A Stochastic Kaczmarz Algorithm for Network Tomography

Gugan Thoppe\textsuperscript{a}, Vivek Borkar\textsuperscript{b}, D. Manjunath\textsuperscript{b}

\textsuperscript{a}School of Technology and Computer Science, Tata Institute of Fundamental Research, Mumbai 400005, India
\textsuperscript{b}Department of Electrical Engineering, Indian Institute of Technology, Powai, Mumbai 400076, India

Abstract

We develop a stochastic approximation version of the classical Kaczmarz algorithm that is incremental in nature and takes as input noisy real time data. Our analysis shows that with probability one it mimics the behavior of the original scheme: starting from the same initial point, our algorithm and the corresponding deterministic Kaczmarz algorithm converge to precisely the same point. The motivation for this work comes from network tomography where network parameters are to be estimated based upon end-to-end measurements. Numerical examples via Matlab based simulations demonstrate the efficacy of the algorithm.

Key words: Kaczmarz algorithm; Stochastic approximation; Network tomography; Online algorithm.

1 Introduction

1.1 Kaczmarz algorithm

Kaczmarz algorithm [20] is a successive projection based iterative scheme for solving ill posed linear systems of equations. Since its introduction, its convergence properties have been extensively analyzed [18] and it has found diverse applications in areas ranging from tomography [29], synchronization in sensor networks [17], to learning and adaptive control [4,27,32]. The original algorithm is deterministic, but some applications, notably network tomography which we describe later, call for a stochastic version. In this article, we introduce and analyze a stochastic approximation version based on the Robbins-Monro paradigm [19] that has become a standard workhorse of signal processing and learning control.

We use the ‘o.d.e.’ approach [12,24] to analyze the scheme and argue that it has the same asymptotic behavior as the original deterministic scheme ‘almost surely’. While we apply our results to network tomography in this article, we believe that this analysis will be of use in other areas mentioned above. In particular, networked control is one potential application area.

A significant development in this line of research has been a randomized Kaczmarz scheme having provable strong convergence properties [34], [23], with recent modifications to further improve performance by weighted sampling [17], [38]. The important difference between these works and ours is as follows. For them, the randomization is over the choice of rows, which is a part of algorithm design and can be chosen at will. In our case, however, a part of the randomness is due to noise and not under our control, as also in the choice of rows which a priori we allow to be uncontrolled.

1.2 Network tomography

Network tomography is inference of spatially localized network behavior using only measurements of end-to-end aggregates. Recent work can be classified into traffic volume and link delay tomography. A basic paradigm in both these is to infer the statistics of the random vector $X$ from an ill posed measurement model $Y = AX$, where the matrix $A$ is assumed to be known a priori. See [11,9,22] for excellent surveys.

In the transportation literature, the aim is to estimate...
the traffic volume on the end-to-end routes assuming access to only traffic volumes on a subset of links [25,5,33,35]. An excellent survey is given in [1]. An analogous problem has been addressed in packet networking [16,36,37]. In all of these works, one sample of $Y$ is assumed available and $X$ is estimated by a suitable regularization.

Link delay tomography deals with estimation of link delay statistics from path delay measurements. Here the network is usually assumed to be in the form of a tree. Multicast probe packets, real or emulated, are sent from the root node to the leaves. For each probe packet, a set of delay measurements for paths from the root node to the leaves is collected. These delays are correlated and this correlation is exploited to estimate the link delay statistics from path delay measurements. Here the simplifying statistical assumption that the samples are IID, we point out later in Section 5 that these can be correlated. While our analysis is under the IID assumption is purely for simplicity of analysis. We point out later that these results extend to much more general situations.) We assume that at each time step $k$, we know only the value of $Z_{k+1}$ and the $Z_{k+1}$-th component of $Y_{k+1}$, i.e. $Y_{k+1}(Z_{k+1}) =: Y_{k+1}$. Our objective is to develop a real-time algorithm, with provable convergence properties, to estimate the moments and cross moments of the random vector $X$.

3 Preliminaries

3.1 Stochastic approximation algorithms

The archetypical stochastic approximation algorithm is

$$x_{k+1} = x_k + \eta_k [h(x_k) + \xi_{k+1}],$$

(2)

where $h : \mathbb{R}^n \to \mathbb{R}^n$ is Lipschitz, $\{\eta_k\}_{k \geq 0}$ is a positive stepsize sequence satisfying $\sum_{k \geq 0} \eta_k = \infty$ and $\sum_{k \geq 0} (\eta_k)^2 < \infty$, and $\xi_{k+1}$ represents noise. As $\eta_k \to 0$, (2) can be viewed as a noisy discretization of the o.d.e.

$$\dot{x}(t) = h(x(t)).$$

(3)

This is the ‘o.d.e. approach’ [12,24]. More specifically, suppose that the following assumptions hold.

(A1) $\{\xi_k\}$ is a square-integrable martingale difference sequence w.r.t. the $\sigma$-fields $\{\mathcal{F}_k\}, \mathcal{F}_k := \sigma(x_0, \xi_1, \ldots, \xi_k),$ satisfying $E[||\xi_{k+1}||^2 | \mathcal{F}_k] \leq L(1 + ||x_k||^2)$ a.s. for some $L > 0$.

(A2) $\forall u, h_{\infty}(u) := \lim_{t \to \infty} h(tu)/c$ exists ($h_{\infty}$ will be necessarily Lipschitz) and the o.d.e. $\dot{x}(t) = h_{\infty}(x(t))$ has origin as its globally asymptotically stable equilibrium.

(A3) $H := \{x \in \mathbb{R}^n : h(x) = 0\} \neq \emptyset$. Also, $\exists$ a continuously differentiable Lyapunov function $\mathcal{L} : \mathbb{R}^n \to \mathbb{R}$ such that $\langle \nabla \mathcal{L}(x), h(x) \rangle < 0$ for $x \notin H$.

Then, as in Chapters 2.3 of [7], we have:

Lemma 1 The iterates $\{x_k\}$ of (2) a.s. converge to $H$. 

3.2 Kaczmarz algorithm

Consider the inverse problem of finding a fixed $v^* \in \mathbb{R}^N$ from $Av^*$, where $A$ is as defined in Section 2. W.l.o.g., let rows of $A$ be of unit norm. Given an approximation $x_0$ of $v^*$, a natural optimization problem to consider is

$$\min_{u \in \mathbb{R}^N} \|u - x_0\|, \text{ subject to } Au = Av^*. \tag{4}$$

Elementary calculation shows that its solution is

$$x^* = x_0 + A'(AA')^{-1}(Av^* - Ax_0). \tag{5}$$

Clearly, $x^* \in A^0 := x_0 + \mathcal{R}_A$. As $A$ has full row rank, $x^*$ is the only point in $A^0$ that satisfies $Au = Av^*$. The Kaczmarz algorithm uses this fact to solve (4).

With prescribed initial point $x_0$, stepsize $\kappa$, and $r_k \equiv (k \mod m) + 1$, its update rule is given by

$$x_{k+1} = x_k + \kappa(\langle a_{r_k}, v^* \rangle - \langle a_{r_k}, x_k \rangle)a_{r_k}. \tag{6}$$

**Theorem 1** [10] If $0 < \kappa < 2$, then $x_k \to x^*$ as $k \to \infty$.

Let $A^* := v^* + \mathcal{R}_A$. Since $A^0$, $A^*$ are translations of $\mathcal{R}_A$, $\text{dist}(x_0, A^*) = \text{dist}(A^0, v^*)$. As $A(x^* - v^*) = 0$, $(x^* - v^*) \perp \mathcal{R}_A$. Thus, $(x^* - v^*) \perp A^0, A^*$. Hence, $\|v^* - x^*\| = \text{dist}(A^0, v^*) = \text{dist}(x_0, A^*)$. Thus we have:

**Lemma 2** For any $\delta > 0$, $\|x^* - v^*\| < \delta$ if and only if $\text{dist}(x_0, A^*) < \delta$.

4 The SAK Algorithm

We develop here a SAK algorithm to estimate $EX$ for the model of Section 2. Let $x_0$, an approximation to $EX$, be given. Observe from (1) that

$$EX = AEX. \tag{7}$$

By rescaling equations, we assume w.l.o.g. that the rows of $A$ are of unit norm. This saves some notation without affecting the analysis. $EX$ not being known exactly, one may estimate it off-line and use the classical Kaczmarz to determine $EX$. From (5), note that the classical Kaczmarz would have converged to

$$x^* = x_0 + A'(AA')^{-1}(EY - Ax_0). \tag{8}$$

As against this off-line scheme, a better alternative is to use an on-line algorithm. Using the notations and assumptions of Section 2, a SAK algorithm to estimate $EX$, based on (6), is:

$$x_{k+1} = x_k + \eta_k [\tilde{Y}_{k+1} - \langle a_{Z_{k+1}}, x_k \rangle]a_{Z_{k+1}}, \tag{9}$$

where $\{\eta_k\}$ is as defined below (2). Note in (9) the noisy measurements $\{\tilde{Y}_k\}$ of the elements of $EY$ and the real time estimates $\{x_k\}$ of $EX$.

We now analyze its behaviour. Clearly, the iterates $\{x_k\}$ of (9) always remain confined to $A^0$, the affine space defined below (5). Since $A$ has full row rank, for each $k \geq 0$, there exists unique $\alpha_k \in \mathbb{R}^m$ such that

$$x_k = x_0 + A'\alpha_k. \tag{10}$$

Thus one can equivalently analyze the algorithm

$$\alpha_{k+1} = \alpha_k + \eta_k (\tilde{Y}_{k+1} - e_{Z_{k+1}}A(x_0 + A'\alpha_k)), \tag{11}$$

where $e_0 = 0, e_{Z_{k+1}}$ is the $m \times m$ matrix with 1 in its $Z_{k+1}$-th diagonal position and zero elsewhere and $\tilde{Y}_{k+1}$ is the $m$-dimensional vector with its $Z_{k+1}$-th position occupied by $\tilde{Y}_{k+1}$ and zero elsewhere.

Let $\gamma_{k+1} = [\tilde{Y}_{k+1} - e_{Z_{k+1}}, A(x_0 + A'\alpha_k)]$.

Denoting $\xi_k = \gamma_{k+1} - \Lambda (EY - A(x_0 + A'\alpha_k))$, where $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_m)$, note that (11) can be rewritten as

$$\alpha_{k+1} = \alpha_k + \eta_k [\Lambda (EY - A(x_0 + A'\alpha_k)) + \xi_{k+1}]. \tag{12}$$

If $h(u) := \Lambda (EY - A(x_0 + A'\alpha)), \theta$, then, clearly, (12) is in the form given in (2). Its limiting o.d.e. is thus

$$\dot{\alpha}(t) = \Lambda (EY - A(x_0 + A'\alpha(t))). \tag{13}$$

**Theorem 2** $\alpha_k \xrightarrow{k \to \infty} \alpha^* := (AA')^{-1}(EY - Ax_0)$.

**Proof.** For each $k \geq 0$, let $F_k := \sigma(\alpha_0, \xi_1, \ldots, \xi_k)$. Lipschitz property of $h$ and (A1) are easily verified. If $h_c(u) := h(cu)/c$, then $h_c(u) \xrightarrow{c \to 0} h(u) := -\Lambda A'\alpha$, pointwise. Let $L_\infty(u) := \|A'\alpha\|^2$. This vanishes only at the origin. Further, for any solution to the o.d.e. $\dot{\alpha}(t) = -\Lambda A'\alpha(t), L_\infty(\alpha(t)) = -2\|\sqrt{\Lambda} A'\alpha(t)\|^2 \leq 0$, again with equality only at the origin. Thus $L_\infty$ is a Lyapunov function for the o.d.e. $\dot{\alpha}(t) = -h_\infty(\alpha(t))$ with the origin as its globally asymptotically stable equilibrium. Thus (A2) holds. By Lemma 1, to exhibit $\alpha_k \to \alpha^*$, it now suffices to show (A3), i.e. $\alpha^*$ is the globally asymptotically stable equilibrium of the o.d.e. given in (13). Towards this, consider the function $L(u) = \|A'(u - \alpha^*)\|^2$. As $A$ has full row rank, $L(u) = 0$ if and only if $u = \alpha^*$. For any solution $\alpha(t)$ of (13), $L(\alpha(t)) = 2\|A'(\alpha(t) - \alpha^*)\|^2$. But $\dot{\alpha}(t) = -\Lambda A'\alpha(t) - \alpha^*$. Thus $\dot{L}(\alpha(t)) \leq 0$ with equality only when $\alpha(t) = \alpha^*$. This shows that $L$ is a Lyapunov function. Thus $\alpha^*$ is the sole globally asymptotically stable equilibrium of (13) as desired.

Because of (10), it follows that the SAK algorithm of (9) converges to $x^*$ of (8), the same point that the corresponding classical Kaczmarz converges to.

5 Extensions

(1) We had assumed $\{Z_k\}$ to be IID. The final result, however, can be established under much more gen-
eral conditions. For example, \( \{ Z_k \} \) can be:

- ergodic Markov, as in Part II, Chapter 1, [6], with \( \lambda_i \)'s the corresponding stationary probabilities.
- asymptotically stationary as in Chapter 6, [21],
- ‘controlled’ Markov, as in Chapter 6, [7], which allows for non-stationarity under the mild restriction that the relative frequency of \( Z_k = i \) remain bounded away from zero a.s. \( \forall i \).

Likewise, \( \{ X_k \} \) can be ergodic Markov or asymptotically stationary as long as it is independent of \( \{ Z_k \} \). In fact, we can allow it to be long range dependent and heavy tailed [3], which is often the case with real communications networks.

(2) The above analysis was for estimation of means. We can also extend it to cover higher moments. For simplicity, we neglect measurement noise from (1) and consider the model \( Y = AX \). Observe that, for any \( q \in \mathbb{N} \) and each \( i \in [m] \),

\[
[Y(i)]^q = \sum_{r \in \Delta_{N,q}} \left( \prod_{i=1}^N [a_iX(i)]^r \right) \prod_{j=1}^r (q_{ij})^{r_j},
\]

where \( r \equiv (r_1, \ldots, r_N) \), \( q_{ij} = \frac{q_{ij}}{1 - r_i - r_N} \) and \( \Delta_{N,q} = \{ r \in \mathbb{Z}_+^N : \sum_i r_i = q \} \). Let \( X^q \) denote (\( \prod_{i=1}^N X(i)^{r_i} \) : \( r \in \Delta_{N,q} \))', \( A^q \) a \( N + q - 1 \)-dimensional vector, and let \( Y^q \equiv (Y(1))^q, \ldots, (Y(m))^q \)'. Also, let \( A^q \) denote the \( m \times (N + q - 1) \) matrix, whose \( (i,j) \)-th entry is the coefficient associated with \( j \)-th component of \( X^q \) as given in (14). The set of relations in (14) can thus be compactly written as

\[
Y^q = A^q X^q.
\]

Note that \( A^q \) is generically full row rank. Hence, (15) is of the same spirit as (1). One can thus use (9), after replacing samples of \( Y_i \) with those of \( Y^q_i \) and \( A \) with \( A^q \), to estimate in real-time \( E(\prod_{j=1}^N X_j^{q_j}) \) for any \( r \in \Delta_{N,q} \). For desired \( r \), the only condition one needs to ensure is that if \( r_{ij}, \ldots, r_{ij} \) are the components of \( r \) that are positive, then \( \exists i \in [m] \) such that \( a_{ij}, \ldots, a_{ij} \) are simultaneously nonzero. Clearly, by choosing appropriate \( q \) in (14), one can estimate the moments of any desired order.

Given finite moment estimates, one can then postulate a maximum entropy distribution. For e.g., if \( E[\|X\|^2] \approx a, E[\|X\|^4] \approx b \), then the maximum entropy distribution is \( \rho^{-1} \exp\left(-\alpha \|x\|^2 + \beta \|x\|^4\right) \), where \( \rho \) is for normalization and \( \alpha, \beta \) are chosen so as to ensure \( E[\|X\|^2] = a, E[\|X\|^4] = b \).

(3) We have taken the process \( \{ Z_k \} \) as given, i.e., not within our control. If instead one can schedule \( \{ Z_k \} \), randomization policies such as [34, 23, 17, 38] can be used to advantage. Further performance improvements are possible by adapting additional averaging as in [28]. We do not pursue this here.

(4) Since (12) is of the form \( \alpha_k = D_k - \eta_k (b + \xi_{k+1}) \) for \( D_k := I - \eta_k \Lambda A' \), suitable vector \( b \) and martingale difference noise \( \xi_k \), we can iterate this to obtain \( \alpha_k = \prod_{m=0}^{k-1} D_m \alpha_0 + \sum_{m=0}^{k-1} \eta_m \prod_{\ell=m+1}^{k-1} D_\ell (b + \xi_{m+1}) \). Note that the matrix \( \Lambda A' \) is similar to the positive definite matrix \( \sqrt{\Lambda} A A' \). Hence, if \( \zeta > 0 \) denotes the minimum eigenvalue of \( \sqrt{\Lambda} A A' \), then \( \prod_{m=0}^{k-1} D_m \lambda \leq \prod_{m=0}^{k-1} (1 - \eta_m \zeta) \leq e^{-\zeta (\sum_{m=0}^k \eta_m)} \), where \( \lambda \) denotes the weighted norm defined by \( \| r \|_\lambda := \left( \sum \frac{r_i^2}{\lambda_i} \right)^{\frac{1}{2}} \). This can be used to obtain estimates for finite time error and convergence rate.

6 Experimental Results

We illustrate the application of SAK algorithm in real time delay tomography for the network of Figure 1. The goal here is to use the measurements of end-to-end delay experienced by probe packets while traversing different paths in the network to obtain, in real time, the estimates of link delay statistics.

In the framework of Section 2, the experimental setup is as follows. A priori we choose six paths in the network. This is described by the path-link matrix (rows \( \equiv \) paths,
columns $\equiv$ links)

$$A = \begin{bmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & \\
\end{bmatrix}.$$  

Its entry $a_{ij}$ is one if link $j$ is present on path $i$. Thus, row four denotes the path that connects the nodes 2-8-11-12-4. The delay a probe packet experiences while traversing link $j$ is a random variable $X(j)$ with arbitrary non-negative distribution. The delay across path $i$ is $Y(i) = \langle a_i, X \rangle + W(i)$, where $W(1), \ldots, W(6)$ are IID standard Gaussian random variables denoting measurement error. We generate a million probe packets, where the $k$th packet is sent along a path whose index, denoted $Z_k$, is chosen uniformly randomly from $\{1, \ldots, 6\}$. Thus each path gets about 167,000 samples. We use $\hat{Y}_k$ to record the delay, packet $k$ experiences while traversing the path $Z_k$. We also run our SAK Algorithm of (9) for a million iterations, first for (1) and then for (15) with $q = 2$. The chosen start point, the actual value and final estimated value of moments are given in Tables 1 and 2. In both cases the initial point satisfies the assumption of Lemma 2 and hence the final estimates are close to the actual values. In Table 2 we give only a subset of results. Figure 2 compares the real-time estimates of expected delay for candidate links 1 and 3 obtained using the SAK algorithm and the averaged SAK algorithm. The iterates of the averaged SAK algorithm are samples averages of the SAK algorithm iterates. Observe that, although we run the simulation for a million packets, the estimates are very nearly the true values in after about 300 iterations. Also, note that the error in the estimates does not decrease monotonically. This is because of the direct use of noisy measurements. The fluctuations, however, get suppressed as the stepsizes decrease with iterations.

Table 1

| LinkId | Initial guess | True expected delay | Final estimate |
|--------|--------------|-------------------|---------------|
| 1      | 00.00        | 50.25             | 45.09         |
| 2      | 00.00        | 26.32             | 33.17         |
| 3      | 12.15        | 41.84             | 39.96         |
| 4      | 00.00        | 09.10             | 11.92         |
| 5      | 25.34        | 23.04             | 19.98         |
| 6      | 00.00        | 48.08             | 46.87         |
| 7      | 00.00        | 41.49             | 39.05         |
| 8      | 00.00        | 49.75             | 50.97         |
| 9      | 00.00        | 34.72             | 37.34         |
| 10     | 00.00        | 03.78             | 07.82         |
| 11     | 28.86        | 44.05             | 42.06         |
| 12     | 39.90        | 48.54             | 53.54         |
| 13     | 00.00        | 29.07             | 26.82         |

Table 2

| Moment | Initial guess | True value | Final estimate |
|--------|--------------|------------|---------------|
| $E(X^2_1)$ | 17388 | 20539 | 20570 |
| $E(X^2_2)$ | 0 | 277.85 | 286.29 |
| $E(X_3X_{10})$ | 15985 | 158.83 | 164.34 |
| $E(X_8X_{12})$ | -126 | 2427.8 | 2390.5 |

References

[1] T. Abrahamsson. Estimation of origin-destination matrices using traffic counts–a literature survey. IIASA Interim Report
