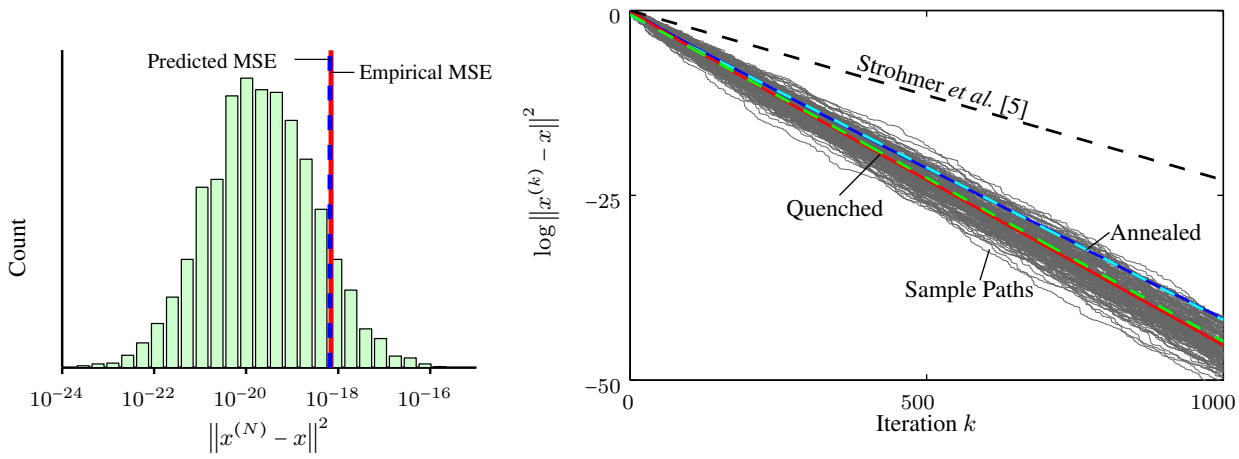

Fig. 1: (a) Histogram of squared errors after the simulation described in Section 2.3. The errors are plotted on a logarithmic scale to show the full range of errors; on a linear scale, the histogram is an L-shaped distribution with a spike at the origin and a long, thin tail. The location of the empirical MSE is overlaid on the histogram (red solid line), as is the exact MSE as given in Proposition 1 (blue dashed line). (b) Of the 3007 simulation trials, the "error trajectories" of 150 randomly-selected trials are plotted here (gray lines). On a logarithmic scale, there is a clear linear trend. Overlaid on these trajectories is the (annealed) average error trajectory (blue solid line) of all 3007 trials, and the prediction based on the annealed error exponent (cyan dashed line). We have also plotted the quenched average error trajectory, i.e. the average of the log of the error (red solid line), and the prediction based on the quenched error exponent (green dashed line) as given in (16). These are much more representative of the typical behavior of the algorithm. The upper bound of Strohmer *et al.* [5] is also shown (black dashed line).

2.3. Error Exponents: Annealed vs. Quenched

Proposition 1 confirms earlier bounds showing that the error decays exponentially. In fact, for generic values of the initial error vector, we have $\mathbb{E}\left\|\boldsymbol{z}^{(N)}\right\|^2 = \exp(-\gamma_a N + o(N))$, where γ_a is the *annealed* error exponent, defined by

$$\gamma_a \stackrel{\text{def}}{=} \lim_{N\to\infty} -\frac{1}{N}\log \mathbb{E}\left\|\boldsymbol{z}^{(N)}\right\|^2. \tag{13}$$

It is not hard to see that $\gamma_a = -\log \lambda_{\max}(R_{\boldsymbol{A}}(\boldsymbol{p}))$, where $\lambda_{\max}(\cdot)$ is the largest eigenvalue of a matrix.

To test our result, we simulated 3007 trials of the Kaczmarz algorithm for solving a linear system of dimension 150×20. The same system was used for each run, as well as the same initial vector. The matrix $\boldsymbol{A}$ was chosen to have independent standard normal entries (note that none of our analysis depends on $\boldsymbol{A}$ being drawn in this way, and similar results can be obtained with other matrices). We tracked the error after every iteration for each run. The row was chosen uniformly at random for each iteration. Figure 1(a) shows a histogram of the errors after 1000 iterations. The histogram was computed and is plotted on a logarithmic scale because of the wide range of resulting errors. The empirical MSE is overlaid on the histogram, as well as our prediction based on Proposition 1.

It is clear that our prediction matches the empirical value quite well. However, it is also clear that there is more to the story. Over 90% of the realizations have an error smaller than the mean, which is more than 10^2 times smaller than the worst realization. It appears that the average error is not necessarily a great representation of the *typical* error; in reality, there are occasional, rare, extreme failures that cause the average error to be much higher than the "typical" error.

A more representative measure of the error's decay rate is the *quenched* error exponent:

$$\gamma_q \stackrel{\text{def}}{=} \lim_{N\to\infty} -\frac{1}{N}\mathbb{E}\log \left\|\boldsymbol{z}^{(N)}\right\|^2. \tag{14}$$

Here, the logarithm of the error is taken *before* the expectation. The annealed and quenched error exponents we have defined are formally similar to Lyapunov exponents of products of random matrices, a problem well-studied by statistical physicists for use in modeling dynamical systems [13]. The terms "annealed" and "quenched" are borrowed from their analysis and have certain physical meanings, but to us they are just convenient names for two interesting quantities.

The quenched error exponent is far more difficult to analyze than the annealed one, a fact well known to the physicists [13, 14]. Jensen's inequality tells us that $\gamma_q \geq \gamma_a$. To obtain more information, physicists often rely on non-rigorous heuristics that are verified numerically or experimentally. One such heuristic is the replica method, which provides an approximation for the quenched Lyapunov exponent [13]. The physicists have their own intuition for this approximation, but our engineer's intuition is quite simple. The quintessential heavy-tailed distribution is the log-normal distribution. So let us assume that the error distribution is $\left\|\boldsymbol{z}^{(N)}\right\|^2 \sim \text{log-}\mathcal{N}(N\mu, N\sigma^2)$. Then $\log\left\|\boldsymbol{z}^{(N)}\right\|^2 \sim \mathcal{N}(N\mu, N\sigma^2)$. The log-normal assumption is supported by the histogram in Figure 1(a): the logarithm of the squared errors appear to follow a Gaussian distribution. The quenched error exponent is seen to be simply $\gamma_q = -\mu$. Now we need to compute the parameters of the distribution. Under these assumptions, $\mathbb{E}\left\|\boldsymbol{z}^{(N)}\right\|^2 = \exp(N[\mu + \frac{1}{2}\sigma^2])$ and $\mathbb{E}\left\|\boldsymbol{z}^{(N)}\right\|^4 = \exp(N[2\mu + 2\sigma^2])$. Solving this system of equations, we obtain:

$$\mu = \frac{1}{N}\left[2\log \mathbb{E}\left\|\boldsymbol{z}^{(N)}\right\|^2 - \frac{1}{2}\log \mathbb{E}\left\|\boldsymbol{z}^{(N)}\right\|^4\right]. \tag{15}$$

Thus, our approximation for the quenched error exponent is

$$\gamma_q \approx 2\gamma_a - \frac{1}{2}\gamma_a^{(2)}, \tag{16}$$

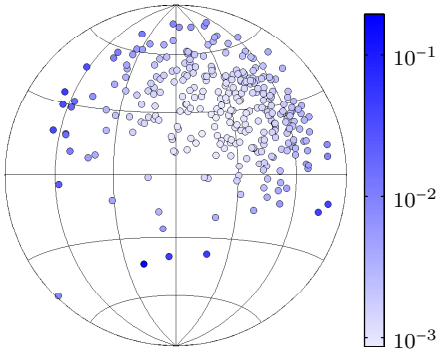

Fig. 2: Optimal selection probabilities for a non-uniform matrix. The plot is an equal-area projection of the entire unit hemisphere in R^3. Each row in the matrix is represented by a point on the plot; the color represents the optimal selection probability computed using `cvx`.

where

$$\gamma_a^{(2)} = \lim_{N\to\infty} -\frac{1}{N}\log \mathbb{E}\left\|\boldsymbol{z}^{(N)}\right\|^4. \tag{17}$$

To compute $\gamma_a^{(2)}$, we define

$$R_{\boldsymbol{A}}^{(2)}(\boldsymbol{p}) = \sum_{i=1}^{m} p_i \left(\boldsymbol{P}_{\boldsymbol{a}_i}^{\perp} \otimes \boldsymbol{P}_{\boldsymbol{a}_i}^{\perp} \otimes \boldsymbol{P}_{\boldsymbol{a}_i}^{\perp} \otimes \boldsymbol{P}_{\boldsymbol{a}_i}^{\perp}\right), \tag{18}$$

and have

$$\gamma_a^{(2)} = -\log \lambda_{\max}\left(R_{\boldsymbol{A}}^{(2)}(\boldsymbol{p})\right). \tag{19}$$

$R_{\boldsymbol{A}}^{(2)}(\boldsymbol{p})$ is an $n^4 \times n^4$ matrix, but it can be applied in time $O(mn^4)$ instead of the naive $O(n^8)$. So finding the largest eigenvalue is not as complex as one might naively expect.

Figure 1(b) illustrates our argument and shows just how good the replica method approximation is. We have plotted, on a logarithmic scale, the error trajectory of many trials as the iterations proceeded. (Only 150 randomly-selected trials are shown to prevent the figure from getting too cluttered). We have also plotted the logarithm of the average error, which matches the linear trendline predicted by the annealed error exponent γ_a, and the average of the logarithm of the error trajectories, which matches the linear trendline predicted by our approximation for the quenched error exponent γ_q. The quenched values are clearly more representative of the typical performance of the algorithm than the annealed ones. The close match indicates that our approximation is valid. For comparison purposes, we have also plotted the upper bound provided by Strohmer *et al.* [5].

3. OPTIMAL ROW-SELECTION PROBABILITIES

Given a matrix $\boldsymbol{A}$, we may wish to choose the row selection probabilities $p_1, p_2, \ldots, p_m$ that provide the fastest convergence. A tractable way to do this is to optimize the annealed error exponent γ_a, which measures the decay rate of the MSE. This is equivalent to the following optimization problem:

$$(p_1, \ldots, p_m) = \underset{\boldsymbol{p}\in\Delta^{n-1}}{\arg\min}\, \lambda_{\max}(R_{\boldsymbol{A}}(\boldsymbol{p})), \tag{20}$$

where Δ^{n-1} is the unit simplex in $\mathbb{R}^n$. The function $\lambda_{\max}(R_{\boldsymbol{A}}(\boldsymbol{p}))$ is convex [15], as is the set Δ^{n-1}, so (20) is a convex optimization problem (more specifically, it is a semidefinite programming problem). Thus, finding the optimal probability distribution $\boldsymbol{p}$ is quite tractable. Note that Dai *et al.* recently considered an optimized randomized Kaczmarz algorithm [11], in which the row-selection probabilities were chosen to optimize a bound on the MSE's decay rate. However, we optimize the *exact* decay rate of the MSE.

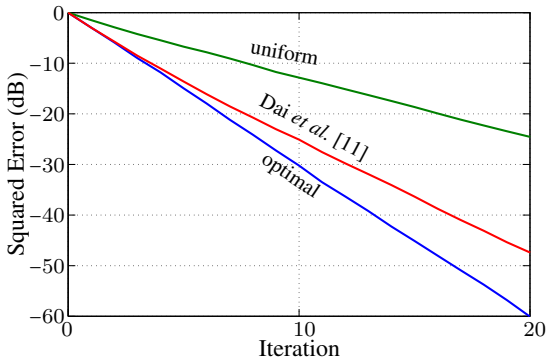

Fig. 3: Quenched average squared errors versus RKA iteration under the uniform, Dai *et al.*'s approximate optimal, and optimal row selection probabilities, for the 1000 x 3 matrix described in the text.The average is taken over 1007 trials.

To illustrate the kind of improvement possible by optimizing the row selection probabilities, and develop some intuition on the optimum choice, we computed the optimal values for a matrix of size 300×3. The elements of the matrix were chosen as independent Gaussian random variables with a variance of 0.5; the columns had means 0.5, 1, and 2, respectively. We used the `cvx` convex optimization software package to compute the optimal row selection probabilities for this matrix [16, 17].

Since the problem is invariant to the scale and sign of each rows, each row in the matrix $\boldsymbol{A}$ can be represented as a point on the unit hemisphere. Thus, the matrix and row probabilities can be illustrated as in Figure 2 by plotting each row as a point on a 2D projection of a unit hemisphere. We used the Lambert equal-area projection, which is measure-preserving and therefore allows us to accurately visualize the sampling density everywhere in the space. The darker points represent rows that are selected with high probability in the optimal selection scheme; the lighter ones are selected with lower probability. We would expect an optimal scheme to choose rows that are far from any other rows with higher probability than rows that are in close proximity to many other rows, in order to reduce redundancy and cover the whole space. The figure conforms to this intuition.

Figure 3 illustrates the improvement of the optimal randomization scheme over simply choosing rows uniformly at random. After 20 iterations, the optimal scheme has an error 36 dB lower than the uniform scheme, and 12 dB lower than the sub-optimal scheme of Dai *et al.*

Of course, in practice, there is a tradeoff between the computation time saved by needing fewer iterations and the computation time spent determining the optimal row selection probabilities in advance. The main purpose of the exact optimization proposed in this work is to develop intuition and validate sub-optimal heuristics. A fast or on-line method for approximating the optimal probabilities would be very beneficial for large-scale problems.

4. CONCLUSIONS

We provided a complete characterization of the randomized Kaczmarz algorithm. This included an exact formula for the MSE and the annealed error exponent characterizing its decay rate, plus an approximation for the quenched error exponent that captures the typical error decay rate. We also explored choosing the row-selection probabilities to achieve the best convergence properties for the algorithm.

5. REFERENCES

[1] S. Kaczmarz, "Angenäherte auflösung von systemen linearer gleichungen," *Bull. Internat. Acad. Polon. Sci. Lettres A*, pp. 335–357, 1937.

[2] R. Gordon, R. Bender, and G. T. Herman, "Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and X-ray photography," *Journal of theoretical Biology*, vol. 29, no. 3, p. 471–481, 1970.

[3] H. Trussell and M. Civanlar, "Signal deconvolution by projection onto convex sets," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '84.*, vol. 9, Mar. 1984, pp. 496–499.

[4] G. T. Herman and L. B. Meyer, "Algebraic reconstruction techniques can be made computationally efficient [positron emission tomography application]," *Medical Imaging, IEEE Transactions on*, vol. 12, no. 3, p. 600–609, 1993.

[5] T. Strohmer and R. Vershynin, "A randomized Kaczmarz algorithm with exponential convergence," *Journal of Fourier Analysis and Applications*, vol. 15, no. 2, p. 262–278, 2009.

[6] Y. Censor, G. T. Herman, and M. Jiang, "A note on the behavior of the randomized Kaczmarz algorithm of Strohmer and Vershynin," *Journal of Fourier Analysis and Applications*, vol. 15, no. 4, pp. 431–436, Aug. 2009.

[7] D. Needell, "Randomized Kaczmarz solver for noisy linear systems," *BIT Numerical Mathematics*, vol. 50, no. 2, pp. 395—403, 2010.

[8] X. Chen and A. M. Powell, "Almost sure convergence of the Kaczmarz algorithm with random measurements," *Journal of Fourier Analysis and Applications*, vol. 18, no. 6, pp. 1195—1214, 2012.

[9] A. Zouzias and N. M. Freris, "Randomized extended Kaczmarz for solving least squares," *SIAM Journal on Matrix Analysis and Applications*, vol. 34, no. 2, pp. 773–793, 2013.

[10] D. Needell and J. A. Tropp, "Paved with good intentions: Analysis of a randomized block Kaczmarz method," *Linear Algebra and its Applications*, vol. 441, pp. 199—221, 2014.

[11] L. Dai, M. Soltanalian, and K. Pelckmans, "On the randomized Kaczmarz algorithm," *IEEE Signal Process. Lett.*, vol. 21, no. 3, pp. 330–333, Mar. 2014.

[12] B. Recht and C. Ré, "Toward a noncommutative arithmetic-geometric mean inequality: Conjectures, case-studies, and consequences," in *Conference on Learning Theory*, 2012.

[13] A. Crisanti, G. Paladin, and A. Vulpiani, *Products of Random Matrices in Statistical Physics*, ser. Springer Series in Solid-State Sciences, M. Cardona, P. Fulde, K. Klitzing, H.-J. Queisser, and H. K. V. Lotsch, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1993, vol. 104, 00378.

[14] J. N. Tsitsiklis and V. D. Blondel, "The lyapunov exponent and joint spectral radius of pairs of matrices are hard—when not impossible—to compute and to approximate," *Mathematics of Control, Signals, and Systems (MCSS)*, vol. 10, no. 1, pp. 31–40, 1997.

[15] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2003.

[16] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.

[17] ——, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control*, ser. Lecture Notes in Control and Information Sciences, V. Blondel, S. Boyd, and H. Kimura, Eds. Springer-Verlag Limited, 2008, pp. 95–110.