IAS SUMMER COLLABORATORS REPORT: PARTIAL INFORMATION METHODS FOR LARGE-SCALE LINEAR SYSTEMS

JAMIE HADDOCK
Iris & Howard Critchell Assistant Professor, Department of Mathematics, Harvey Mudd College

ANNA MA
Assistant Professor, Department of Mathematics, University of California, Irvine

With the computational advances of recent decades, the size and complexity of datasets regularly analyzed and employed in learning pipelines have skyrocketed. Consequently, scientists continue searching for better, faster, and more efficient tools for using increasingly high-dimensional data. At the core of many scientific computing and learning applications are subroutines that require approximating solutions to linear systems. Traditionally, direct methods were frequently used for solving linear systems. However, these direct approaches quickly become impractical, making way for memory-efficient iterative approaches. Iterative approaches such as the Kaczmarz [2, 4] or Gauss-Seidel [3] algorithms require only one row or column of the matrix at a time to iteratively approximate the solution to linear systems $A\boldsymbol{x} = \boldsymbol{y}$ for $A \in \mathbb{R}^{m\times n}$. This can be immensely beneficial when m and n are very large. For example, suppose a computer has a limited storage of approximately $\mathcal{O}(g)$ bits. If $mn \gg g$, the matrix is too large to load into memory, making it impossible to utilize direct methods. As long as $m, n < g$, row or column action methods can be used.

Row and column action methods, such as the ones mentioned above, have a limitation in that they require the full row or column of a matrix per iteration. What happens if both $m, n > g$, i.e., when neither entire rows nor entire columns can be utilized?

1. Progress

During our visit, we considered the case where the measurement matrix A is partitioned into row and column blocks, and one only has access to a single subblock in each iteration. Let $A\boldsymbol{x} = \boldsymbol{y}$, such that $A \in \mathbb{R}^{m\times n}$ $m \gg n$ denote a consistent linear system. Let $\mathcal{R} \subseteq [m]$ and $\mathcal{C} \subseteq [n]$ to denote subsets of row and column indices and let $I_{\mathcal{R}} \in \{0,1\}^{m\times m}$ (resp., $I_{\mathcal{C}} \in \{0,1\}^{n\times n}$) be a diagonal matrix with ones in the $(i,i)^{th}$ entry for all $i \in \mathcal{R}$ (resp., $i \in \mathcal{C}$). Let $\mathcal{R}_1, ..., \mathcal{R}_k$ and $\mathcal{C}_1, ..., \mathcal{C}_\ell$ denote a partitioning of $[m]$ and $[n]$, respectively.

Our work at IAS considered two methods. The first method encompasses classical iterative update methods, including Kaczmarz, Gauss-Seidel, and their average block variants [1], given special choices of row and/or column partitions.

Algorithm 1 General Iterative Framework

procedure METHOD($A, \boldsymbol{y}, \{\mathcal{R}_i\}_{i=1}^k, \{\mathcal{C}_j\}_{j=1}^\ell, \alpha$)
  Initialize $\boldsymbol{x}_0 = \boldsymbol{0}$, $\boldsymbol{r}_0 = \boldsymbol{y} - A\boldsymbol{x}_0$
  for $t = 1, \ldots, T$ **do**
    Sample a row block $i \in [k]$, sample a column block $j \in [\ell]$, $V_{i,j,t} = I_{\mathcal{R}_i} A I_{\mathcal{C}_j}$
    $\boldsymbol{r}_t = \boldsymbol{r}_{t-1} - \alpha A V_{i,j,t}^\top \boldsymbol{r}_{t-1}$
    $\boldsymbol{x}_t = \boldsymbol{x}_{t-1} + \alpha V_{i,j,t}^\top \boldsymbol{r}_{t-1}$
  return $\boldsymbol{x}_T$

One challenge of working with this method is that in order to guarantee that the updates will converge to the solution of the system $A\boldsymbol{x} = \boldsymbol{y}$, we require that the updates maintain the correctness of the residual estimate $\boldsymbol{r}_t$. This is only guaranteed when either complete rows or full columns (or blocks of these) are used in each iteration. Thus, to develop methods for our setting of interest (when only partial information about

E-mail addresses: jhaddock@g.hmc.edu, anna.ma@uci.edu.

either rows or columns is available), we developed a new method that uses the iterate history $H(i,j,t)$ to correct the residual estimate. Our proposed algorithm is as follows.

Algorithm 2 Partial Information Gradient Descent

procedure PI-GD($A, \boldsymbol{y}, \{\mathcal{R}_i\}_{i=1}^k, \{\mathcal{C}_j\}_{j=1}^\ell, \alpha$)
  Initialize $\boldsymbol{x}_0 = \boldsymbol{0}$, $\boldsymbol{z}_0 = \boldsymbol{y} - A\boldsymbol{x}_0$
  for $t = 1, \dots, T$ **do**
    Sample a row block $i \in [k]$
    Sample a column block $j \in [\ell]$
    $V_{i,j,t} = I_{\mathcal{R}_i} A I_{\mathcal{C}_j}$
    $\boldsymbol{z}_t = \boldsymbol{z}_{t-1} - V_{i,j,t}\boldsymbol{x}_{t-1} + V_{i,j,t}\boldsymbol{x}_{H(i,j,t)}$
    $\boldsymbol{x}_t = \boldsymbol{x}_{t-1} + \alpha V_{i,j,t}^\top \boldsymbol{z}_t$
  return $\boldsymbol{x}_T$

We made progress on proving convergence guarantees for Algorithm 2. To start, we began by proving a relationship between the Pi-GD iterates $\boldsymbol{x}_t$ and classical gradient descent iterates $\boldsymbol{x}_t^{GD}$ where gradient descent is applied to the least squares objective $F(\boldsymbol{x}) = \frac{1}{2}\|A\boldsymbol{x} - \boldsymbol{y}\|^2$ under two simplifying assumptions. The first assumption is that sub-blocks of A, determined by $\mathcal{R}_i$ and $\mathcal{C}_j$, are selected in a deterministic and cyclic fashion. Doing so allows for easier control over the use of iterative history term $\boldsymbol{x}_{H(i,j,t)}$. The second assumption imposes a constraint on the partitioning of the matrix A into mutually orthogonal blocks. This assumption creates independence in the recursion of the proposed method, simplifying the computation. With these assumptions, we showed that after every kl iterations, the iterate coincides with one gradient descent update. In other words, Algorithm 2 can be interpreted as a delayed gradient descent method.

2. Future work

Going forward, we aim to extend our convergence analysis to settings with more relaxed assumptions. Our current analysis relies upon the assumptions that the blocks are provided to us with cyclic control (that is, we see each block precisely once in kl many iterations of Algorithm 2 and in the same order each kl iterations) and that the blocks obey a mutual orthogonality condition.

We have initial results relaxing the block mutual orthogonality assumption to a mild incoherence assumption, but we hope to explore the possibility of even more general settings.

As for relaxing the block cyclic control assumption, we expect that our analysis will extend to the case where the blocks are not provided entirely cyclically, but instead are provided so that each block is seen precisely once in kl iterations (although possibly in different orders within each kl-iteration block). We further hope to generalize to the setting where the blocks are uniformly randomly sampled. In this case, each block will be seen in kl iterations *in expectation*, but of course this will not be observed in any kl iterations with high probability. This setting is far more challenging and will require different analysis techniques than those we have developed.

Acknowledgments

We are immensely grateful to IAS for their support for our two-week stay. This was an invaluable time for in-person collaboration that has allowed us to push the project forward significantly. We would especially like to thank Michelle Huguenin, Madeleine Perez, and the rest of the staff who make stays like this so smooth!

References

[1] K. Du and X.-H. Sun. Pseudoinverse-free randomized block iterative algorithms for consistent and inconsistent linear systems. *arXiv preprint arXiv:2011.10353*, 2020.

[2] S. Kaczmarz. Angenäherte auflösung von systemen linearer gleichungen. *Bull. Int. Acad. Polon. Sci. Lett. Ser. A*, pages 335–357, 1937.

[3] D. Leventhal and A. S. Lewis. Randomized methods for linear constraints: convergence rates and conditioning. *Math. Oper. Res.*, 35(3):641–654, 2010.

[4] T. Strohmer and R. Vershynin. A randomized Kaczmarz algorithm with exponential convergence. *J. Fourier Anal. Appl.*, 15(2):262, 2009.