Numer. Math. Theor. Meth. Appl.
doi: 10.4208/nmtma.OA-2018-0066

Vol. **12**, No. 1, pp. 1-20
February 2019

Stochastic Gradient Descent for Linear Systems with Missing Data

Anna Ma[1,*]and Deanna Needell [2]

[1] *Claremont Graduate University, Claremont, CA 91711, USA*
[2] *University of California, Los Angeles, Los Angeles CA 90095, USA*

Received 25 May 2018; Accepted (in revised version) 21 July 2018

Abstract. Traditional methods for solving linear systems have quickly become impractical due to an increase in the size of available data. Utilizing massive amounts of data is further complicated when the data is incomplete or has missing entries. In this work, we address the obstacles presented when working with large data and incomplete data simultaneously. In particular, we propose to adapt the Stochastic Gradient Descent method to address missing data in linear systems. Our proposed algorithm, the Stochastic Gradient Descent for Missing Data method (mSGD), is introduced and theoretical convergence guarantees are provided. In addition, we include numerical experiments on simulated and real world data that demonstrate the usefulness of our method.

AMS subject classifications: 65F10, 65F20, 68W20

Key words: Linear systems, missing data, iterative methods, least squares problems.

1. Introduction

When handling large amounts of data, it may not be possible to load the entire matrix (data set) into memory, as typically required by matrix inversions or matrix factorization. This has led to the study and advancement of stochastic iterative methods with low memory footprints such as Stochastic Gradient Descent, Randomized Kaczmarz, and Randomized Gauss-Seidel [13, 16, 18, 23]. The need for algorithms that can process large amounts of information is further complicated by incomplete or missing data, which can arise due to, for example, attrition, errors in data recording, or cost of data acquisition. Standard methods for treating missing data, which include data imputation [6, 7], matrix completion [3, 11, 12, 19], and maximum likelihood estimation [5, 15] can be wasteful, create biases, or be impractical for extremely large amounts of data. This work simultaneously addresses both issues of large-scale and missing data.

[*] Corresponding author. *Email addresses:* a4ma@ucsd.edu (A. Ma), deanna@math.ucla.edu (D. Needell)

Consider the system of linear equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$[1], where $\boldsymbol{A} \in \mathbb{C}^{m\times n}$ is a large, full-rank, overdetermined ($m > n$) matrix. Suppose that $\boldsymbol{A}$ is not known entirely, but instead only some of its entries are available. As a concrete example, suppose $\boldsymbol{A}$ is the rating matrix from the survey of m users about n service questions, and $\boldsymbol{b}$ contains the m "overall" ratings from each user (which is fully known). Each user may not answer all of the individual service questions, but a company wishes to understand how each question affects the overall rating of the user. That is, given partial knowledge of $\boldsymbol{A}$, one wishes to uncover $\boldsymbol{x}_\star = \arg\min_{\boldsymbol{x}} \frac{1}{2m}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|^2$.

Let $\tilde{\boldsymbol{A}} = \boldsymbol{D} \circ \boldsymbol{A}$ where $\boldsymbol{A}$ denotes the full matrix, and $\circ$ be the element-wise product, $\boldsymbol{D}$ denotes a binary matrix (1 indicating the availability of an element and 0 indicating a missing entry). Formally, one wants to solve the following optimization program:

$$\begin{aligned} &\text{Given } \tilde{\boldsymbol{A}}, \boldsymbol{b} \quad \text{s.t. } \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \ \textit{and} \ \tilde{\boldsymbol{A}} = \boldsymbol{D} \circ \boldsymbol{A}, \\ &\text{Find } \boldsymbol{x}_\star = \arg\min_{\boldsymbol{x}\in\mathcal{W}} \frac{1}{2m}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|^2, \end{aligned} \tag{1.1}$$

where $\mathcal{W}$ is a convex domain containing the solution $\boldsymbol{x}_\star$ (e.g. a ball with large enough radius).

Contributions. This work presents a stochastic iterative projection method for solving large-scale linear systems with missing data. We provide theoretical bounds for the proposed method's performance and demonstrate its usefulness on simulated and real world data sets.

1.1. Stochastic Gradient Descent

Stochastic iterative methods such as Randomized Kaczmarz (RK) and Stochastic Gradient Descent (SGD) have gained interest in recent years due to their simplicity and ability to handle large-scale systems. Originally discussed in [20], SGD has proved to be particularly popular in machine learning [1, 2, 24]. SGD minimizes an objective function $F(\boldsymbol{x})$ over a convex domain $\mathcal{W}$ using unbiased estimates for the gradient of the objective, i.e., using $f_i(\boldsymbol{x})$ such that $\mathbb{E}[\nabla f_i(\boldsymbol{x})] = \nabla F(\boldsymbol{x})$. At each iteration, a random unbiased estimate, $\nabla f_i(\boldsymbol{x})$, is drawn and the minimizer of $F(\boldsymbol{x})$ is estimated with:

$$\boldsymbol{x}_k = \mathcal{P}_{\mathcal{W}}\left(\boldsymbol{x}_{k-1} - \alpha_k \nabla f_i(\boldsymbol{x}_{k-1})\right), \tag{1.2}$$

where α_k is an appropriately chosen step size, or learning rate, at iteration k and $\mathcal{P}_{\mathcal{W}}$ denotes the projection onto the convex set $\mathcal{W}$. To solve an overdetermined linear system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, one approach is to minimize the least-squares objective function $F(\boldsymbol{x}) = \frac{1}{2m}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|^2 = \frac{1}{m}\sum_{i=1}^m f_i(\boldsymbol{x})$, where $f_i(\boldsymbol{x}) = \frac{1}{2}(\boldsymbol{A}_i\boldsymbol{x} - \boldsymbol{b}_i)^2$, $\boldsymbol{A}_i$ denotes the i^{th} row of $\boldsymbol{A}$, and $\boldsymbol{b}_i$ denotes the i^{th} entry of $\boldsymbol{b}$. In this setting, a random *row* of the matrix $\boldsymbol{A}$ is selected and

[1] The linear system is not assumed to be consistent; we will use the notation $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ to denote a general linear system.