MULTIPAR: Supervised Irregular Tensor Factorization with Multi-task Learning for Computational Phenotyping

Yifei Ren YIFEIREN13@GMAIL.COM
Emory University, United States

Jian Lou JIAN.LOU@HOIYING.NET
Zhejiang University, China

Li Xiong LXIONG@EMORY.EDU
Emory University, United States

Joyce C Ho OYCE.C.HO@EMORY.EDU
Emory University, United States

Xiaoqian Jiang XIAOQIAN.JIANG@UTH.TMC.EDU
Health Science Center of University of Texas, United States

Sivasubramanium Venkatraman Bhavani SIVASUBRAMANIUM.BHAVANI@EMORY.EDU
Emory University, United States

Abstract

Tensor factorization has received increasing interest due to its intrinsic ability to capture latent factors in multi-dimensional data with many applications including Electronic Health Records (EHR) mining. PARAFAC2 and its variants have been proposed to address irregular tensors where one of the tensor modes is not aligned, e.g., different patients in EHRs may have different length of records. PARAFAC2 has been successfully applied to EHRs for extracting meaningful medical concepts (phenotypes). Despite recent advancements, current models' predictability and interpretability are not satisfactory, which limits its utility for downstream analysis. In this paper, we propose MULTIPAR: a supervised irregular tensor factorization with multi-task learning for computational phenotyping. 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. We conduct extensive experiments on two real-world temporal EHR datasets to demonstrate that MULTIPAR is scalable and achieves better tensor fit with more meaningful subgroups and stronger predictive performance compared to existing state-of-the-art methods. The implementation of MULTIPAR is available[1].

Keywords: tensor factorization, electronic health records, PARAFAC2, multi-task learning

1. Introduction

Tensor factorization has received increasing interest due to its intrinsic ability to capture the multi-dimensional structure in the data. It has a wide range of applications including social network analysis Spiegel et al. (2011); Fernandes et al. (2021), health data mining Wang et al. (2015a); Afshar et al. (2019); Ren et al. (2020); Yin et al. (2020); Afshar et al. (2018); Yi et al. (2021); Meyers et al. (2022), recommender systems Karatzoglou et al. (2010), and signal processing Sidiropoulos et al. (2016). Canonical Polyadic (CP) Carroll and Chang (1970); Harshman (1970), Tucker Snyder et al. (1979), and tensor singular value decomposition (SVD) Kilmer et al. (2013); Kilmer and Martin (2011) are popular regular tensor factorization methods, where each mode of the tensor has a fixed size. However, in real-world cases, different people in recommender systems or patients in health data may have different lengths of records,

1. https://github.com/yifeiren13/MULTIPAR

which can not be handled by regular tensor factorization methods. PARAFAC2 Harshman (1972); Kiers et al. (1999) has been proposed for factorizing irregular tensors, where one of the mode sizes is not fixed. We introduce our motivating application for Electronic Health Records (EHRs) mining below for irregular tensor factorization.

EHRs are patient-centered records collected from a variety of institutions, hospitals, and pharmaceutical companies over a long period of time. EHR mining can significantly improve the ability to diagnose diseases and reduce or even prevent medical errors, thereby improving patient outcomes Yadav et al. (2017). However, directly using the raw, massive, high-dimensional, and longitudinal EHRs is challenging. For example, a single disease can consist of several heterogeneous subgroups yet be coded with the same diagnosis category, e.g., hypertension can be divided into hypertension resolver, hypertension, and prehypertension subgroups. Thus, researchers and healthcare practitioners seek to identify *phenotypes*, or disease subgroups, to better understand differences in biological mechanisms and treatment responses, which can lead to more effective and precise treatment.

The tensor factorization technique has been widely used to capture the multi-dimensional structure in EHRs for computational phenotyping Ho et al. (2014b,a); Wang et al. (2015a); Afshar et al. (2019); Ren et al. (2020); Yin et al. (2020); Afshar et al. (2018). Compared to traditional clustering-based approaches, tensor factorization can mine concise and potentially more interpretable latent information between multiple attributes (e.g., diagnosis and medications) in addition to clustering patients into subgroups. One key characteristic of EHR data is that different patients have different visit lengths, varying disease states, and varying time gaps between consecutive visits. Hence, PARAFAC2 has been applied to extract phenotypes from EHR data.

As shown in Figure 1, the EHR 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. 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. Figure 1 also illustrates the computational phenotyping process using PARAFAC2. Each slice of the irregular tensor X 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 R phenotypes. Each row of the 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.

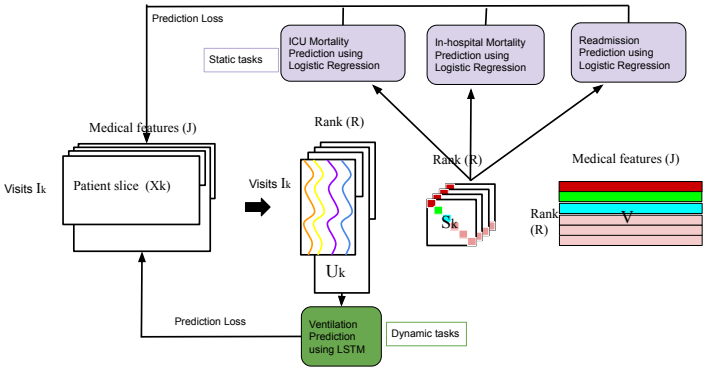


Figure 1: Overview of MULTIPAR on MIMIC-EXTRACT dataset

Various works have proposed improvements to the basic PARAFAC2 model. SPARTan Perros et al. (2017) modified the MTTKRP calculation order of PARAFAC2 to handle large and sparse data. COPA Afshar et al. (2018) further introduced various constraints to improve the interpretability of the factor matrices for more meaningful phenotype extraction. REPAIR Ren et al. (2020) and LogPar Yin et al. (2020) added a low-rankness constraint to improve PARAFAC2 robustness to missing data. Despite these advances, current PARAFAC2 models 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 Afshar et al. (2018); Yin et al. (2020); Ren et al. (2020); Perros et al. (2017); Kim et al. (2017b) 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.

To address these limitations, we propose MULTIPAR: a supervised irregular tensor factorization with multi-task learning for both phenotype extraction and predictive learning, as shown in Figure 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 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, as shown in Figure 1.

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. In addition, by sharing the representation across tasks, the learned phenotypes can generalize better for each task. 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.

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 an 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.

2. Background

2.1. Irregular Tensor Factorization

Definition 1 *(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. where* $\mathbf{Q}_k \in \mathbf{R}^{I_k \times R}$ *to be orthogonal,* $\mathbf{I}_k \in \mathbf{R}^{R \times R}$ *is the identity matrix and* R *is the target rank of the PARAFAC2 decomposition.*

Given a tensor representing the EHR data as in Figure 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.

2.2. Supervised and Multi-task learning Framework

Supervision can enhance traditional unsupervised tasks such as clustering for a wide range of applications, e.g., graph learning Wang et al. (2015b), pattern classification Liu et al. (2018b). Wang et al. proposed a supervised feature extraction framework using discriminative clustering to improve the

graph learning model's clustering accuracy Wang et al. (2015b). Liu et al. proposed a supervised minimum similarity projection framework using the lowest correlation representation to improve the pattern model's classification accuracy Liu et al. (2018b). Over the past years, multi-task learning (MTL) Zhang and Yang (2018, 2017) 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 Baxter (2000); Thrun (1999). MTL has been used successfully across all applications, from natural language processing Collobert and Weston (2008); Tsoumakas and Katakis (2009) and speech recognition Deng et al. (2013); Chen and Mak (2015) to computer vision Girshick (2015) and drug discovery Ramsundar et al. (2015); Harutyunyan et al. (2019). However, no current work has considered improving the predictability of tensor factorization using MTL.

3. Proposed Method

In this section, we present the MULTIPAR model in the context of EHR phenotyping and its optimization.

3.1. Problem Formulation

We formalize the objective function for the MULTIPAR model in Definition 2. Given an irregular tensor $\mathbf{X}$, our goal is to factorize it into U_k, S_k and V using supervised irregular tensor factorization to improve predictability and interpretability. Hence our objective function consists of several components. The PARAFAC2 loss for $\mathbf{X}$ ensures the reconstructed tensor closely approximates the original tensor. The static outcomes loss and dynamic outcomes loss are to ensure the phenotypes are supervised by the downstream prediction tasks, including both static and dynamic tasks. Static tasks have one-time or static labels, and dynamic tasks have continuously changing or temporal dynamic labels for each time stamp. An approximate uniqueness constraint ensures uniqueness of the solution of the tensor factorization. Finally, for EHR 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 including: non-negativity for $\mathbf{S}_k$, sparsity for $\mathbf{V}$. We explain each of the loss components and constraints in detail below.

Definition 2 *(MULTIPAR objective function)*

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

where $k = 1, ..., K$, $\mathbf{H}, \{\mathbf{S}_k\}, \mathbf{I} \in \mathbb{R}^{R\times R}$. c_1 is the non-negativity constraint, and c_2 is the sparsity penalty.

PARAFAC2 loss. The PARAFAC2 tensor factorization loss can ensure the reconstructed tensor closely approximates the original tensor. To accommodate different data types, the PARAFAC2 loss can be any smooth loss function, e.g., Least square loss, Poisson loss Hong et al. (2020) and Rayleigh Loss Hong et al. (2020).

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 denote it as a static outcome prediction task. For illustrative purposes, we use a logistic regression model on the $\mathbf{S}$ 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 at a given future time can be used to measure the model's predictability. For illustrative purposes, we use the long short-term memory (LSTM) model on the $\mathbf{U}$ 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 into the objective function.

Approximate uniqueness constraint. The optimization of the original PARAFAC2 model adopts the alternating direction method of multipliers (AO-ADMM) Roald et al. (2021), 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 $\mathbf{Q}_k^\top \mathbf{Q}_k = \mathbf{I}$. Similar to LogPar Yin et al. (2020), to optimize $\mathbf{Q}_k$, we relax the uniqueness constraint to $\|\mathbf{Q}_k^\top \mathbf{Q}_k - \mathbf{I}\|_F^2$.

Non-negativity on S. The diagonal matrix $\mathbf{S}$ indicates the importance of membership of patient k in each one of the R phenotypes. Since only non-negative membership values make sense, we zero out the negative values in $\mathbf{S}$.

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, or hard thresholding, will cap the number of non-zero values in a matrix. The l_1 regularization norm, or soft thresholding, will shrink matrix values towards zero. As hard thresholding is a non-convex optimization problem that can not be optimized by the SGD framework, we adopt soft thresholding, which is convex and can be migrated into the SGD optimization framework.

SDW: Smooth dynamic weight selection. Our objective function consists of several main losses from the tensor factorization and the multiple predictive tasks. Each of these losses is associated with a weight. 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. Hence a key challenge for our framework is how to tune the weights for different loss terms and tasks. We introduce a novel unified and dynamic weight selection method for weighing the tensor factorization and predictive tasks, where the tensor factorization is considered as one task, to achieve an overall optimized result.

The DWA weight selection Liu et al. (2018a) was proposed to dynamically change the task weights at each epoch by considering the rate of change of the loss over the epoch, however, 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. While Definition 2 shows ρ_1 as the weight for tensor loss, ρ_2 and ρ_3 as the accumulative weights for static and dynamic tasks respectively, here we use $Weight_n(t)$ to denote the weight for each individual task n in epoch t. 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:

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

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

$$Weight_n(t) := N \frac{exp(\sum_{j=1}^{m}(\omega_n(t-j)/C)/m)}{\sum_{i=1}^{N} exp(\sum_{j=1}^{m}(\omega_i(t-j)/C)/m)}. \tag{3}$$

Intuitively, each task weight is dynamically updated based on a "smoothed" descending rate of the loss, the higher the rate (i.e. the more the task contributes to the optimization objective), the higher the weight for the task in the next epoch. We use C to control the softness distribution between different tasks. If C is large enough, the weight for each task will be uniform. Different from Hinton et al. (2015), 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). The main rationale for this smoothing is 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.

Algorithm 1 Optimization Framework for MULTIPAR

Require: Input tensor $\mathcal{X}$; Model parameters ρ_1-ρ_3, ϱ_1-ϱ_2; Interpretability constraint types c_1, c_2; Initial rank estimation R.

1: **while** Not reach convergence criteria **do**
2: Update $\{\mathbf{U}_k\}$ using eq.(5) by SGD;
3: Update $\{\mathbf{Q}_k\}$ using eq.(6) by SGD;
4: Update $\mathbf{H}$ using eq.(7) by SGD;
5: Update $\mathbf{S}_k$ using eq.(8) by Proximal/Projected SGD;
6: Update $\mathbf{V}$ using eq.(10) by Proximal/Projected SGD;
7: Calculate weight for each prediction task using eq. 2 and eq.3 by SDW;
8: **end while**

Ensure: Phenotype factor matrices $\{\mathbf{U}_k\} = \{\mathbf{Q}_k\mathbf{H}\}, \{\mathbf{S}_k\}, \mathbf{V}$.

3.2. Optimization

To solve the optimization problem in Eq. (1), MULTIPAR follows an alternative optimization strategy where we optimize one variable individually with all other variables fixed. According to whether the subproblem is differentiable, we group the variables into two groups: pure smooth subproblems which can be directly solved by SGD, and proximal mapping-based smooth subproblems Parikh and Boyd (2014). The proximal map is a key building block for optimizing nonsmooth regularized objective functions, e.g., the $\|\cdot\|_1$ ℓ_1-norm regularization function for inducing sparsity and $\|\cdot\|_*$ nuclear norm regularization for inducing low-rankness. We present the optimization details and convergence analysis in the appendix 6.1.

4. Experiment

4.1. Dataset

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

eICU[2] **Pollard et al. (2018):** 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. We select 202 diagnosis codes that have the highest frequency, as in Kim et al. (2017a). 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 task indicating whether a patient needs to be ventilated at the time of the worst respiratory rate, we will use "vent-res" as the name for this task.

MIMIC-EXTRACT[3] **Wang et al. (2020):** 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 for every visit.

4.2. Evaluation Metrics

In order to test the tensor reconstruction quality of MULTIPAR model, we adopt the $FIT \in (-\infty, 1]$ score Bro et al. (1999) as the quality measure (the higher the better):

$$FIT = 1 - \frac{\sum_{k=1}^{K} \|\mathbf{X}_k - \mathbf{U}_k\mathbf{S}_k\mathbf{V}^T\|^2}{\sum_{k=1}^{K} \|\mathbf{X}_k\|^2}. \quad (4)$$

The original tensor, denoted as $\{\mathbf{X}_k\}$, 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.

4.3. Methods for Comparison

We compare MULTIPAR with three baseline methods: SPARTan, COPA (two state-of-the-art irregular tensor factorization methods) and singlePAR (a single-task version of MULTIPAR).

- **SPARTan Perros et al. (2017) - scalable PARAFAC2**: A tensor factorization method

2. https://eicu-crd.mit.edu

3. https://github.com/MLforHealth/MIMIC_Extract/

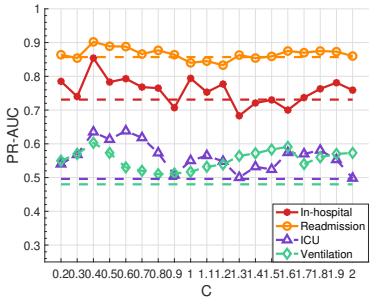


Figure 2: PR-AUC score using different C

for fitting large and sparse irregular tensor data. It only considers the tensor reconstruction loss.

- **COPA Afshar et al. (2018) - scalable PARAFAC2 with additional regularizations**: An irregular tensor factorization method that introduces various constraints/regularizations to improve the interpretability of the factor matrices. 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). The weight of tensor factorization and prediction tasks are also tuned using SDW.

4.4. Implementation Details

The multi-task dynamic weight selection of MULTIPAR 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 dataset under different ranks. Figure 2 shows the MIMIC-EXTRACT dataset result when rank $= 50$ and m is fixed to 5. The dashed line is the PR-AUC when all the tasks have equal weight. C is set to $\frac{1}{\sqrt{N}}$ in the remaining experiments to achieve the best result.

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

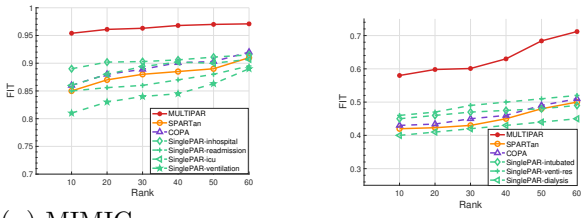
	m=1	m=3	m=5	m=8	m=10
In-hospital mortality prediction task	0.740	0.789	0.854	0.783	0.768
Readmission prediction task	0.872	0.893	0.902	0.892	0.853
ICU mortality prediction task	0.626	0.638	0.635	0.583	0.571
Ventilation prediction task	0.600	0.605	0.603	0.591	0.587
Convergence epoch	200	187	98	110	150

Table 1: Experiment result of PR-AUC and convergence epochs when m varies

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 $m = 5$. The result is shown in Table 1.

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.



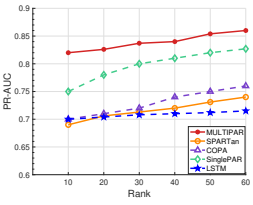
(a) MIMIC-EXTRACT (b) eICU

Figure 3: FIT score on MIMIC-EXTRACT and eICU dataset

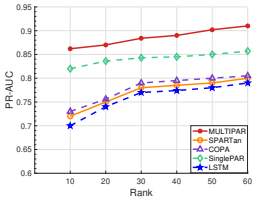
First, we compare MULTIPAR's FIT with the baseline models on two datasets shown in Figure 3. MULTIPAR optimizes all prediction tasks and tensor factorization together. SPARTan and COPA first finish the tensor factorization, and then predict the downstream prediction tasks. SinglePAR optimizes the evaluated task and tensor factorization together. As Figure 3(a) and 3(b) shows, MULTIPAR outperforms all 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 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.

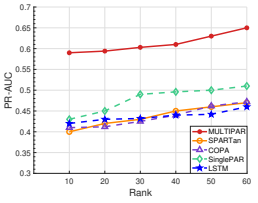
SinglePAR performs better than SPARTan and COPA on most of the ranks but is left behind COPA on large ranks. SinglePAR jointly optimizes prediction task and tensor factorization together. We can see that certain tasks benefit the tensor FIT while others may guide the tensor factorization into a suboptimal 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 for tensor reconstruction.



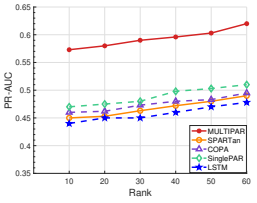
(a) In-hospital Mortality Prediction



(b) Readmission Prediction



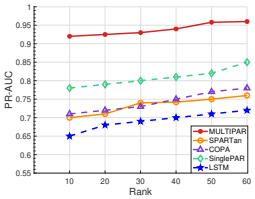
(c) ICU Mortality Prediction



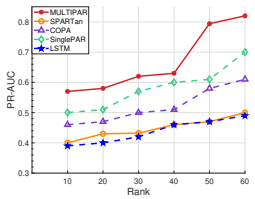
(d) Ventilation Prediction

Figure 4: PR-AUC for prediction tasks on MIMIC-EXTRACT

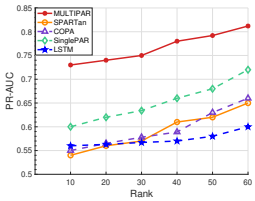
Predictability analysis. A logistic regression model is trained on the patient importance membership matrix $\mathbf{S}_k$ for static outcome prediction tasks and an LSTM model is trained on the temporal evolution matrix $\mathbf{U}_k$ for dynamic outcome prediction task. LSTM is a variant of the recurrent neural network



(a) Dialysis Prediction



(b) Venti-res Prediction



(c) Intubated Prediction

Figure 5: PR-AUC for prediction tasks on eICU dataset

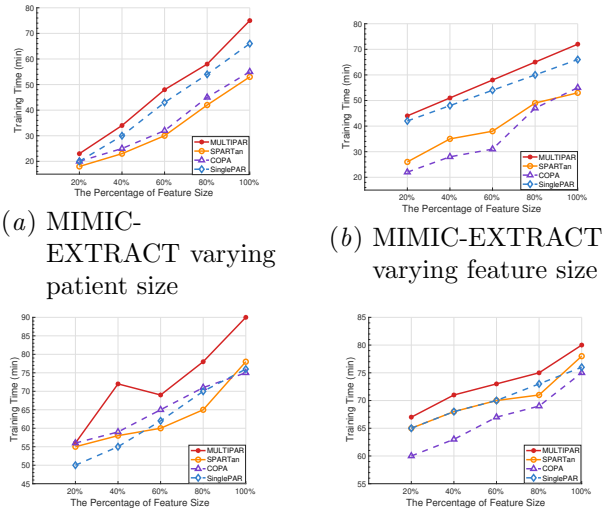
(RNN) that mitigates the gradient vanishing problem in traditional RNNs.

In the MIMIC-EXTRACT dataset, only the ventilation prediction task is a dynamic task, and all the tasks in the eICU dataset are static tasks. In order to illustrate the benefit of using the latent factors as features for a downstream prediction model, we also include an LSTM model trained using the original EHR data. The input to the LSTM model is an irregular tensor which contains k different patients, 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 and different visit for the dynamic task.

We evaluate the prediction accuracy as a function of the tensor factorization rank. As shown in Figures 4 and 5, MULTIPAR outperforms the other methods. In Figure 4, when the rank is 10, MULTIPAR outperforms the best baseline methods SinglePAR by 17%, 18%, 20% and 22% 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 of the extracted phenotypes. Moreover, SinglePAR always outperforms COPA, SPARTan, and LSTM, which shows that the supervised learning framework can improve predictability. The figures also illustrate 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 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 train 50 epochs of both methods. Although MULTIPAR adds MTL, it still exhibits linear scalability similar to SinglePAR. While MTL adds some additional training time, it is not significantly more as the maximum added time is 8 minutes. Moreover, the experiment result is consistent with our computational complexity analysis, the per-iteration computational complexity is linear with respect to number of patients J and feature size K. The results can be found in Figure 6.



(a) MIMIC-EXTRACT varying patient size

(b) MIMIC-EXTRACT varying feature size

(c) eICU varying patient size

(d) eICU varying feature size

Figure 6: Training time on MIMIC-EXTRACT and eICU with varying patient size and feature size

Interpretability analysis. Finally, we did an interpretability analysis of MULTIPAR on the MIMIC-EXTRACT dataset. We first illustrate the phenotypes discovered by MULTIPAR in Appendix 6.3. 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, abnormal renal and liver function, normal blood counts and serum electrolytes, and abnormal vital signs. We then test MULTIPAR's ability to find meaningful subgroup temporal trajectories in Appendix 6.3, which can help clinical care experts make precise prescriptions and treatments for specific subgroups of patients.

5. Conclusion

We proposed MULTIPAR: a supervised irregular tensor factorization with multi-task learning to jointly optimize the tensor factorization and multiple downstream prediction tasks. It is built on three major contributions: a supervised framework for PARAFAC2 tensor factorization and downstream prediction tasks; a new multi-task learning method combining tensor factorization and multiple prediction models; and a novel weight selection method for supervised multi-task optimization. We conducted extensive experiments and demonstrated that MULTIPAR can extract more meaningful phenotypes from EHR data with higher predictability for downstream tasks compared to state-of-the-art methods in a scalable way. In the future, we plan to incorporate low-rankness constraints for robustness against missing data, incorporate more sophisticated regularization constraints to capture the complex and nonlinear temporal relationships, and conduct more thorough and larger scale interpretability analysis.

Acknowledgement

This research is supported by National Science Foundation (NSF) under CNS-2124104, HCC-2302968, and National Institutes of Health (NIH) under R01LM013712, R01ES033241, UL1TR002378.

References

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

Ardavan Afshar, Ioakeim Perros, Joyce Ho, J. Sun, Walter Stewart, Haesun Park, Christopher DeFilippi, and Sherry Yan. Taste: Temporal and static tensor factorization for phenotyping electronic health records. 11 2019. doi: 10.1145/3368555.3384464.

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

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

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.

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.

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.

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.

Sofia Fernandes, Hadi Fanaee-T, and João Gama. Tensor decomposition for analysing time-evolving social networks: an overview. *Artificial Intelligence Review*, 54:1–26, 04 2021. doi: 10.1007/s10462-020-09916-4.

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

R. Harshman. Foundations of the parafac procedure: Models and conditions for an "explanatory" multimodel factor analysis. 1970.

R. Harshman. Parafac2: Mathematical and technical notes. 1972.

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.

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

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 2014a. doi: 10.1016/j.jbi.2014.07.001.

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 2014b. doi: 10.1145/2623330.2623658.

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.

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.

Henk Kiers, Jos Berge, and Rasmus Bro. Parafac2—part i. a direct fitting algorithm for the parafac2 model. *Journal of Chemometrics*, 13:275–294, 05 1999. doi: 10.1002/(SICI)1099-128X(199905/08)13:3/43.3.CO;2-2.

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.

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.

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

Yejin Kim, Robert El-Kareh, Jimeng Sun, Hwanjo Yu, and Xiaoqian Jiang. Discriminative and distinct phenotyping by constrained tensor factor-

ization. *Scientific Reports*, 7, 12 2017b. doi: 10.1038/s41598-017-01139-y.

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

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 2018b. doi: 10.1109/ICPR.2018.8545267.

Benjamin Meyers, Vincent Lee, Lauren Dennis, Julia Wallace, Vanessa Schmithorst, Jodie Votava-Smith, Vidya Rajagopalan, Elizabeth Herrup, Tracy Cook-Baust, Nhu Tran, Jill Hunter, Daniel Licht, J. Gaynor, Dean Andropoulos, Ashok Panigrahy, and Rafael Ceschin. Harmonization of multicenter diffusion tensor tractography in neonates with congenital heart disease: Optimizing post-processing and application of combat. *Neuroimage: Reports*, 2:100114, 09 2022. doi: 10.1016/j.ynirp.2022.100114.

Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends in optimization*, 1(3):127–239, 2014.

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.

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

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

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.

Marie Roald, Carla Schenker, Rasmus Bro, Jérémy Cohen, and Evrim Acar. An ao-admm approach to constraining parafac2 on all modes. 10 2021.

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.

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.

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.

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

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.

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.

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, 2015a.

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

Yangyang Xu and Wotao Yin. Block stochastic gradient iteration for convex and nonconvex optimization. *SIAM Journal on Optimization*, 25(3):1686–1716, 2015.

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.

Zheyuan Yi, Yilong Liu, Yujiao Zhao, Linfang Xiao, Alex Leong, Yanqiu Feng, and Fei Chen. Joint calibrationless reconstruction of highly undersampled multicontrast mr datasets using a low-rank hankel tensor completion framework. *Magnetic resonance in medicine*, 85, 02 2021. doi: 10.1002/mrm.28674.

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.

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.

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

6. First Appendix

6.1. Optimization

To solve the optimization problem in Eq. (1), MULTIPAR follows an alternative optimization strategy where we optimize one variable individually with all other variables fixed. According to the subproblem differentiable, we group the variables into two groups: pure smooth subproblems which can be directly solved by SGD and proximal mapping-based smooth subproblems Parikh and Boyd (2014). Proximal map is a key building block for optimizing nonsmooth regularized objective functions, e.g., the $\|\cdot\|_1$ ℓ_1-norm regularization function for inducing sparsity and $\|\cdot\|_*$ nuclear norm regularization for inducing low-rankness. In the following, we omit the iteration number for brevity in notation.

6.1.1. 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

$$\begin{aligned}\arg\min_{\mathbf{U}_k} &\sum_{(i,j)\in\Omega} \rho_1 L(\mathbf{X}_{ijk}, \{\mathbf{U}_k\mathbf{S}_k\mathbf{V}^\top\}_{ijk}) \\ &+ \varrho_1\|\mathbf{U}_k - \mathbf{Q}_k\mathbf{H}\|_F^2 + \rho_3 L_3(\mathbf{U}_k).\end{aligned} \tag{5}$$

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

$$\arg\min_{\mathbf{Q}_k} \varrho_1\|\mathbf{U}_k - \mathbf{Q}_k\mathbf{H}\|_F^2 + \varrho_2\|\mathbf{Q}_k^\top\mathbf{Q}_k - \mathbf{I}\|_F^2. \tag{6}$$

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

$$\arg\min_{\mathbf{H}} \sum_{k=1}^{K} \|\mathbf{U}_k - \mathbf{Q}_k\mathbf{H}\}\|_F^2 \tag{7}$$

6.1.2. Proximal Mapping-base Smooth Subproblems Updates

For the nonsmooth subproblems, we propose a proximal mapping-based Parikh and Boyd (2014) update, which include the following two parts.

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

$$\begin{aligned}\arg\min_{\mathbf{S}_k} &\sum_{(i,j)\in\Omega} \rho_1 L(\mathbf{X}_{ijk}, \{\mathbf{U}_k\mathbf{S}_k\mathbf{V}^\top\}_{ijk}) + \rho_2 L_2(\mathbf{S}_k) \\ &+ c_1(\mathbf{S}_k).\end{aligned} \tag{8}$$

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

$$\mathbf{S}_k = \max(0, \mathbf{S} - \lambda\mathbf{G}[\mathbf{S}_k]), \tag{9}$$

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

$$\arg\min_{\mathbf{V}} \sum_{k=1}^{K} \sum_{(i,j)\in\Omega} \rho_1 L(\mathbf{X}_{ijk}, \{\mathbf{U}_k\mathbf{S}_k\mathbf{V}^\top\}_{ijk}) + c_2\|\mathbf{V}\|_1. \tag{10}$$

We use soft-thresholding operator to update $\mathbf{V}$, where each step takes the following form: $\texttt{soft}-\texttt{thresholding}(\mathbf{V} - \lambda\mathbf{G}[\mathbf{V}]) = sign((\mathbf{V} - \lambda\mathbf{G}[\mathbf{V}]))((\mathbf{V} - \lambda\mathbf{G}[\mathbf{V}]) - \frac{c_2}{\lambda})$, where λ

is the step-size and $G[\mathbf{V}]$ 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 1.

6.2. Complexity and Convergence Analysis

The following theorem summarizes the computational complexity of Algorithm 1.

Theorem 3 *(Per-iteration computational complexity of MULTIPAR algorithm) For an input tensor $\mathbf{O}_k : \mathbb{R}^{I_k\times J}$, for $k = 1, ..., K$ and initial target rank estimation R, Algorithm 1's per-iteration complexity is $\mathcal{O}(3R^2JK)$.*

Proof MULTIPAR's per-iteration complexity breaks down as follows: Line 2 costs $\mathcal{O}(R(R+J+K))$; Line 3 costs $\mathcal{O}(\min\{R^2I, RI^2\})$, where I denotes the maximum among $\{I_k\}$; Line 4,5,6 cost $\mathcal{O}(R^2(R + J + K))$. As a result, the per-iteration complexity is $\mathcal{O}(3R^2JK)$. ■

Theorem 4 *(Convergence Rate of MULTIPAR algorithm) Let $\Phi[t] = (\mathbf{Q}[t], \mathbf{H}[t], \mathbf{S}[t], \mathbf{V}[t])$ be the iterates of MULTIPAR at iteration t, and $\mathcal{L} = L_1 + L_2 + L_3$ be the loss function of MULTIPAR. Under certain conditions, MULTIPAR will converge in terms of $\lim_{t\mapsto+\infty}\mathbb{E}[\|\mathcal{L}(\nabla\mathcal{L}(\Phi[t]))\|] = 0$.*

Proof *(Sketch of proof) MULTIPAR is a multi-block SGD algorithm for nonconvex optimization in nature, which has been extensively studied in optimization literature. The above theorem follows the convergence results established in Xu and Yin (2015).* ■

6.3. 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 2. We set rank to 4, and use the $\mathbf{V}$ matrix to select the most important vital signs in each phenotype based on the weight. $\mathbf{V}$ matrix represents the membership of medical features in each phenotype, and the "weight" column in Table 2 is the weight in the V matrix. We then use the $\mathbf{S}$ matrix to find the patient subgroup of each phenotype, and calculate the average value of the vital signs shown in the "Average value" column. 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, abnormal renal and liver function, normal blood counts and serum electrolytes, and abnormal vital signs.

Phenotype 1 (Normal vital signs)	**Weight**	**Average Value**
Oxygen saturation	1.52	98.5
Systolic blood pressure	0.91	112.7
Heart rate	0.82	82.5
Mean blood pressure	0.79	81.2
Diastolic blood pressure	0.65	76.3
Respiratory rate	0.57	18.6
Co2 (etco2, pco2, etc.)	0.43	24.2
Phenotype 2 (Abnormal renal and liver function)	**Weight**	**Average Value**
Alanine aminotransferase	11.51	83.1
Blood urea nitrogen	9.64	42.3
Alkaline phosphate	8.01	153.2
Asparate aminotransferase	5.18	90.1
Albumin	3.90	3.2
Bicarbonate	2.76	17
Mean blood pressure	1.59	85
Phenotype 3 (Normal Blood Counts and Serum Electrolytes)	**Weight**	**Average Value**
Mean corpuscular hemoglobin concentration	7.54	32.1
Sodium	4.93	135.2
Mean corpuscular hemoglobin	3.62	30.8
Mean corpuscular volume	3.41	93.2
Chloride	2.73	103
Hemoglobin	1.04	12.8
Hematocrit	0.62	33.2
Phenotype 4 (Abnormal vital signs)	**Weight**	**Average Value**
Glascow coma scale total	2.13	6.7
Oxygen saturation	1.41	85
Systolic blood pressure	1.30	153.1
Temperature	1.29	37.5
Heart rate	1.03	115
Mean blood pressure	0.93	95
Diastolic blood pressure	0.84	82

Table 2: Phenotypes discovered by MULTIPAR.

The phenotypes discovered by the supervised single task model SinglePAR strongly overlap with each other as shown in Tables 3, 4, 5, and 6. Since we are incorporating in-hospital mortality prediction task in Table 3, most of the phenotypes discovered by SinglePAR are abnormal in vital signs. COPA discovered phenotypes shown in Table 7 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 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.

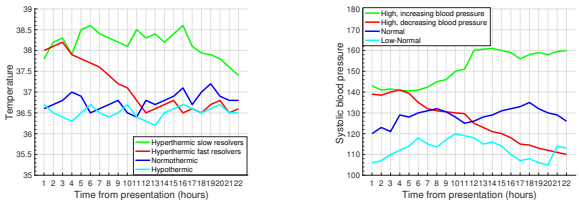
We then test MULTIPAR's ability to find meaningful subgroup temporal trajectories, which can help clinical care experts make precise prescriptions and treatments for specific subgroup of patients. We select the patients with the number of observations

Table 3: 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

equal to 22 for visualization purposes. We select the four phenotypes for the temperature feature and systolic blood pressure feature, then use the **S** matrix to find the patient subgroup for each phenotype, and print the average value trajectory.

From Figure 7, 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 7(b), 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.



(a) Temperature (b) Blood pressure

Figure 7: Temporal Trajectory

Table 4: 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

Table 5: 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 6: 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

Table 7: MIMIC-EXTRACT phenotypes discovered by COPA.

Phenotype 1	
Mean corpuscular hemoglobin concentration	Sodium
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