AltGDmin: Alternating GD and Minimization for Partly-decoupled (Federated) Optimization

Other titles in Foundations and Trends® in Optimization

Gradient-Based Algorithms for Zeroth-Order Optimization
Prashanth L. A. and Shalabh Bhatnagar
ISBN: 978-1-63828-544-1

Integer Programming Games
Margarida Carvalho, Gabriele Dragotto, Andrea Lodi and Sriram Sankaranarayanan
ISBN: 978-1-63828-516-8

Multi-agent Online Optimization
Deming Yuan, Alexandre Proutiere and Guodong Shi
ISBN: 978-1-63828-482-6

An Invitation to Deep Reinforcement Learning
Bernhard Jaeger and Andreas Geiger
ISBN: 978-1-63828-440-6

Constrained Reinforcement Learning with Average Reward Objective: Model-Based and Model-Free Algorithms
Vaneet Aggarwal, Washim Uddin Mondal and Qinbo Bai
ISBN: 978-1-63828-396-6

Stochastic Optimization Methods for Policy Evaluation in Reinforcement Learning
Yi Zhou and Shaocong Ma
ISBN: 978-1-63828-370-6

AltGDmin: Alternating GD and Minimization for Partly-decoupled (Federated) Optimization

Namrata Vaswani
Iowa State University
namrata@iastate.edu

the essence of knowledge
Boston — Delft

Foundations and Trends® in Optimization

Published, sold and distributed by:
now Publishers Inc.
PO Box 1024
Hanover, MA 02339
United States
Tel. +1-781-985-4510
www.nowpublishers.com
sales@nowpublishers.com

Outside North America:
now Publishers Inc.
PO Box 179
2600 AD Delft
The Netherlands
Tel. +31-6-51115274

The preferred citation for this publication is

N. Vaswani. *AltGDmin: Alternating GD and Minimization for Partly-decoupled (Federated) Optimization*. Foundations and Trends® in Optimization, vol. 8, no. 4, pp. 333–414, 2025.

ISBN: 978-1-63828-581-6

Foundations and Trends® in Optimization
Volume 8, Issue 4, 2025

Editorial Scope

Foundations and Trends® in Optimization publishes survey and tutorial articles in the following topics:

- algorithm design, analysis, and implementation (especially, on modern computing platforms
- models and modeling systems, new optimization formulations for practical problems
- applications of optimization in machine learning, statistics, and data analysis, signal and image processing, computational economics and finance, engineering design, scheduling and resource allocation, and other areas

Information for Librarians

Foundations and Trends® in Optimization, 2025, Volume 8, 4 issues. ISSN paper version 2167-3888. ISSN online version 2167-3918. Also available as a combined paper and online subscription.

Contents

1 Introduction 3
1.1 Partly Decoupled Optimization Examples 5

I Existing Optimization Solutions and AltGDmin 10

2 Commonly Used Optimization Algorithms 11
2.1 The Optimization Problem and its Federated Version . . . 11
2.2 Gradient Descent (GD) 12
2.3 Block Coordinate Descent and Alternating Minimization (AltMin) 13
2.4 Non-linear Least Squares (NLLS) 15
2.5 Algorithm Initialization 15

3 AltGDmin for Partly Decoupled Optimization Problems 17
3.1 Partly-decoupled Optimization: Precise Definition 17
3.2 Alternating GD and Minimization (AltGDmin) 18

II AltGDmin for Partly-decoupled Low Rank (LR) Recovery Problems: Algorithms and Guarantees **21**

4 AltGDmin for Three LR Matrix Recovery Problems **22**
4.1 Notation 22
4.2 LRCS, LRPR, and LRMC Problems 22
4.3 Federation 25
4.4 AltGDmin for LRCS: Algorithm and Guarantees 25
4.5 AltGDmin for LRPR: Algorithm and Guarantees 30
4.6 AltGDmin for LRMC: Algorithm and Guarantees 32

III AltGDmin Analysis – Overall Proof Technique and Details for LR Problems **36**

5 General Proof Approach for any Problem **37**

6 AltGDmin for LR Problems: Overall Proof Ideas **40**
6.1 AltGDmin for any LR Matrix Recovery Problem 41
6.2 Proof Approach: Clean and Noise-free Case 41
6.3 Proof Approach: Noisy Gradient Approach to Deal with Nonlinear or Noisy or Attack-prone Cases 45

7 AltGDmin for LR Problems: Proof Details **47**
7.1 Key Results Used 47
7.2 Analyzing the Initialization Step 48
7.3 Clean Noise-free Case 50
7.4 Nonlinear or Noisy or Attack-prone or Outlier Corrupted Settings 53

8 Linear Algebra and Random Matrix Theory Preliminaries **56**
8.1 Linear Algebra: Maximum and Minimum Singular Value and the Induced 2-norm 56
8.2 Linear Algebra: Wedin and Davis-Kahan $\sin\Theta$ Theorems . 57
8.3 Probability Results: Markov's Inequality and its Use to Prove Concentration Bounds 58
8.4 Probability Results: Chernoff Bounding Idea 59

8.5 Probability Results: Bounds on Sums of Independent Scalar r.v.s (Scalar Concentration Bounds) 60
8.6 Probability Results: Epsilon Netting Argument Used for Extending Union Bound to Uncountable but Compact Sets 61
8.7 Probability Results: Bounding Sums of Independent Matrix r.v.s (Matrix Concentration Bounds) 62

IV Open Questions: AltGDmin and Generalized-AltGDmin for Other Partly-decoupled Problems 65

9 Open Questions 66
9.1 Guarantees for a General Optimization Problem 66
9.2 Generalized AltGDmin 66
9.3 Robust PCA and Extensions: A Partly Decoupled Example Problem for Generalized AltGDmin 67
9.4 Partly Decoupled Tensor LR: Tensor LR Slicewise Sensing . 68
9.5 Partly Decoupled Not-differentiable Problems 70

Appendix 73

References 76

AltGDmin: Alternating GD and Minimization for Partly-decoupled (Federated) Optimization

Namrata Vaswani

Iowa State University, USA; namrata@iastate.edu

ABSTRACT

This monograph describes a novel optimization solution framework, called alternating gradient descent (GD) and minimization (AltGDmin), that is useful for many problems for which alternating minimization (AltMin) is a popular solution. AltMin is a special case of the block coordinate descent algorithm that is useful for problems in which minimization w.r.t one subset of variables keeping the other fixed is closed form or otherwise reliably solved. Denote the two blocks/subsets of the optimization variables $\boldsymbol{Z}$ by $\boldsymbol{Z}_{slow}, \boldsymbol{Z}_{fast}$, i.e., $\boldsymbol{Z} = \{\boldsymbol{Z}_{slow}, \boldsymbol{Z}_{fast}\}$. AltGDmin is often a faster solution than AltMin for any problem for which (i) the minimization over one set of variables, $\boldsymbol{Z}_{fast}$, is much quicker than that over the other set, $\boldsymbol{Z}_{slow}$; and (ii) the cost function is differentiable w.r.t. $\boldsymbol{Z}_{slow}$. Often, the reason for one minimization to be quicker is that the problem is "decoupled" for $\boldsymbol{Z}_{fast}$ and each of the decoupled problems is quick to solve. This decoupling is also what makes AltGDmin communication-efficient for federated settings.

Important examples where this assumption holds include (a) low rank column-wise compressive sensing (LRCS), low

Namrata Vaswani (2025), "AltGDmin: Alternating GD and Minimization for Partly-decoupled (Federated) Optimization", Foundations and Trends® in Optimization: Vol. 8, No. 4, pp 333–414. DOI: 10.1561/2400000051.

rank matrix completion (LRMC), (b) their outlier-corrupted extensions such as robust PCA, robust LRCS and robust LRMC; (c) phase retrieval and its sparse and low-rank model based extensions; (d) tensor extensions of many of these problems such as tensor LRCS and tensor completion; and (e) many partly discrete problems where GD does not apply – such as clustering, unlabeled sensing, and mixed linear regression. LRCS finds important applications in multi-task representation learning and few shot learning, federated sketching, and accelerated dynamic MRI. LRMC and robust PCA find important applications in recommender systems, computer vision and video analytics.

1

Introduction

This monograph describes a novel algorithmic framework, called Alternating Gradient Descent (GD) and Minimization or AltGDmin for short, that is useful for optimization problems that are "partly decoupled" [37]. Consider the optimization problem $\min_{\boldsymbol{Z}} f(\boldsymbol{Z})$. This is partly-decoupled if we can split the set of optimization variables $\boldsymbol{Z}$ into two blocks, $\boldsymbol{Z} = \{\boldsymbol{Z}_{slow}, \boldsymbol{Z}_{fast}\}$, so that the minimization over $\boldsymbol{Z}_{fast}$, keeping $\boldsymbol{Z}_{slow}$ fixed, is decoupled. This means that it can be solved by solving many smaller-dimensional, and hence much faster, minimization problems over disjoint subsets of $\boldsymbol{Z}_{fast}$. That over $\boldsymbol{Z}_{slow}$, keeping $\boldsymbol{Z}_{fast}$ fixed, may or may not be decoupled. We provide examples below and define this mathematically in Section 3.1.

For problems for which one of the two minimizations is decoupled, and hence fast, while the other is not, AltGDmin often provides a much faster solution than the well-known Alternating Minimization (AltMin) [7, 19] approach. Even if both problems are decoupled, AltGDmin still often has a communication-efficiency advantage over AltMin when used in distributed or federated settings. This is the case when the data is distributed across the nodes in such a way that the decoupled minimization over a subset of $\boldsymbol{Z}_{fast}$ also depends on the subset of data available at a node; so this can be solved locally.

Federated learning is a setting in which multiple distributed nodes or entities or clients collaborate to solve a machine learning (ML) problem and where different subsets of the data are acquired at the different nodes. Each node can only communicate with a central server or service provider that we refer to as "center" in this monograph. Communication-efficiency is a key concern with all distributed algorithms, including federated ones. Privacy is another key concern in federated learning. Both concerns dictate that the data observed or measured at each node/client be stored locally and not be shared with the center. Summaries of it can be shared with the center. The center typically aggregates the received summaries and broadcasts the aggregate to all the nodes [29]. In this monograph, "privacy" only means the following: the nodes' raw data cannot be shared with the center and the algorithm should be such that the center cannot reconstruct the entire unknown true signal (vector/matrix/tensor).

One of the challenges in federated learning is developing algorithms that are resilient to adversarial attacks on the nodes; resilience to Byzantine attacks is especially critical. An important challenge in distributed computing settings (data is available centrally, but is distributed to nodes, e.g., over the cloud, to parallelize and hence speed up the computing) is to have algorithms that are resilient to stragglers (some worker nodes occasionally slowing down or failing) [45, 49]. As will become clear in this monograph, the design of both attack resilient and straggler resilient modifications of AltGDmin is also efficient. One example of Byzantine attack resilient AltGDmin is studied in [46].

Monograph organization. This monograph begins by giving some examples of partly decoupled optimization problems and their applications below. In Section 2, we provide a short overview of some of the popular optimization algorithms - gradient descent (GD), block coordinate descent and AltMin, and nonlinear least squares – and when these work well. All these are iterative algorithms that need an initialization. We describe common initialization approaches as well. Then, in Section 3, we precisely define a partly decoupled problem and develop and discuss the AltGDmin algorithmic framework. In the second part of this monograph, in Section 4, we provide the AltGDmin algorithm details, including initialization, for three important LR matrix recovery problems - LR column-wise sensing, LR phase retrieval and LR matrix completion. We also state and discuss the theoretical sample and iteration complexity guarantees that we can prove for these problems. The iteration complexity helps provide total computational and communication complexity bounds. The third part of this monograph discusses proof techniques. We first provide the general proof approach that can be used to analyze the AltGDmin in Section 5 and then describe the key ideas for LR problems in Section 6. Details are in Section 7. Preliminaries used in these proofs are provided and explained in Section 8. This section provides a short overview of the most useful linear algebra and random matrix theory topics from [53] and [16]. In the last part of this monograph, Section 9 describes open questions including other problems where AltGDmin or its generalization may be useful.

1.1 Partly Decoupled Optimization Examples

We provide a few examples of partly decoupled problems.

Low rank column-wise compressive sensing (LRCS). This problem involves recovering an $n \times q$ rank-r matrix $\boldsymbol{X}^*$, with $r \ll \min(n, q)$, from column-wise undersampled (compressive) measurements, $\boldsymbol{y}_k := \boldsymbol{A}_k \boldsymbol{x}_k^*$, $k \in [q]$. The matrices $\boldsymbol{A}_k$ are dense (non-sparse) matrices that are known. Each $\boldsymbol{y}_k$ is an m-length vector with $m < n$. Let $\boldsymbol{Y} := [\boldsymbol{y}_1, \boldsymbol{y}_2, \dots, \boldsymbol{y}_q]$ denote the observed data matrix. We can solve this problem by considering the squared loss function. It then becomes a problem of finding a matrix $\boldsymbol{X}$ of rank at most r that minimizes $\sum_{k=1}^q \|\boldsymbol{y}_k - \boldsymbol{A}_k \boldsymbol{x}_k\|_2^2$. Suppose that r or an upper bound on it is known. This problem can be converted into an unconstrained, and smaller dimensional, one by factorizing $\boldsymbol{X}$ as $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{B}$, where $\boldsymbol{U}$ and $\boldsymbol{B}$ are matrices with r columns and rows respectively. Thus, the goal is to solve

$$\arg\min_{\boldsymbol{U},\boldsymbol{B}} f(\boldsymbol{U}, \boldsymbol{B}) := \arg\min_{\boldsymbol{U},\boldsymbol{B}} \sum_{k=1}^{q} \|\boldsymbol{y}_k - \boldsymbol{A}_k \boldsymbol{U} \boldsymbol{b}_k\|_2^2. \tag{1.1}$$

Notice that $\boldsymbol{b}_k$ appears only in the k-th term of the above summation. Thus, if we needed to minimize over $\boldsymbol{B}$, while keeping $\boldsymbol{U}$ fixed, the problem decouples column-wise. The opposite is not true. We refer to such a problem as a partly decoupled problem.

In solving the above problem iteratively, there can be numerical issues because $\boldsymbol{U}\boldsymbol{B} = \boldsymbol{U}\boldsymbol{R}\boldsymbol{R}^{-1}\boldsymbol{B}$ for any $r \times r$ invertible matrix $\boldsymbol{R}$. The norm of $\boldsymbol{U}$ could keep increasing over iterations while that of $\boldsymbol{B}$ decreases or vice versa. To prevent this, either the cost function is modified to include a norm balancing term, e.g., as in [56], or one orthonormalizes the estimate of $\boldsymbol{U}$ after each update.

Three important practical applications where the LRCS problem occurs include (i) federated sketching [3, 17, 22, 23, 44, 48, 55], (ii) accelerated (undersampled) dynamic MRI with the low rank (LR) model on the image sequence, and (iii) multi-task linear representation learning to enable few shot learning [18, 20, 46, 50]. In fact, some works refer to the LRCS problem as multi-task representation learning. (iv) The LRCS problem also occurs in for parameter estimation in multi-task linear bandits [33].

Low rank phase retrieval (LRPR). This is the phaseless extension of LRCS [37, 39, 40] but it was studied in detail before LRCS was studied. This involves solving

$$\arg\min_{\boldsymbol{U},\boldsymbol{B}} f(\boldsymbol{U},\boldsymbol{B}) := \arg\min_{\boldsymbol{U},\boldsymbol{B}} \sum_{k=1}^{q} \|\boldsymbol{y}_k - |\boldsymbol{A}_k\boldsymbol{U}\boldsymbol{b}_k|\|_2^2 \tag{1.2}$$

where $|.|$ computes the absolute value of each vector entry. LRPR finds applications in dynamic Fourier ptychography [26, 27].

LR matrix completion (LRMC). In this case, the cost function is partly decoupled w.r.t. both $\boldsymbol{U}$ and $\boldsymbol{B}$ (keeping the other fixed). This involves recovering a LR matrix from a subset of its observed entries. Letting Ω denote the set of observed matrix entries, and letting $\mathcal{P}_\Omega$ denote the linear projection operator that returns a matrix of size $n \times q$ with the unobserved entries set to zero, this can be expressed as a problem of learning $\boldsymbol{X}^*$ from $\boldsymbol{Y} := \mathcal{P}_\Omega(\boldsymbol{X}^*)$. Letting the unknown $\boldsymbol{X}$ as $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{B}$ as above, the optimization problem to solve now becomes:

$$\begin{aligned}\arg\min_{\boldsymbol{U},\boldsymbol{B}} f(\boldsymbol{U},\boldsymbol{B}) &:= \|\boldsymbol{Y} - \mathcal{P}_\Omega(\boldsymbol{X}^*)\|_F^2 \\ &= \sum_{k=1}^{q} \|\boldsymbol{y}_k - \mathcal{P}_{\Omega_k}(\boldsymbol{U}\boldsymbol{b}_k)\|_2^2 \\ &= \sum_{j=1}^{n} \|\boldsymbol{y}^j - \mathcal{P}_{\Omega^j}(\boldsymbol{u}^{j\top}\boldsymbol{B})\|_2^2 \end{aligned} \tag{1.3}$$

with $\boldsymbol{B} = [\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_k, \ldots \boldsymbol{b}_q]$, $\boldsymbol{U}^\top = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_j, \ldots, \boldsymbol{u}_n]$, $\Omega_k := \{j : (j,k) \in \Omega\}$ and $\Omega^j := \{k : (j,k) \in \Omega\}$. Notice that the above problem is decoupled over $\boldsymbol{B}$ for a given $\boldsymbol{U}$, and vice-versa. LRMC finds important applications in recommender systems' design, survey data analysis, and video inpainting [11]. LRMC also finds applications in parameter estimation for reinforcement learning, in particular for filling in the missing entries of its state transition probability matrix.

Other partly-decoupled examples. Other examples of partly decoupled problems include non-negative matrix factorization, sparse PCA, robust PCA and extensions (robust LRCS and robust LRMC), tensor LR slice-wise sensing and its robust extension, and LR tensor completion; and certain partly discrete problems – clustering, shuffled or unlabeled sensing, and mixed linear regression. We describe these in Section 9.

1.1.1 Detailed Description of Some Applications

Why the LR model? Medical image sequences change slowly over time and hence these are well modeled as forming a low-rank matrix with each column of the matrix being one vectorized image [5, 34]. The same is often also true for similar sets of natural images and videos [12, 36]. The matrix of user ratings of different products, e.g., movies, is modeled as a LR matrix under the commonly used hypothesis that the ratings are explained by much fewer factors than the number of users, q, or products, n [11]. In fact, many large matrices are well modeled as being LR [51]; these model any image sequence or product ratings or survey dataset, in which most of the differences between the different images or ratings or survey data, q, are explained by only a small number r of factors.

MRI. In MRI, which is used in medicine for cross-sectional imaging of human organs, after some pre-processing, the acquired data can be modeled as the 2D discrete Fourier transform (FT) of the cross-section being imaged. This is acquired one FT coefficient (or one row or line of coefficients) at a time [9, 35]. The choice of the sampled coefficients can be random or it may be specified by carefully designed trajectories. The goal is to reconstruct the image of the cross-section from this acquired data. If we can reconstruct accurately from fewer samples, it means that the acquisition can be speeded up. This is especially useful for dynamic MRI because it can improve the temporal resolution for imaging the changes over time, e.g. the beating heart. Accelerated dynamic MRI involves doing this to recover a sequence of q images, $\boldsymbol{x}_k^*$, $k \in [q]$, say, of the beating heart or of brain function as brain neurons respond to a stimuli, or of the vocal tract (larynx) as a person speaks, from undersampled DFT measurements $\boldsymbol{y}_k$, $k \in [q]$. Here $\boldsymbol{x}_k^*$ is a vectorized image. The matrices $\boldsymbol{A}_k$ are the partial Fourier matrices represented by the 2D DFT (or sometimes the FT in case of radial sampling) computed at the specified frequencies.

Multi-task learning. Multi-task representation learning refers to the problem of jointly estimating the model parameters for a set of related tasks. This is typically done by learning a common lower-dimensional "representation" for all of their feature vectors. This learned representation can then be used for solving the meta-learning or learning-to-learn problem: learning model parameters in a data-scarce environment. This strategy is referred to as "few-shot" learning. In recent work [20], a very interesting low-dimensional linear representation was introduced and the corresponding low rank matrix learning optimization problem was defined. This linear case will be solved if we can solve (1.1). Simply said, this can be understood as a problem of jointly learning the coefficients' for q related linear regression problems, each with their own dataset $\boldsymbol{A}_k$, and with the regression vectors $\boldsymbol{x}_k^*$ being correlated (so that low rank is a good model on the matrix formed by these vectors, $\boldsymbol{X}^*$). Once the "common representation" (the column span subspace matrix $\boldsymbol{U}$) can be estimated, we can solve a new linear regression problem that is related (correlated) with these hold ones by only learning a new r-dimensional vector $\boldsymbol{b}_k$ for it.

Federated sketching. For the vast amounts of data acquired on smartphones/other devices, there is a need to compress/sketch it before it can be stored or transmitted. The term "sketch" refers to a compression approach, where the compression end is very inexpensive [3, 17, 22, 23, 44, 48, 55]. A common approach to sketching, that is especially efficient in distributed settings, is to multiply each vectorized image by a different independent $m \times n$ random matrix (typically random Gaussian or Rademacher matrix) with $m < n$, and to store or transmit this sketch.

Appendix

A

Partly Decoupled Optimization Problem: Most General Definition

Consider an optimization problem $\arg\min_{\boldsymbol{Z}} g(\boldsymbol{Z})$. We say the problem is decoupled if it can be solved by solving smaller dimensional problems over disjoint subsets of $\boldsymbol{Z}$. To define this precisely, observe that any function $g(\boldsymbol{Z})$ can be expressed as a composition of γ functions, for a $\gamma \geq 1$,

$$g(\boldsymbol{Z}) = h(f^1(\boldsymbol{Z}), f^2(\boldsymbol{Z}), \ldots f^\gamma(\boldsymbol{Z})),$$

Here $h(., ., ..)$ is a function of γ inputs. This is true always since we can trivially let $\gamma = 1$, $h(\boldsymbol{Z}) = \boldsymbol{Z}$ and $f^1(\boldsymbol{Z}) = g(\boldsymbol{Z})$.

We say that the optimization problem is decoupled if, for a $\gamma > 1$, $\boldsymbol{Z}$ can be split into γ disjoint subsets

$$\boldsymbol{Z} = [\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots \boldsymbol{Z}_\gamma]$$

so that

$$\begin{aligned}\arg\min_{\boldsymbol{Z}} g(\boldsymbol{Z}) = &[\arg\min_{\boldsymbol{Z}_1} f^1(\boldsymbol{Z}_1), \arg\min_{\boldsymbol{Z}_2} f^2(\boldsymbol{Z}_2), \ldots, \arg\min_{\boldsymbol{Z}_\ell} f^\ell(\boldsymbol{Z}_\ell), \\ &\ldots \arg\min_{\boldsymbol{Z}_\gamma} f^\gamma(\boldsymbol{Z}_\gamma))]\end{aligned}$$

Observe that, in general, arg min is a set and the notation $[\mathcal{S}_1, \mathcal{S}_2, \ldots \mathcal{S}_\gamma]$ is short for their Cartesian product $\mathcal{S}_1 \times \mathcal{S}_2 \times \ldots \mathcal{S}_\gamma$. In words, the set $\arg\min_{\boldsymbol{Z}} f(\boldsymbol{Z}) = \{[\hat{\boldsymbol{Z}}_1, \hat{\boldsymbol{Z}}_2, \dots \hat{\boldsymbol{Z}}_\gamma] : \hat{\boldsymbol{Z}}_1 \in \arg\min_{\boldsymbol{Z}_1} f^1(\boldsymbol{Z}_1), \hat{\boldsymbol{Z}}_2 \in \arg\min_{\boldsymbol{Z}_2} f^2(\boldsymbol{Z}_2), \dots, \hat{\boldsymbol{Z}}_\gamma \in \arg\min_{\boldsymbol{Z}_\gamma} f^\gamma(\boldsymbol{Z}_\gamma)\}$.

If $g(\boldsymbol{Z})$ is strongly convex, then the arg min is one unique minimizer $\hat{\boldsymbol{Z}}$. In this case, the decoupled functions have a unique minimizer too and $\arg\min_{\boldsymbol{Z}_1} f^1(\boldsymbol{Z}_1)$ returns $\hat{\boldsymbol{Z}}_1$ and so on, and $\hat{\boldsymbol{Z}} = [\hat{\boldsymbol{Z}}_1, \hat{\boldsymbol{Z}}_2, \dots, \hat{\boldsymbol{Z}}_\gamma]$. Data-decoupled means that the above holds and that $ef^\ell(\boldsymbol{Z}_\ll)$ depends only on a disjoint subset $\mathcal{D}_\ell$ of the data $\mathcal{D}$. Let $\mathcal{D} = [\mathcal{D}_1, \mathcal{D}_2, \dots \mathcal{D}_\gamma]$. We use a subscript to denote the data. Data-decoupled means that

$$\begin{aligned}\arg\min_{\boldsymbol{Z}} f(\boldsymbol{Z}) =& [\arg\min_{\boldsymbol{Z}_1} f^1_{\mathcal{D}_1}(\boldsymbol{Z}_1), \arg\min_{\boldsymbol{Z}_2} f^2_{\mathcal{D}_2}(\boldsymbol{Z}_2), \dots, \arg\min_{\boldsymbol{Z}_\ell} f^\ell_{\mathcal{D}_\ell}(\boldsymbol{Z}_\ell),\\ &\dots \arg\min_{\boldsymbol{Z}_\gamma} f^\gamma_{\mathcal{D}_\gamma}(\boldsymbol{Z}_\gamma))]\end{aligned}$$

Most practical problems that are decoupled are often also data-decoupled. *Henceforth we use the term "decoupled" to also mean data-decoupled.*

Partly-decoupled is a term used for optimization problems for which the unknown variable $\boldsymbol{Z}$ can be split into two parts, $\boldsymbol{Z} = \{\boldsymbol{Z}_{slow}, \boldsymbol{Z}_{fast}\}$, so that the optimization over one keeping the other fixed is "easy" (closed form, provably correct algorithm exists, or fast). Decoupled and data-decoupled w.r.t. $\boldsymbol{Z}_{fast}$ means that decoupling holds only for minimization over $\boldsymbol{Z}_{fast}$. To be precise, let

$$\boldsymbol{Z}_{fast} = [(\boldsymbol{Z}_{fast})_1, (\boldsymbol{Z}_{fast})_2, \dots (\boldsymbol{Z}_{fast})_\gamma] \text{ and } \mathcal{D} = [\mathcal{D}_1, \mathcal{D}_2, \dots \mathcal{D}_\gamma]$$

Then,

$$\begin{aligned}\arg\min_{\boldsymbol{Z}_{fast}} f(\boldsymbol{Z}_{slow}, \boldsymbol{Z}_{fast}) =& [\arg\min_{(\boldsymbol{Z}_{fast})_1} f^1_{\mathcal{D}_1}(\boldsymbol{Z}_{slow}, (\boldsymbol{Z}_{fast})_1), \dots,\\ &\arg\min_{(\boldsymbol{Z}_{fast})_\ell} f^\ell_{\mathcal{D}_\ell}(\boldsymbol{Z}_{slow}, (\boldsymbol{Z}_{fast})_\ell),\\ &\dots \arg\min_{\boldsymbol{Z}_\gamma} f^\gamma_{\mathcal{D}_\gamma}(\boldsymbol{Z}_{slow}, (\boldsymbol{Z}_{fast})_\gamma))]\end{aligned}$$

All the examples of partly decoupled optimization problems that we discuss in this work are those for which $g(\boldsymbol{Z}) = h(f^1, f^2, \dots f^\gamma) = \sum_{\ell=1}^\gamma f^\ell$ is a sum of the γ functions f^ℓ. In this case, partly decoupled problems means that

$$\min_{\boldsymbol{Z}_{fast}} f(\boldsymbol{Z}_{slow}, \boldsymbol{Z}_{fast}) = \sum_\ell \min_{(\boldsymbol{Z}_{fast})_\ell} f^\ell_{\mathcal{D}_\ell}(\boldsymbol{Z}_{slow}, \boldsymbol{Z}_{fast_\ell})$$

References

[1] A. A. Abbasi, A. Tasissa, and S. Aeron, "R-local unlabeled sensing: Improved algorithm and applications," in *IEEE Intl. Conf. Acoustics, Speech, Sig. Proc. (ICASSP)*, pp. 5593–5597, 2022.

[2] A. A. Abbasi and N. Vaswani, "Efficient federated low rank matrix completion," *IEEE Trans. Info. Th.*, 2025. URL: https://ieeexplore.ieee.org/document/10975055.

[3] F. P. Anaraki and S. Hughes, "Memory and computation efficient pca via very sparse random projections," pp. 1341–1349, 2014.

[4] S. Babu, S. Aviyente, and N. Vaswani, "Tensor low rank column-wise compressive sensing for dynamic imaging," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, pp. 1–5, 2023.

[5] S. Babu, S. G. Lingala, and N. Vaswani, "Fast low rank compressive sensing for accelerated dynamic MRI," *IEEE Trans. Comput. Imag.*, 2023.

[6] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[7] C. L. Byrne, "Alternating minimization and alternating projection algorithms: A tutorial," *Journal of Optimization Theory and Applications*, vol. 156, no. 3, 2013, pp. 554–566.

[8] T. Cai, X. Li, and Z. Ma, "Optimal rates of convergence for noisy sparse phase retrieval via thresholded wirtinger flow," *The Annals of Statistics*, vol. 44, no. 5, 2016, pp. 2221–2251.

[9] E. Candes and T. Tao, "Near optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Trans. on Information Theory*, vol. 52(12), 2006, pp. 5406–5425.

[10] E. J. Candes, X. Li, and M. Soltanolkotabi, "Phase retrieval via wirtinger flow: Theory and algorithms," *IEEE Trans. Info. Th.*, vol. 61, no. 4, 2015, pp. 1985–2007.

[11] E. J. Candes and B. Recht, "Exact matrix completion via convex optimization," *Found. of Comput. Math*, no. 9, 2008, pp. 717–772.

[12] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, 2011.

[13] A.-L. Cauchy, "Méthode générale pour la résolution des systèmes d'équations simultanées," French, *Comptes Rendus de l'Académie des Sciences*, vol. 25, 1847, pp. 536–538.

[14] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky, "Rank-sparsity incoherence for matrix decomposition," *SIAM Journal on Optimization*, vol. 21, 2 2011.

[15] Y. Chen and E. Candes, "Solving random quadratic systems of equations is nearly as easy as solving linear systems," pp. 739–747, 2015.

[16] Y. Chen, Y. Chi, J. Fan, C. Ma, *et al.*, "Spectral methods for data science: A statistical perspective," *Foundations and Trends® in Machine Learning*, vol. 14, no. 5, 2021, pp. 566–806.

[17] Y. Chen, Y. Chi, and A. J. Goldsmith, "Exact and stable covariance estimation from quadratic sampling via convex programming," *IEEE Transactions on Information Theory*, vol. 61, no. 7, 2015, pp. 4034–4059.

[18] L. Collins, H. Hassani, A. Mokhtari, and S. Shakkottai, "Exploiting shared representations for personalized federated learning," in *International conference on machine learning*, PMLR, pp. 2089–2099, 2021.

[19] I. Csiszár, "Information geometry and alternating minimization procedures," *Statistics and Decisions, Dedewicz*, vol. 1, 1984, pp. 205–237.

[20] S. S. Du, W. Hu, S. M. Kakade, J. D. Lee, and Q. Lei, "Few-shot learning via learning the representation, provably," 2021.

[21] A. Ghosh, A. Pananjady, A. Guntuboyina, and K. Ramchandran, "Max-affine regression: Parameter estimation for gaussian designs," *IEEE Transactions on Information Theory*, vol. 68, no. 3, 2021, pp. 1851–1885.

[22] A. C. Gilbert, J. Y. Park, and M. B. Wakin, "Sketched svd: Recovering spectral features from compressive measurements," *arXiv preprint arXiv:1211.0361*, 2012.

[23] A. C. Gilbert, M. J. Strauss, J. A. Tropp, and R. Vershynin, "One sketch for all: Fast algorithms for compressed sensing," in *Proceedings of ACM Symposium on Theory of Computing (STOC)*, pp. 237–246, 2007.

[24] G. H. Golub and C. F. Van Loan, "Matrix computations," *The Johns Hopkins University Press, Baltimore, USA*, 1989.

[25] M. Hardt and E. Price, "The noisy power method: A meta algorithm with applications," *Advances in neural information processing systems*, vol. 27, 2014.

[26] J. Holloway, M. S. Asif, M. K. Sharma, N. Matsuda, R. Horstmeyer, O. Cossairt, and A. Veeraraghavan, "Toward long-distance subdiffraction imaging using coherent camera arrays," *IEEE Trans Comput Imaging*, vol. 2, no. 3, 2016, pp. 251–265.

[27] G. Jagatap, Z. Chen, S. Nayer, C. Hegde, and N. Vaswani, "Sample efficient fourier ptychography for structured data," *IEEE Transactions on Computational Imaging*, vol. 6, 2019, pp. 344–357.

[28] P. Jain and P. Netrapalli, "Fast exact matrix completion with finite samples," in *Conference on Learning Theory*, PMLR, pp. 1007–1034, 2015.

[29] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, 2021, 1–210.

[30] S. M. Kay, *Fundamentals of statistical processing: Estimation theory.* 1993.

[31] R. H. Keshavan, A. Montanari, and S. Oh, "Matrix completion from a few entries," *IEEE transactions on information theory*, vol. 56, no. 6, 2010, pp. 2980–2998.

[32] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, 2009, pp. 455–500. DOI: 10.1137/07070111X.

[33] J. Lin, S. Moothedath, and N. Vaswani, "Fast and sample efficient multi-task representation learning in stochastic contextual bandits," in *International Conference on Machine Learning*, PMLR, pp. 30 227–30 251, 2024.

[34] S. G. Lingala, Y. Hu, E. DiBella, and M. Jacob, "Accelerated dynamic mri exploiting sparsity and low-rank structure: Kt slr," *IEEE Transactions on Medical Imaging*, vol. 30, no. 5, 2011, pp. 1042–1054.

[35] M. Lustig, J. M. Santos, D. L. Donoho, and J. M. Pauly, "K-t sparse: High frame rate dynamic MRI exploiting spatio-temporal sparsity," in *Conf. of International Society for Magnetic Resonance in Medicine(ISMRM), Seattle, Washington*, May 2006.

[36] P. Narayanamurthy and N. Vaswani, "Provable dynamic robust pca or robust subspace tracking," *IEEE Transactions on Information Theory*, vol. 65, no. 3, 2018, pp. 1547–1577.

[37] S. Nayer and N. Vaswani, "Fast and sample-efficient federated low rank matrix recovery from column-wise linear and quadratic projections," *IEEE Trans. Info. Th.*, 2023.

[38] S. Nayer, P. Narayanamurthy, and N. Vaswani, "Phaseless PCA: Low-rank matrix recovery from column-wise phaseless measurements," in *International Conference on Machine Learning*, PMLR, pp. 4762–4770, 2019.

[39] S. Nayer, P. Narayanamurthy, and N. Vaswani, "Provable low rank phase retrieval," *IEEE Transactions on Information Theory*, vol. 66, no. 9, 2020, pp. 5875–5903.

[40] S. Nayer and N. Vaswani, "Sample-efficient low rank phase retrieval," *IEEE Transactions on Information Theory*, vol. 67, no. 12, 2021, pp. 8190–8206.

[41] P. Netrapalli, P. Jain, and S. Sanghavi, "Low-rank matrix completion using alternating minimization," 2013.

[42] P. Netrapalli, P. Jain, and S. Sanghavi, "Phase retrieval using alternating minimization," in *Neur. Info. Proc. Sys. (NeurIPS)*, pp. 2796–2804, 2013.

[43] P. Netrapalli, U. N. Niranjan, S. Sanghavi, A. Anandkumar, and P. Jain, "Non-convex robust pca," in *Neur. Info. Proc. Sys. (NeurIPS)*, 2014.

[44] H. Qi and S. M. Hughes, "Invariance of principal components under low-dimensional random projection of the data," in *19th IEEE International Conference on Image Processing*, pp. 937–940, 2012.

[45] A. Ramamoorthy, R. Meng, and V. Girimaji, "Leveraging partial stragglers within gradient coding," *Advances in Neural Information Processing Systems*, vol. 37, 2024, pp. 60 382–60 402.

[46] A. P. Singh and N. Vaswani, "Byzantine resilient and fast federated few-shot learning," in *Forty-first International Conference on Machine Learning*, 2024.

[47] M. Slawski and E. Ben-David, "Linear regression with sparsely permuted data," *Electronic Journal of Statistics*, vol. 13, 2019, pp. 1–36.

[48] R. S. Srinivasa, K. Lee, M. Junge, and J. Romberg, "Decentralized sketching of low rank matrices," pp. 10 101–10 110, 2019.

[49] R. Tandon, Q. Lei, A. G. Dimakis, and N. Karampatziakis, "Gradient coding: Avoiding stragglers in distributed learning," in *International Conference on Machine Learning*, PMLR, pp. 3368–3376, 2017.

[50] K. K. Thekumparampil, P. Jain, P. Netrapalli, and S. Oh, "Statistically and computationally efficient linear meta-representation learning," *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 18 487–18 500.

[51] M. Udell and A. Townsend, "Why are big data matrices approximately low rank?" *SIAM Journal on Mathematics of Data Science*, vol. 1, no. 1, 2019, pp. 144–160.

[52] N. Vaswani, "Efficient federated low rank matrix recovery via alternating gd and minimization: A simple proof," *IEEE Trans. Info. Th.*, 2024.

[53] R. Vershynin, *High-dimensional probability: An introduction with applications in data science*, vol. 47. Cambridge university press, 2018.

[54] P.-Å. Wedin, "Perturbation bounds in connection with singular value decomposition," *BIT Numerical Mathematics*, vol. 12, no. 1, 1972, pp. 99–111.

[55] D. P. Woodruff *et al.*, "Sketching as a tool for numerical linear algebra," *Foundations and Trends® in Theoretical Computer Science*, vol. 10, no. 1–2, 2014, pp. 1–157.

[56] X. Yi, D. Park, Y. Chen, and C. Caramanis, "Fast algorithms for robust pca via gradient descent," in *Neur. Info. Proc. Sys. (NeurIPS)*, 2016.

[57] Q. Zheng and J. Lafferty, "Convergence analysis for rectangular matrix completion using burer-monteiro factorization and gradient descent," *arXiv preprint arXiv:1605.07051*, 2016.

[58] K. Zhong, P. Jain, and I. S. Dhillon, "Mixed linear regression with multiple components," in *Neur. Info. Proc. Sys. (NeurIPS)*, vol. 29, 2016.