## Distribution Agreement

Signature:

________________________________________ ______________

Yifei Ren Date

Temporal Irregular Tensor Factorization and Prediction for Health Data Analysis

By

Yifei Ren
Doctor of Philosophy

Computer Science and Informatics

---

Li Xiong, Ph.D.
Advisor

---

Joyce C Ho, Ph.D.
Advisor

---

Liang Zhao, Ph.D.
Committee Member

---

Xiaoqian Jiang, Ph.D.
Committee Member

Accepted:

---

Kimberly Jacob Arriola, Ph.D, MPH.
Dean of the James T. Laney School of Graduate Studies

---

Date

Temporal Irregular Tensor Factorization and Prediction for Health Data Analysis

By

Yifei Ren
B.S., Southeast University, Jiangsu, 2015
M.Sc., University of Michigan, MI, 2017

Advisor: Li Xiong, Ph.D. ; Joyce C Ho, Ph.D.

An abstract of
A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Computer Science and Informatics
2021

Tensors are a popular algebraic structure for a wide range of applications, due to their exceptional capability to model multidimensional relationships of the data. Among them, regular tensors with aligned dimensions for all modes have been extensively studied, for which various tensor factorization structures are proposed depending on the applications. However, regular tensor decomposition is incapable of handling many real-world cases involving time, due to its irregularity. Electronic health records (EHRs) are often generated and collected across a large number of patients featuring distinctive medical conditions and clinical progress over a long period of time, which results in unaligned records along the time dimension. PARAFAC2 has been re-popularized for successfully extracting meaningful medical concepts (phenotypes) from EHRs by irregular tensor factorization. However, efforts still need to overcome the limitations of the current PARAFAC2 model, including lack of robustness against missing values, lack of modeling of non-linear temporal dependencies, and lack of consideration of the downstream tasks. We propose 1) robust temporal PARAFAC2 for irregular tensor factorization and completion with potential missing and erroneous values; 2) generalized, low-rank recurrent neural network (RNN) regularized robustly irregular tensor factorization for more accurate temporal modeling, which is flexible enough to choose from a variate of losses to best suit different types of data in practice; 3) supervised irregular tensor factorization framework with multi-task learning for both phenotype extraction and predictive learning which enables information sharing between different prediction tasks and further improve downstream prediction performance.

# Temporal Irregular Tensor Factorization and Prediction for Health Data Analysis

By

Yifei Ren

Advisor: Li Xiong, Ph.D. ; Joyce C Ho, Ph.D.

A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Computer Science and Informatics
2021

# Acknowledgments

I would like to express my sincere gratitude to my advisors, Dr. Li Xiong and Dr. Joyce C Ho for their invaluable guidance throughout this work. Dr. Xiong and Dr. Ho have always being there to offer timely and thoughtful advice across every single aspect of the Ph.D. journey. I am honored and grateful to work towards my Ph.D. under their guidance.

I would also like to thank Dr. Liang Zhao and Dr. Xiaoqian Jiang who helped to shed lights on many of my research ideas and had priceless advice. I would also like to thank my mentor Jian Lou. Our discussions and collaborations has been extremely supportive every step of the way.

I would like to thank my daughter Julia Li for coming into our lives and brings us so much happiness. I also want to thank my husband, Junxuan Li. Your support, love and thoughtfulness mean the world to me. Finally, I would like to thank my parents, Xingguo Ren and Hong Yang, and parents-in-law, Zhenti Li and Cuizhi Gu for their unconditional love and sacrifices they have made in order to enable me to follow my dreams.

# Contents

# List of Figures

# List of Tables

# List of Algorithms

# Chapter 1

# Introduction and Background

Recent years have witnessed a global interest in mining Electronic Health Records (EHRs) to improve healthcare and advance medical research [76]. EHRs consist of detailed information such as diagnoses, laboratory test results, and medication prescriptions, for large patient populations. However, directly using raw EHR data is challenging due to its multi-dimensional and complex structure, and massive data amount. In addition, clinical scientists are interested in breaking apart heterogeneous syndromes into subgroups, i.e. *phenotypes*, such as diseases and disease subtypes, for better understandings of the differences in biological mechanisms and treatment responses, which could lead to more effective and precise treatment. Therefore, raw EHR data are often mapped to concise and meaningful medical concepts (i.e., phenotypes) [7], which can be used for cohort (patient subgroup) identification and healthcare quality measurement. The ideal phenotype should concisely represent complex interactions between different aspects of the patients (e.g., diagnosis, medications, and lab results). Thus, computational phenotyping, the transformation from EHRs to phenotypes, can be viewed as a form of dimension reduction, where each phenotype forms a latent space.

Due to tensor's intrinsic capability to model multi-dimensional relationships of the data, tensor decomposition-based computational phenotyping models have been proposed to automatically extract phenotypes [28, 27, 73, 32, 41]. Compared to traditional clustering-based approaches, tensor analysis not only can cluster patients into subgroups but also can capture the interactions between the multiple attributes (e.g, specific procedures used to treat a disease) and extract concise and potentially more interpretable patterns in the latent spaces. Tensors are a popular algebraic structure for a wide range of applications beyond health data mining, e.g., social network analysis [40, 66], recommender system [33] and signal processing [64]. Regular tensors with aligned dimensions for all modes have been extensively studied. Many regular tensor factorization structures have been proposed: Canonical Polyadic (CP) [10, 22, 26], Tucker [65], and tensor singular value decomposition (SVD) [35, 34]. However, regular tensor decomposition is incapable of handling many real-world data involving time due to the irregularity in the time dimension. A concrete example is EHR data that different patients may have the different numbers of visits.

Table 1.1: Sample EHRs data (extracted from MIMIC-EXTRACT dataset, circles means missing observations).

| **patient 1 visit time** | **Albumin** | **Blood urea nitrogen** | **Chloride** | **Glascow coma scale** | **Oxygen saturation** | **Sodium** | **......** |
|---|---|---|---|---|---|---|---|
| 0 | 1.8 | O | O | O | 97.5 | O | ...... |
| 1 | O | O | O | 11 | O | 135 | ...... |
| 2 | O | O | 109 | O | O | O | ...... |
| 3 | O | 24 | O | O | 91.4 | O | ...... |
| **patient 2 visit time** | **Albumin** | **Blood urea nitrogen** | **Chloride** | **Glascow coma scale** | **Oxygen saturation** | **Sodium** | **......** |
| 0 | O | 22 | O | O | O | O | ...... |
| 1 | O | 18 | O | O | 95 | O | ...... |

*Example 1.* Consider an EHR database that captures $K$ patients. Table 1.1 shows two patients' EHR records. The number of visits can be of different sizes across patients. Besides the irregularity, EHR records are also prone to missing observations because of many practical reasons, for example, equipment failure, inaccurate infor-

mation recording, and inexperienced medical staff. Circles in Table 1.1 means missing observations.

Recently, PARAFAC2 [23] has been re-popularized for successfully extracting meaningful medical concepts (phenotypes) from such temporal EHR by irregular tensor factorization. Figure 1.1 illustrates the computational phenotyping process using PARAFAC2 for the data. Each patient record can be captured using a binary, numeric, or count matrix $X_k$, where each matrix value represents the measurement associated with a particular feature for a particular visit. The entire data can be represented as an irregular tensor where each slice $X_k$ represents the information of patient $k$ with $I_k$ visits and $J$ medical features. The irregular tensor $X_k$ will be factorized by PARAFAC2 to three-factor matrices. $\mathbf{U}_k \in \mathbb{R}^{I_k \times R}$ captures temporal evolution of the $R$ phenotypes for patient $k$. $\mathbf{V} \in \mathbb{R}^{J \times R}$ contains the phenotypes. Each row of $V$ matrix represents one latent and potentially interpretable phenotype. Each medical feature is represented with a weight indicating its contribution to the phenotype in each row. $\mathbf{S}_k \in \mathbb{R}^{R \times R}$ is a diagonal matrix with the importance membership of patient $k$ in each one of the $R$ phenotypes. The right side table in figure 1.1 shows three example phenotypes represented in the $V$ matrix with the top 4 highest weighted medical features in each phenotype (the example is from the MIMIC-EXTRACT dataset and weight is shown in parenthesis). Each phenotype represents a set of related medical features which can suggest meaningful subgroups.

A scalable PARAFAC2 model was proposed in [51] to handle large and sparse data. Afshar et al. further introduced various constraints to improve the interpretability of the factor matrices for more meaningful phenotype extraction [2]. Despite these improvements, existing PARAFAC2 methods suffered from three major limitations: 1) they are not robust to missing and erroneous elements in the data; 2) they fail to model the non-linear temporal dependency of patients' disease states, and are designed only for a single data type – numeric or binary; 3) they are completely un-

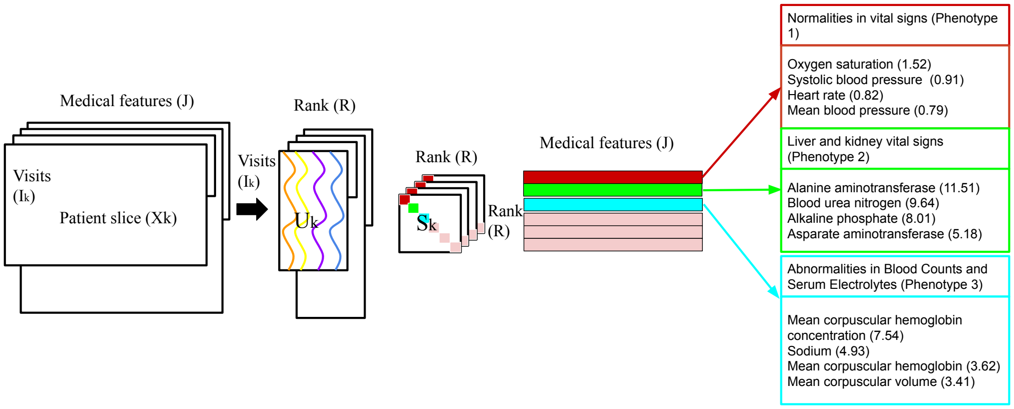

Medical features (J)
Visits
($I_k$)
Patient slice (Xk)
Visits
($I_k$)
Rank (R)
$U_k$
Rank (R)
$S_k$
Rank
(R)
Medical features (J)
Normalities in vital signs (Phenotype 1)
Oxygen saturation (1.52)
Systolic blood pressure (0.91)
Heart rate (0.82)
Mean blood pressure (0.79)
Liver and kidney vital signs (Phenotype 2)
Alanine aminotransferase (11.51)
Blood urea nitrogen (9.64)
Alkaline phosphate (8.01)
Asparate aminotransferase (5.18)
Abnormalities in Blood Counts and Serum Electrolytes (Phenotype 3)
Mean corpuscular hemoglobin concentration (7.54)
Sodium (4.93)
Mean corpuscular hemoglobin (3.62)
Mean corpuscular volume (3.41)

Figure 1.1: PARAFAC2 tensor factorization

supervised, i.e., they attempt to learn the latent factors to best recover the original observations without considering downstream predictive tasks. While there are models that use extracted phenotypes for predictive tasks, they are trained separately and only consider a single prediction task, which ignores auxiliary information from other predictive tasks.

## 1.1 Research Contributions

This thesis proposes a framework extending PARAFAC2 for robust, better temporal modeling, and supervised tensor factorization and prediction for health data analysis. It includes several contributions:

1. We propose a robust PARAFAC2 tensor factorization method for irregular tensors with a new low-rank regularization function to handle potentially missing and erroneous entries in the input tensor (address the limitation 1).

2. We propose a generalized, low-rank Recurrent Neural Network (RNN) regularized robust irregular tensor factorization for more accurate temporal modeling, which is flexible enough to choose from a variate of losses to best suit different types of data in practice (address the limitation 2).

3. We propose a supervised irregular tensor factorization framework with multi-task learning for both phenotype extraction and predictive learning, which can yield not only more meaningful phenotypes but also better predictive accuracy (address the limitation 3).

### 1.1.1 Robust Temporal PARAFAC2 for Irregular Tensor Factorization and Completion (Chapter 2)

First, We study the unexplored robust irregular tensor factorization and completion with potential missing and erroneous values. Existing PARAFAC2 methods are not robust to missing and erroneous elements in the data, which severely limits its applicability to practical temporal EHR data analysis. For regular tensor factorization frameworks, robust mechanisms are well developed to handle missing and erroneous data, among which the robust low-rank tensor minimization (RLTM) is one of the most successful approaches [1, 42, 46, 19, 62, 21, 49, 48]. Different low-rank regularization functions are adopted by these methods, which vary according to different types of tensor factorization. However, it is still unknown how to impose low-rank regularization for PARAFAC2 and design an explicit RLTM mechanism to handle missing entries and remove erroneous entries.

To fill this gap, we propose **REPAIR**, a **R**obust t**E**mporal **PA**FAFAC2 for **IR**regular tensor factorization and completion method, which is the first robust irregular tensor recovery method. Given each patient input data with erroneous and missing entries, REPAIR performs RLTM to separate out the erroneous entries from the underlying clean and completed components, and uses the clean tensor from a common low-rank space for PARAFAC2 based candidate phenotype extraction. We achieve this by addressing two main challenges: First, specific low-rank regularizations need to be designed for PARAFAC2 to suit its decomposition structure which has not been explored in existing work. Second, the robust factorization needs to incorporate additional constraints such as temporal smoothness, non-negativity, and sparsity [2] to obtain more meaningful and accurate phenotypes.

We evaluate **REPAIR** on two real-world temporal EHR datasets with a set of experiments, which verify the improved recovery and factorization robustness against missing and erroneous values. Through two case studies: identification of higher-risk patient subgroups, and in-hospital mortality prediction, we further demonstrate the superior utility of the factorization outputs of REPAIR to facilitate downstream temporal EHR data analysis.

### 1.1.2 RNN Regularized Robust Irregular Tensor Factorization and Completion (Chapter 3)

Next, inspired by the non-linear modeling capability of deep neural networks, we focus on extending a generalized PARAFAC2 model to capture complex temporal relationships for different patients. The existing PARAFAC2 tensor factorization methods only impose linear and human-defined temporal regularization functions, which fail to capture non-linear and complex temporal information in real-world scenarios. Moreover, current PARAFAC2 models are designed only for a single data type – numeric or binary. Thus, there is a lack of flexibility of PARAFAC2 models for other data types.

To address these limitations, we propose **REBAR**, a **R**NN R**E**gularized Ro**B**ust P**AR**AFAC2 Irregular Tensor Factorization and Completion model. REBAR has a new hybrid optimization framework using stochastic gradient descent and proximal average that can handle multiple regularizations and generalized loss functions. Moreover, REBAR accommodates a wide selection of regularization, including statistical learning-based, deep learning-based, and composite, to better capture the intrinsic nature of the irregular temporal EHR data. We also introduce a new optimization framework to fully exploit the parallel computing capability of modern GPUs to boost efficiency.

We evaluate **REBAR** on three real-world temporal EHR datasets with a set of experiments, which verify the improved recovery and factorization robustness against missing values. Through three case studies: interpretation of the dynamic subphenotypes trajectory, downstream prediction analysis, and scalability analysis, we further demonstrate that REBAR can robustly and scalably extract meaningful and high predictability phenotypes with missing data.

### 1.1.3 Supervised Irregular Tensor Factorization Framework with Multi-task Learning (Chapter 4)

Last but not least, we tackle the challenge of improving the predictability of the current PARAFAC2 model in Chapter 4. Current PARAFAC2 models [51, 2] are completely unsupervised and only attempt to learn the latent factors to best recover the original observations. Some works have considered using the latent factors as features for downstream prediction tasks (e.g., in-hospital mortality or hospital readmission prediction using extracted phenotypes), and achieved limited performance gain than using the raw data as features. This is because the tensor factorization does not take advantage of the downstream labels, the extracted factors, while interpretable, may not be the most representative or discriminating for downstream prediction tasks. In addition, current work [2, 51] using tensor factorization for predictive tasks only consider a single task (e.g., in-hospital mortality prediction) and ignore useful information from other prediction tasks.

We propose **MULTIPAR**: a supervised irregular tensor factorization framework with multi-task learning for both phenotype extraction and predictive learning. MULTIPAR jointly optimizes the tensor factorization and downstream prediction together, so that the factorization can be “supervised” or informed by the predictive tasks. In addition, we use a multi-task framework to leverage information from multiple predictive tasks. It provides flexibility to incorporate both one-time or static (e.g. in-hospital mortality prediction) and continuously changing or dynamic (e.g. the need for ventilation) outcomes. To achieve this, the temporal features from $\mathbf{U}$ matrix are used for dynamic prediction, and the features from $\mathbf{S}$ matrix are used for static prediction.

Our main hypothesis is that such a supervised multi-task framework can yield not only more meaningful phenotypes but also better predictive accuracy than performing tensor factorization independently followed by predictive learning using the phenotypes extracted from the tensor. Our empirical studies on two large publicly available EHR datasets with representative predictive tasks (both static and dynamic) and different models (e.g. logistic regression and recurrent neural networks) verified this hypothesis.

# Chapter 2

# REPAIR: Robust Temporal PARAFAC2 for Irregular Tensor Factorization and Completion

## 2.1 Overview

Existing PARAFAC2 methods are unable to robustly handle erroneousness and missing data which are prevalent in clinical practice. In this chapter, we propose **REPAIR**, a **R**obust t**E**mporal **PA**RAFAC2 for **IR**regular tensor factorization and completion method, to complete irregular tensor and extract phenotypes in the presence of missing and erroneous values. As it is shown in Figure 2.1, given each patient input data $\mathbf{O}_k$ with erroneous and missing entries, REPAIR performs RLTM to separate out the erroneous entries $\mathbf{E}_k$ from the underlying clean and completed components $\mathbf{X}_k$, and uses the clean tensor from a common low-rank space for PARAFAC2 based candidate phenotype extraction, i.e. $\mathbf{X}_k \approx \mathbf{U}_k S_k \mathbf{V}^\top$. We achieve this by addressing two main challenges: First, specific low-rank regularizations need to be designed for PARAFAC2 to suit its decomposition structure which has not been explored in existing work. Second, the robust factorization needs to incorporate additional constraints such as temporal smoothness, non-negativity and sparsity [2] to obtain more meaningful and accurate phenotypes.

We summarize our contributions below:

1. We propose a robust PARAFAC2 tensor factorization method for irregular tensors with a new low-rank regularization function to handle potentially missing and erroneous entries in the input tensor. This is the first work that explicitly handles missing and erroneous data for irregular tensor factorization.

2. We design an efficient two-phase optimization to simultaneously: 1) learn and complete the clean underlying tensor by decomposing the original tensor into the underlying low-rank tensor and the sparse error tensor; and 2) extract phenotypes by factorizing the clean tensor. The phenotype extraction phase incorporates many practical constraints for improving interpretability of the extracted phenotypes, including temporal smoothness, non-negativity and sparsity.

3. We evaluate **REPAIR** on two real-world temporal EHR datasets with a set of experiments, which verify the improved recovery and factorization robustness against missing and erroneous values. Through two case studies: identification of higher-risk patient subgroups, and in-hospital mortality prediction, we further demonstrate the superior utility of the factorization outputs of REPAIR to facilitate downstream temporal EHR data analysis.

## 2.2 Preliminaries and Backgrounds

In this section, we define the notations, present background on robust low-rank tensor minimization followed by PARAFAC2 and its application for temporal EHR phenotyping. Table 2.1 summarizes commonly used notations.

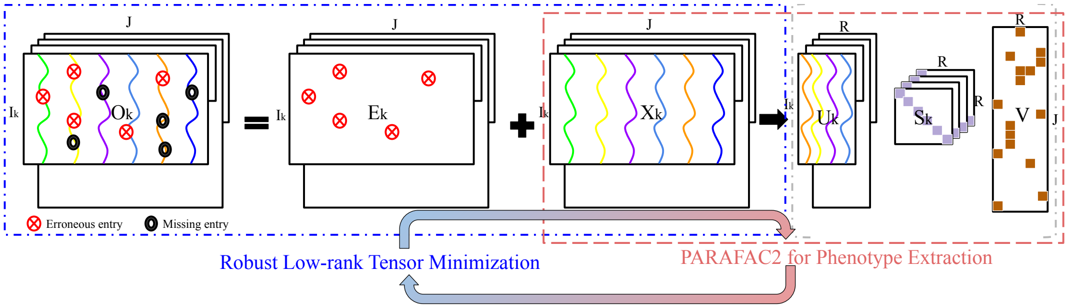

J
Ik
Ok
Erroneous entry
Missing entry
Ek
Xk
Uk
Sk
R
V
Robust Low-rank Tensor Minimization
PARAFAC2 for Phenotype Extraction

Figure 2.1: Overview of REPAIR: robust irregular tensor PARAFAC2 factorization for EHR phenotyping on input patients' data $\boldsymbol{\mathcal{O}}$. $\mathbf{O}_k$ contains erroneous and missing entries, which can be decomposed into erroneous $\mathbf{E}_k$ and clean and completed components denoted by $\mathbf{X}_k$. $\mathcal{P}_\Omega(\mathbf{O}_k) = \mathcal{P}_\Omega(\mathbf{E}_k + \mathbf{X}_k)$. The underling clean tensor is decomposed by PARAFAC2 into $\mathbf{X}_k \approx \mathbf{U}_k S_k \mathbf{V}^\top$.

Table 2.1: Symbols and notations used in chapter 2

| Symbol | Definition |
|---|---|
| $\mathbf{a}, \mathbf{A}, \boldsymbol{\mathcal{A}}$ | Vector, Matrix, Tensor |
| $\mathbf{A}_k$ | $k$-th frontal slice of $\boldsymbol{\mathcal{A}}$ |
| $\boldsymbol{\mathcal{A}}_{(n)}$ | Mode-$n$ matricization of $\boldsymbol{\mathcal{A}}$ |
| $\|\cdot\|_1$ | $\ell_1$-norm |
| $\|\cdot\|_F$ | Frobenius norm |
| $\|\cdot\|_*$ | Nuclear norm |
| $*$ | Hadamard (element-wise) multiplication |
| $\odot$ | Khatri Rao product |
| $\circ$ | Outer product |
| $\langle\cdot,\cdot\rangle$ | Inner product |

For temporal EHR, let the observed tensor be $\boldsymbol{\mathcal{O}} = \{\mathbf{O}_k\} \in \{\mathbb{R}^{I_k \times J}\}$ (c.f. leftmost tensor in Figure 2.1) with 3 modes, where each frontal slice $\mathbf{O}_k$ represents patient $k$'s record of $J$ types of diagnosis, treatments or lab test results (along mode 2), across $I_k$ clinical encounters (along mode 1) varying from patient to patient. The aim of temporal EHR phenotyping is to discover medical concepts by making use of all $K$ frontal slices, i.e. the information of all K patients, and discerning as much inter-relationship across different patients (i.e. cross frontal slice) as possible.

### 2.2.1 Robust Low-rank Tensor Factorization and Completion

For regular tensors (i.e. assuming $\{\mathbf{O}_k\}$ are aligned in all dimensions), the robust low-rank tensor minimization (RLTM) is one of the most successful approaches to handle incomplete and corrupted input tensors. For such a regular tensor $\mathcal{O}$, RLTM separates it into an underlying clean and completed tensor $\mathcal{X}$ and an error tensor $\mathcal{E}$. In practice, the clean part is often low-rank while the erroneous part is sparse. Thus, RLTM imposes a low-rank regularization function $\|\cdot\|_{lr}$ and a sparsity regularization function $\|\cdot\|_1$ on $\mathcal{X}$ and $\mathcal{E}$, correspondingly:

$$\underset{\mathcal{X},\mathcal{E}}{\operatorname{argmin}} \|\mathcal{X}\|_{lr} + \rho_0\|\mathcal{E}\|_1, \; s.t. \; \mathcal{P}_\Omega(\mathcal{O}) = \mathcal{P}_\Omega(\mathcal{X}+\mathcal{E}), \tag{2.1}$$

where $\Omega$ is the index set of non-missing entries and $\mathcal{P}_\Omega$ keeps entries in $\Omega$ and zeros out others (i.e., missing entries), $\rho_0$ is a balancing constant. RLTM is a multidimensional extension to the robust low-rank matrix minimization [9], but it is intrinsically more difficult. The main challenge lies in introducing a proper low-rank definition and designing an effective and efficient low-rank regularization. Unlike a low-rank matrix, the low-rank definition for tensor is not unique and should be adapted according to each tensor decomposition model (e.g., CP, Tucker, tensor SVD).

For example, Tucker model defines the rank of $\mathcal{X}$ based on the matrix rank of its matricization, i.e. the vector $(rank(\mathcal{X}_{(1)}), rank(\mathcal{X}_{(2)}),\ rank(\mathcal{X}_{(3)}))$. CP decomposes $\mathcal{X} \in \mathbb{R}^{I_1\times I_2\times I_3}$ into the sum of $R$ rank-one tensors by $\mathcal{X} = \sum_{r=1}^{R} \mathbf{A}(:,r)\circ\mathbf{B}(:,r)\circ\mathbf{C}(:,r)$, where $\mathbf{A},\mathbf{B},\mathbf{C}$ are factorization matrices and the smallest $R$ to achieve such decomposition is defined to be the rank $R^*$ of $\mathcal{X}$ under CP model. It is difficult to accurately estimating $R^*$ for CP (in fact, NP-hard to determine), as well as to deal with matrix rank used by Tucker. More tractable relaxations are then proposed with various low-rank regularization functions [19, 46, 62, 42].

Despite their varieties, the existing low-rank regularization functions are designed for regular tensor factorization models and cannot be applied to an irregular tensor factorization model like PARAFAC2. In fact, they are not even well-defined on irregular tensors and PARAFAC2. Thus, there lacks tractable and effective low-rank regularization for PARAFAC2 applicable to large-scale irregular tensors.

### 2.2.2 PARAFAC2 for Temporal EHR

PARAFAC2 is the state-of-the-art tensor factorization structure for irregular tensors that do not align naturally along one of its modes. The classic PARAFAC2 (c.f. Fig. 2.1 red box) for irregular tensor $\{\mathbf{X}_k\}$ is formalized below [38]:

**Definition 1.** *(Classic PARAFAC2 model)*

$$\underset{\{\mathbf{U}_k\},\{\mathbf{S}_k\},\mathbf{V}}{\operatorname{argmin}} \sum_{k=1}^{K} \frac{1}{2}\|\mathbf{X}_k - \mathbf{U}_k\mathbf{S}_k\mathbf{V}^\top\|_F^2,$$

*s.t.* $\mathbf{U}_k = \mathbf{Q}_k\mathbf{H}, \mathbf{Q}_k^\top\mathbf{Q}_k = \mathbf{I}, \mathbf{S}_k$ *is diagonal, where* $\mathbf{Q}_k \in \mathbb{R}^{I_k \times R}$ *is orthogonal,* $\mathbf{I}_k \in \mathbb{R}^{R \times R}$ *is the identity matrix and* $R$ *is the target rank of the PARAFAC2 decomposition.*

For temporal EHR data, the factorization matrices have the following interpretation:

$\triangleright$ $\mathbf{U}_k \in \mathbb{R}^{I_k \times R}$ contains temporal evolution for patient $k$: the $r$-th column of $\mathbf{U}_k$ indicates the evolution of the $r$-th phenotype for all $I_k$ clinical visits for patient $k$.

$\triangleright$ $\mathbf{V} \in \mathbb{R}^{J \times R}$ reflects the phenotypes. Each non-zero entry of $V$ indicates the membership of the corresponding $j$-th medical feature in the $r$-th phenotype.

$\triangleright$ $\mathbf{S}_k \in \mathbb{R}^{R \times R}$ is a diagonal matrix with the importance membership of patient $k$ in each one of the $R$ phenotypes. It is often organized into $\mathbf{W} \in \mathbb{R}_+^{R \times K}$ with each row of $\mathbf{W}$ composed by the diagonal of $\mathbf{S}_k$, i.e. $\mathbf{W}(:,k) = \texttt{diag}(\mathbf{S}_k)$.

SPARTan [51] scales PARAFAC2 to large temporal EHR phenotyping by introducing a sparse MTTKRP (abbreviated for Matricized-Tensor-Times-Khatri-Rao-Product) module, which takes advantage of the high input sparsity to reduce the per-iteration cost. Following its efficiency improvement, COPA [2] further introduces various constraints/regularizations to improve the interpretability of the factor matrices for more meaningful pheonotype extraction. For example, COPA introduces the M-spline constraint [58] to $\mathbf{U}_k$ to capture the temporal smoothness, non-negative constraint to $\mathbf{S}_k$ to get positive weight, and sparsity (e.g., $\ell_1$ norm regularization) to $\mathbf{V}$ to induce sparse phenotype definitions.

In sum, despite their improvements on computational efficiency and output interpretability, existing PARAFAC2 methods do not explicitly address the problem of extracting meaningful phenotypes from EHR datasets with moderate ratio of missing and error entries, which severely limits them from more robust clinical usage.

## 2.3 Proposed Method

### 2.3.1 Low-rank Regularization for PARAFAC2

As mentioned, effective low-rank regularization has not been studied for irregular tensors. Recent work [75] proposes to recover each of $\{\mathbf{X}_k\}$'s frontal slices matrix by matrix by robust low-rank matrix completion techniques [8, 47]. The drawback of this approach is that it cannot capture the internal structural correlations across frontal slices, i.e. common information among patients, for temporal EHR phenotyping. As can be seen from our experiments, this approach does not provide satisfactory recovery performance. On the contrary, we propose to impose the low-rankness on $\{\mathbf{X}_k\}$ through adding nuclear norm constraints on the internal factorization matrices $\mathbf{H}, \mathbf{V}, \mathbf{W}$, which are shared by all frontal slices thus capable of capturing cross-slice information.

**Definition 2.** *For irregular tensor*

$$\mathcal{X} = \{\mathbf{X}_k\} \approx \texttt{PARAFAC2}(\{\mathbf{Q}_k\}, \mathbf{H}, \mathbf{V}, \mathbf{W}),$$

*the low-rank regularization function is defined as*

$$\|\boldsymbol{\mathcal{X}}\|_{lr} := \|\mathbf{H}\|_* + \|\mathbf{V}\|_* + \|\mathbf{W}\|_*. \tag{2.2}$$

Our low-rank regularization function enjoys the following nice properties: 1) it is natural to the decomposition structure of the PARAFAC2 model; 2) it can effectively recover the underlying clean and completed tensor $\{\mathbf{X}_k\}$ by capturing cross frontal slice information.

### 2.3.2 REPAIR: Model

Having defined the low-rank regularization function in Definition 2, we formalize the objective function for the REPAIR model in Definition 3. It applies the RLTM framework (i.e. eq.(2.1)) to PARAFAC2, which separates the underlying clean and completed tensor $\boldsymbol{\mathcal{X}} = \{\mathbf{X}_k\}$ and the erroneous tensor $\boldsymbol{\mathcal{E}} = \{\mathbf{E}_k\}$ given the missing and corrupted observation tensor $\boldsymbol{\mathcal{O}} = \{\mathbf{O}_k\}$. Meanwhile, REPAIR decomposes $\{\mathbf{X}_k\}$ into PARAFAC2 structure. The tensor recovery of $\boldsymbol{\mathcal{X}}$ is enforced by the linear constraint between $\boldsymbol{\mathcal{O}}$, $\boldsymbol{\mathcal{X}}$, $\boldsymbol{\mathcal{E}}$ in eq (2.4), low-rank regularization for $\boldsymbol{\mathcal{X}} = \{\mathbf{X}_k\}$ and sparsity constraint for $\boldsymbol{\mathcal{E}}$ in the second row of eq (2.3). The tensor factorization of $\boldsymbol{\mathcal{X}}$ is enforced by the PARAFAC2 loss for $\boldsymbol{\mathcal{X}}$, the temporal smoothness, nonnegativity, and sparsity constraints in the first row of eq (2.3) and additional constraints in eq (2.5).

For EHR phenotype discovery, various constraints should be imposed on the factorization matrices to yield meaningful and high-interpretability phenotypes. The REPAIR model accommodates such interpretability-purposed constraints in eq.(2.3) including: temporal smoothness for $c_1(\mathbf{H})$, non-negativity for $\{c_2(\mathbf{S}_k)\}$, sparsity for $c_3(\mathbf{V})$.

**Definition 3.** *(REPAIR objective function)*

$$
\begin{aligned}
&\underset{\mathbf{Q}_k,\mathbf{H},\mathbf{S}_k,\mathbf{V}}{\operatorname{argmin}} \sum_{k=1}^{K} \big( \overbrace{\|\mathbf{X}_k - \mathbf{U}_k\mathbf{S}_k\mathbf{V}^\top\|_F^2}^{\textit{PARAFAC2 loss for } \mathcal{X}} + \overbrace{\rho_0\|\mathcal{P}_\Omega(\mathbf{E}_k)\|_1 \big)}^{\textit{sparsity for } \mathcal{E}} + \\
&\overbrace{\rho_1\|\mathbf{H}\|_* + \rho_2\|\mathbf{V}\|_* + \rho_3\|\mathbf{W}\|_*}^{\textit{low-rankness for } \mathcal{X}} + \overbrace{c_1(\mathbf{H})}^{\textit{smoothness}} + \overbrace{\sum_{k=1}^{K} c_2(\mathbf{S}_k)}^{\textit{nonnegativity}} + \overbrace{c_3(\mathbf{V})}^{\textit{sparsity}},
\end{aligned}
\tag{2.3}
$$

$$
\textit{s.t. for } k = 1, ..., K, \ \overbrace{\mathcal{P}_\Omega(\mathbf{O}_k) = \mathcal{P}_\Omega(\mathbf{X}_k + \mathbf{E}_k)}^{\textit{linear constraint between } \mathcal{O},\mathcal{X},\mathcal{E}}, \tag{2.4}
$$

$$
, \quad \underbrace{\mathbf{U}_k = \mathbf{Q}_k\mathbf{H}, \ \mathbf{Q}_k^\top\mathbf{Q}_k = \mathbf{I}}_{\textit{constraints for PARAFAC2 decomposition}} \tag{2.5}
$$

*where* $\mathbf{H}, \{\mathbf{S}_k\}, \mathbf{I} \in \mathbb{R}^{R\times R}, \ \mathbf{Q}_k \in \mathbb{R}^{I_k\times R}$.

### 2.3.3 REPAIR: Optimization

To solve the REPAIR model, a straightforward approach is to introduce auxiliary variables for the low-rank and interpretability regularizations, then solve the problem by multi-block Alternating Direction Method of Multipliers (ADMM) [5]. Inspired by the more flexible Alternating Optimization ADMM (AO-ADMM) [30], we design a two-phase alternative optimization algorithm to accommodate more constraints. The REPAIR optimization proceeds by iterating between the two phases: I) updating the factorization matrices $\{\mathbf{Q}_k\}, \mathbf{H}, \mathbf{V}, \mathbf{W}$; II) separating the $\mathcal{X}$ and $\mathcal{E}$ from $\mathcal{O}$. For I), we factorize the intermediate (inaccurate) recovered tensor $\mathcal{X}$ by solving an approximated PARAFAC2; for II), we follow standard ADMM to convert the linear constraint of eq.(2.4) by introducing Lagrangian dual variable $\{\Gamma_O^k\}$ to get rid of the constraint in eq.(2.4) as shown in Definition 3. This way, REPAIR can accommodate a variety of constraints for each factorization for better interpretability. Also, the optimizations for each factor are more independent, which makes it easier to deal with.

**Definition 4.** *The augmented Lagrangian dual objective is,*

$$
\begin{aligned}
&\sum_{k=1}^{K}\Big(\|\mathbf{X}_k-\mathbf{Q}_k\mathbf{H}\mathbf{S}_k\mathbf{V}^\top\|_F^2-\langle\Gamma_O^k,\mathbf{O}_k-\mathbf{X}_k-\mathbf{E}_k\rangle\\
&+\frac{\eta_O^k}{2}\|\mathbf{O}_k-\mathbf{X}_k-\mathbf{E}_k\|_F^2+\rho_0\|\mathcal{P}_\Omega(\mathbf{E}_k)\|_1\Big)\\
&+\big(\rho_1\|\mathbf{H}\|_*+\rho_2\|\mathbf{V}\|_*+\rho_3\|\mathbf{W}\|_*\big)+\big(c_1(\mathbf{H})+c_2(\mathbf{W})+c_3(\mathbf{V})\big)\\
&s.t.\ \mathbf{S}_k=\texttt{diag}(\mathbf{W}(k,:)),\ \ \mathbf{Q}_k^\top\mathbf{Q}_k=\mathbf{I},\ \ for\ k=1,...,K.
\end{aligned}
$$

**Phase I: Approximated PARAFAC2**

In the first phase, we update the factorization matrices $\{\mathbf{Q}_k\},\mathbf{H},\mathbf{V},\mathbf{W}$ with $\{\mathbf{X}_k\}$ and $\{\mathbf{E}_k\}$ fixed, which can be intuitively seen as decomposing the latest recovered tensor $\{\mathbf{X}_k\}$ into PARAFAC2. In practice, we observe that it is enough to run PARAFAC2 by one iteration in this phase to achieve the overall convergence, which avoids heavy computation of solving precise PARAFAC2.

**Update $\mathbf{Q}_k$:** To update $\mathbf{Q}^k$, we need Lemma 1 below:

**Lemma 1.** *The Orthogonal Procrustes problem is:*

$$\mathbf{Q}^{\#}=\underset{\mathbf{Q}:\mathbf{Q}^\top\mathbf{Q}=\mathbf{I}}{\operatorname{argmin}}\|\mathbf{Q}\mathbf{A}-\mathbf{B}\|_F^2,$$

*which has the closed-form solution:* $\mathbf{Q}^{\#}=\mathbf{P}\mathbf{Z}^\top$*, where* $[\mathbf{P},\Sigma,\mathbf{Z}]=\texttt{svd}(\mathbf{B}\mathbf{A}^\top)$ *and* $\texttt{svd}(\cdot)$ *is singular value decomposition.*

When applied to the update of $\mathbf{Q}_k$, with other factors fixed, we have

$$\mathbf{Q}_k=\underset{\mathbf{Q}_k:\mathbf{Q}_k^\top\mathbf{Q}_k=\mathbf{I}}{\operatorname{argmin}}\|\mathbf{X}_k-\mathbf{Q}_k\mathbf{H}\mathbf{S}_k\mathbf{V}^\top\|_F^2. \tag{2.6}$$

Table 2.2: Additional symbols for REPAIR optimization

| Symbol | Definition |
|---|---|
| $\rho_0, \rho_1, \rho_2, \rho_3$ | Balancing hyper-parameters |
| $\mathbf{H}^l, \mathbf{V}^l, \mathbf{W}^l$ | Auxiliary variable for low-rank constr. |
| $\Gamma_H^l, \Gamma_W^l, \Gamma_V^l$ | Lagrangian dual for low-rank constr. |
| $\eta_H^l, \eta_W^l, \eta_V^l$ | Lagrangian constant for low-rank constr. |
| $\mathbf{H}^c, \mathbf{V}^c, \mathbf{W}^c$ | Auxiliary variable for interpretability constr. |
| $\Gamma_H^c, \Gamma_W^c, \Gamma_V^c$ | Lagrangian dual for interpretability constr. |
| $\eta_H^c, \eta_W^c, \eta_V^c$ | Lagrangian constant for interpretability constr. |

Let $\mathbf{B} = \mathbf{X}_k$ and $\mathbf{A} = \mathbf{H}\mathbf{S}_k\mathbf{V}^\top$ and by Lemma 1:

$$\mathbf{Q}_k = \mathbf{P}_k\mathbf{Z}_k^\top, \text{ where}[\mathbf{P}_k, \Sigma, \mathbf{Z}_k] = \texttt{svd}(\mathbf{X}_k\mathbf{V}\mathbf{S}_k\mathbf{H}^\top). \tag{2.7}$$

**Update H:** After obtaining $\{\mathbf{Q}_k\}$, we denote $\mathbf{Y}_k = \mathbf{Q}_k^\top\mathbf{X}_k$, for $k = 1, ..., K$, and let $\boldsymbol{\mathcal{Y}}$ be the tensor with $\mathbf{Y}_k$ being its frontal slice. We then update $\mathbf{H}, \mathbf{V}, \mathbf{W}$ alternatively by solving three constrained least squares sub-problems. Due to the symmetry of the three sub-problems, we elaborate the update for $\mathbf{H}$ as an example.

$$\mathbf{H} = \underset{\mathbf{H}}{\operatorname{argmin}} \|\boldsymbol{\mathcal{Y}}_{(1)} - \mathbf{H}(\mathbf{V} \odot \mathbf{W})^\top\|_F^2 + \rho_1\|\mathbf{H}\|_* + c_1(\mathbf{H}).$$

We introduce two auxiliary variables $\mathbf{H}^l$ and $\mathbf{H}^c$ to separate the low-rank and interpretability constraints:

$$\underset{\mathbf{H},\mathbf{H}^l,\mathbf{H}^c}{\operatorname{argmin}} \|\boldsymbol{\mathcal{Y}}_{(1)} - \mathbf{H}(\mathbf{V} \odot \mathbf{W})^\top\|_F^2 + \rho_1\|\mathbf{H}^l\|_* + c_1(\mathbf{H}^c),$$
$$s.t. \ \mathbf{H}^l = \mathbf{H}, \ \mathbf{H}^c = \mathbf{H}.$$

The above can be solved by ADMM after introducing Lagrangian dual variable

$\Gamma_H^l, \Gamma_H^c$ and constants $\eta_H^l, \eta_H^c$, correspondingly:

$$\begin{aligned}\underset{\mathbf{H},\mathbf{H}^l,\mathbf{H}^c}{\operatorname{argmin}} \|\boldsymbol{\mathcal{Y}}_{(1)} - \mathbf{H}(\mathbf{V}\odot\mathbf{W})^\top\|_F^2 + \rho_1\|\mathbf{H}^l\|_* + c_1(\mathbf{H}^c)\\ - \langle\Gamma_H^l, \mathbf{H}-\mathbf{H}^l\rangle + \frac{\eta_H^l}{2}\|\mathbf{H}-\mathbf{H}^l\|_F^2\\ - \langle\Gamma_H^c, \mathbf{H}-\mathbf{H}^c\rangle + \frac{\eta_H^c}{2}\|\mathbf{H}-\mathbf{H}^c\|_F^2.\end{aligned}$$

To solve it by ADMM, we have the following update sequence for $\mathbf{H}, \mathbf{H}^l, \mathbf{H}^c$ and dual $\Gamma_H^l, \Gamma_H^c$:

$$\begin{aligned}\mathbf{H} = \Big(\boldsymbol{\mathcal{Y}}_{(1)}(\mathbf{V}\odot\mathbf{W}) + \Gamma_H^l + \Gamma_H^c + \eta_H^l\mathbf{H}^l + \eta_H^c\mathbf{H}^c\Big)\\ \cdot\Big((\mathbf{V}^\top\mathbf{V}) * (\mathbf{W}^\top\mathbf{W}) + (\eta_H^l + \eta_H^c)\mathbf{I}\Big)^\dagger,\end{aligned} \tag{2.8}$$

where $\odot$ is the Khatri Rao product, $*$ is the Hadamard product and $\dagger$ is the pseudo-inverse.

$$\mathbf{H}^l = \underset{\mathbf{H}^l}{\operatorname{argmin}} \frac{\eta_H^l}{2}\|\mathbf{H}^l - \mathbf{H}\|_F^2 - \langle\Gamma_H^l, \mathbf{H}^l - \mathbf{H}\rangle + \rho_1\|\mathbf{H}^l\|_*,$$

which has the proximal operator [50] with respect to the nuclear norm $\|\cdot\|_*$, a.k.a. singular value thresholding [8], as its closed-form solution:

$$\mathbf{H}^l = \mathtt{prox}_{\frac{\rho_1}{\eta_H^l}\|\cdot\|_*}(\mathbf{H} + \frac{\Gamma_H^l}{\eta_H^l}) = \mathbf{P}\mathtt{Diag}(\max\{0, \boldsymbol{\sigma} - \frac{\rho_1}{\eta_H^l}\})\mathbf{Z}^\top, \tag{2.9}$$

where $[\mathbf{P}, \mathtt{Diag}(\boldsymbol{\sigma}), \mathbf{Z}] = \mathtt{svd}(\mathbf{H} + \frac{\Gamma_H^l}{\eta_H^l})$.

$$\mathbf{H}^c = \underset{\mathbf{H}^c}{\operatorname{argmin}} \frac{\eta_H^c}{2}\|\mathbf{H}^c - \mathbf{H}\|_F^2 - \langle\Gamma_H^c, \mathbf{H}^c - \mathbf{H}\rangle + c_1(\mathbf{H}^c),$$

which has the proximal operator with respect to the constraint function $c_1(\cdot)$ as its closed-form solution:

$$\mathbf{H}^c = \mathtt{prox}_{\frac{1}{\eta_H^c}c_1}(\mathbf{H} + \frac{\Gamma_H^c}{\eta_H^c}). \tag{2.10}$$

The Lagrangian dual variables are update as follows:

$$\Gamma_H^c = \Gamma_H^c - \eta_H^c(\mathbf{H} - \mathbf{H}^c); \tag{2.11}$$

$$\Gamma_H^l = \Gamma_H^l - \eta_H^l(\mathbf{H} - \mathbf{H}^l). \tag{2.12}$$

**Update $\mathbf{V}, \mathbf{W}$:** The update for $\mathbf{V}$ (along with $\mathbf{V}^l, \mathbf{V}^c$) and $\mathbf{W}$ (along with $\mathbf{W}^l, \mathbf{W}^c$) are similar to $\mathbf{H}$:

$$\begin{aligned}
\mathbf{V} &= \big(\boldsymbol{\mathcal{Y}}_{(2)}(\mathbf{H} \odot \mathbf{W}) + \Gamma_V^l + \Gamma_V^c + \eta_V^l \mathbf{V}^l + \eta_V^c \mathbf{V}^c\big) \\
&\quad \cdot \big((\mathbf{H}^\top \mathbf{H}) * (\mathbf{W}^\top \mathbf{W}) + (\eta_V^l + \eta_V^c)\mathbf{I}\big)^\dagger; \\
\mathbf{V}^l &= \texttt{prox}_{\frac{\rho_3}{\eta_V^l}\|\cdot\|_*}(\mathbf{V} + \frac{\Gamma_V^l}{\eta_V^l}); \mathbf{V}^c = \texttt{prox}_{\frac{1}{\eta_V^c}c_3}(\mathbf{V} + \frac{\Gamma_V^c}{\eta_V^c}); \\
\Gamma_V^c &= \Gamma_V^c - \eta_V^c(\mathbf{V} - \mathbf{V}^c); \Gamma_V^l = \Gamma_V^l - \eta_V^l(\mathbf{V} - \mathbf{V}^l).
\end{aligned} \tag{2.13}$$

$$\begin{aligned}
\mathbf{W} &= \big(\boldsymbol{\mathcal{Y}}_{(3)}(\mathbf{V} \odot \mathbf{H}) + \Gamma_W^l + \Gamma_W^c + \eta_W^l \mathbf{W}^l + \eta_W^c \mathbf{W}^c\big) \\
&\quad \cdot \big((\mathbf{V}^\top \mathbf{V}) * (\mathbf{H}^\top \mathbf{H}) + (\eta_W^l + \eta_W^c)\mathbf{I}\big)^\dagger; \\
\mathbf{W}^l &= \texttt{prox}_{\frac{\rho_2}{\eta_W^l}\|\cdot\|_*}(\mathbf{W} + \frac{\Gamma_W^l}{\eta_W^l}); \mathbf{W}^c = \texttt{prox}_{\frac{1}{\eta_W^c}c_2}(\mathbf{W} + \frac{\Gamma_W^c}{\eta_W^c}); \\
\Gamma_W^c &= \Gamma_W^c - \eta_W^c(\mathbf{W} - \mathbf{W}^c); \Gamma_W^l = \Gamma_W^l - \eta_W^l(\mathbf{W} - \mathbf{W}^l).
\end{aligned} \tag{2.14}$$

### 2.3.4 Phase II: robust underlying tensor recovery

In this second phase, we alternatively update the low-rank tensor $\{\mathbf{X}_k\}$ which is the underlying clean and completed tensor, and the sparse tensor $\{\mathbf{E}_k\}$ which is the corrupted tensor, as well as the Lagrangian dual variable $\{\Gamma_O^k\}$.

**Update $\{\mathbf{X}_k\}$:** It amounts to

$$\mathbf{X}_k = \underset{\mathbf{X}_k}{\operatorname{argmin}} \|\mathbf{X}_k - \mathbf{Q}_k\mathbf{H}\mathbf{S}_k\mathbf{V}^\top\|_F^2 - \langle \Gamma_O^k, \mathbf{O}_k - \mathbf{X}_k - \mathbf{E}_k\rangle \\ + \frac{\eta_O^k}{2}\|\mathbf{O}_k - \mathbf{X}_k - \mathbf{E}_k\|_F^2,$$

which has the solution

$$\mathbf{X}_k = \mathbf{Q}_k\mathbf{H}\mathbf{S}_k\mathbf{V}^\top - \Gamma_O^k + \eta_O^k(\mathbf{O}_k - \mathbf{E}_k). \tag{2.15}$$

**Update $\{\mathbf{E}_k\}$:** The update of $\mathbf{E}_k$ separates into $P_\Omega(\mathbf{E}_k)$ and $P_{\Omega^\perp}(\mathbf{E}_k)$:

$$\mathcal{P}_\Omega(\mathbf{E}_k) = \mathcal{P}_\Omega(\mathtt{prox}_{\frac{\rho_0}{\eta_O^k}\|\cdot\|_1}(\mathbf{O}_k - \mathbf{X}_k - \frac{1}{\eta_O^k}\Gamma_O^k)), \tag{2.16}$$

where $\mathtt{prox}_{\frac{\rho_0}{\eta_O^k}\|\cdot\|_1}(\cdot)$ is the proximal operator for the $\ell_1$-norm, a.k.a. soft-thresholding.

$$\mathcal{P}_{\Omega^\perp}(\mathbf{E}_k) = \mathcal{P}_{\Omega^\perp}(\mathbf{O}_k - \mathbf{X}_k). \tag{2.17}$$

**Update $\{\Gamma_O^k\}$:** This is Lagriangian dual variable update:

$$\Gamma_O^k = \Gamma_O^k - \eta_O^k(\mathbf{O}_k - \mathbf{X}_k - \mathbf{E}_k). \tag{2.18}$$

The complete REPAIR algorithm is summarized in Algorithm 1.

**Algorithm 1** Optimization framework for REPAIR

```
Input: Input tensor $\mathfrak{O}$; Model parameters $\rho_0$-$\rho_3$; Optimization parameters $\eta$'s; Interpretability constraint types $c_1, c_2, c_3$; Initial rank estimation $R$.
while Not reach convergence criteria do
    %% Phase I begins
    while Not reach inner loop max do
        Update $\{\mathbf{Q}_k\}$ by eq.(2.7);
        Update $\mathbf{H}, \mathbf{V}, \mathbf{W}$-related variables sequantially;
    end while
    %% Phase II begins
    Update $\{\mathbf{X}_k\}$ by eq.(2.15);
    Update $\{\mathbf{E}_k\}$ by eq.(2.16)&(2.17);
    Update $\{\Gamma_O^k\}$ by eq.(2.18).
end while
Output: Phenotype factor matrices $\{\mathbf{U}_k\} = \{\mathbf{Q}_k\mathbf{H}\}, \{\mathbf{S}_k\}, \mathbf{V}$; Recovered tensor $\{\mathbf{X}_k\}$.
```

## 2.4 Experimental Evaluation

### 2.4.1 Experiment Setup

#### Datasets

We evaluate REPAIR on two real-world publicly-available temporal EHR datasets: CMS[1] and MIMIC-III[2].

**CMS:** Centers for Medicare and Medicaid Services (CMS) contains synthesized data of Medicare beneficiaries in 2008 and their claims from 2008 to 2010. We construct a three-mode tensor with patients (along mode-3), diagnosis or ICD9 codes (along mode-2), and clinical visits (along mode-1). Each tensor value $\mathbf{O}_{ijk}$ indicates the number of times a patient $k$ has a diagnosis $j$ during visit $i$. We keep records of patients with at least 2 hospital visits. The resulting number of patients is 50,000 with 284 features (diagnosis categories) and the maximum number of observations for a patient is 1500. The number of non-zero elements is 49 million. 89% of the

[1] `https://www.cms.gov/Research-Statistics-Data-and-Systems/Downloadable-Public-Use-Files/SynPUFs/DE_Syn_PUF.html`

[2] `https://mimic.physionet.org/`

non-zero elements are 1, and 11% are 2.

**MIMIC-III:** The intensive care unit (ICU) dataset is collected between 2001 and 2012. Similar to CMS, we construct the three-mode tensor and keep records of patients with at least 2 hospital visits. We select 202 ICD-9 codes that have the highest frequency as in [36]. The resulting number of patients is 2323 with 202 features (diagnosis codes) and the maximum number of observations for a patient is 41. The number of non-zero entries is 3 million. 96% of non-zero elements are 1, and 4% are 2.

### Methods for Comparison

Since there are no existing robust methods for irregular tensor factorization with missing and erroneous data, we compare with two groups of methods: 1) state-of-the-art irregular tensor factorization methods, which however have no mechanisms to handle missing and erroneous data; 2) we adapt existing robust methods for regular tensor factorization to irregular tensors for comparison.

**1) Irregular tensor factorization methods**

- **SPARTan [51]- scalable PARAFAC2**: A recently-proposed methodology for fitting PARAFAC2 on large and sparse data. It does not explicitly address missing or erroneous data.

- **COPA [2]- scalable PARAFAC2 with additional regularizations**: A state-of-the-art irregular tensor factorization method. It further introduces various constraints/regularizations to improve the interpretability of the factor matrices for more meaningful pheonotype extraction.

**2) Adapted robust regular tensor factorization methods**

- **CP-WOPT [1] - robust method for regular tensors**: CP-WOPT is a robust method for *regular* tensors which uses a weighted optimization method for CP tensor completion and factorization with incomplete data. To make it work with irregular tensors, we first zero-pad the irregular tensors to aligned ones and then apply CP-WOPT.

- **lrmcR [47] + COPA - robust method for matrix completion**: lrmcR [47] is a robust low-rank matrix completion method. To make it work for irregular tensors, we apply lrmcR to recover the frontal slices one by one and then apply COPA for phenotype extraction.

**Implementation details**

REPAIR[3] is implemented in Matlab R2019a and includes functionalities from the Tensor Toolbox[4]. We utilize the Parallel Computing Toolbox of Matlab. For CMS dataset, 30 workers are used; and for MIMIC-III, 4 workers are used. We report the hyper-parameters of REPAIR in the experiment in Table 2.3. The code of COPA and SPARTan are publicly available at: `https://github.com/aafshar/COPA`; `https://github.com/kperros/SPARTan`. For the COPA related methods, we use the same regularizations $c_1, c_2, c_3$ with REPAIR, as given in Defintion 3.

We evaluate recovery accuracy and robustness of the tensor factorization against various conditions of missing and erroneous values. We empirical study the convergence behaviour of all compared methods. In case studies, we evaluate the quality of the factorization matrices (i.e. extracted phenotypes) for downstream analysis via: 1) identification of higher-risk patient sub-groups; 2) in-hospital mortality prediction.

## 2.4.2 Tensor Factorization Robustness

In order to test the robustness of REPAIR model against missing and error entries, we randomly add missing values and error entries into the two datasets. We design

[3] `https://github.com/Emory-AIMS/Repair`

[4] `https://www.tensortoolbox.org/`

Table 2.3: Parameters for CMS and MIMIC-III

| Parameter | CMS | MIMIC-III |
|---|---|---|
| $\rho_0$ | 1e-3 | 1e-3 |
| $\rho_1$ | 1e-3 | 1e-3 |
| $\rho_2$ | 1e-4 | 1e-4 |
| $\rho_3$ | 1e-4 | 1e-4 |
| $c_1$ | 253 | 270 |
| $c_3$ | 0.0000085 | 0.0000085 |

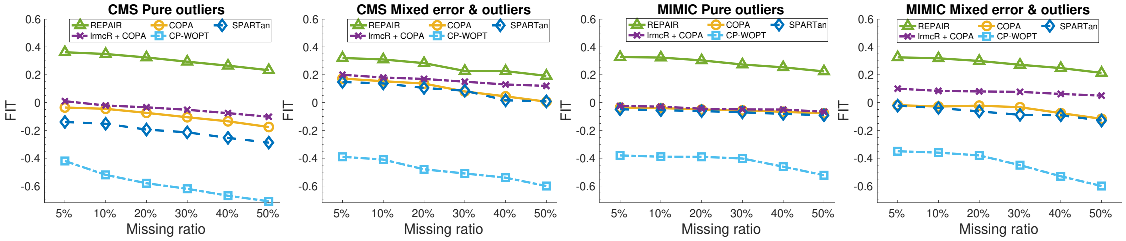


Figure 2.2: Robustness against varying ratio of missing entries

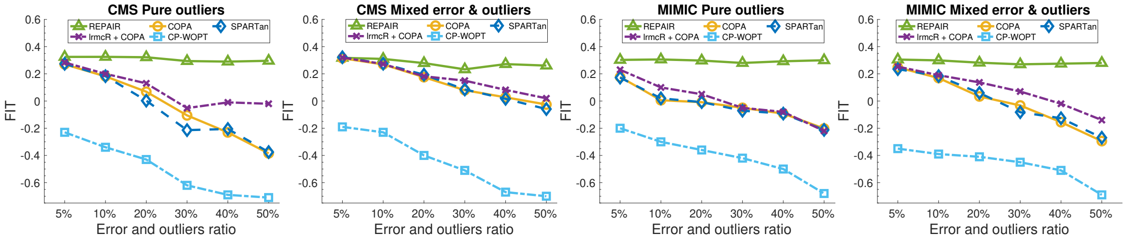


Figure 2.3: Robustness against varying ratio of erroneous entries

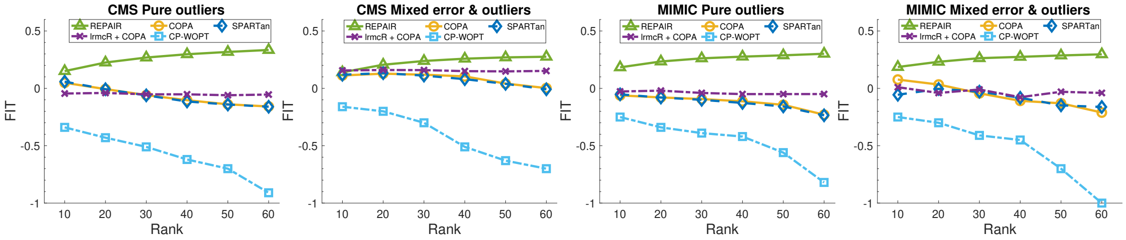


Figure 2.4: Impact of varying rank estimation

two types of errors. The first is referred as pure outliers, where we randomly pick tensor entries and set their values to be 4, which largely deviates from normal values (1 and 2 in these datasets). The second is mixed error, where we randomly pick certain entries and set their values to be 3 or 4 (outliers) with half probability, and 1 or 2 (normal values but flipped from the original value) with half probability. The original uncorrupted tensor denoted as $\{\mathbf{G}_k\}$ serves as the ground truth. We adopt the $FIT \in (-\infty, 1]$ score [6] as the quality measure (the higher the better):

$$FIT = 1 - \frac{\sum_{k=1}^{K} \|\mathbf{G}_k - \mathbf{U}_k \mathbf{S}_k \mathbf{V}^T\|^2}{\sum_{k=1}^{K} \|\mathbf{G}_k\|^2}. \tag{2.19}$$

In the following experiment, we run each setting for 5 different random initialization and report the average $FIT$. When the compared methods' $FIT$ drop below 0 (i.e. fail to recover), we report the averaged highest $FIT$ before the algorithm diverges.

### Robustness against Varying Ratio of Missing Entries

We first evaluate the impact of varying missing ratios on the robustness of the methods with fixed 30% error ratios as Figure 2.2 shows. If no error and missing entries are added into data sets, REPAIR, COPA, SPARTAN and lrmcR + COPA methods can achieve similar $FIT$ scores around 0.42 (please note that it is a typical FIT range for this task, e.g., [2]). However, the four baselines' $FIT$ scores quickly drop as the missing ratio increases, in many cases below 0, which indicates baselines fail to recover the tensor even with small missing ratios. Repair outperforms all methods significantly. lrmcR+COPA performances slightly better than COPA thanks to its completion of the slices. lrmcR + COPA and COPA perform better than SPARTan thanks to its additional temporal constraints. CP-WOPT performs the worst, since it does not address the irregularity of the tensors, even when it explicitly deals with missing data, which indicates the importance of addressing the irregularity. We also observe that pure outlier's performances are often better than mixed error cases, as pure outliers is easier for REPAIR model to separate the error entries.

Table 2.4: Basis number for CMS and MIMIC-III

| Rank R | CMS | MIMIC-III |
|---|---|---|
| 10 | 102 | 140 |
| 20 | 190 | 200 |
| 30 | 215 | 220 |
| 40 | 253 | 270 |
| 50 | 270 | 320 |
| 60 | 320 | 360 |

**Robustness against Varying Ratio of Erroneous Entries**

We set the missing ratio to be 30%, and change the error ratio from 5% to 50%. Figure 2.3 shows the $FIT$ scores of different methods with respect to varying error ratios for the two data sets under two error cases. With increasing error ratios, four baselines' recovery performance drop dramatically, while REPAIR enjoys a robust performance with an average $FIT$ around 0.32.

**Impact of Varying Initial Target Rank Estimation**

We set missing and error ratios both to 30% and vary the initial rank estimation $R$. The detailed $c_1$ (basis function number used by M-spline function for promoting temporal smoothness) for different data sets and various ranks are shown in Table 2.4. With a higher rank $R$, the $FIT$ of REPAIR slightly increases while always outperforming all other methods as Figure 2.4 shows. This is because the low-rank regularization function is able to iteratively decrease the target rank during the optimization (e.g. by soft-thresholding the singular values) and make it approach the optimal one.

### 2.4.3 Convergence Comparison.

Figure 2.5 shows the convergence comparison of REPAIR, SPARTan, COPA, lrmcR + COPA, CP-WOPT on CMS with missing ratio 10% and mixed error ratio 20% (under this setting all algorithms can recover the tensor without failure). By Figure

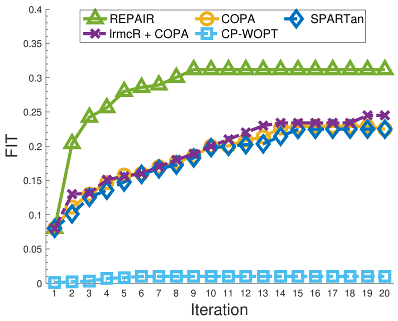


Figure 2.5: Convergence comparison of REPAIR, SPARTan, COPA, lrmcR + COPA, CP-WOPT

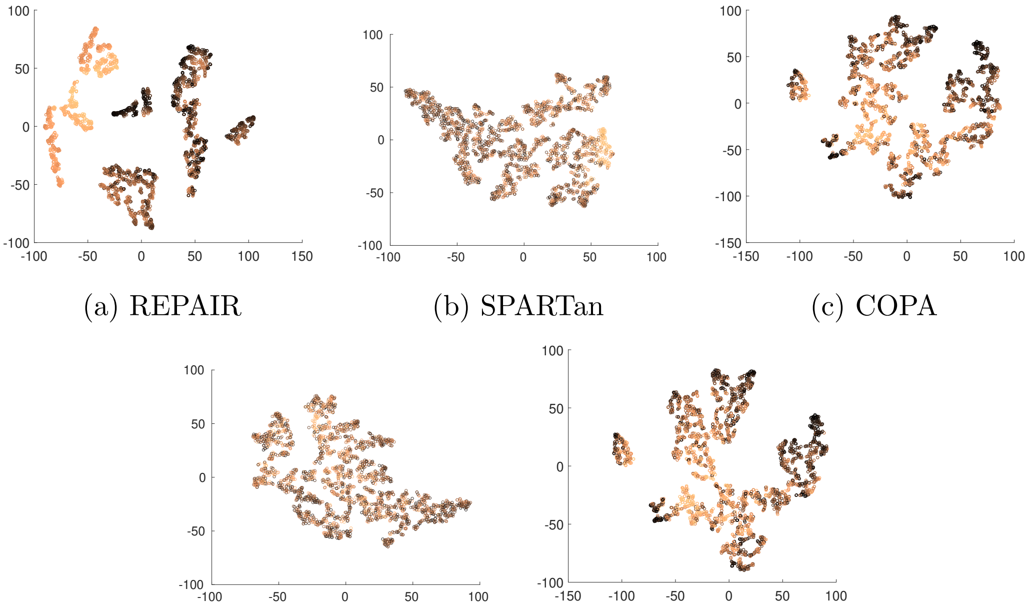
2.5, REPAIR flats around 9-10 iterations (with a higher FIT score than baselines), while it takes baselines 14-15 iterations. This shows that REPAIR not only enjoys more robust recovery, but also faster convergence.

### 2.4.4 Quality of the Extracted Phenotypes: Two Case Studies

The previous experiments show the robustness of REPAIR in terms of how well the factorization matrices (i.e. the extracted phenotypes) recover the ground truth tensor under the $FIT$ metric. In this subsection, our goal is to evaluate how meaningful and useful the extracted phenotypes are. We use MIMIC-III for this set of experiments and set both missing and error ratios to 30%.

#### Identification of Higher-risk Patient Subgroups

The low-dimensional patient representations of PARAFAC2 are effective in distinguishing between higher and lower mortality risk patients [52]. We attempt to test if REPAIR can identify higher-risk patient subgroups if the data contains erroneous and missing entries. The $k$-th row of patient-by-phenotypes matrix $\mathbf{W} \in \mathbb{R}^{k \times R}$ contains the diagonal of $\mathbf{S}_k$, which indicates importance membership of patient $k$ in each of the



(a) REPAIR

(b) SPARTan

(c) COPA

(d) CP-WOPT

(e) lrmcR + COPA

Figure 2.6: tSNE visualization of patient representations learned by REPAIR, SPARTan, COPA, CP-WOPT, and lrmcR+COPA. Each point represents a patient, the color corresponding to the weight of the “oncological conditions” phenotype (lighter means higher weight).

phenotypes. We select the largest-variance column among $\mathbf{S}_k$, which is called the “oncological conditions” phenotype. We set $R = 4$, and use the tSNE [71] software to reduce 4-dimensional vectors to 2-dimensional space, and color each point corresponding to the weight of the “oncological conditions” phenotype (lighter means higher weight). As Figure 2.6 shows, REPAIR can successfully split the patients into two sub-groups while the baselines fail to distinguish the patients.

We perform clustering using K-means (with $k = 2$) on the tSNE result. For the clusters learned by REPAIR, higher risk cluster (corresponding to the left light sub group in Figure 2.6a) and the lower-risk cluster (corresponding to the right dark sub group in Figure 2.6a) are 68.79%, 49.91% respectively. We summarize the average mortality risk of the higher-risk cluster, lower-risk cluster, and their difference in Table 2.5. REPAIR can achieve 18.88% difference, which has the best discriminative capability among all compared methods. In addition, our 18.88%-difference is com-

| Method | REPAIR | SPARTan | COPA | CP-WOPT | lrmcR + COPA |
|---|---|---|---|---|---|
| Higher-risk Mortality Rate | 68.79% | 59.86% | 60.03% | 59.60% | 60.55% |
| Lower-risk Mortality Rate | 49.91% | 59.13% | 58.92% | 59.43% | 58.45% |
| Difference | **18.88%** | 0.83% | 1.11% | 0.5% | 2.1% |

Table 2.5: Summary of average mortality risk of the higher-risk cluster, lower-risk cluster, and their difference. The two clusters are obtained by k-means clustering ($k = 2$). REPAIR can achieve 18.88% difference, which has the best discriminative capability among all compared methods, under the setting of adding 30% erroneous and 30% missing entries.

parable to the 21%-difference reported in [52], which is the journal extension of the SPARTan algorithm [51], and has a clinical expert's endorsement. Because of the extra error and missing entries, our setting is more challenging than [52]. In sum, it shows our method is robust enough to achieve clinical meaningful result comparable to [52].

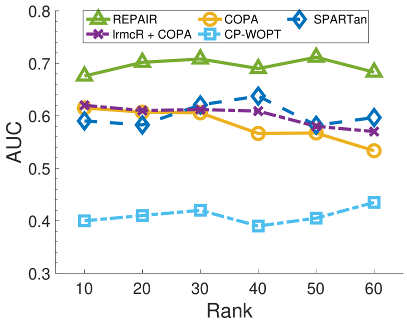


Figure 2.7: In-hospital mortality prediction in AUC. REPAIR outperforms 17% in terms of prediction performance comparing to the best baseline method lrmcR + COPA

.

### In-hospital Mortality Prediction

We also measure REPAIR's phenotype extraction quality under missing and error entries by the predictive power of the discovered phenotypes. A logistic regression model is trained using the patients' membership indicator $S_k$ as features, which is then utilized for predicting in-hospital mortality. We use five 70-30 train-test splits and evaluate the model using the area under the receiver operating characteristic curve

(AUC). As Figure 2.7 shows, the average score of lrmcR + COPA is 0.605, which performs best among four baselines. REPAIR's average score is 0.703, and offers a 17% prediction performance improvement when compared to lrmcR + COPA, which verifies the robustness and usefulness of the extracted phenotypes.

# Chapter 3

# REBAR: RNN Regularized Robust Irregular Tensor Factorization and Completion

## 3.1 Overview

The existing PARAFAC2 tensor factorization methods only impose linear and human-defined temporal regularization functions, which fail to capture non-linear and complex temporal information in real-world scenarios. With the prevalence of deep neural networks, a natural idea is to further enhance robustness, phenotype representations, and predictability using these models. Besides irregularity, EHRs are also prone to missing entries.

In this chapter, we propose **REBAR**, an **R**NN R**E**gularized Ro**B**ust P**AR**AFAC2 Irregular Tensor Factorization and Completion, to complete an irregular tensor and extract phenotypes in the presence of missing values. As shown in Figure 3.1, given an irregular tensor containing missing entries, $Ok$, we add a low-rankness constraint and RNN regularization together to reconstruct $Ok$, and then extract factor matrices

$U_k, S_k, V^\top$ for further downstream analysis and phenotype interpretation.

REBAR has the following appealing features that distinguish it from previous PARAFAC2 methods. (1) It generalizes the loss functions from the sole choice of the least square norm to any smooth loss function, which better suits input tensors with various data types. (2) It accommodates a wide selection of regularization, including statistical learning-based e.g., $l_1$ norm and nuclear norm, deep learning-based e.g., RNN regularization, and composite, to better capture the intrinsic nature of the irregular temporal EHR data. (3) It introduces new optimization geared to fully exploit the parallel computing capability of modern GPUs to boost efficiency.

In summary, we list our main contributions below:

1. We propose a robust RNN and low-rank regularized PARAFAC2 tensor factorization method for irregular tensors to handle potentially missing entries in the input tensor.

2. We introduce a new generalized PARACA2 model with generic loss functions that enable the user to adapt REBAR to suit the data type.

3. We propose a new hybrid optimization framework for PARAFAC2 using stochastic gradient descent and proximal average that can handle multiple regularizations and supports a generalized loss function.

4. We evaluate **REBAR** on three real-world temporal EHR datasets with a set of experiments, which verify the improved recovery and factorization robustness against missing values. Through three case studies: interpretation of the dynamic subphenotypes trajectory, downstream prediction analysis and scalability analysis, we further demonstrate that REBAR can robustly and scalably extract meaningful and high predictability phenotypes with missing data.

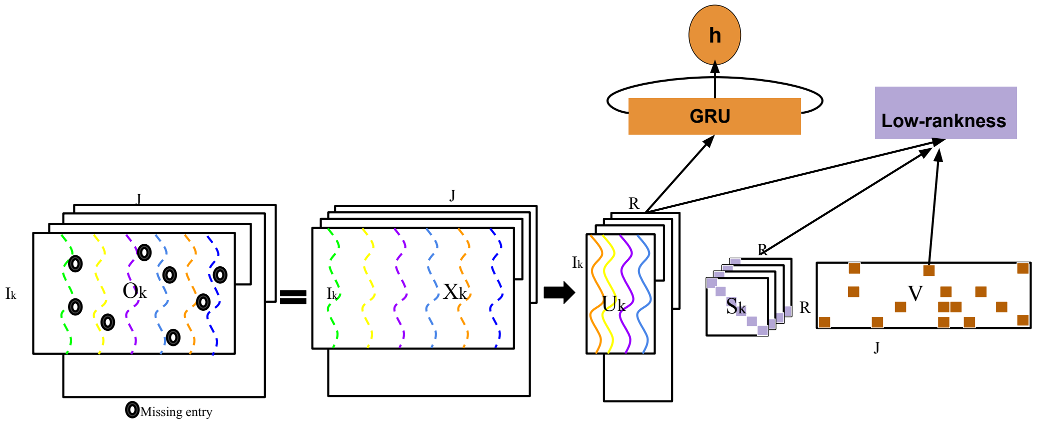


Figure 3.1: REBAR overview

## 3.2 Preliminaries and Backgrounds

### 3.2.1 PARAFAC2

First, we introduce the necessary tensor operations relating to PARAFAC2. Table 3.1 summarizes the notations used throughout the chapter.

A tensor's order, also known as ways or modes, is defined as the number of its dimensions (e.g., vectors are 1-order tensors and matrices are 2-order tensors). Extracting a fiber means fixes all modes but one. For example, a matrix column is a mode-1 fiber. Extracting a slice means fixing all modes but two. In particular, the $\mathbf{X}(:,:,k)$ slices of a third-order tensor $\mathbf{X}$ are called the frontal ones and we denote them as $\mathbf{X}_k$. Tensor unfolding, or matricization, is a fundamental operation and a building block for most tensor methods. It logically reorganizes tensors into other forms without changing the values themselves. The mode-$n$ matricization of an N-order tensor $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ is denoted by $\mathbf{X}_{(n)} \in \mathbb{R}^{I_n \times I_1 I2 \ldots I_{n-1} I_{n+1} \ldots I_N}$ and arranges the mode-$n$ fibers of the tensor as columns of the resulting matrix.

CP decomposition [10, 22, 26], also known as PARAFAC, is one of the most popular tensor factorization methods. It approximates a tensor into a sum of $R$ rank-one tensors. $R$ is the rank of tensor $\mathbf{X} \in \mathbb{R}^{k \times I \times J}$, which can be expressed as:

Table 3.1: Symbols and notations used in chapter 3

| Symbol | Definition |
|---|---|
| $\mathbf{x}, \mathbf{X}, \mathcal{X}$ | Vector, Matrix, Tensor |
| $\mathbf{X}_k$ | $k$-th frontal slice of $\mathcal{X}$ |
| $\mathcal{X}_{(n)}$ | Mode-$n$ matricization of $\mathcal{X}$ |
| $\|\cdot\|_1$ | $\ell_1$-norm |
| $\|\cdot\|_F$ | Frobenius norm |
| $\|\cdot\|_*$ | Nuclear norm |
| $\mathbf{U}_k$ | The temporal factor matrix for the $k^th$ subject |
| $\mathbf{S}_k$ | The weighting vector for the $k^th$ subject |
| $\mathbf{V}$ | The latent factor matrix for the features |
| $\mathbf{I}_k$ | The temporal length of the $k^th$ subject |
| $R$ | Number of target Rank |
| $*$ | Hadamard (element-wise) multiplication |
| $\odot$ | Khatri Rao product |
| $\circ$ | Outer product |
| $\langle\cdot,\cdot\rangle$ | Inner product |

$$\mathbf{X} \approx \sum_{r=1}^{R} u_r \circ v_r \circ w_r \tag{3.1}$$

$u_r \in \mathbb{R}^k$, $v_r \in \mathbb{R}^I$, and $w_r \in \mathbb{R}^J$ are column vectors. Stacking the column vectors $u_r, v_r, w_r$ into their respective matrices, $\mathbf{U} = [u_1, ...u_R]$, $\mathbf{V} = [v_1, ...v_R]$, $\mathbf{W} = [w_1, ...w_R]$, will yield factor matrices. The basic intuition of CP is to find R latent concepts to represent the original tensor. Yet CP decomposition can not deal with any irregular or incomparable mode.

PARAFAC2 is the state-of-the-art tensor factorization framework for effective handling the irregular tensor, i.e., tensors that do not align along one of its mode, and is formally defined as follows:

**Definition 5.** *(Classic PARAFAC2 model)*

$$\underset{\{\mathbf{U}_k\},\{\mathbf{S}_k\},\mathbf{V}}{\operatorname{argmin}} \sum_{k=1}^{K} \frac{1}{2}\|\mathbf{X}_k - \mathbf{U}_k\mathbf{S}_k\mathbf{V}^\top\|_F^2,$$

*s.t.* $\mathbf{U}_k = \mathbf{Q}_k\mathbf{H}, \mathbf{Q}_k^\top\mathbf{Q}_k = \mathbf{I}, \mathbf{S}_k$ *is diagonal, where* $\mathbf{Q}_k \in \mathbb{R}^{I_k \times R}$ *is orthogonal,* $\mathbf{I}_k \in \mathbb{R}^{R\times R}$ *is the identity matrix and* $R$ *is the target rank of the PARAFAC2 decomposition.*

where $k = 1, ..., K, \mathbf{U}_k \in \mathbb{R}^{I_k\times R}, \mathbf{S}_k \in \mathbb{R}^{R\times R}$ is diagonal and $\mathbf{V} \in \mathbb{R}^{J\times R}$. In order to enforce uniqueness, Harshman [22] imposed the constraint $\mathbf{U}^T\mathbf{U}_k = \Phi, \forall k$. This is equivalent to each $\mathbf{U}_k$ being decomposed as $\mathbf{U}_k = \mathbf{Q}_K\mathbf{H}$, where $\mathbf{Q}_k \in \mathbb{R}^{I_k \times R}$, $\mathbf{Q}_k^\top \mathbf{Q}_k = \mathbf{I}$, and $\mathbf{H} \in \mathbb{R}^{R\times R}$ Note that $\mathbf{Q}_k$ has orthonormal columns and $\mathbf{H}$ is invariant regardless of $k$.

For temporal EHR data, the factorization matrices have the following medical interpretations:

- $\mathbf{U}_k \in \mathbb{R}^{I_k \times R}$ captures temporal evolution for patient $k$: the $r$-th column of $\mathbf{U}_k$ indicates the evolution of the $r$-th phenotype for all $I_k$ clinical visits for patient $k$.
- $\mathbf{V} \in \mathbb{R}^{J\times R}$ contains the phenotypes. Each non-zero entry of $V$ indicates the membership of the corresponding $j$-th medical feature in the $r$-th phenotype.
- $\mathbf{S}_k \in \mathbb{R}^{R\times R}$ is a diagonal matrix with the importance membership of patient $k$ in each one of the $R$ phenotypes. It is often organized into $\mathbf{W} \in \mathbb{R}_+^{R\times K}$ with each row of $\mathbf{W}$ composed by the diagonal of $\mathbf{S}_k$, i.e. $\mathbf{W}(:, k) = \mathtt{diag}(\mathbf{S}_k)$.

The current PARAFAC2 models are designed for a single data type. For example, COPA [2] and REPAIR [60] are specifically designed for numeric data (e.g., loss function of mean squared error). Yin et al. [80] proposed a non-negative positive-unlabeled loss (PULoss) specific for binary data. However, in practice, data storage can differ across different EHR systems. What is stored as a numeric in one location might be a binary representation somewhere else. Thus, there is a lack of generalized PARAFAC2 model.

### 3.2.2 RNN

Deep Neural Networks have recently been successfully applied to many fields, e.g. character recognition [53], image classification or labeling [63], location prediction [43, 78], and text classification [55], etc. Recurrent Neural Networks (RNNs) are feedforward neural networks, which include edges between different timestamps. Such connectivity allows RNNs to capture the notion of time. RNN has the intrinsic capability to succinctly represent the non-linear temporal information in sequential data and time series. Here we briefly introduce the basic idea of RNN. Given a sequence $x = \{x_1, x_2, \ldots x_c\}$, the target can be another sequence $\{y_1, y_2, \ldots y_c\}$ or a single variable $y$. To capture the temporal dynamics of the sequence, RNNs use hidden variables $h = \{h_1, h_2, \ldots h_c\}$ to encode the information of the input $x$ and memorize the information of previous steps. Specifically, at each timestamp $t$, RNNs update the hidden variable $h_t$ based on previous state $h_{t-1}$ and current value $x_t$, as in Equation (3.2). Here $\sigma$ is an activation function, and $f_W, f_U$ are recurrent neural layers weighted by $W, U$. The prediction $y$ is usually calculated from the last hidden states $h_c$ ($c$ is the length of the sequence) through another neural layer, e.g., $y = \sigma(W \cdot h_c + b)$.

$$h_t = \sigma(f_W(h_{t-1}), f_U(x_t)) \tag{3.2}$$

As summarized in [56], deep learning models, including RNN, outperforms all other models for mortality and length-of-stay, especially when the raw clinical time series is utilized. Che et al. [11] introduce an RNN model to analyze multivariate clinical time series, and Choi et al. [14] applied an RNN model to predict clinical events. Our intuition is to leverage the modeling benefits of RNN by introducing RNN regularization along with the temporal evolution of PARAFAC2, which can ease of interpretability in the form of identifying useful subgroup characteristics and further increase predictability.

### 3.2.3 Proximal Mapping and Proximal Averaging

Proximal map [50] is a key building block for optimizing nonsmooth regularized objective functions, e.g., the $\|\cdot\|_1$ $\ell_1$-norm regularization function for inducing sparsity and $\|\cdot\|_*$ nuclear norm regularization for inducing low-rankness.

**Definition 6.** *(Proximal Map [50]) The proximal map for a close convex nonsmooth regularizer $h$ is defined as follows,*

$$\mathtt{Prox}_h^{\eta}(\mathbf{x}) := \arg\min_{\mathbf{y}} \frac{1}{2\eta}\|\mathbf{y}-\mathbf{x}\|_2^2 + h(\mathbf{y}), \tag{3.3}$$

*where $\eta$ is a constant parameter.*

The following presents two examples of the proximal map, which are utilized in our algorithm.

**Example 1.** *(Soft Thresholding) Let $h = \|\mathbf{x}\|_1$, then the proximal mapping has the closed-form solution, as follows*

$$\mathtt{Prox}_{\|\cdot\|_1}^{\eta}(\mathbf{x}_j) = \mathtt{sign}(\mathbf{x}_j) \cdot \max(0, |\mathbf{x}_j| - \eta), \tag{3.4}$$

*for $j = 1, ..., d$.*

**Example 2.** *(Singular Value Thresholding [8]) Let $h = \|\mathbf{X}\|_*$, then the proximal mapping has the closed-form solution, as follows*

$$\mathtt{Prox}_{\|\cdot\|_*}^{\eta}(\mathbf{X}) = \mathbf{U}\mathtt{Diag}(s_1, ..., s_d)\mathbf{V}^{\top}, \tag{3.5}$$

*where $[\mathbf{U}, \mathtt{Diag}(\boldsymbol{\sigma}), \mathbf{V}] = \mathtt{svd}(\mathbf{X})$ and $\mathbf{s} = \mathtt{Prox}_{\|\cdot\|_1}^{\eta}(\boldsymbol{\sigma})$.*

The proximal averaging technique is introduced by [3] to optimize in the presence of a composite regularizer that is in the form of the average of $J$ simpler nonsmooth regularizers, e.g., $\rho_1\|\cdot\|_1 + \rho_2\|\cdot\|_*$ with $\rho_1 + \rho_2 = 1$.

**Definition 7.** *(Proximal Average) Let $h$ be a composite regularizer in the form of $h(\mathbf{x}) := \sum_{i=1}^{J} \omega_i \cdot h_i(\mathbf{x})$, where $\sum \omega_{i=1}^{J} = 1$ and $h_i(\mathbf{x})$ are simple nonsmooth convex* *regularizers admitting the proximal map* $\mathtt{Prox}^{\eta}_{h_i}(\mathbf{x})$*. The proximal average of* $h$ *is the unique semi-continuous convex function* $\hat{h}(\mathbf{x})$ *such that the proximal map of* $\hat{h}(\mathbf{x})$ *is*

$$\mathtt{Prox}^{\eta}_{\hat{h}} = \sum_{i=1}^{J} \omega_i \mathtt{Prox}^{\eta}_{h_i}(\mathbf{x}), \tag{3.6}$$

*where each* $\mathtt{Prox}^{\eta}_{h_i}(\mathbf{x})$ *admits closed-from computation.*

## 3.3 Related Work

SPARTan [51] scaled PARAFAC2 to large and sparse irregular tensors by introducing a sparse MTTKRP (abbreviated for Matricized-Tensor-Times-Khatri-Rao-Product) module to reduce the per-iteration cost. Following its efficiency improvement, COPA [2] introduced various constraints/regularizations to improve the interpretability of the factor matrices. For example, COPA proposed the M-spline constraint [57] to $\mathbf{U}_k$ to capture the temporal smoothness, non-negative constraint to $\mathbf{S}_k$ to avoid negative weights, and $\ell_1$-norm regularization of $\mathbf{V}$ to induce sparse phenotype definitions. Despite their improvements in computational efficiency and output interpretability, COPA and SPARTan did not explicitly address the problem of missing entries in the input tensor, which severely limits them from more robust clinical usage.

REPAIR [60] and LogPar [80] address missing entries in PARAFAC2. REPAIR [60] added low-rank regularization to PARAFAC2 to address missing entries. Inspired by the robust low-rank tensor minimization (RLTM), the state-of-the-art mechanism for dealing with missing and error entries, REPAIR separated the corrupted input tensor into a clean, completed tensor and an error tensor. Since the clean tensor is often low-rank, REPAIR added low-rank regularization (i.e., nuclear norm) on the clean tensor and sparsity regularization ($\ell_1$-norm regularization) on the error tensor. It then proposed a novel two-phase optimization alternative direction method of multipliers (ADMM) approach to solve the low-rank regularized PARAFAC2 model.

LogPar considered binary data with a one-class missing value scenario. LogPar modeled the binary irregular tensor with the Bernoulli distribution parameterized by an underlying real-valued tensor. Then they approximated the underlying tensor with a positive-unlabeled learning loss function to account for the missing values. However, both models are suitable for one type of data and cannot be easily adapted for composite regularization of the factor matrices.

CNTF treated each patient's data as an individual tensor, used CP decomposition to find the factor matrices, and used RNN to regularize the latent factor evolution [79]. The RNN model was used to model the non-linear temporal dependency in patient progressions and can also integrate higher-order information. However, CNTF assumes interactions among modalities which may not be always the case as demonstrated by the empirical results in[2] and [56].

## 3.4 Proposed Method

In this section, we present the REBAR model and its optimization, which has the following appealing features that distinguish it from previous PARAFAC2 methods. (1) It generalizes the loss functions from the sole choice of the least square norm to any smooth loss function, which better suits input tensors with various data types. (2) It accommodates a wide selection of regularization, including statistical learning-based e.g., $l_1$ norm and nuclear norm, deep learning-based e.g., RNN regularization, and composite, to better capture the intrinsic nature of the irregular temporal EHR data. (3) It introduces new optimization geared to fully exploit the parallel computing capability of modern GPUs to boost efficiency.

### 3.4.1 Problem Formulation

We formalize the objective function for the REBAR model in Definition 8. The PARAFAC2 loss for $\mathcal{X}$ ensures the reconstructed tensor closely approximates the original tensor, the low-rankness for $\mathcal{X}$ enforces the underlying complete tensor is separated from missing values, the RNN loss can better capture the temporal patterns in the data, and an approximate uniqueness constraint ensures tensor factorization uniqueness. For EHR phenotype discovery, various constraints can be imposed on the factorization matrices to yield meaningful and high-interpretability phenotypes. The REBAR model accommodates such interpretability-purposed constraints in eq.(3.7) including: non-negativity for $c_1(\mathbf{S}_k)$, sparsity for $c_2(\mathbf{V})$. We explain each of the loss components and constraints in details below.

**Definition 8.** *(REBAR objective function)*

$$\begin{aligned}
&\underset{\mathbf{Q}_k,\mathbf{H},\mathbf{S}_k,\mathbf{V}}{\operatorname{argmin}} \sum_{k=1}^{K} \sum_{(i,j)\in\Omega} \overbrace{L(\mathbf{X}_{ijk}, \{\mathbf{U}_k\mathbf{S}_k\mathbf{V}^\top\}_{ijk})}^{\textit{PARAFAC2 loss for } \mathcal{X}} + \overbrace{\sum_{k=1}^{K} \mathtt{RNN}(\mathbf{U}_k)}^{\textit{RNN loss}} \\
&+ \overbrace{\rho_1\|\mathbf{H}\|_* + \rho_2\|\mathbf{V}\|_* + \rho_3\|\mathbf{W}\|_*}^{\textit{low-rankness for } \mathcal{X}} \\
&+ \overbrace{\varrho_1 \sum_{k=1}^{K} \left(\|\mathbf{U}_k - \mathbf{Q}_k\mathbf{H}\|_F^2 + \varrho_2\|\mathbf{Q}_k^\top\mathbf{Q}_k - \mathbf{I}\|_F^2\right)}^{\textit{approximate uniqueness constraint}} \\
&+ \sum_{k=1}^{K} c_1(\mathbf{S}_k) + c_2\|\mathbf{V}\|_1 + \overbrace{\varrho_3\|\mathbf{W} - \overline{\mathbf{W}}\|_F^2}^{\textit{auxiliary } \overline{W} \textit{ for disentangling constraints}},
\end{aligned} \tag{3.7}$$

$$\textit{s.t. for } k = 1, ..., K, \overbrace{\mathbf{S}_k = \mathtt{diag}(\overline{W}(k,:)), \mathbf{S}_k \textit{ is diagonal}}^{\textit{relation between } \mathbf{S},\mathbf{W}} \tag{3.8}$$

*where* $\mathbf{H}, \{\mathbf{S}_k\}, \mathbf{I} \in \mathbb{R}^{R\times R}$*,* $\mathbf{Q}_k \in \mathbb{R}^{I_k\times R}$*,* $\Omega$ *denotes the index of the non-missing entries,* $c_1$ *is the nonnegativity constraint, and* $c_2\|\mathbf{V}\|_1$ *is the sparsity penalty.*

**Generalized PARAFAC2.** To accommodate different data types, we extend PARAFAC2

Table 3.2: Examples of tensor data types and loss functions

| **Data Type** | **Loss Function** |
|---|---|
| Binary | Positive Unlabeled loss [80] |
| Count | Poisson Loss [29] |
| Numerical | Least Square Loss |
| Strictly positive data | Rayleigh Loss [29] |

to a more generalized form by introducing a general loss function $\sum_{k=1}^{K}\sum_{(i,j)\in\Omega} L(\mathbf{X}_{ijk}, \{\mathbf{U}_k\mathbf{S}_k\mathbf{V}^\top\}_{ijk})$ with any smooth loss function $L$, rather than limiting it to be the least squared loss in Definition 5. By being capable of switching between various loss functions, our generalized PARAFAC2 can better suit different input data types. Table 3.2 lists several example data types and their corresponding loss function.

**RNN regularization.** In order to model the temporal dependency in phenotype progression, we regard each patient's temporal evolution matrix $\mathbf{U}_k \in \mathbb{R}^{I_k \times R}$ as a multivariate time series with each variable describing the progression of the corresponding phenotype for patient $k$. For each timestamp, we use the RNN model to predict $\mathbf{U}_k^t$ given the previous stage $\mathbf{U}_k^{t-1}$, and minimize the Mean Square Error (MSE) between the real and predicted value. The RNN regularization term is written as:

$$\mathtt{RNN}(\mathbf{U}_k) = \frac{1}{\mathbf{I}_k}\sum_{t=2}^{\mathbf{I}_k} \|g(\mathbf{U}_k^{t-1}) - \mathbf{U}_k^t\|_2^2, \tag{3.9}$$

where $g(\mathbf{U}_k^{t-1})$ is the prediction output given by the RNN model.

A key feature of our model is that the RNN regularization is jointly optimized with the PARAFAC2 tensor factorization to enforce the patient temporal evolution matrix is consistent with the regularity captured by RNN as well as recovering the temporal tensor.

**Sparsity on V.** $\mathbf{V}$ matrix indicates the importance membership of each of the medical features in the corresponding phenotypes. Introducing sparsity constraint on factor matrix $\mathbf{V}$ will provide coherent and sparse EHR phenotypes, which can improve interpretability and increase robustness. $l_0$ and $l_1$ norms are two popular regularization techniques. The $l_0$ regularization norm, also known as hard thresholding, will cap the number of non-zero values in a matrix. The $l_1$ regularization norm, also known as soft thresholding, will shrink matrix values towards zero. Hard thresholding is a non-convex optimization problem, and soft thresholding is a convex relaxation of the $l_0$ norm. We choose $l_1$ norm to migrate it into SGD optimization framework.

**Non-negativity on S.** The diagonal matrix values in $\mathbf{S}$ indicate the importance membership of patient $k$ in each one of the $R$ phenotypes (how much a patient $k$ is associated with or exhibit a particular phenotype). Since we only care about non-negative memberships, we zero out the negative values in $\mathbf{S}$, which significantly improves the interpretability.

**Low-rankness for $\boldsymbol{\mathcal{X}}$.** The robust low-rank tensor minimization (RLTM) can successfully recover the original tensor with missing values by imposing a low-rank regularization function on the original tensor. It seperate underlying completed tensor from the corrupted tensor. In practice, the completed part is often low-rank. As studied in REPAIR [60], adding low-rankness on $\boldsymbol{\mathcal{X}}$ via nuclear norm constraints on the factor matrices $\mathbf{H}, \mathbf{V}, \mathbf{W}$ can improve robustness to missing entries.

**Approximate uniqueness constraint.** Similar to [80], we relax the uniqueness constraint of $\mathbf{Q}_k^\top \mathbf{Q}_k = \mathbf{I}$ and introduce the regularization of $\|\mathbf{Q}_k^\top \mathbf{Q}_k - \mathbf{I}\|_F^2$, which enables us to use stochastic gradient descent to optimize $\mathbf{Q}_k$. The relaxation and the SGD optimization facilitate the adoption of mainstream deep learning platforms like Pytorch, thus making full use of the parallel computation feature of modern GPUs.

**Disentangling constraint.** Each row of $\mathbf{W}$ is composed by the diagonal of $\mathbf{S}_k$. Since we already have low-rankness constraint on $\mathbf{W}$ and non-negativity constraint on $\mathbf{S}$, we add a auxiliary parameter $\overline{\mathbf{W}}$, where for different patient $k$, $\mathbf{S}_k = \mathtt{diag}(\overline{W}(k,:))$ to separate $\mathbf{S}$ and $\mathbf{W}$.

### 3.4.2 Optimization

To solve the optimization problem in Eq. (3.7), REBAR follows an alternative optimization strategy where we optimize one variable individually with all other variables fixed. The variables to be optimized can be categorized into three groups according to whether the subproblem is purely smooth, or proximal mapping-based smooth, or multiple nonsmooth subproblems. In particular, when dealing with multiple nonsmooth functions regularized subproblems, we introduce the proximal average-based technique as a replacement for the AO-ADMM approach adopted in the previous PARAFAC2 works [2, 60]. As a result, REBAR can take advantage of the parallel computing feature of GPU to boost efficiency. Moreover, the implementation is significantly simpler. In the following, we omit the iteration number for brevity in notation.

**Pure Smooth Subproblems Updates**

For the pure smooth subproblems, we use stochastic gradient descent to update the variables, which include the following three parts:

**Update of $\mathbf{U}_k$.** The subproblem of $\mathbf{U}_k$ takes the form as follows

$$\begin{aligned} &\arg\min_{\mathbf{U}_k} \sum_{(i,j)\in\Omega} L(\mathbf{X}_{ijk}, \{\mathbf{U}_k\mathbf{S}_k\mathbf{V}^\top\}_{ijk}) \\ &+ \varrho_1\|\mathbf{U}_k - \mathbf{Q}_k\mathbf{H}\|_F^2 + \mathtt{RNN}(\mathbf{U}_k) \end{aligned} \tag{3.10}$$

**Update of $\mathbf{Q}_k$.** The subproblem of $\mathbf{Q}_k$ takes the form as follows

$$\arg\min_{\mathbf{Q}_k} \varrho_1\|\mathbf{U}_k - \mathbf{Q}_k\mathbf{H}\|_F^2 + \varrho_2\|\mathbf{Q}_k^\top\mathbf{Q}_k - \mathbf{I}\|_F^2 \tag{3.11}$$

**Algorithm 2** Optimization Framework for REBAR

```
Input: Input tensor $\mathcal{X}$; Model parameters $\rho_1$-$\rho_3$, $\varrho_1$-$\varrho_3$; Interpretability constraint
    types $c_1, c_2$ and RNN sub-model; Initial rank estimation $R$.
while Not reach convergence criteria do
    Update $\{\mathbf{U}_k\}$ using eq.(3.10) by SGD;
    Update $\{\mathbf{Q}_k\}$ using eq.(3.11) by SGD;
    Update RNN using eq.(3.12) by SGD;
    Update $\mathbf{W}$ using eq.(3.13) by Proximal/Projected SGD;
    Update $\mathbf{S}_k$ using eq.(3.15) by Proximal/Projected SGD;
    Update $\mathbf{H}$ using eq.(3.17) by Proximal/Projected SGD;
    Update $\mathbf{V}$ using eq.(3.19) by Proximal averaging SGD;
end while
Output: Phenotype factor matrices $\{\mathbf{U}_k\} = \{\mathbf{Q}_k\mathbf{H}\}, \{\mathbf{S}_k\}, \mathbf{V}$.
```

**Update of `RNN`.** The subproblem of RNN model parameters takes the form as follows

$$\arg\min_{\texttt{RNN}} \sum_{k=1}^{K} \texttt{RNN}(\mathbf{U}_k). \tag{3.12}$$

### Proximal Mapping-base Smooth Subproblems Updates

For the nonsmooth subproblems, we propose a proximal mapping-based update, which include the following three parts.

**Update of $\mathbf{W}$.** The subproblem of $\mathbf{W}$ takes the form as follows

$$\arg\min_{\mathbf{W}} \sum_{k=1}^{K} \|\mathbf{W} - \overline{\mathbf{W}}\}\|_F^2 + \rho_3 \|\mathbf{W}\|_*. \tag{3.13}$$

We use the following closed-form update

$$\mathbf{W} = \texttt{Prox}^{\rho_3}_{\|\cdot\|_*}(\overline{\mathbf{W}}]). \tag{3.14}$$

**Update of $\mathbf{S}_k$ and $\overline{\mathbf{W}}$.** The subproblem of $\mathbf{S}_k$ takes the form as follows

$$\arg\min_{\mathbf{S}_k} \sum_{(i,j)\in\Omega} L(\mathbf{X}_{ijk}, \{\mathbf{U}_k\mathbf{S}_k\mathbf{V}^\top\}_{ijk}) + c_1(\mathbf{S}_k). \tag{3.15}$$

We use projected stochastic gradient descent to update $\mathbf{S}_k$, where each step takes the following form

$$\mathbf{S}_k = \max(0, \mathbf{V} - \lambda \mathbf{G}[\mathbf{S}_k]), \tag{3.16}$$

where $G[\mathbf{V}]$ denotes the stochastic gradient of the smooth part $\sum_{(i,j)\in\Omega} L(\mathbf{X}_{ijk}, \{\mathbf{U}_k\mathbf{S}_k\mathbf{V}^\top\}_{ijk})$ with respect to $\mathbf{S}_k$. Finally, we let $\mathtt{diag}(\overline{W}(k,:)) = \mathbf{S}_k$.

**Update of H.** The subproblem of $\mathbf{H}$ takes the form as follows

$$\arg\min_{\mathbf{H}} \sum_{k=1}^{K} \|\mathbf{U}_k - \mathbf{Q}_k\mathbf{H}\}\|_F^2 + \rho_1 \|\mathbf{H}\|_*. \tag{3.17}$$

We use proximal stochastic gradient descent to update $\mathbf{H}$, where each step takes the following form

$$\mathbf{H} = \mathtt{Prox}_{\|\cdot\|_*}^{\frac{\rho_1}{\lambda}}(\mathbf{V} - \lambda \mathbf{G}[\mathbf{H}]), \tag{3.18}$$

where $G[\mathbf{H}]$ denotes the stochastic gradient of the smooth part $\sum_{k=1}^{K} \|\mathbf{U}_k - \mathbf{Q}_k\mathbf{H}\}\|_F^2$ with respect to $\mathbf{H}$.

### Multiple Nonsmooth Subproblems Updates

For multiple nonsmooth functions regularized subproblem, We propose a proximal averaging-based update.

**Update of V.** The subproblem of $\mathbf{V}$ takes the form as follows

$$\arg\min_{\mathbf{V}} \sum_{k=1}^{K} \sum_{(i,j)\in\Omega} L(\mathbf{X}_{ijk}, \{\mathbf{U}_k\mathbf{S}_k\mathbf{V}^\top\}_{ijk}) + \rho_2 \|\mathbf{V}\|_* + c_2 \|\mathbf{V}\|_1. \tag{3.19}$$

We use proximal average stochastic gradient descent [81] to update $\mathbf{V}$, where each step takes the following form

$$\begin{aligned} \mathbf{V} = &\frac{\rho_2}{\rho_2 + c_2}\texttt{Prox}_{\|\cdot\|_*}^{\frac{\rho_2}{\lambda}}(\mathbf{V} - \lambda\mathbf{G}[\mathbf{V}]) \\ &+ \frac{c_2}{\rho_2 + c_2}\texttt{Prox}_{\|\cdot\|_1}^{\frac{c_2}{\lambda}}(\mathbf{V} - \lambda\mathbf{G}[\mathbf{V}]), \end{aligned} \tag{3.20}$$

where $G[\mathbf{V}]$ denotes the stochastic gradient of the smooth part $\sum_{k=1}^{K}\sum_{(i,j)\in\Omega} L(\mathbf{X}_{ijk}, \{\mathbf{U}_k\mathbf{S}_k\mathbf{V}^\top\}_{ijk})$ with respect to $\mathbf{V}$.

**The complete algorithm.** The optimization procedure is summarized in Algorithm 2.

## 3.5 Experimental Evaluation

### 3.5.1 Dataset

We use three datasets to test REBAR.

**MIMIC-III [1] [31]:** The intensive care unit (ICU) dataset is collected between 2001 and 2012. We keep records of patients with at least 10 hospital visits and construct a three-mode tensor. We select 405 medical NDC codes and 202 diagnosis codes that have the highest frequency as in [36]. The resulting number of patients is 5133 with 607 features (medication codes) and the maximum number of observations for a patient is 172. 21% patient mortality flag is positive.

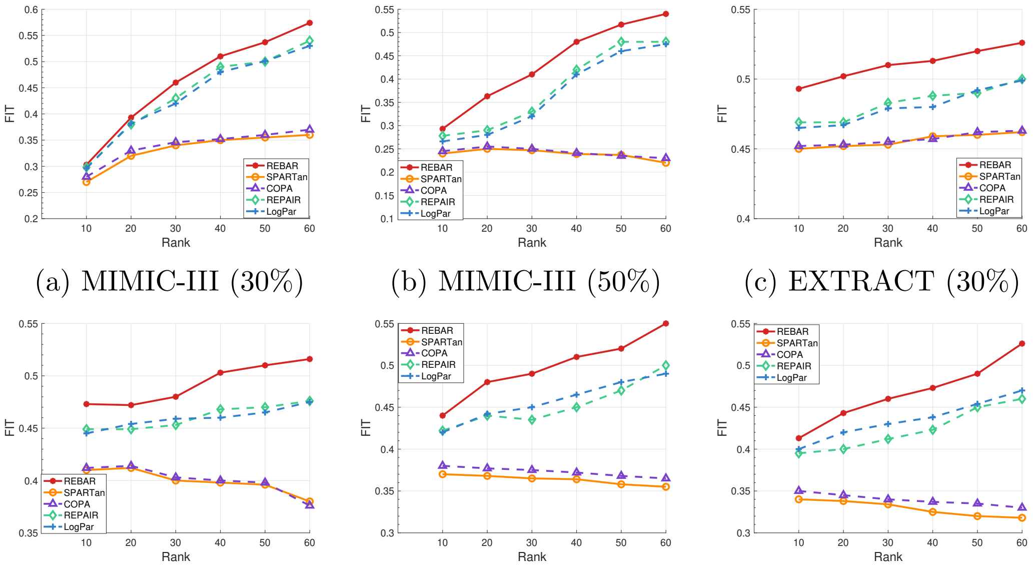
**MIMIC-EXTRACT [2] [72]:** MIMIC-Extract, an open-source pipeline for transforming raw electronic health record (EHR) data in MIMIC-III into data frames that are directly usable in common machine learning pipelines. We use the vitals labs mean table, which contains 34,472 patients with 104 features (Vital lab codes). The maximum number of observations for a patient is 240. We further normalize the data to [0,1]. 10% patient mortality flag is positive.

[1] https://mimic.physionet.org/

[2] https://github.com/MLforHealth/MIMIC_Extract/

Table 3.3: Feature discretion ranges for the 6 features in PhysioNet sepsis dataset

| **Feature** | **Low Value** | **Normal Value** | **High Value** |
|---|---|---|---|
| Heart rate (HR) | HR< 60 beats/minute | 60 beats/minute $<$ HR $<$ 90 beats/minute | HR $>$ 90 beats/ minute |
| Temperature (Temp) | Temp$< 36.0^{\circ}C$ | $36.0C$ <Temp$< 38.0^{\circ}C$ | Temp$> 38.0^{\circ}C$ |
| Respiratory rate (Resp) | Resp $< 6 \times 10^9$/I | $6 \times 10^9$/I $<$ Resp $< 20 \times 10^9$/I | Resp $> 20 \times 10^9$/I |
| Mean arterial pressure (MAP) | MAP $<$ 65 mmHg | 65 mmHg <MAP $<$ 100 mmHg | MAP $>$ 100 mmHg |
| Oxygen saturation (O2Sat) | O2Sat $<$ 95% | O2Sat $>$ 95 % | - |
| Systolic blood pressure (SBP) | SBP $<$ 120 mmHg | 120 mmHg <SBP $<$ 140 mmHg | SBP $>$ 140 mmHg |



(a) MIMIC-III (30%)

(b) MIMIC-III (50%)

(c) EXTRACT (30%)

(d) EXTRACT (50%)

(e) Sepsis (30%)

(f) Sepsis (50%)

Figure 3.2: Fit on the different datasets with the missing ratio in parenthesis (missing percentage %).

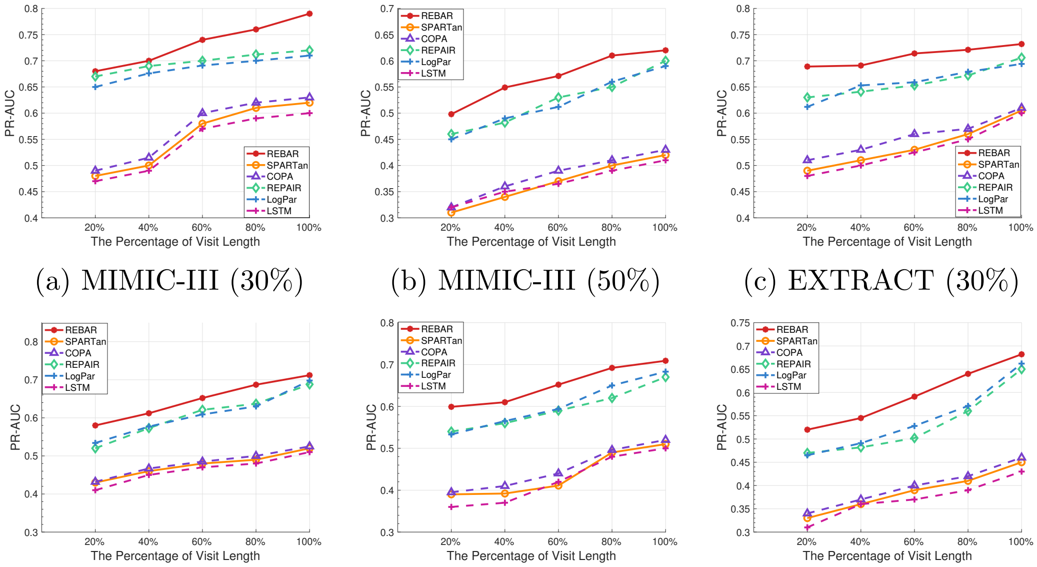
**PhysioNet Sepsis Dataset** [3]**[61]:** PhysioNet 2019 Early Prediction of Sepsis from Clinical Data Challenge is an open-access dataset. It contains 20,336 patients with 40 time-dependent variables such as HR, O2Sat, Temp, etc. Since most of the features are extremely sparse, we select 6 dense features and then discretize the variables using criteria in [17]. The detailed values can be found in table 3.3. The maximum number of observations for a patient is 336. 17% patient sepsis flag is positive.

[3] `https://archive.physionet.org/users/shared/challenge-2019/`

### Methods for Comparison

We compare REBAR with four baseline methods: SPARTan and COPA are two state-of-the-art irregular tensor factorization methods with different temporal regularizations, REPAIR and LogPar are two state-of-the-art robust irregular tensor factorization methods to handle missing entries.

- **SPARTan [51] - scalable PARAFAC2**: A method for fitting PARAFAC2 on large and sparse data. It does not include temporal regularization, and it also does not address missing data.
- **COPA [2] - scalable PARAFAC2 with additional regularizations**: An irregular tensor factorization method that introduces various constraints/regularizations to improve the interpretability of the factor matrices. Temporal smoothness is enforced by modeling $Uk$ to be the linear combination of a set of temporal basis functions generated by the M-spline. This method requires a pre-computation of the spline functions.
- **REPAIR[60] - robust method for irregular tensors**: A recently-proposed robust method adding effective low-rank regularization to address missing and error entries. It uses the same temporal regularization as COPA.
- **LogPar [80] - robust method for matrix completion**: A method addressing missing data in binary irregular tensors. LogPar uses a positive-unlabeled learning loss function to account for the missing values. It uses an exponential term to adaptively weigh the regularization based on the time gap between two visits with the intuition that steps closer in time generally should be closer in the latent space.



(a) MIMIC-III (30%) (b) MIMIC-III (50%) (c) EXTRACT (30%)

(d) EXTRACT (50%) (e) Sepsis (30%) (f) Sepsis (50%)

Figure 3.3: PR-AUC for downstream prediction tasks on the different datasets with the missing ratio in parenthesis (missing percentage %). For MIMIC-III and MIMIC-Extract, the classification task is mortality prediction while Sepsis is sepsis prediction.

### 3.5.2 Implementation Details

MIMIC-III contains count data, so we use Poisson loss [29], and MIMIC-EXTRACT contains numerical data, so we select mean squared error loss. PhysioNet Sepsis data is binary after the discretization, so we use the non-negative positive-unlabeled loss [80]. To determine the best RNN models, we tested LSTM, GRU, and vanilla RNN, and found a single-layer GRU network [13] with 100 hidden units gives the best result. Vanilla RNN failed to capture long-term temporal dependencies and suffers from the gradient vanishing problem. Although LSTM is the most complex RNN model, it tends to over-fit on small datasets.

### 3.5.3 Experiment Result

#### Tensor Factorization Robustness

In order to test the robustness of REBAR model against missing entries, we randomly add missing values into the two datasets. The original uncorrupted tensor, denoted as $\{\mathbf{G}_k\}$, serves as the ground truth. We adopt the $FIT \in (-\infty, 1]$ score [6] as the quality measure (the higher the better):

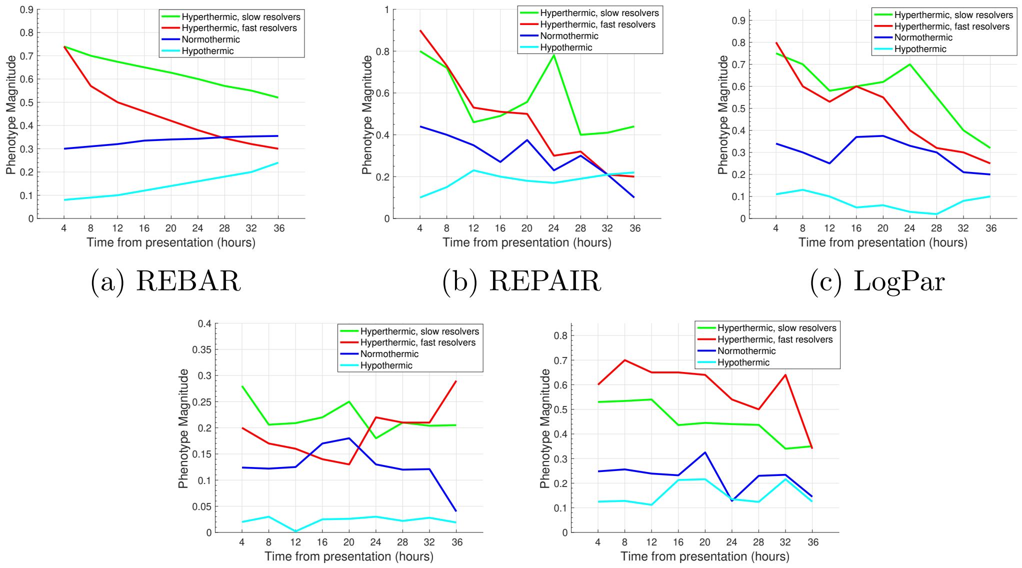
$$FIT = 1 - \frac{\sum_{k=1}^{K} \|\mathbf{G}_k - \mathbf{U}_k \mathbf{S}_k \mathbf{V}^T\|^2}{\sum_{k=1}^{K} \|\mathbf{G}_k\|^2}. \tag{3.21}$$

In the following experiment, we run each set for 5 different random initializations and report the average $FIT$. We set the missing ratio to 30% and 50%, then test model completion performance under different target ranks, $R$, from 10 to 60.

As Figure 3.2 shows, REBAR outperforms all the other baseline methods on all datasets under both missing ratio settings. In particular, REBAR achieves a FIT score of 0.574 and 0.524 on MIMIC-III when the missing ratio equals 30% and 50% respectively, a 10% relative improvement when compared to the best baseline model REPAIR. REBAR shows the same outstanding performance with 7% and 10% improvement to the best baseline model for the MIMIC-EXTRACT (REPAIR) and Sepsis (LogPar) datasets respectively. LogPar and REPAIR perform better than COPA and SPARTan, because COPA and SPARTan has no regularizations (e.g. by soft-thresholding the singular values) to handle missing entries. COPA performs slightly better than SPARTan, because of the linear temporal smooth regularization imposed in COPA. LogPar outperforms REPAIR on Sepsis dataset, but is left behind REPAIR on MIMIC-III and MIMIC-EXTRACT. This demonstrates the importance of appropriately tuning the loss function as Sepsis is a binary dataset, and the non-negative positive-unlabeled loss in LogPar is more suitable for such data.

It is also noteworthy to discuss the FIT score trend as the rank varies. REBAR, REPAIR, and LogPar show increasing FIT score as rank increases, however, COPA and SPARTan shows decreasing or flat FIT score as rank increases. This demonstrates the benefit of the low-rank regularization that can be less reliance on the user specification rank in REBAR, REPAIR, and LogPar. As the rank increases, the low-rank regularization function can iteratively decrease the target rank, and find the optimal one in the latent space. COPA and SPARTan do not have low-rank regularization, so a lower target rank is better for these methods to recover from missing values.

Comparing different missing ratios, REBAR demonstrates superior robustness when the missing ratio increases to 50% from 30%. It only has 6%, 2% and 4% FIT score decreases on MIMIC-III, MIMIC-EXTRACT, and Sepsis datasets, respectively. In contrast, the best baseline model, i.e., REPAIR for MIMIC-III, REPAIR for MIMIC-EXTRACT, and LogPar on Sepsis dataset, suffer 12%, 7% and 8% completion performance drop, respectively. Though REPAIR and LogPar have low-rank regularization, they fail to handle non-linear temporal information, which is the main reason for these large performance drops.



(a) REBAR

(b) REPAIR

(c) LogPar

(d) COPA

(e) SPARTan

Figure 3.4: Dynamic subphenotype trajectories using different PARAFAC2 models

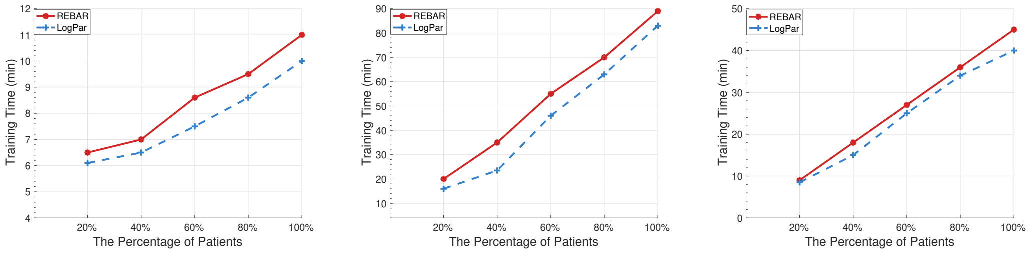
Table 3.4: Temperature measurements of temperature trajectory groups

| | **All Patients** | **Hyperthermic, Slow Resolvers** | **Hyperthermic, Fast Resolvers** | **Normothermic** | **Hypothermic** |
|---|---|---|---|---|---|
| Number of Patients | 298 | 46 | 71 | 101 | 80 |
| Mean temperature,$^{\circ}C$ | 36.6 | 37.4 | 36.8 | 36.6 | 36.0 |
| Min temperature,$^{\circ}C$ | 35.1 | 36.1 | 35.8 | 35.9 | 35.1 |
| Max temperature,$^{\circ}C$ | 39 | 39 | 38.5 | 37.4 | 36.7 |

### Downstream Prediction Analysis

We further evaluate the derived phenotypes' predictability power via a downstream prediction task. We predict in-hospital mortality on MIMIC-III and MIMIC-EXTRACT datasets using the in-hospital death flag, and predict if a patient will have sepsis on PhysioNet Sepsis dataset. We split the data with a proportion of 8:2 as training and test sets and use PR-AUC (Area Under the Precision-Recall Curve) score to evaluate the predictive power. A logistic regression model is trained on the patient importance membership matrix, $\mathbf{S}_k$. Besides tensor models, we also include a non-tensor prediction model. We directly train a long short-term memory (LSTM) model using irregular tensor. LSTM is a variant of RNN that mitigate the gradient vanishing problem in traditional RNNs. Its memory cells contain three types of non-linear gates, namely input gate, output gate and forget gate, which can regulate the flow of signals into and out of the cell and learn long-term dependencies. The input is a the irregular tensor which contains $k$ different patient, and each patient information $X_k$ consists with $I_K$ visits and $J$ medical features. The output is the prediction label for different patient.

In the real-world scenario, predicting in-hospital mortality or if the patient will have a certain kind of disease using early-stage data is a crucial problem. It makes no sense to predict in-hospital mortality using full in-hospital data, as it can not give any useful information to help health care workers provide accurate and precise treatment for patients to avoid death. We vary the visit length percentage to mimic this real-



(a) MIMIC-III (b) MIMIC-EXTRACT (c) Sepsis

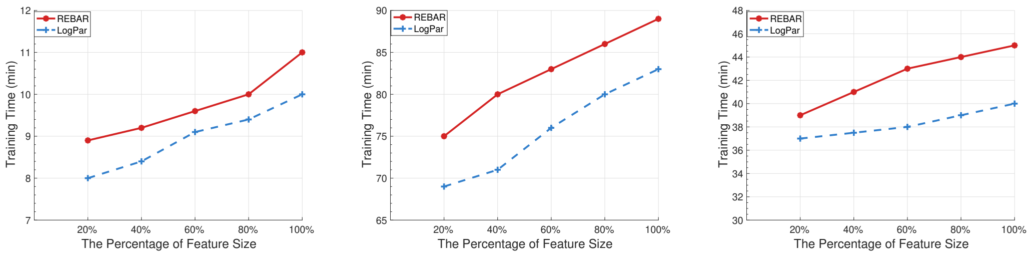
Figure 3.5: Training time on the different datasets varying patients size with the missing ratio equal to 30 %.

world setting. As shown in Figure 3.3, REBAR outperforms the other methods. When the visit length ratio is 60%, REBAR outperforms the best baseline methods by 8%, 9% and 16% in Figure 3.3b, 3.3d, and 3.3f respectively. This demonstrates strong predictability even with missing values. Because of the RNN regularization, REBAR can learn the non-linear temporal dependence and also use the whole time trajectory to improve missing value recovery. COPA and SPARTan are left behind REBAR, REPAIR, and LogPar because of the lack of low-rank regularization to handle missing values. Since COPA has an additional temporal smoothness constraint compared to SPARTan, it performs slightly better than SPARTan. Non-tensor based LSTM model performs worst on three datasets because it lacks tensor factorization to filter out noises.

We also vary the missing ratio on the different datasets. As the missing ratio increases from 30% to 50%, REBAR's PR-AUC drops 6%, 3%, and 4% on MIMIC-III, MIMIC-EXTRACT, and Sepsis dataset, respectively. This is because the average visit length $I_k$ is the smallest in MIMIC-III, so GRU's expressive power is limited.

### Interpretation of the Dynamic Subphenotypes Trajectory

We demonstrate the effectiveness of REBAR to find dynamic phenotype trajectories with the missing ratio of 30% using MIMIC-EXTRACT. We select the patients with the number of observations equal to 36 for visualization purposes. The rank is set to



(a) MIMIC-III (b) MIMIC-EXTRACT (c) Sepsis

Figure 3.6: Training time on the different datasets varying feature size with the missing ratio equal to 30 %.

50. The factor matrix **V** reflects the membership of the corresponding $j$-th medical feature in $r$-th phenotype. We select the top 4 largest weighted phenotypes for the temperature feature and plot the temporal trajectory of selected phenotypes using the **U** matrix by taking the average of the selected patients' phenotype magnitude value in Figure 3.4. As shown in Figure 3.4a, the high temperature phenotypes (Hyperthermic phenotype slow resolvers and Hyperthermic phenotype fast resolvers) are higher than the normal temperature phenotype (Normothermic phenotype) and low temperature phenotype (Hypothermic phenotype), and exhibit a decreasing trend as time increases. However, there is no clear trend for Hyperthermic, Normothermic, and Hypothermic phenotypes in COPA and SPARTan because they overfit to the zero entries. Although REPAIR and LogPar also display similar decreasing trends, there are sudden spikes in the temporal pattern which can hinder the interpretability and clinical meaningfulness. This is because REPAIR and LogPar's temporal smoothness fail to capture long-term temporal information. We compare the four phenotypes' weight in the **U** matrix for each patient and find corresponding patient groups of REBAR. The temperature measurements for the patient groups are calculated. REBAR can correctly find temperature subgroups using extracted phenotypes as shown in Table 3.4. As verified by a critical care clinician, our model provides better and accurate treatment for different patient groups. The other methods fail to find meaningful subgroups, so they are not presented in this paper. As time increases in Figure 3.4,

temporal trajectory shows non-linear dependency, REBAR can successfully capture the non-linear information, filter out the noise, and shows a more clinical meaningful temporal trajectory.

### Scalability Analysis

Adding deep learning based methods on PARAFAC2 framework can raise some concerns on computational time and scalability issues on large datasets. We measure the training time of REBAR on different data size and different feature size and compare to the state-of-the-art robust PARAFAC2 method LogPar. We use two Titan RTX GPUs, each GPU has 24 GB of RAM, and rain 50 epochs of both methods with a missing ratio of 30%

In Figure 3.5 and 3.6, we show the total training time. LogPar shows linear scalability as the number of patients and features grows. Although REBAR adds deep learning based constraint, it still has linear scalability as LogPar. Although deep learning regularization adds some additional training time, but not significantly more, maximum of the added time is 10 minutes.

MIMIC-III, MIMIC-EXTRACT and Sepsis has increased 57%, 154% and 150% in training time as patients size grows from 40% to 100%, whereas, MIMIC-III, MIMIC-EXTRACT and Sepsis has increased 20%, 11% and 9.7% in training time as feature size grows from 40% to 100%, which shows patients size is the key factor for scaling up REBAR.

### Phenotype Presentation

Finally, we present the phenotype discovered by REBAR on MIMIC-EXTRACT and MIMIC-III in Table 3.5 and 3.6 with the missing ratio set to 30%. We set rank to be 3, and sort each column of $\mathbf{V}$ matrix to get the top weighted phenotype features. It is important to note there is no post-processing in these extracted phenotypes. The

Table 3.5: MIMIC-EXTRACT phenotypes discovered by REBAR.

| | |
|---|---|
| **Abnormalities in Vital Signs** | |
| Glascow Coma Scale Total | Oxygen Saturation |
| Temperature | Systolic Blood Pressure |
| **Metabolic Syndrome** | |
| Glucose | Systolic Blood Pressure |
| Weight | Mean Blood Pressure |
| **Abnormalities in Blood Counts** | |
| Sodium | Mean Corpuscular Hemoglobin Concentration |
| Mean Corpuscular Volume | Mean Corpuscular Hemoglobin |

presented phenotypes have been endorsed by the critical care expert. Moreover, the expert provided the labels to reflect the associated medical concept.

Table 3.6: MIMIC-III phenotypes discovered by REBAR. The red color corresponds to diagnosis and blue color corresponds to medications.

| | |
|---|---|
| **Cardiovascular Disturbances** | |
| Cardiovascular syph NEC | Coronary atherosclerosis of autologous vein bypass graft |
| Atrial fibrillation | Coronary atherosclerosis of native coronary artery |
| Metoprolol | Metoprolol Tartrate |
| Labetalol | Acetaminophen |
| Propofol | Furosemide |
| **Electrolyte Disturbances** | |
| Functional diarrhea | Electrolyte and fluid disorders not elsewhere classified |
| Vomiting alone | Iron deficiency anemia secondary to blood loss (chronic) |
| Potassium Chloride | Magnesium Sulfate |
| Neutra-Phos | Hydromorphone |
| Aspirin EC | Atorvastatin |
| **Gastrointestinal Disturbances** | |
| Gastrointestinal vessel anomaly | Hemorrhage of gastrointestinal tract, unspecified |
| Vomiting alone | Malignant neoplasm of body of stomach |
| Ipratropium Bromide Neb | Fentanyl Citrate |
| Ranitidine | Lactulose |
| Metronidazole | Milk of Magnesia |

# Chapter 4

# MULTIPAR: Supervised Irregular Tensor Factorization Framework with Multi-task Learning

## 4.1 Overview

Current PARAFAC2 model's predictability and interpretability are not satisfactory, which limits its utility for downstream analysis. In this chapter, we propose **MULTIPAR**: a supervised irregular tensor factorization framework with multi-task learning. MULTIPAR is flexible to incorporate both static (e.g. in-hospital mortality prediction) and continuous or dynamic (e.g. the need for ventilation) tasks. By supervising the tensor factorization with downstream prediction tasks and leveraging information from multiple related predictive tasks, MULTIPAR can yield not only more meaningful phenotypes but also better predictive performance for downstream tasks.

As shown in figure 4.1. MULTIPAR jointly optimizes the tensor factorization and downstream prediction together, so that the factorization can be "supervised" or informed by the predictive tasks. In addition, we use a multi-task framework to leverage information from multiple predictive tasks. It provides flexibility to incorporate both static and dynamic outcomes. To achieve this, the temporal features from **U** matrix is used for dynamic prediction and the features from **S** matrix are used for static prediction, as shown in the figure 4.1.

In summary, we list our main contributions below:

1. We propose a supervised framework for PARAFAC2 tensor factorization and downstream prediction tasks such that the factorization can be "supervised" or informed by the predictive tasks.

2. We use a multi-task framework to leverage information from multiple predictive tasks and provide flexibility to incorporate both static and dynamic tasks and different models (e.g. logistic regression and recurrent neural networks).

3. We introduce a novel unified and dynamic weight selection method for weighing the tensor factorization and predictive tasks during the optimization process, where the tensor factorization is considered as one task, to achieve overall optimized result.

4. We evaluate MULTIPAR's tensor reconstruction quality, predictability, scalability, and interpretability on two real-world temporal EHR datasets through a set of experiments, which verify MULTIPAR can identify more meaningful subgroups and yield stronger predictive performance compared to existing state-of-the-art approaches.

## 4.2 Preliminaries and Backgrounds

In this section, we first introduce the necessary background of tensor operations. Table 4.1 summarizes the notations used throughout the chapter.

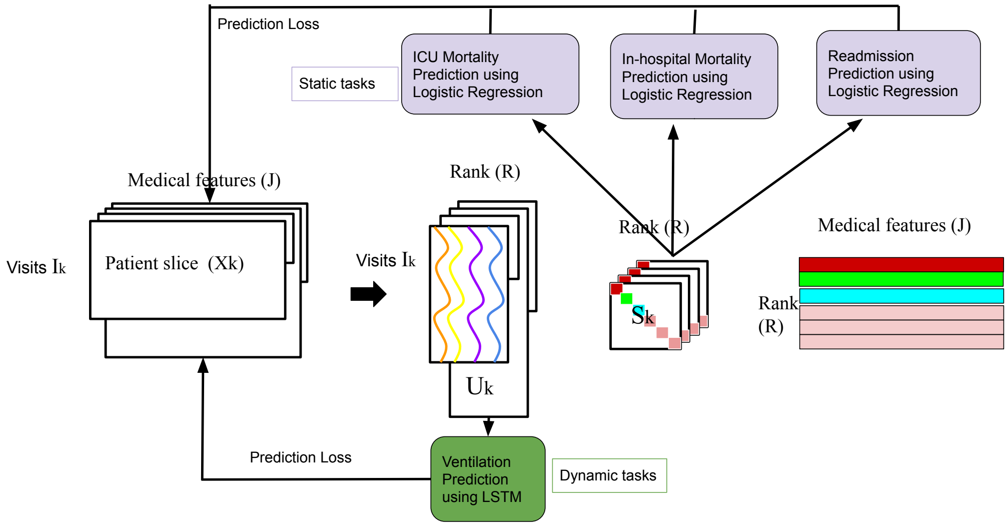

Prediction Loss
Static tasks
ICU Mortality Prediction using Logistic Regression
In-hospital Mortality Prediction using Logistic Regression
Readmission Prediction using Logistic Regression
Medical features (J)
Rank (R)
Rank (R)
Medical features (J)
Visits Ik
Patient slice (Xk)
Visits Ik
Sk
Rank (R)
Uk
Prediction Loss
Ventilation Prediction using LSTM
Dynamic tasks

Figure 4.1: Overview of MULTIPAR on MIMIC-EXTRACT dataset

The mode or order of an tensor is the number of dimensions of a tensor (e.g., vectors are 1-order tensors and matrices are 2-order tensors). Extracting a fiber refers to a vector derived from the tensor by fixing all modes but one. For example, a matrix column is a mode-1 fiber. Extracting a slice refers to fixing all modes but two. In particular, the $\mathbf{X}(:,:,k)$ slices of a third order tensor $\mathbf{X}$ are called the frontal ones, and we denote them as $\mathbf{X}_k$. Tensor unfolding, or matricization, is a fundamental operation and a building block for most tensor methods. It logically reorganizes tensors into other forms without changing the values themselves. The mode-$n$ matricization of an N-order tensor $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ is denoted by $\mathbf{X}_{(n)} \in \mathbb{R}^{I_n \times I_1 I2 \ldots I_{n-1} I_{n+1} \ldots I_N}$ and arranges the mode-$n$ fibers of the tensor as columns of the resulting matrix.

In tensor operations, scalar multiplication means the scalar is multiplied to every element in the tensor. The inner product of two tensors means to multiply each element of the first tensor by the corresponding element of the second tensor, and adding the results. The outer product of two coordinate vectors is a matrix, which is obtained by multiplying each element of the first vector by each element of the second vector. The Frobenius norm, sometimes also called the Euclidean norm, is defined as the square root of the inner product of the tensor, which is the sum of the absolute

Table 4.1: Symbols and notations used in chapter 4

| Symbol | Definition |
|---|---|
| $\mathbf{x}, \mathbf{X}, \mathcal{X}$ | Vector, Matrix, Tensor |
| $\mathbf{X}_k$ | $k$-th frontal slice of $\mathcal{X}$ |
| $\mathcal{X}_{(n)}$ | Mode-$n$ matricization of $\mathcal{X}$ |
| $\lVert \cdot \rVert_1$ | $\ell_1$-norm |
| $\lVert \cdot \rVert_F$ | Frobenius norm |
| $\mathbf{U}_k$ | The temporal factor matrix for the $k^t h$ subject |
| $\mathbf{S}_k$ | The weighting vector for the $k^t h$ subject |
| $\mathbf{V}$ | The latent factor matrix for the features |
| $\mathbf{I}_k$ | The temporal length of the $k^t h$ subject |
| $R$ | Number of target Rank |
| $*$ | Hadamard (element-wise) multiplication |
| $\odot$ | Khatri Rao product |
| $\circ$ | Outer product |
| $\langle \cdot, \cdot \rangle$ | Inner product |

squares of its elements.

### 4.2.1 PARAFAC2

The most popular tensor factorization method is CP decomposition [10, 22, 26], also known as PARAFAC. It approximates a tensor into a sum of $R$ rank-one tensors. $R$ is the rank of tensor $\mathbf{X} \in \mathbb{R}^{k \times I \times J}$, which can be expressed as:

$$\mathbf{X} \approx \sum_{r=1}^{R} u_r \circ v_r \circ w_r \tag{4.1}$$

where $u_r \in \mathbb{R}^k$, $v_r \in \mathbb{R}^I$, and $w_r \in \mathbb{R}^J$ are column vectors, and $\circ$ denotes the outer product. $\mathbf{U} = [u_1, ...u_R]$, $\mathbf{V} = [v_1, ...v_R]$, $\mathbf{W} = [w_1, ...w_R]$ are factor matrices. The basic idea of CP decomposition is to find R latent concepts to approximate the original tensor. However, since each mode of CP decomposition is fixed size, it can not handle irregular tensor factorization, where one mode in the irregular tensor has unfixed size.

PARAFAC2 model is the state-of-the-art tensor factorization framework for irregular tensor, i.e., tensors that do not align along one of its modes. The PARAFAC2 model decomposes each frontal slice of the irregular tensor $\mathbf{X}_k$ as $\mathbf{U}_k \mathbf{S}_k \mathbf{V}^\top$, where $\mathbf{U}_k \in \mathbb{R}^{I_k \times R}$, $\mathbf{S}_k \in \mathbb{R}^{R \times R}$ is diagonal and $\mathbf{V} \in \mathbb{R}^{J \times R}$. $R$ is the target rank. Uniqueness is an important property in tensor factorization models that ensures the solution is not an arbitrarily rotated version of the actual latent factors. In order to enforce uniqueness, Harshman [22] imposed the constraint $\mathbf{U}^T\mathbf{U}_k = \Phi, \forall k$. This is equivalent to each $\mathbf{U}_k$ being decomposed as $\mathbf{U}_k = \mathbf{Q}_K\mathbf{H}$, where $\mathbf{Q}_k \in \mathbb{R}^{I_k \times R}$, $\mathbf{Q}_k^\top\mathbf{Q}_k = \mathbf{I}$, and $\mathbf{H} \in \mathbb{R}^{R \times R}$. Note that $\mathbf{Q}_k$ has orthonormal columns and $\mathbf{H}$ is invariant regardless of $k$.

Given the above modeling, the standard algorithm to fit PARAFAC2 solves the following optimization problem:

**Definition 9.** *(Original PARAFAC2 model)*

$$\underset{\{\mathbf{U}_k\},\{\mathbf{S}_k\},\mathbf{V}}{\operatorname{argmin}} \sum_{k=1}^{K} \frac{1}{2}\|\mathbf{X}_k - \mathbf{U}_k\mathbf{S}_k\mathbf{V}^\top\|_F^2,$$

*subject to:* $\mathbf{U}_k = \mathbf{Q}_k\mathbf{H}, \mathbf{Q}_k^\top\mathbf{Q}_k = \mathbf{I}, \mathbf{S}_k$ *is diagonal.*

Given a tensor representing the EHRs data as in figure 4.1 where each slice $X_k$ represents the information of patient $k$ with $I_k$ visits and $J$ medical features, PARAFAC2 decomposes the irregular tensor $\mathbf{X}$ into the factorization matrices which have the following interpretations:

- $\mathbf{U}_k \in \mathbb{R}^{I_k \times R}$ represents the temporal trajectory of $I_k$ clinical visits in each one of the $R$ phenotypes.
- $\mathbf{V} \in \mathbb{R}^{J \times R}$ represents the relationship between medical features and phenotypes.
- $\mathbf{S}_k \in \mathbb{R}^{R \times R}$ represents the relationship between patients and phenotypes. Each column in S represents one phenotype, and if a patient has the highest weight in a specific phenotype, it means the patient is mostly associated with or exhibits a particular phenotype.

SPARTan [51] was developed to decompose large-scale sparse datasets, and COPA [2] extended SPARTan to further enhance the interpretability by adding more constraints, e.g., smoothness on $\mathbf{U}_k$ and sparsity on $\mathbf{V}$. REPAIR [60] and LogPar [80]

Table 4.2: Comparison of existing PARAFAC2-based models

| **PARAFAC2 model** | **Scalability** | **Predictability** | **Robustness** | **Interpretability** |
|---|---|---|---|---|
| Original PARAFAC2 | x | x | x | x |
| SPARTan [51] | ✓ | x | x | x |
| COPA [2] | ✓ | x | x | ✓ |
| REPAIR [60] | ✓ | x | ✓ | ✓ |
| LogPar [80] | ✓ | x | ✓ | x |
| MULTIPAR | ✓ | ✓ | x | ✓ |

add low-rankness constraints to improve the robustness of PARAFAC2 model to handle missing values. However, no current work has considered using downstream tasks to supervise and improve the predictability of PARAFAC2 model as table 4.2 shows.

### 4.2.2 Supervised and Multi-task learning Framework

The supervised machine learning model has shown great performance compared to the non-supervised one for a wide range of applications, e.g., graph learning [74], pattern classification [45], and tensor factorization [37]. In graph learning paper [74], they propose a supervised feature extraction framework using discriminative clustering to improve model's clustering accuracy. In patter classification paper [45], they propose a supervised minimum similarity projection framework using lowest correlation representation to improve model's classification accuracy. In tensor factorization paper [37], they introduce a novel supervised tensor factorization using diagnosis cluster structure, which can significantly improve model's discriminative power.

Over past years, multi-task learning (MTL) [84, 83] has attracted much attention in the artificial intelligence and machine learning communities. Traditional machine learning frameworks solve a single learning task each time, which ignores commonalities and differences across different tasks. MTL aims to learn multiple related tasks jointly so that the knowledge contained in one task can be leveraged by other tasks, with the hope of improving generalization performance by learning a shared representation [4, 68]. MTL has been used successfully across all applications, from natural language processing [16, 70] and speech recognition [18, 12] to computer vision [20] and drug discovery [59, 24]. However, no current work has considered improving predictability of tensor factorization using MTL.

There are other machine learning paradigms that are related to MTL, e.g., transfer learning [77, 69, 67], and multi-label learning [82, 70], but these have significant differences compared to MTL. In MTL, there is no distinction among different tasks and the aim is to improve the performance of all the tasks. However, transfer learning separate the tasks into two different groups: target task and source tasks. The aim of transfer learning is to improve the performance of a target task with the help of source tasks. In our case, we adopt MTL because our goal is to improve both tensor factorization quality (to extract meaningful and representative latent factors) and prediction task accuracy. In multi-label learning, there are multiple static labels associated with each data point. In our case, we have both static and dynamic outcomes, which fits in MTL.

MTL models' performance is strongly dependent on the relative weighting between each task's loss. There are several weighting strategies available in the MTL framework. The most naive way is to uniformly combine the losses from the different tasks, which is called vanilla MTL. Dynamic weight average [44] will dynamically calculate the loss ratio of different epochs, and assign the weight accordingly. Uncertainty weighting methods [15, 39] use homoscedastic (task) uncertainty to calculate the weight for each task. We introduce a unified and dynamic weight selection method for weighing the tensor factorization and predictive tasks, where the tensor factorization is considered as one task, we calculate the average summation of the loss over several epochs, and dynamically calculate the weight for each task by considering the loss change rate over several epochs, which can minimize the noise caused by the noisy nature of Stochastic Gradient Descent (SGD), and improve the convergence speed.

## 4.3 Proposed Method

In this section, we present the MULTIPAR model in the context of EHR pheynotyping and its optimization. The general framework is applicable to any irregular tensor factorization and predictive learning tasks.

### 4.3.1 Problem Formulation

We formalize the objective function for the MULTIPAR model in Definition 10. The PARAFAC2 loss for $\mathcal{X}$ ensures the reconstructed tensor closely approximates the original tensor. The static outcomes loss and dynamic outcomes loss are separate prediction tasks. Static outcome prediction tasks have a one-time or static labels, and dynamic outcome prediction tasks have a continuously changing or temporal dynamic labels for each time stamp. An approximate uniqueness constraint ensures tensor factorization uniqueness. For EHRs phenotype discovery, various constraints can be imposed on the factorization matrices to yield meaningful and high-interpretability phenotypes. The MULTIPAR model accommodates such interpretability-purposed constraints in eq. (4.2) including: non-negativity for $c_1(\mathbf{S}_k)$, sparsity for $c_2(\mathbf{V})$. We explain each of the loss components and constraints in detail below.

**Definition 10.** *(MULTIPAR objective function)*

$$
\begin{aligned}
&\underset{\mathbf{Q}_k,\mathbf{H},\mathbf{S}_k,\mathbf{V}}{\operatorname{argmin}} \sum_{k=1}^{K} \sum_{(i,j)\in\Omega} \overbrace{\rho_1 L_1(\mathbf{X}_{ijk}, \{\mathbf{U}_k \mathbf{S}_k \mathbf{V}^\top\}_{ijk})}^{\textit{PARAFAC2 loss for } \mathcal{X}} \\
+ \quad & \overbrace{\rho_2 L_2(\mathbf{S}_k)}^{\textit{static outcomes loss}} \quad + \quad \overbrace{\rho_3 L_3(\mathbf{U}_k)}^{\textit{dynamic outcomes loss}} \\
+ \quad & \overbrace{\varrho_1 \|\mathbf{U}_k^\top \mathbf{U}_k - \Phi\|_F^2}^{\textit{approximate uniqueness constraint}} \\
+ \quad & \overbrace{\sum_{k=1}^{K} c_1(\mathbf{S}_k)}^{\textit{non-negativity constraint}} \quad + \quad \overbrace{c_2 \|\mathbf{V}\|_1}^{\textit{sparsity constraint}}
\end{aligned}
\tag{4.2}
$$

$$s.t.\ for\ k = 1, ..., K, \tag{4.3}$$

*where* $\mathbf{H}, \{\mathbf{S}_k\}, \mathbf{I} \in \mathbb{R}^{R \times R}$*.* $c_1$ *is the nonnegativity constraint, and* $c_2\|\mathbf{V}\|_1$ *is the sparsity penalty.*

### PARAFAC2 loss

The PARAFAC2 tensor factorization loss can ensure the reconstructed tensor closely approximate the original tensor. To accommodate different data types, the PARAFAC2 loss can be any smooth loss function, e.g., Least square loss, Poisson loss [29] and Rayleigh Loss [29].

### Static outcomes loss

Previous PARAFAC2 models separate the PARAFAC2 training process and downstream prediction process. For example, in-hospital mortality prediction accuracy may be used as the metric to measure the predictability of the phenotypes extracted by the model. In the MULTIPAR model, we optimize the downstream prediction tasks and tensor factorization together by adding the prediction losses of the prediction tasks to the objective function. If the prediction task has one label per patient, we denoted it as a static outcome prediction task. For illustrative purposes, we use a logistic regression model on the **S** matrix to predict static outcome tasks, and add the cross-entropy loss to the objective function. In fact, any differentiable loss function (e.g., square loss, exponential loss) can be incorporated in the objective function.

### Dynamic outcomes loss

Different from static outcomes, dynamic outcomes have labels at each timestamp. For example, predicting whether a patient will be on a ventilator can also be used to measure the model's predictability. For illustrative purposes, we use the long short-term memory (LSTM) model on the **U** matrix to predict each patient's dynamic outcome labels, and add the loss of the LSTM model to the objective function. Similar to the static outcome loss, other models (e.g., gated recurrent units, vanilla recurrent neural networks) and their associated loss functions can be incorporated in the objective function.

### Approximate uniqueness constraint

The optimization of the original PARAFAC2 model adopts the AO-ADMM framework, which can not make full use of the parallel computation feature of GPUs. To adopt mainstream deep learning frameworks like Pytorch and Tensorflow, we use a stochastic gradient descent (SGD) based optimization approach. The uniqueness constraint in the original PARAFAC2 model is $\mathbf{U}_k^\top \mathbf{U}_k = \Phi$. Similar to LogPar [80], to optimize $\mathbf{U}_k$ we relax the uniqueness constraint to $\|\mathbf{U}_k^\top \mathbf{U}_k - \Phi\|_F^2$.

### Sparsity on V

The $\mathbf{V}$ matrix captures the association between a medical feature and a particular phenotype. In order to improve interpretability, we introduce a sparsity constraint on the $\mathbf{V}$ matrix. $l_0$ and $l_1$ norms are two popular sparsity regularization techniques. The $l_0$ regularization norm, also relaxed by hard thresholding, will cap the number of non-zero values in a matrix. The $l_1$ regularization norm, also relaxed by soft thresholding, will shrink matrix values towards zero. As hard thresholding is a non-convex optimization problem which can not be optimized by the SGD framework, we adopt the soft thresholding, which is convex and can be migrated into the SGD optimization framework.

### Non-negativity on S

The diagonal matrix $\mathbf{S}$ indicates the importance membership of patient $k$ in each one of the $R$ phenotypes. Since only non-negative membership values makes sense, we zero out the negative values in **S**, which significantly improves the interpretability.

### SDW: Smooth dynamic weight selection

Numerous deep learning applications benefit from MTL with multiple regression and classification objectives. Yet the performance of MTL is strongly dependent on the relative weighting between each task's loss. Our objective function consists of several losses from the tensor factorization and the predictive tasks. Each of this loss is associated with a weight. Tensor factorization is considered as a special task. Definition 10 shows $\rho_1$ as the weight for tensor loss, $\rho_2$ and $\rho_3$ as the accumulative weights for static and dynamic tasks respectively, here we use $\rho_n(t)$ to denote the weight for each individual task $n$ in epoch $t$. A key challenge is how to tune these weights for different tasks. While the DWA weight selection [44] was proposed to dynamically change the task weights at each epoch by considering the rate of change of the loss over the epoch, the noisy nature of SGD weights can cause drastic fluctuations in the task weights between epochs. This can cause oscillating behavior between the various tasks and impedes convergence of the algorithm. Therefore, we propose a novel smooth dynamic weight selection method to choose the weight for each task. We first calculate the relative descending rate of each task loss and denote it as $\omega_n(t-1)$. $t$ here represents an epoch index:

$$\omega_n(t-1) = \frac{Loss_n(t-1)}{Loss_n(t-2)} \tag{4.4}$$

We then calculate the weight for each task using the following equation:

$$\rho_n(t) := \frac{exp(\sum_{j=1}^{m}(\omega_n(t-j)/C)/m)}{\sum_{i=1}^{N} exp(\sum_{j=1}^{m}(\omega_i(t-j)/C)/m)/N} \tag{4.5}$$

Similar to [25], we use $C$ to control the softness distribution between different tasks. If $C$ is large enough, the weight for each task will be uniformly weighted.

Different from [25], we introduce $m$, the weight update window size. The task weights are updated as an average over several epochs from iteration $t$ to $t+m$ (instead of using one iteration) to reduce the SGD update uncertainty and training data selection randomness. Finally, a softmax operator, which is multiplied by the number of tasks $N$, ensures the sum of the weight equals $N$. For $t=1$, we initialize all the weights to 1.

### 4.3.2 Optimization

To solve the optimization problem in Eq. (4.2), MULTIPAR follows an alternative optimization strategy where we optimize one variable individually with all other variables fixed. According to the subproblem smoothness, we group the variables into two groups: pure smooth subproblems and proximal mapping-base smooth subproblems. In the following, we omit the iteration number for brevity in notation.

**Pure Smooth Subproblems Updates.**

For the pure smooth subproblems, we use SGD to update the variables, which include the following three parts:

**Update of $\mathbf{U}_k$.** The subproblem of $\mathbf{U}_k$ takes the form as follows

$$\arg\min_{\mathbf{U}_k} \sum_{(i,j)\in\Omega} \rho_1 L(\mathbf{X}_{ijk}, \{\mathbf{U}_k \mathbf{S}_k \mathbf{V}^\top\}_{ijk}) + \rho_3 L_3(\mathbf{U}_k) \tag{4.6}$$

**Proximal Mapping-base Smooth Subproblems Updates**

For the nonsmooth subproblems, we propose a proximal mapping-based update, which include the following two parts.

**Algorithm 3** Optimization Framework for MULTIPAR

```
Input: Input tensor $\mathcal{X}$; Model parameters $\rho_1$-$\rho_3$, $\varrho_1$; Interpretability constraint types $c_1, c_2$; Initial rank estimation $R$.
while Not reach convergence criteria do
    Update $\{\mathbf{U}_k\}$ using eq.(4.6) by SGD;
    Update $\mathbf{S}_k$ using eq.(4.7) by Proximal/Projected SGD;
    Update $\mathbf{V}$ using eq.(4.9) by Proximal/Projected SGD;
    Calculate weight for each prediction task using eq.(4.4) and eq.(4.5) by SDW;
end while
Output: Phenotype factor matrices $\mathbf{U}_k, \mathbf{S}_k, \mathbf{V}$.
```

**Update of $\mathbf{S}_k$.** The subproblem of $\mathbf{S}_k$ takes the form as follows

$$\arg\min_{\mathbf{S}_k} \sum_{(i,j)\in\Omega} \rho_1 L(\mathbf{X}_{ijk}, \{\mathbf{U}_k\mathbf{S}_k\mathbf{V}^\top\}_{ijk}) + \rho_2 L_2(\mathbf{S}_k) + c_1(\mathbf{S}_k). \tag{4.7}$$

We use projected SGD to update $\mathbf{S}_k$, where each step takes the following form

$$\mathbf{S}_k = \max(0, \mathbf{S} - \lambda \mathbf{G}[\mathbf{S}_k]), \tag{4.8}$$

where $G[\mathbf{S}_k]$ denotes the stochastic gradient of the smooth part $\sum_{(i,j)\in\Omega} \rho_1 L(\mathbf{X}_{ijk}, \{\mathbf{U}_k\mathbf{S}_k\mathbf{V}^\top\}_{ijk}) + \rho_1 L_2(\mathbf{S}_k)$ with respect to $\mathbf{S}_k$.

**Update of $\mathbf{V}$.** The subproblem of $\mathbf{V}$ takes the form as follows

$$\arg\min_{\mathbf{V}} \sum_{k=1}^{K} \sum_{(i,j)\in\Omega} \rho_1 L(\mathbf{X}_{ijk}, \{\mathbf{U}_k\mathbf{S}_k\mathbf{V}^\top\}_{ijk}) + c_2\|\mathbf{V}\|_1. \tag{4.9}$$

We use soft-thresholding operator to update $\mathbf{V}$, where each step takes the following form: $\texttt{soft}-\texttt{thresholding}(\mathbf{V} - \lambda\mathbf{G}[\mathbf{V}])$. $G[\mathbf{V}]$ denotes the stochastic gradient of the smooth part $\sum_{k=1}^{K}\sum_{(i,j)\in\Omega} \rho_1 L(\mathbf{X}_{ijk}, \{\mathbf{U}_k\mathbf{S}_k\mathbf{V}^\top\}_{ijk})$ with respect to $\mathbf{V}$.

**The complete algorithm.** The optimization procedure is summarized in Algorithm 3.

## 4.4 Experimental Evaluation

### 4.4.1 Dataset

We use two real-world datasets to evaluate MULTIPAR in terms of its reconstruction quality, predictive performance, interpretability, and scalability.

**eICU** [1] **[54]:** The eICU Collaborative Research Database is a freely available multicenter database for critical care research. It contains variables used to calculate the Acute Physiology Score (APS) III for patients. APS-III is an established method of summarizing patient severity of illness on admission to the ICU. We select 202 diagnosis codes that have the highest frequency, as in [36]. The resulting number of unique ICU visits is 145426. The maximum number of observations for a patient is 215. We select three static outcome prediction tasks, including intubated prediction, ventilation prediction, and dialysis prediction. The ventilation prediction here is a static prediction tasks indicated whether a patient need to be ventilated at the time of the worst respiratory rate, we will use ”vent-res” as the name for this task.

**MIMIC-EXTRACT** [2] **[72]:** MIMIC-Extract is an open-source pipeline for transforming raw EHR data in MIMIC-III into data frames that are directly usable in common machine learning pipelines. We use the vitals labs mean table, which contains 34,472 patients with 104 features (Vital lab codes). The maximum number of observations for a patient is 240. We further normalize the data to [0,1]. We select three static outcome prediction tasks, including in-hospital mortality prediction, readmission prediction, ICU mortality prediction, and one dynamic outcome prediction task, which is ventilation prediction.

[1] `https://eicu-crd.mit.edu`

[2] `https://github.com/MLforHealth/MIMIC_Extract/`

### 4.4.2 Evaluation Metrics

In order to test the tensor reconstruction quality of MULTIPAR model, we adopt the $FIT \in (-\infty, 1]$ score [6] as the quality measure (the higher the better):

$$FIT = 1 - \frac{\sum_{k=1}^{K} \|\mathbf{G}_k - \mathbf{U}_k \mathbf{S}_k \mathbf{V}^T\|^2}{\sum_{k=1}^{K} \|\mathbf{G}_k\|^2}. \tag{4.10}$$

The original tensor, denoted as $\{\mathbf{G}_k\}$, serves as the ground truth. $\mathbf{U}_k, \mathbf{S}, \mathbf{V}$ are factor matrices after the MULTIPAR tensor factorization.

We evaluate the derived phenotypes' predictability power using the PR-AUC score of the prediction tasks. We split the data with a proportion of 8:2 as training and test sets and use PR-AUC score to evaluate the predictive power.

### 4.4.3 Methods for Comparison

We compare MULTIPAR with three baseline methods: SPARTan, COPA, and singlePAR. SPARTan and COPA are two state-of-the-art irregular tensor factorization methods. We also compare against a supervised single task PARAFAC2, which is a single-task version of MULTIPAR.

- **SPARTan [51] - scalable PARAFAC2**: A tensor factorization method for fitting large and sparse irregular tensor data. It only considers the tensor reconstruction loss.

- **COPA [2] - scalable PARAFAC2 with additional regularizations**: An irregular tensor factorization method that introduces various constraints/regularizations to improve the interpretability of the factor matrices. It only considers the tensor reconstruction loss. For both SPARTan and COPA, the extracted phenotypes are used for training the models for the downstream predictive tasks.

- **SinglePAR - supervised single task PARAFAC2**: The supervised irregular tensor factorization with single prediction task (single task version of MULTIPAR).

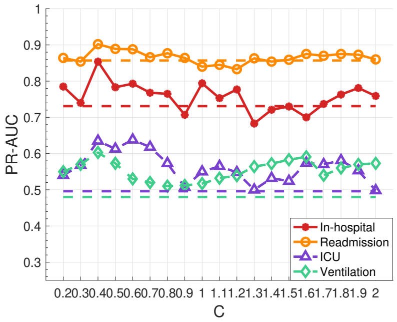


Figure 4.2: PR-AUC score using different C

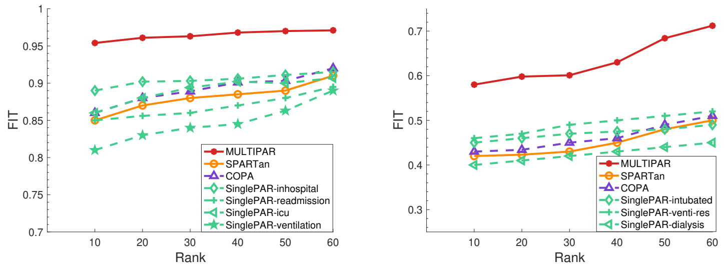
### 4.4.4 Implementation Details

The performance of MTL is strongly dependent on the relative weighting between each task's loss. In order to present the best performance of MULTIPAR, SDW has two hyper-parameters that need to be tuned. $C$ controls the softness distribution between different tasks, and $m$ is the weight update window size. In order to find the best $C$, we vary $C$ from 0.2 to 2, and compare the prediction tasks' PR-AUC scores on different data set under different ranks. Figure 4.2 shows the MIMIC-EXTRACT dataset result when rank $= 50$ and $m$ is fixed to 5. In our empirical experiments, when $C = \frac{1}{\sqrt{N}}$, MULTIPAR shows the best performance.

We vary the weight update window size $m$ from 1 to 10, and compare the convergence speed and PR-AUC score. We fix $C = \frac{1}{\sqrt{N}}$, and plot the tensor loss in each epoch and set the maximum number of epochs to be 200. When $m = 1$, it does not converge after 200 epochs. When $m = 5$, it requires the least number of epochs to converge (when the total loss plateaus). Although when $m = 3$, some prediction tasks' PR-AUC scores are slightly better than $m = 5$, it requires too many epochs to converge. Thus, in our experiments below, we adopt $m = 5$.

Table 4.3: Experiment result of PR-AUC and convergence epochs when $m$ varies

| | **m=1** | **m=3** | **m=5** | **m=8** | **m=10** |
|---|---|---|---|---|---|
| In-hospital mortality prediction task | 0.740 | 0.789 | 0.854 | 0.783 | 0.768 |
| Readmission prediction task | 0.872 | 0.893 | 0.902 | 0.892 | 0.853 |
| ICU mortality prediction task | 0.626 | 0.638 | 0.635 | 0.583 | 0.571 |
| Ventilation prediction task | 0.600 | 0.605 | 0.603 | 0.591 | 0.587 |
| Convergence epoch | 200 | 187 | 98 | 110 | 150 |



(a) FIT score on EXTRACT (b) FIT score on eICU

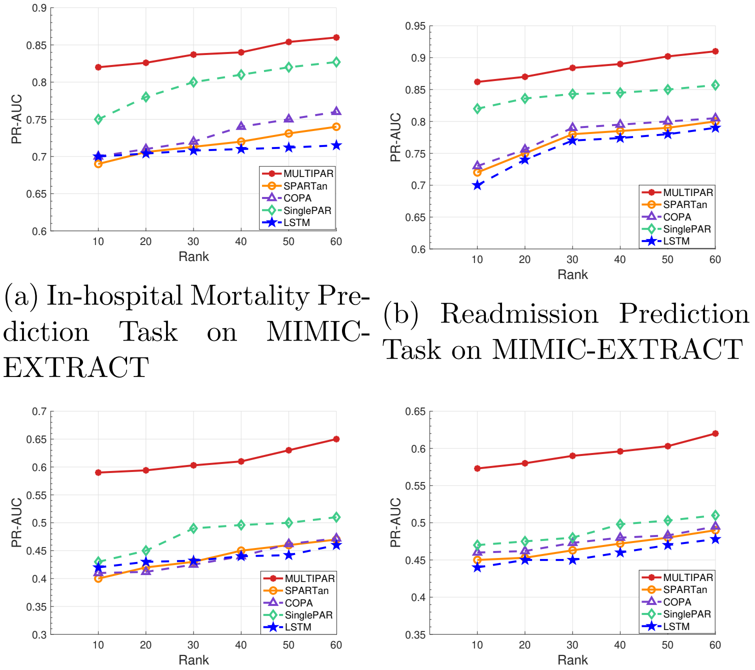
Figure 4.3: FIT on MIMIC-EXTRACT and eICU dataset.

### 4.4.5 Experiment Result

**Tensor reconstruction quality analysis**

For the following experiments on tensor reconstruction quality, we run each method for 5 different random initializations and report the average $FIT$. In addition, we evaluate model completion performance under different target ranks, $R$, from 10 to 60, and run 200 epochs.

First, we compare MULTIPAR model's FIT with the baseline models on two datasets shown in figure 4.3. MULTIPAR model optimizes all prediction tasks and tensor factorization together. SPARTan and COPA first finish the tensor factorization, and then predict the downstream prediction tasks. As figure 4.3a and 4.3b shows, MULTIPAR outperforms all the other baseline methods on all datasets. In particular, MULTIPAR achieves a FIT score of 0.97 and 0.71 on MIMIC-EXTRACT and eICU respectively, a 13% and 40% relative improvement when compared to the best baseline model SinglePAR, which shows the strong tensor reconstruction ability



(a) In-hospital Mortality Prediction Task on MIMIC-EXTRACT

(b) Readmission Prediction Task on MIMIC-EXTRACT

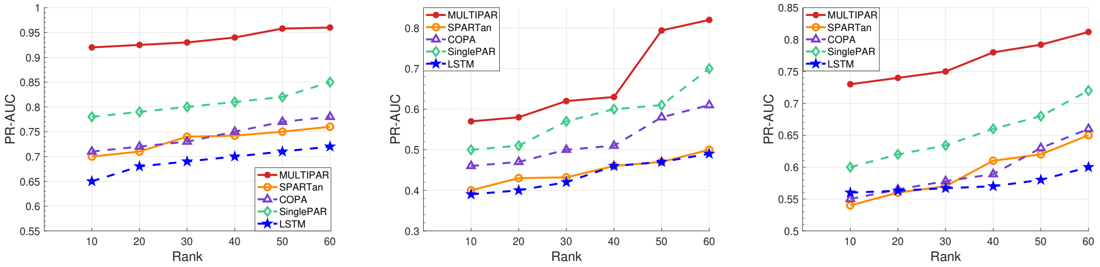
(c) ICU Mortality Prediction Task on MIMIC-EXTRACT

(d) Ventilation Prediction Task on MIMIC-EXTRACT

Figure 4.4: PR-AUC for prediction tasks on MIMIC-EXTRACT

of MULTIPAR, thanks to the “supervision” of the multiple predictive tasks. COPA performs better than SPARTan because it introduces various regularizations on the factor matrices, which can slightly improve the tensor reconstruction ability.

SinglePAR is a single task version of MULTIPAR. In sub-figure 4.3a, SinglePAR adopts the in-hospital mortality perdition, readmission prediction, ICU mortality prediction, and ventilation prediction tasks respectively. SinglePAR performs better than SPARTan and COPA on most of the ranks but is left behind COPA on large ranks. SinglePAR jointly optimizes prediction task and tensor factorization together. We can see that certain tasks benefit the tensor FIT while others may guide the tensor factorization into a suboptimal direction and degrade the tensor reconstruction quality. Although MULTIPAR model is supervised, thanks to the MTL, it can use all of the available outcomes across the different tasks to learn generalized representations of the data that are useful in the tensor reconstruction context.



(a) Dialysis Prediction Task on eICU

(b) Venti-res Prediction Task on eICU

(c) Intubated Prediction Task on eICU

Figure 4.5: PR-AUC for prediction tasks on eICU dataset

It is also noteworthy to discuss the FIT score trend as a function of the rank. All of the methods show better FIT score as rank increases because a large rank can preserve more information in the hidden space when doing the tensor factorization, thus having better reconstruction quality.

### Predictability analysis

A logistic regression model is trained on the patient importance membership matrix $\mathbf{S}_k$ for static outcome prediction tasks and an LSTM model is trained on the temporal evolution matrix $\mathbf{U}_k$ for dynamic outcome prediction task. LSTM is a variant of the recurrent neural network (RNN) that mitigates the gradient vanishing problem in traditional RNNs. Its memory cells contain three types of non-linear gates, namely input gate, output gate and forget gate, which can regulate the flow of signals into and out of the cell and learn long-term dependencies. Moreover, LSTM can process varying-length input data.

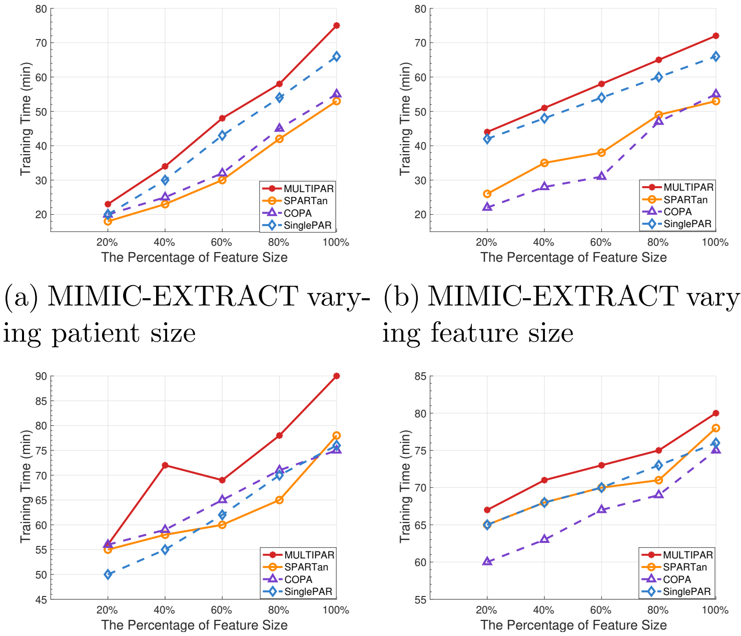
In the MIMIC-EXTRACT dataset, only the ventilation prediction task is a dynamic outcome prediction task, and all the tasks in the eICU dataset are static outcome prediction tasks. In order to illustrate the benefit of using the latent factors as features for a downstream prediction model, we also include a LSTM model trained using the original EHR data. The reason why we choose LSTM model is because the original EHR data contains different length patients' visit data, and LSTM model can handle the varying size input temporal data. The input to the LSTM model is an irregular tensor which contains $k$ different patient, and each patient information $X_k$ consists of $I_K$ visits and $J$ medical features. The output is the prediction label for the different patients or different time stamps.

We evaluate the prediction accuracy as a function of the tensor factorization rank. As shown in figure 4.4 and 4.5, MULTIPAR outperforms the other methods. In figure 4.4, when the rank is 10, MULTIPAR outperforms the best baseline methods SinglePAR by 17%, 18%, 20% and 22% in figure 4.4a, 4.4b, 4.4c and 4.4d for each of the tasks respectively. This demonstrates MULTIPAR's strong generalization ability across multiple prediction tasks by leveraging the shared information between different tasks as well as the strong predictive power by the extracted phenotypes. Moreover, SinglePAR always outperforms COPA, SPARTan, and LSTM, which shows that the supervised learning framework can improve predictability. The figure also illustrates the important role PARAFAC2 plays as the non-tensor based LSTM model performs the worst because it lacks the ability to filter out noise in the raw EHR.

### Scalability analysis

Adding MTL on the PARAFAC2 framework can raise some concerns related to potential scalability issues on large datasets. Therefore, we evaluated the computational time of MULTIPAR compared with the other baseline methods using different data sizes and different feature sizes. We use two Titan RTX GPUs, each GPU has 24 GB of RAM, and rain 50 epochs of both methods.

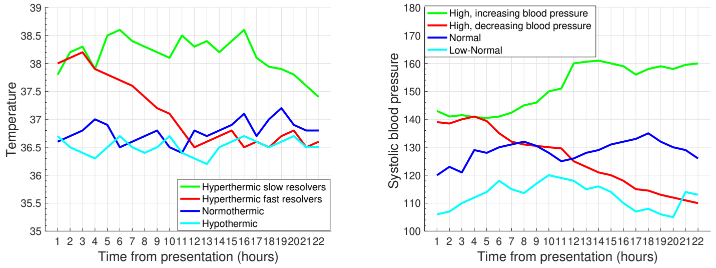
In Figure 4.6, we show the total training time. COPA, SPARTan, and SinglePAR shows linear scalability as the number of patients and features grows. Although MULTIPAR adds MTL, it still exhibits linear scalability similar to SinglePAR. Although MTL adds some additional training time, it is not significantly more as the maximum added time is 8 minutes.



(a) MIMIC-EXTRACT varying patient size (b) MIMIC-EXTRACT varying feature size

(c) eICU varying patient size (d) eICU varying feature size

Figure 4.6: Training time on MIMIC-EXTRACT and eICU varying patient size and feature size



(a) Temperature (b) Systolic blood pressure

Figure 4.7: Temporal Trajectory

Table 4.4: MIMIC-EXTRACT phenotypes discovered by MULTIPAR.

| **Phenotype 1 (Normal vital signs)** | **Weight** | **Average Value** | **Weighted Average Value** |
|---|---|---|---|
| Oxygen saturation | 1.52 | 98.5 | 149.72 |
| Systolic blood pressure | 0.91 | 112.7 | 101.92 |
| Heart rate | 0.82 | 82.5 | 67.24 |
| Mean blood pressure | 0.79 | 81.2 | 63.99 |
| Diastolic blood pressure | 0.65 | 76.3 | 49.6 |
| Respiratory rate | 0.57 | 18.6 | 10.602 |
| Co2 (etco2, pco2, etc.) | 0.43 | 24.2 | 10.4 |
| **Phenotype 2 (Abnormal renal and liver function)** | **Weight** | **Average Value** | **Weighted Average Value** |
| Alanine aminotransferase | 11.51 | 83.1 | 956.481 |
| Blood urea nitrogen | 9.64 | 42.3 | 407.77 |
| Alkaline phosphate | 8.01 | 153.2 | 1224 |
| Asparate aminotransferase | 5.18 | 90.1 | 466.718 |
| Albumin | 3.90 | 3.2 | 12.48 |
| Bicarbonate | 2.76 | 17 | 46.92 |
| Mean blood pressure | 1.59 | 85 | 135.15 |
| **Phenotype 3 (Normal Blood Counts and Serum Electrolytes)** | **Weight** | **Average Value** | **Weighted Average Value** |
| Mean corpuscular hemoglobin concentration | 7.54 | 32.1 | 242.0 |
| Sodium | 4.93 | 135.2 | 666.53 |
| Mean corpuscular hemoglobin | 3.62 | 30.8 | 111.49 |
| Mean corpuscular volume | 3.41 | 93.2 | 317.8 |
| Chloride | 2.73 | 103 | 281.19 |
| Hemoglobin | 1.04 | 12.8 | 13.3 |
| Hematocrit | 0.62 | 33.2 | 20.58 |
| **Phenotype 4 (Abnormal vital signs)** | **Weight** | **Average Value** | **Weighted Average Value** |
| Glascow coma scale total | 2.13 | 6.7 | 14.271 |
| Oxygen saturation | 1.41 | 85 | 119.85 |
| Systolic blood pressure | 1.30 | 153.1 | 199.03 |
| Temperature | 1.29 | 37.5 | 48.37 |
| Heart rate | 1.03 | 115 | 118.45 |
| Mean blood pressure | 0.93 | 95 | 88.35 |
| Diastolic blood pressure | 0.84 | 82 | 68.88 |

### Interpretability analysis

Finally, we did an interpretability analysis of MULTIPAR on the MIMIC-EXTRACT dataset. We first illustrate the phenotypes discovered by MULTIPAR in table 4.4. We set rank to 4, and use the **V** matrix to select the most important vital signs in each phenotype based on the weight. **V** matrix represent the membership of medical features in each one of the phenotype, and the "weight" column in table 4.4 is the weight in the $V$ matrix. We then use the **S** matrix to find the patient subgroup of each phenotype, and calculate the average value of the vital signs shown in the "Average value" column, and the "Weight average value" column is calculated by weight multiplying average value. It is important to note that there is no post-processing in these extracted phenotypes. A critical care expert reviewed and endorsed the presented phenotypes which suggest collective characteristics such as normal vital signs,

Table 4.5: MIMIC-EXTRACT phenotypes discovered by SinglePAR incorporating in-hospital mortality prediction.

| **Phenotype 1** | |
|---|---|
| Oxygen saturation | Systolic blood pressure |
| Heart rate | PH |
| Mean blood pressure | Diastolic blood pressure |
| **Phenotype 2** | |
| Oxygen saturation | Systolic blood pressure |
| PH | Mean blood pressure |
| Heart rate | Diastolic blood pressure |
| **Phenotype 3** | |
| Temperature | Glascow coma scale total |
| Oxygen saturation | Systolic blood pressure |
| Heart rate | Mean blood pressure |
| **Phenotype 4** | |
| Glascow coma scale total | Heart rate |
| Temperature | Systolic blood pressure |
| Mean blood pressure | PH |

Table 4.6: MIMIC-EXTRACT phenotypes discovered by incorporating icu mortality prediction.

| **Phenotype 1** | |
|---|---|
| Oxygen saturation | Systolic blood pressure |
| Heart rate | respiratory rate |
| Mean blood pressure | Diastolic blood pressure |
| **Phenotype 2** | |
| Hemoglobin | PH |
| Sodium | chloride |
| Mean corpuscular volume | Co2 (etco2, pco2, etc.) |
| **Phenotype 3** | |
| Oxygen saturation | Systolic blood pressure |
| Heart rate | Respiratory rate |
| Mean blood pressure | Diastolic blood pressure |
| **Phenotype 4** | |
| Temperature | Glascow coma scale total |
| Oxygen saturation | Systolic blood pressure |
| Heart rate | Mean blood pressure |

abnormal renal and liver function, normal blood counts and serum electroytes, and abnormal vital signs.

The phenotypes discovered by the supervised single task model SinglePAR strongly overlap with each other shown in table 4.5, 4.6, 4.7, and 4.8. Since we are incorporating in-hospital mortality prediction task in table 4.5, most of the phenotypes discovered by SinglePAR are abnormal in vital signs. COPA discovered phenotypes shown in table 4.9 contain more information compared to SinglePAR, which makes sense because a supervised model may guide the tensor factorization to a specific direction geared toward the task and cause information loss. However, MULTIPAR does not

Table 4.7: MIMIC-EXTRACT phenotypes discovered by SinglePAR incorporating readmission prediction.

| **Phenotype 1** | |
|---|---|
| Temperature | Glascow coma scale total |
| Oxygen saturation | Systolic blood pressure |
| Heart rate | Mean blood pressure |
| **Phenotype 2** | |
| Oxygen saturation | Systolic blood pressure |
| Heart rate | Mean blood pressure |
| Diastolic blood pressure | Respiratory rate |
| **Phenotype 3** | |
| Sodium | Chloride |
| Oxygen saturation | PH |
| Hemoglobin | Hear rate |
| **Phenotype 4** | |
| Oxygen saturation | Systolic blood pressure |
| Heart rate | Mean blood pressure |
| Diastolic blood pressure | PH |

Table 4.8: MIMIC-EXTRACT phenotypes discovered by SinglePAR incorporating ventilation prediction

| **Phenotype 1** | |
|---|---|
| Oxygen saturation | Systolic blood pressure |
| Heart rate | Respiratory rate |
| Mean blood pressure | Diastolic blood pressure |
| **Phenotype 2** | |
| Respiratory rate | Sodium |
| Temperature | Mean corpuscular volume |
| Chloride | PH |
| **Phenotype 3** | |
| Oxygen saturation | Systolic blood pressure |
| Heart rate | Mean blood pressure |
| Diastolic blood pressure | Respiratory rate |
| **Phenotype 4** | |
| Temperature | Glascow coma scale total |
| Oxygen saturation | Diastolic blood pressure |
| Heart rate | Mean blood pressure |

have information loss compared to COPA, it even provides a new phenotype (phenotype 2: abnormal in renal and liver function) which is not discovered by COPA. This verifies the benefit of MTL in MULTIPAR, which can leverage information from multiple tasks to avoid local optimum.

We then test MULTIPAR's ability to find meaningful subgroup temporal trajectories, which can help clinical care experts make precise prescriptions and treatments for specific subgroup of patients. We select the patients with the number of observations equal to 22 for visualization purposes. The rank is set to 4. We select the four phenotypes for the temperature feature and systolic blood pressure feature, then use

Table 4.9: MIMIC-EXTRACT phenotypes discovered by COPA.

| | |
|---|---|
| **Phenotype 1** | |
| Sodium | Mean corpuscular volume |
| Mean corpuscular hemoglobin | Oxygen saturation |
| Mean corpuscular volume | Chloride |
| **Phenotype 2** | |
| Temperature | Oxygen saturation |
| Systolic blood pressure | Heart rate |
| Mean blood pressure | Diastolic blood pressure |
| **Phenotype 3** | |
| Glascow coma scale total | Temperature |
| Oxygen saturation | Systolic blood pressure |
| Heart rate | Mean blood pressure |
| **Phenotype 4** | |
| Oxygen saturation | Systolic blood pressure |
| Mean blood pressure | Diastolic blood pressure |
| Heart rate | Respiratory rate |

the **S** matrix to find the patient subgroup for each phenotype, and print the average value trajectory.

From figure 4.7a, we can see that the four patient subgroups (clusters) exhibit very different temporal trajectories in the temperature. Our clinical expert interpreted that the green line suggests a hyperthermic slow resolver patient subgroup which exhibits a slow decreasing trend as time increases, the red line suggests a hyperthermic fast resolver patient subgroup, which exhibits a fast decreasing trend as time increases, the dark blue line suggests a normothermic patient subgroup and the light blue line is a hypothermic patient subgroup. For the systolic blood pressure trajectory shown in figure 4.7b, the green subgroup has high, increasing blood pressure, the red subgroup has high, decreasing blood pressure, the dark blue and light blue subgroups have consistently normal and low-normal blood pressure, respectively.

# Chapter 5

# Conclusion

In this dissertation, we demonstrate three PARAFAC2 irregular tensor factorization methods for health data analysis that address the major limitations in current models. In chapter 2, we developed a robust unsupervised PARAFAC2 method to handle missing and erroneous EHR data. In chapter 3, we proposed an RNN regularized robust PARAFAC2 method for more accurate temporal modeling. In chapter 4, we built a supervised PARAFAC2 framework with multi-task learning for more meaningful phenotypes and better predictive accuracy. In the following of this Chapter, we specify future work directions focusing on EHR-based phenotyping, which is a central topic of this dissertation.

**Robust supervised multi-task learning.** The first future work direction could be to further enhance the robustness of the supervised multi-task learning framework. A possible approach is adding the low-rankness constraint on the factor matrices.

**Incorporating additional EHR data domains.** Our work has been mostly focusing on utilizing EHR structured code information. Incorporating clinical text using the (NLP) natural language processing technique is an additional target for our future work. Choosing an appropriate NLP model and scalability issue is the main challenge.

# Bibliography

[1] Evrim Acar, Daniel M Dunlavy, Tamara G Kolda, and Morten Mørup. Scalable tensor factorizations for incomplete data. *Chemometrics and Intelligent Laboratory Systems*, 2011.

[2] Ardavan Afshar, Ioakeim Perros, Evangelos E Papalexakis, Elizabeth Searles, Joyce Ho, and Jimeng Sun. Copa: Constrained parafac2 for sparse & large datasets. In *CIKM*, 2018.

[3] Heinz H Bauschke, Rafal Goebel, Yves Lucet, and Xianfu Wang. The proximal average: basic theory. *SIAM Journal on Optimization*, 19(2):766–785, 2008.

[4] Jonathan Baxter. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12, 05 2000. doi: 10.1613/jair.731.

[5] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 2011.

[6] Rasmus Bro, Claus A. Andersson, and Henk A. L. Kiers. Parafac2—part ii. modeling chromatographic data with retention time shifts. *Journal of Chemometrics*, 1999.

[7] Jacqueline Brothier, Peter Speltz, Luke Rasmussen, Melissa Basford, Omri Gottesman, Peggy Peissig, Jennifer Pacheco, Gerard Tromp, Jyotishman Pathak,

David Carrell, Stephen Ellis, Todd Lingren, Will Thompson, Guergana Savova, Jonathan Haines, Dan Roden, Paul Harris, and Joshua Denny. Phekb: A catalog and workflow for creating electronic phenotype algorithms for transportability. *Journal of the American Medical Informatics Association*, 23:ocv202, 03 2016. doi: 10.1093/jamia/ocv202.

[8] Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 2010.

[9] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *JACM*, 2011.

[10] J. Carroll and J. Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of "eckart-young" decomposition. *Psychometrika*, 35:283–319, 1970.

[11] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific Reports*, 8, 04 2018. doi: 10.1038/s41598-018-24271-9.

[12] Dongpeng Chen and Brian Mak. Multitask learning of deep neural networks for low-resource speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23:1–1, 07 2015. doi: 10.1109/TASLP.2015.2422573.

[13] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. 06 2014. doi: 10.3115/v1/D14-1179.

[14] Edward Choi, Taha Bahadori, and J. Sun. Doctor ai: Predicting clinical events via recurrent neural networks. In *Machine learning for healthcare conference*, pages 301–318, 2016.

[15] Roberto Cipolla, Yarin Gal, and Alex Kendall. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. pages 7482–7491, 06 2018. doi: 10.1109/CVPR.2018.00781.

[16] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. pages 160–167, 01 2008. doi: 10.1145/1390156.1390177.

[17] Pål Comstedt, Merete Storgaard, and Annmarie Lassen. The systemic inflammatory response syndrome (sirs) in acutely hospitalised medical patients: a cohort study. *Scandinavian journal of trauma, resuscitation and emergency medicine*, 17:67, 12 2009. doi: 10.1186/1757-7241-17-67.

[18] li Deng, Geoffrey Hinton, and Brian Kingsbury. New types of deep neural network learning for speech recognition and related applications: An overview. pages 8599–8603, 10 2013. doi: 10.1109/ICASSP.2013.6639344.

[19] Silvia Gandy, Benjamin Recht, and Isao Yamada. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems*, 2011.

[20] Ross B. Girshick. Fast r-cnn. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015.

[21] Donald Goldfarb and Zhiwei Qin. Robust low-rank tensor recovery: Models and algorithms. *SIAM Journal on Matrix Analysis and Applications*, 2014.

[22] R. Harshman. Foundations of the parafac procedure: Models and conditions for an ”explanatory” multi-model factor analysis. 1970.

[23] R. Harshman. Parafac2: Mathematical and technical notes. 1972.

[24] Hrayr Harutyunyan, Hrant Khachatrian, David Kale, and Aram Galstyan. Multitask learning and benchmarking with clinical time series data. *Scientific Data*, 6, 06 2019. doi: 10.1038/s41597-019-0103-9.

[25] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.

[26] F. L. Hitchcock. The expression of a tensor or a polyadic as a sum of products. *Journal of Mathematics and Physics*, 6:164–189.

[27] Joyce Ho, Joydeep Ghosh, Steve Steinhubl, Walter Stewart, Joshua Denny, Bradley Malin, and J. Sun. Limestone: High-throughput candidate phenotype generation via tensor factorization. *Journal of Biomedical Informatics*, 52, 12 2014. doi: 10.1016/j.jbi.2014.07.001.

[28] Joyce Ho, Joydeep Ghosh, and J. Sun. Marble: High-throughput phenotyping from electronic health records via sparse nonnegative tensor factorization. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 08 2014. doi: 10.1145/2623330.2623658.

[29] David Hong, Tamara G. Kolda, and Jed A. Duersch. Generalized canonical polyadic tensor decomposition. *SIAM Review*, 62(1):133–163, Jan 2020. ISSN 1095-7200. doi: 10.1137/18m1203626. URL `http://dx.doi.org/10.1137/18M1203626`.

[30] Kejun Huang, Nicholas D Sidiropoulos, and Athanasios P Liavas. A flexible and efficient algorithmic framework for constrained matrix and tensor factorization. *IEEE Transactions on Signal Processing*, 2016.

[31] Alistair Johnson, Tom Pollard, Lu Shen, Li-wei Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Celi, and Roger Mark.

Mimic-iii, a freely accessible critical care database. *Scientific Data*, 3:160035, 05 2016. doi: 10.1038/sdata.2016.35.

[32] U. Kang, Evangelos Papalexakis, Abhay Harpale, and Christos Faloutsos. Gigatensor: Scaling tensor analysis up by 100 times - algorithms and discoveries. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 08 2012. doi: 10.1145/2339530.2339583.

[33] Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver. Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering. pages 79–86, 01 2010. doi: 10.1145/1864708.1864727.

[34] Misha Kilmer and Carla Martin. Factorization strategies for third-order tensors. *Linear Algebra and Its Applications - LINEAR ALGEBRA APPL*, 435, 08 2011. doi: 10.1016/j.laa.2010.09.020.

[35] Misha Kilmer, Karen Braman, Ning Hao, and Randy Hoover. Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging. *SIAM Journal on Matrix Analysis and Applications*, 34, 01 2013. doi: 10.1137/110837711.

[36] Yejin Kim, Robert El-Kareh, Jimeng Sun, Hwanjo Yu, and Xiaoqian Jiang. Discriminative and distinct phenotyping by constrained tensor factorization. *Scientific Reports*, 2017.

[37] Yejin Kim, Robert El-Kareh, Jimeng Sun, Hwanjo Yu, and Xiaoqian Jiang. Discriminative and distinct phenotyping by constrained tensor factorization. *Scientific Reports*, 7, 12 2017. doi: 10.1038/s41598-017-01139-y.

[38] Tamara Kolda and Brett Bader. Tensor decompositions and applications. *SIAM Review*, 2009.

[39] Lukas Liebel and Marco Körner. Auxiliary tasks in multi-task learning. *CoRR*, abs/1805.06334, 2018. URL `http://arxiv.org/abs/1805.06334`.

[40] Yu-Ru Lin, J. Sun, Paul Castro, Ravi Konuru, Hari Sundaram, and Aisling Kelliher. Metafac: Community discovery via relational hypergraph factorization with propinquity dynamics. In *KDD*, pages 527–536, 2009. doi: 10.1145/1557019.1557080.

[41] Hanpeng Liu, Yaguang Li, Michael Tsang, and Yan Liu. Costco: A neural tensor completion model for sparse tensors. pages 324–334, 07 2019. ISBN 978-1-4503-6201-6. doi: 10.1145/3292500.3330881.

[42] Ji Liu, Przemyslaw Musialski, Peter Wonka, and Jieping Ye. Tensor completion for estimating missing values in visual data. *IEEE Transactions on pattern analysis and machine intelligence*, 2012.

[43] Qiang Liu, Shu Wu, LiangWang, and Tieniu Tan. Predicting the next location: A recurrent model with spatial and temporal contexts. In *AAAI*, pages 194–200, 2016.

[44] Shikun Liu, Edward Johns, and Andrew Davison. End-to-end multi-task learning with attention. 03 2018.

[45] Xiaofeng Liu, Zhaofeng Li, Lingsheng Kong, Zhihui Diao, Junliang Yan, Yang Zou, Chao Yang, Ping Jia, and Jane You. A joint optimization framework of low-dimensional projection and collaborative representation for discriminative classification. pages 1493–1498, 08 2018. doi: 10.1109/ICPR.2018.8545267.

[46] Yuanyuan Liu, Fanhua Shang, Licheng Jiao, James Cheng, and Hong Cheng. Trace norm regularized candecomp/parafac decomposition with missing data. *IEEE Transactions on Cybernetics*, 2015.

[47] Canyi Lu, Jiashi Feng, Shuicheng Yan, and Zhouchen Lin. A unified alternating direction method of multipliers by majorization minimization. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):527–541, 2017.

[48] Canyi Lu, Jiashi Feng, Wei Liu, Zhouchen Lin, Shuicheng Yan, et al. Tensor robust principal component analysis with a new tensor nuclear norm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[49] Cun Mu, Bo Huang, John Wright, and Donald Goldfarb. Square deal: Lower bounds and improved relaxations for tensor recovery. In *ICML*, 2014.

[50] Neal Parikh, Stephen Boyd, et al. Proximal algorithms. *Foundations and Trends® in Optimization*, 2014.

[51] Ioakeim Perros, Evangelos E Papalexakis, Fei Wang, Richard Vuduc, Elizabeth Searles, Michael Thompson, and Jimeng Sun. Spartan: Scalable parafac2 for large & sparse data. In *KDD*, 2017.

[52] Ioakeim Perros, Evangelos E. Papalexakis, Richard Vuduc, Elizabeth Searles, and Jimeng Sun. Temporal phenotyping of medically complex children via parafac2 tensor factorization. *Journal of Biomedical Informatics*, 2019.

[53] Dr. Yusuf Perwej and chaturvedi Ashish. Machine recognition of hand written characters using neural networks. *International Journal of Computer Applications*, 14, 01 2011. doi: 10.5120/1819-2380.

[54] Tom Pollard, Alistair Johnson, Jesse Raffa, Leo Celi, Roger Mark, and Omar Badawi. The eicu collaborative research database, a freely available multi-center database for critical care research. *Scientific Data*, 5:180178, 09 2018. doi: 10.1038/sdata.2018.178.

[55] P. Prasanna and Dr Rao. Text classification using artificial neural networks. *International Journal of Engineering and Technology(UAE)*, 7:603–606, 01 2018. doi: 10.14419/ijet.v7i1.1.10785.

[56] Sanjay Purushotham, Chuizheng Meng, Zhengping Che, and Yan Liu. Benchmark of deep learning models on large healthcare mimic datasets. *Journal of Biomedical Informatics*, 83, 10 2017. doi: 10.1016/j.jbi.2018.04.007.

[57] James Ramsay. Monotone regression splines in action. *Statistical Science*, 3, 11 1988. doi: 10.1214/ss/1177012761.

[58] James O Ramsay et al. Monotone regression splines in action. *Statistical science*, 3(4):425–441, 1988.

[59] Bharath Ramsundar, Steven Kearnes, Patrick Riley, Dale Webster, David Konerding, and Vijay Pande. Massively multitask networks for drug discovery. *arXiv:1502.02072*, 02 2015.

[60] Yifei Ren, Jian Lou, Li Xiong, and Joyce Ho. Robust irregular tensor factorization and completion for temporal health data analysis. In *CIKM*, pages 1295–1304, 2020. doi: 10.1145/3340531.3411982.

[61] Matthew Reyna, Christopher Josef, Russell Jeter, Supreeth Shashikumar, M Brandon Westover, Shamim Nemati, and Gari Clifford. Early prediction of sepsis from clinical data: The physionet/computing in cardiology challenge 2019. *Critical Care Medicine*, 48:1, 12 2019. doi: 10.1097/CCM.0000000000004145.

[62] Bernardino Romera-Paredes and Massimiliano Pontil. A new convex relaxation for tensor completion. In *NIPS*, 2013.

[63] M. Seetha, Iyyanki Muralikrishna, Bulusu Deekshatulu, B. Malleswari, and

P. HEGDE. Artificial neural networks and other methods of image classification. 4, 11 2007.

[64] N.D. Sidiropoulos, Lieven Lathauwer, Xiao Fu, Kejun Huang, Evangelos Papalexakis, and Christos Faloutsos. Tensor decomposition for signal processing and machine learning. *IEEE Transactions on Signal Processing*, PP, 07 2016. doi: 10.1109/TSP.2017.2690524.

[65] Conrad Snyder, Henry Law, and Peter Pamment. Calculation of tucker's three-mode common factor analysis. *Behavior Research Methods and Instrumentation*, 11:609–611, 11 1979. doi: 10.3758/BF03201400.

[66] Stephan Spiegel, Jan Clausen, Sahin Albayrak, and Jérôme Kunegis. Link prediction on evolving data using tensor factorization. In *2009 ICDM Workshops*, pages 100–110, 05 2011. ISBN 978-3-642-28319-2. doi: 10.1007/978-3-642-28320-8_9.

[67] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning, 08 2018.

[68] Sebastian Thrun. Is learning the n-th thing any easier than learning the first? *Advances in Neural Information Processing Systems (NIPS)*, 09 1999.

[69] Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura. Attention-based wav2text with feature transfer learning. pages 309–315, 12 2017. doi: 10.1109/ASRU.2017.8268951.

[70] Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, 3:1–13, 09 2009. doi: 10.4018/jdwm.2007070101.

[71] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 2008.

[72] Shirly Wang, Matthew B. A. McDermott, Geeticka Chauhan, Michael C. Hughes, Tristan Naumann, and Marzyeh Ghassemi. Mimic-extract: A data extraction, preprocessing, and representation pipeline for MIMIC-III. In *ACM CHIL*, page 222–235, 2020.

[73] Yichen Wang, Robert Chen, Joydeep Ghosh, Joshua C. Denny, Abel N. Kho, You Chen, Bradley A. Malin, and Jimeng Sun. Rubik: Knowledge guided tensor factorization and completion for health data analytics. In *KDD*, pages 1265–1274, 2015.

[74] Zhangyang Wang, Yingzhen Yang, Shiyu Chang, Jinyan(Leo) Li, and Simon Fong. A joint optimization framework of sparse coding and discriminative clustering. 06 2015.

[75] Kun Xie, Can Peng, Xin Wang, Gaogang Xie, and Jigang Wen. Accurate recovery of internet traffic data under dynamic measurements. In *IEEE INFOCOM*, 2017.

[76] Pranjul Yadav, Michael S. Steinbach, Vipin Kumar, and György J. Simon. Mining electronic health records: A survey. *CoRR*, abs/1702.03222, 2017. URL `http://arxiv.org/abs/1702.03222`.

[77] Qiang Yang, Yu Zhang, Wenyuan Dai, and Sinno Jialin Pan. *Transfer Learning*. Cambridge University Press, 2020. doi: 10.1017/9781139061773.

[78] Di Yao, Chao Zhang, Jianhui Huang, and Jingping Bi. Serm: A recurrent model for next location prediction in semantic trajectories. In *CIKM*, page 2411–2414, 2017.

[79] Kejing Yin, Dong Qian, Kwok-Wai Cheung, Benjamin Fung, and Jonathan Poon. Learning phenotypes and dynamic patient representations via rnn regularized collective non-negative tensor factorization. In *AAAI*, pages 1246–1253, 2019. doi: 10.1609/aaai.v33i01.33011246.

[80] Kejing Yin, Ardavan Afshar, Joyce Ho, Kwok-Wai Cheung, Chao Zhang, and Jimeng Sun. Logpar: Logistic parafac2 factorization for temporal binary data with missing values. In *KDD*, pages 1625–1635, 08 2020. doi: 10.1145/3394486.3403213.

[81] Yaoliang Yu. Better approximation and faster algorithm using the proximal average. In *NIPS*, pages 458–466, 2013.

[82] Min-Ling Zhang and Zhi-Hua Zhou. A review on multi-label learning algorithms. *Knowledge and Data Engineering, IEEE Transactions on*, 26:1819–1837, 08 2014. doi: 10.1109/TKDE.2013.39.

[83] Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, PP, 07 2017. doi: 10.1109/TKDE.2021.3070203.

[84] Yu Zhang and Qiang Yang. An overview of multi-task learning. *National Science Review*, 5:30–43, 01 2018. doi: 10.1093/nsr/nwx105.