Notes on Randomized Kaczmarz

Mark Schmidt

April 9, 2015

1 Problem Definition

- **Input**: we are given m equalities $a_i^T x = b_i$, where each $b_i \in \mathbb{R}$ and each $a_i \in \mathbb{R}^n$ for $i = 1, 2, \ldots, m$.
- **Output**: a point $x_* \in \mathbb{R}^n$ that satisfies all m inequalities, $a_i^T x_* = b_i$ (we assume such a point exists).

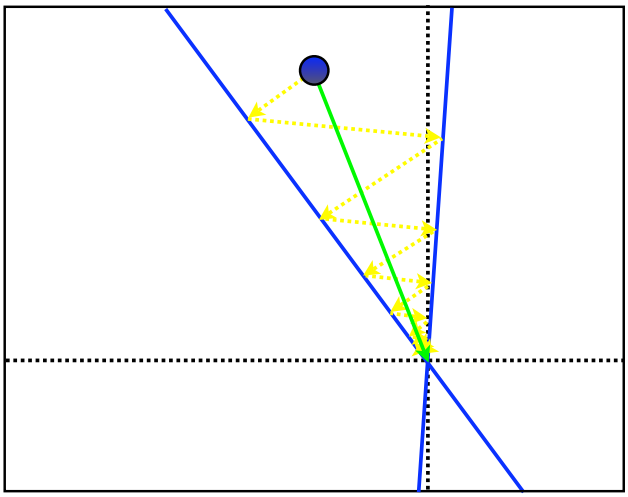
Written in matrix form, we want solve a linear system $Ax = b$. In this notation, element i of b is given by b_i (so $b \in \mathbb{R}^m$) and each row i of A is given by a_i^T (so $A \in \mathbb{R}^{m \times n}$).

2 Kaczmarz Algorithm

The Kaczmarz algorithm requires an initial guess x_0 and generates a sequence of iterations $\{x_k\}$ that converge to x_*. Specifically, given x_k we generate x_{k+1} as the solution to the problem

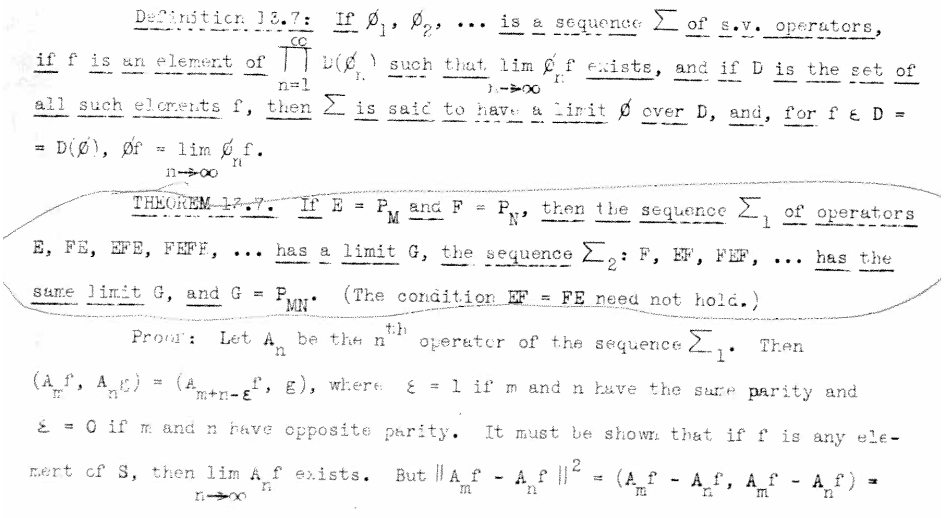
$$x_{k+1} = \underset{\{x \mid a_i^T x = b_i\}}{\operatorname{argmin}} \|x - x_k\|.$$

That is, the algorithm sets x_{k+1} to the closest value to x_k that satisfies the constraint $a_i^T x_{k+1} = b_i$. We call this operation the 'projection' of x_k onto the set $\{x \mid a_i^T x_{k+1} = b_i\}$. As long make sure we never stop selecting any i, the algorithm converges to x^*.
Below is a picture of the algorithm in action. The blue point is x_0, the yellow lines show the sequence of projections, and the green line shows the overall trend towards the intersection of the two lines.



2.1 History of the method

According to Wikipedia, the method was proposed in 1937 by Polish mathematician Stefan Kaczmarz. It is a very popular approach in the field of image reconstruction, where it was re-invented by Gordon et al. under the name algebraic reconstruction technique (ART). They are also sometimes called 'row-action' method, since each iteration only involves one row of the matrix. Another name used by Bertsekas is 'component-solution' methods and you may hear the expressions 'cyclic projection' or 'successive projection' (for obvious reasons). The Kaczmarz algorithm is also closely-related to an earlier result in some lecture notes of von Neumann originally distributed in 1933 but published in 1950. That work considers two subspaces, and shows that alternately projecting onto the two subspaces converges to the projection of the initial point onto the intersection of the subspaces. You can get the published version from the UBC library, I've scanned the result below so that you can appreciate modern typesetting:



Definition 13.7: If $\phi_1, \phi_2, \ldots$ is a sequence $\sum$ of s.v. operators, if f is an element of $\prod_{n=1}^{\infty} D(\phi_n)$ such that $\lim_{n\to\infty} \phi_n f$ exists, and if D is the set of all such elements f, then $\sum$ is said to have a limit ϕ over D, and, for $f \in D = D(\phi)$, $\phi f = \lim_{n\to\infty} \phi_n f$.

THEOREM 13.7. If $E = P_M$ and $F = P_N$, then the sequence $\sum_1$ of operators E, FE, EFE, FEFE, ... has a limit G, the sequence $\sum_2$: F, EF, FEF, ... has the same limit G, and $G = P_{MN}$. (The condition EF = FE need not hold.)

Proof: Let A_n be the n^{th} operator of the sequence $\sum_1$. Then $(A_m f, A_n g) = (A_{m+n-\varepsilon} f, g)$, where $\varepsilon = 1$ if m and n have the same parity and $\varepsilon = 0$ if m and n have opposite parity. It must be shown that if f is any element of S, then $\lim_{n\to\infty} A_n f$ exists. But $\| A_m f - A_n f \|^2 = (A_m f - A_n f, A_m f - A_n f) =$

There are bunch of interesting generalizations of the Kaczmarz method. For example, there are variants that try to speed up the convergence rate, there are variants that allow linear inequalities instead of just equalities, there are variants that find a point in the intersection of convex sets, there are variants that use a general Bregman divergences instead of the Euclidean norm, there are variants that solve certain optimization problems, and there are variants that do the projections in parallel. An informal survey of these methods is given in Section 5 of my notes on 'big-n' problems (where you can also get the citations mentioned in this document):
`http://www.cs.ubc.ca/~schmidtm/Documents/2012_Notes_BigN.pdf`

2.2 Should we use this method? What does this have to do with randomization?

There are an enormous number of ways to solve linear systems, so how does this compare to other strategies? The key advantage of the method is that the iterations are extremely simple and cheap, you only do operations with one row of the matrix (and the iterations are faster when a_i is sparse). This is especially appealing with the number of rows m is very large, since the iteration cost is actually independent of m. This is a key advantage over most of the standard strategies for solving linear systems.
However, in order for the method to be useful we need to know that it doesn't need too many iterations to reach an accurate solution. Work by Deutsch & Hundal in the 1980s and 1990s, as well as Galantai during the 2000s show that the convergence rate depends on a certain measure of the 'angle' between the sets (see the link above for the full references). If you look at the figure from the previous page, you can get an intuition for why this angle will affect the convergence rate; in that figure you would go faster if the lines

where closer to perpendicular and you would go slower if the lines were closer to being parallel. Although these works are very impressive, it is hard to compare the angle criterion to the convergence rates of other methods for solving linear systems which depend on things like the singular values of the matrix A.
In 2009, Strohmer and Vershynin analyzed the convergence rate of the Kaczmarz algorithm with random selection of the rows i. Using random selection was not new and was being used in practice for decades before this work, but with a simple and elegant argument Strohmer and Vershynin showed that randomized selection achieves a fast convergence rate that depends on quantities that are more-closely related to the singular values of the matrix A. This nice result has got a lot of attention, and led to a variety of interesting works in this area (including being an inspiration for some of my own work).

3 Randomized Kaczmarz

Let $\sigma_j(A)$ denote singular value j of A, organized in decreasing order so that $\sigma_1(A) \geq \sigma_2(A) \geq \ldots \sigma_{\min\{m,n\}}(A)$. In their analysis, Strohmer and Vershynin use the extra assumption that $\sigma_n(A) > 0$. This implies that $m \geq n$, that A has n independent columns, and that there can be at most one solution x_* (recall that we assumed at least one solution exists). Note that these assumptions are not actually needed to show convergence of the Kaczmarz algorithm, but as far as I know there is no convergence rate analysis of the randomized algorithm without this assumption.

3.1 Simple form of the iteration

First, we'll derive a nicer formula for x^{k+1}. By noting that squaring the (non-negative) objective and dividing it by 2 does not change the argmin, we re-write the iteration as the solution to the problem

$$x_{k+1} = \underset{\{x|a_i^T x = b_i\}}{\operatorname{argmin}} \frac{1}{2}\|x - x_k\|^2,$$

which is strongly-convex (implying that the solution is unique). To derive the solution of this problem, we write the Lagrangian as

$$L(x, \lambda) = \frac{1}{2}\|x - x_k\|^2 + \lambda(a_i^T x - b_i).$$

The solution of the problem is the stationary point of the Lagrangian. Taking the gradient and equating its components to zero gives

$$\begin{aligned} \nabla_x L(x, \lambda) &= x_{k+1} - x_k + \lambda a_i = 0, \\ \nabla_\lambda L(x, \lambda) &= a_i^T x_{k+1} - b_i = 0. \end{aligned}$$

From the first equality, we have

$$x_{k+1} = x_k - \lambda a_i,$$

and plugging this into the second equality gives

$$a_i^T(x_k - \lambda a_i) - b_i = 0,$$

which means that $\lambda = \frac{a_i^T x_k - b_i}{a_i^T a_i} = \frac{a_i^T x_k - b_i}{\|a_i\|^2}$ and the iteration can be written as

$$x_{k+1} = x_k - \frac{a_i^T x_k - b_i}{\|a_i\|^2} a_i, \tag{1}$$

which is clearly quite simple to implement.

3.2 Linear Convergence and Outline of Proof

We will derive a bound of the form

$$\mathbb{E}[\|x_k - x_*\|^2] \leq \rho^k \|x_0 - x_*\|^2,$$

for some $\rho < 1$. This shows that the expected error is cut by a fixed fraction on each iteration. This is called 'linear convergence', 'geometric convergence' (because the error goes down as a geometric sequence), or 'exponential convergence'. The term 'exponential' convergence comes from writing $\rho = (1 - \delta)$ for some $\delta < 1$ and then using the convexity inequality $1 - \delta \leq e^{-\delta}$ to get $(1 - \delta)^k = \exp(-\delta k)$. This term can be somewhat misleading because 'exponential' convergence sounds much better than 'linear' convergence, but they are basically the same.

This convergence rate implies that the number of iterations to reach an accuracy of ϵ is at most $O(\log(1/\epsilon))$. It's not as fast as the superlinear convergence of some other methods, but on the other hand this is probably the best we can expect from a method that only looks at one row at a time. To show a rate like the above, we will first show that

$$\mathbb{E}[\|x_k - x_*\|^2] \leq \rho \|x_{k-1} - x_*\|^2,$$

where the expectation is only taken with respect to the selection of the random i on iteration k. We then altnernate between taking expectations with respect to previous random choices and applying the inequality until we get back to x_0,

$$\begin{aligned}
\mathbb{E}[\|x_k - x_*\|^2] &= \mathbb{E}[\mathbb{E}[\|x_k - x_*\|^2]] \\
&\leq \mathbb{E}[\rho \|x_{k-1} - x_*\|^2] \\
&= \rho \mathbb{E}[\|x_{k-1} - x_*\|^2] \\
&\leq \rho(\rho \|x_{k-2} - x_*\|^2) \\
&= \rho^2 \|x_{k-2} - x_*\|^2 \\
&\cdots \\
&\leq \rho^k \|x_0 - x_*\|^2.
\end{aligned}$$

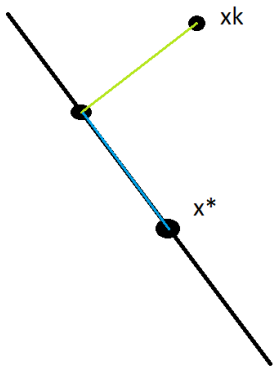
Technically, we are being really sloppy above in how we define the expectations and expectations of expectations, but you can do this more carefully by defining a sequence of σ-fields $\mathcal{F}_k$ and using the law of total expectation (also known as the tower rule) to formally show that the above reasoning is sound.

3.3 Measure of progress for arbitrary selection of i

We want to establish a relationship between $\|x_{k+1} - x_*\|$ and $\|x_k - x_*\|$, and a logical way to do this is to start from $\|x_k - x_*\|^2$ then add/subtract x_{k+1} inside the norm and expand the square,

$$\begin{aligned}
\|x_k - x_*\|^2 &= \|x_k - x_{k+1} + x_{k+1} - x_*\|^2 \\
&= \|(x_k - x_{k+1}) + (x_{k+1} - x_*)\|^2 \\
&= \|x_k - x_{k+1}\|^2 + \|x_{k+1} - x_*\|^2 - 2\langle x_k - x_{k+1}, x_{k+1} - x_* \rangle.
\end{aligned}$$

The first two terms on the right side are quantities we expect to show up, but the last term looks unpleasant. Fortunately, for this algorithm the last term is 0 because $(x_k - x_{k+1})$ and $(x_{k+1} - x_*)$ are orthogonal. Below is a picture showing why they will be orthogonal in two dimensions,



and we can show this formally by observing that $x_k - x_{k+1} = \gamma a_i$ for some scalar γ so we have

$$(x_k - x_{k+1})^T(x_{k+1} - x_*) = \gamma a_i^T(x_{k+1} - x_*) = \gamma(a_i^T x_{k+1} - a_i^T x_*) = \gamma(b_i - b_i) = 0,$$

where we used that both x_{k+1} and x_* solve equality i. Using that this inner product is zero and re-arranging, we get

$$\|x_{k+1} - x_*\|^2 = \|x_k - x_*\|^2 - \|x_{k+1} - x_k\|^2. \tag{2}$$

This holds for any choice of the hyper-plane i to project onto and is almost what we want. Now we just need to make that the quantity $\|x_{k+1} - x_k\|$ is big enough to guarantee a fixed amount of progress, and we can do this by an appropriate selection of i.

3.4 Linear convergence with uniform sampling

The simplest randomized scheme for selecting i is to choose each possible row i with probability $1/m$. Let's take the expectation of (2) with this choice,

$$\begin{aligned}
\mathbb{E}[\|x_{k+1} - x_*\|] &= \|x_k - x_*\|^2 - \mathbb{E}[\|x_{k+1} - x_k\|^2] \\
&= \|x_k - x_*\|^2 - \mathbb{E}\left[\left\|\frac{a_i^T x_k - b_i}{\|a_i\|^2} a_i\right\|^2\right] && \text{definition of iteration, Equation (1)} \\
&= \|x_k - x_*\|^2 - \mathbb{E}\left[\frac{(a_i^T x_k - b_i)^2}{\|a_i\|^4}\|a_i\|^2\right] && \text{take scalars outside norm} \\
&= \|x_k - x_*\|^2 - \mathbb{E}\left[\frac{(a_i^T x_k - a_i^T x_*)^2}{\|a_i\|^2}\right] && \text{cancel terms and use } a_i^T x_* = b_i \\
(*) \qquad &= \|x_k - x_*\|^2 - \sum_{i=1}^m \frac{1}{m}\frac{(a_i^T(x_k - x_*))^2}{\|a_i\|^2} && \text{definition of expectation} \\
&\leq \|x_k - x_*\|^2 - \sum_{i=1}^m \frac{1}{m}\frac{(a_i^T(x_k - x_*))^2}{\|A\|_{\infty,2}^2} && \text{defining } \|A\|_{\infty,2}^2 \triangleq \max_i\{\|a_i\|^2\} \\
&= \|x_k - x_*\|^2 - \frac{1}{m\|A\|_{\infty,2}^2}\sum_{i=1}^m (a_i^T(x_k - x_*))^2 && \|A\|_{\infty,2} \text{ does not depend on } i \\
&= \|x_k - x_*\|^2 - \frac{1}{m\|A\|_{\infty,2}^2}\|A(x_k - x_*)\|^2 && \|Az\|^2 = \sum_{i=1}^m (a_i^T z)^2 \\
&= \left(1 - \frac{1}{m\|A\|_{\infty,2}^2}\frac{\|A(x_k - x_*)\|^2}{\|x_k - x_*\|^2}\right)\|x_k - x_*\|^2 && \text{common factor of } \|x_k - x_*\|^2
\end{aligned}$$

Note that at this point, we haven't made any assumption about A except that an x_* exists. In particular, we haven't yet used any assumption that x_* is unique, so the above inequality actually holds for any solution x_*. To get the final expression we now use that

$$\frac{||z||}{||Az||} \leq \sup_{Ax \neq 0} \frac{||x||}{||Ax||} = \frac{1}{\sigma_n(A)}.$$

and thus if we also have $z \neq 0$ that

$$\frac{\|Az\|}{\|z\|} \geq \sigma_n(A).$$

We can do the substituion $z = x_k - x_*$ and use the inequality above if we assume that $x_k \neq x_*$ and $Ax_k \neq Ax_*$ (this is reasonable since if either of these conditions are true then x_k solves the problem and the bound we want to show holds trivially), which gives the final result

$$\mathbb{E}[\|x_{k+1} - x_*\|] \leq \left(1 - \frac{\sigma_n(A)^2}{m\|A\|_{\infty,2}^2}\right)\|x_k - x_*\|^2.$$

Since we require $\rho < 1$ for the rate to be meaningful, we need $\sigma_n(A)$ to satisfy

$$0 < \sigma_n(A)^2 \leq m\|A\|_{\infty,2}^2.$$

The lower bound holds because we assumed $\sigma_n(A) > 0$, and the upper bound is true because $\sigma_n(A) \leq \sigma_1(A) \leq \|A\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n a_{ij}^2} \leq \sqrt{m \max_i\{\sum_{j=1}^n a_{ij}^2\}} = \sqrt{m}\|A\|_{\infty,2}$, where we use that the Frobenius norm

$$\|A\|_F \triangleq \sqrt{\sum_{i=1}^m \sum_{j=1}^n a_{ij}^2} = \sqrt{\text{Tr}(A^T A)} = \sqrt{\sum_{i=1}^{\min\{m,n\}} \sigma_i(A)^2},$$

is always at least as large $\sigma_1(A)$.

3.5 Linear convergence with non-uniform sampling

Instead of sampling each row i with probability $1/m$, Strohmer and Verhsynin consider sampling each row with probability $\frac{\|a_i\|^2}{\sum_{j=1}^m \|a_j\|^2} = \frac{\|a_i\|^2}{\|A\|_F^2}$. Using this sampling strategy, the argument is the same up to the line marked $(*)$, and at this line you will get an $\|a_i\|^2$ term in the numerator that cancels the corresponding term in the denominator (and you will get $\|A\|_F^2$ in the denominator instead of m). This avoids the need to use inequality the $\|a_i\| \leq \|A\|_{\infty,2}$. Proceeding as before, you will eventually get to the rate

$$\mathbb{E}[\|x_{k+1} - x_*\|] \leq \left(1 - \frac{\sigma_n(A)^2}{\|A\|_F^2}\right)\|x_k - x_*\|^2.$$

We have $\|A\|_F^2 \leq m\|A\|_{\infty,2}^2$ so this is always at least as fast as the uniform sampling strategy, and it will be faster if any two rows don't have the same norm.

If you think it's weird to sample the rows proportional to their norms, you would be right. The solution is invariant to scaling of the row norms, as long as you scale the corresponding elements of b by the same amounts, so it's strange that the algorithm depends on the row norms. Also note that instead of sampling non-uniformly, you could scale each row so that it has a norm of one, and then do uniform sampling. But note that this changes $\sigma_n(A)^2$.