Machine Learning (2024) 113:5939–5980
https://doi.org/10.1007/s10994-024-06545-8

SWoTTeD: an extension of tensor decomposition to temporal phenotyping

Hana Sebia[1] · Thomas Guyet[1] · Etienne Audureau[2,3]

Received: 3 October 2023 / Revised: 28 March 2024 / Accepted: 30 March 2024 /
Published online: 30 April 2024

Abstract

Tensor decomposition has recently been gaining attention in the machine learning community for the analysis of individual traces, such as Electronic Health Records. However, this task becomes significantly more difficult when the data follows complex temporal patterns. This paper introduces the notion of a temporal phenotype as an arrangement of features over time and it proposes SWoTTeD (**S**liding **W**ind**o**w for **T**emporal **Te**nsor **D**ecomposition), a novel method to discover hidden temporal patterns. SWoTTeD integrates several constraints and regularizations to enhance the interpretability of the extracted phenotypes. We validate our proposal using both synthetic and real-world datasets, and we present an original usecase using data from the Greater Paris University Hospital. The results show that SWoTTeD achieves at least as accurate reconstruction as recent state-of-the-art tensor decomposition models, and extracts temporal phenotypes that are meaningful for clinicians.

Keywords Tensor decomposition · Unsupervised learning · Temporal data · Health data

Editor: Hendrik Blockeel.

✉ Thomas Guyet
thomas.guyet@inria.fr

Hana Sebia
hana.sebia@inria.fr

Etienne Audureau
etienne.audureau@aphp.fr

[1] AIstroSight, Inria, Hospices Civils de Lyon, Université Claude Bernard Lyon 1, 56 Bld Niels Bohr, Villeurbanne 69603, France

[2] CEpiA Team, INSERM U955, IMRB, Univ Paris Est Creteil, Créteil, France

[3] Public Health Department, AP-HP (Assistance Publique-Hôpitaux de Paris), Henri Mondor University Hospital, Créteil, France

1 Introduction

A tensor is a natural representation for multidimensional data. Tensor decomposition is a historic statistical tool for analyzing such complex data. The popularization of efficient and scalable machine learning techniques has made them attractive for real-world data (Perros et al., 2017). It has therefore been successfully investigated in a number of areas, such as signal processing, chemometrics, neuroscience, communication or psychometrics (Fanaee-T and Gama, 2016). Technically, tensor decomposition simplifies a multidimensional tensor into simpler tensors by learning latent variables in an unsupervised fashion (Anandkumar et al., 2014). Latent variables are unobserved features that capture hidden behaviors of a system. Such variables are difficult to extract from complex multidimensional data due to (1) multiple interactions between dimensions and (2) intertwined occurrences of hidden behaviors.

Recently, several approaches based on tensor decomposition have shown their effectiveness and their interest for computational phenotyping from Electronic Health Records (EHR) (Becker et al., 2023). The hidden recurrent patterns that are discovered in these data are called *phenotypes*. These phenotypes are of particular interest to (1) describe the real practices of medical units and (2) support hospital administrators to improve their care management. For example, a better description of care pathways of COVID-19 patients at the beginning of the pandemic may help clinicians to improve care management of future epidemic waves. This example motivates to apply such data analytic tools on a cohort of patients from the Greater Paris University Hospitals (see the case study in Sect. 7).

The main limitation of existing tensor decomposition techniques is the definition of a phenotype as a mixture of medical events without considering the temporal dimension. This means that all events occur at the same time. In this case, a care pathway is viewed as a succession of independent daily cares. Nonetheless, it seems more realistic to interpret a care pathway as mixtures of *treatments*, *i.e.* sequences of cares. For example, COVID-19 patients hospitalized with acute respiratory distress syndrome are treated for several problems during the same visit: viral infection, respiratory syndromes and hemodynamic problems. On the one hand, a treatment of the viral infection involves the administration of drugs for several days. On the other hand, the acute respiratory syndrome also requires continuous monitoring for several days. A patient's care pathway can then be abstracted as a mixture of these treatments. Some approaches proposed to capture the temporal dependencies between daily phenotypes using temporal regularization (Yin et al., 2019) but the knowledge provided to the clinician are still daily phenotypes.

In this article, we present SWoTTeD (**S**liding **W**ind**o**w for **T**emporal **Te**nsor **D**ecomposition), a tensor decomposition technique based on machine learning to extract temporal phenotypes. Contrary to a classical daily phenotype, a temporal phenotype describes the arrangement of drugs/procedures over a time window of several days. Drawing a parallel with sequential pattern mining, the state-of-the-art methods extract itemsets from sequences while SWoTTeD extracts sub-sequences. Thus, temporal phenotyping significantly enhances the expressivity of computational phenotyping. Following the principle of tensor decomposition, SWoTTeD discovers temporal phenotypes that accurately reconstruct an input tensor with a time dimension. It allows the overlapping of distinct occurrences of phenotypes to represent asynchronous starts of treatments. To the best of our knowledge, SWoTTeD is the first extension of tensor decomposition to temporal phenotyping. We evaluate the proposed model using both synthetic and real-world data. The results show that SWoTTeD outperforms the state-of-the-art tensor decomposition models in terms of reconstruction accuracy and noise robustness. Furthermore, the qualitative analysis shows that the discovered phenotypes are clinically meaningful.

In summary, our main contributions are as follows:

1. We extend the definition of tensor decomposition to temporal tensor decomposition. To the best of our knowledge, this is the first extension of tensor factorization that is capable of extracting temporal patterns. A comprehensive review is provided to position our proposal within the existing approaches in different fields of machine learning.
2. We propose a new framework, denoted as SWoTTeD, for extracting temporal phenotypes through the resolution of an optimization problem. This model also introduces a novel regularization term that enhances the quality of the extracted phenotypes. SWoTTeD has been extensively tested on synthetic and real-world datasets to provide insights into its competitive advantages. Additionally, we offer an open-source, well-documented, and efficient implementation of our model.
3. We demonstrate the utility of temporal phenotypes through a real-world case study.

The remainder of the article is organized as follows: the next section presents the state of the art of machine learning techniques related to tensor decomposition in the specific case of temporal tensor. Section 3 introduces the new problem of temporal phenotyping, then Sect. 4 presents SWoTTeD to solve it. The evaluation of this model is detailed in three sections. We begin by introducing the experimental setup in Sect. 5, followed by the presentation of reproducible experiments and results conducted on synthetic and real-world datasets. Lastly, Sect. 7 presents a case study on a COVID-19 dataset.

2 Related work

Discovering hidden patterns, a.k.a *phenotypes*[1] in our work, from longitudinal data is a fundamental issue of data analysis. This problem has been more especially investigated for the analysis of EHR data which are complex and require to be explored to discover hidden patterns providing insights about patient cares. With this objective, tensor decomposition has been widely used and proven to extract concise and interpretable patterns (Becker et al., 2023).

In this related work, we enlarge the scope and also review different machine learning techniques that have been recently proposed to address the problem of patient phenotyping. As our proposal is based on the principles of tensor decomposition, we start by reviewing techniques derived from tensor decomposition and that have been applied to EHR. Then, we present methods that targeted a similar objective, but with alternative modeling techniques. They both share the task of uncovering hidden patterns in temporal sequences using unsupervised methods.

Notations

In the remaining of this article, $[K]=\{1,\dots,K\}$ denotes the set of the K first non-zero integers. $\mathbb{N}^*$ denotes the strictly positive natural numbers. Curvy capital letters ($\mathcal{X}$) denote

[1] The term of phenotype usually denotes a set of traits that characterizes a disease. This notion is here extended to traits observed through the EHR systems. Thus, a phenotype is a set of observations in EHR data characterizing a treatment or a disease.

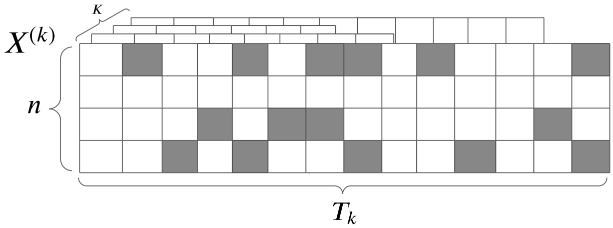


Fig. 1 Illustration of an irregular tensor $\mathcal{X} = \{X^{(k)}\}_{k\in[K]}$ representing a collection of K patients stays. Each patient has its own duration T_k but share the same set of cares (in rows). A black cell at position (i, t) (*i.e.* $x_{i,t}^{(k)} = 1$) indicates that the i-th care occurs at the time t

tensors (or irregular tensors), bold capital letters ($\boldsymbol{X}$) are matrices, bold lowercase letters ($\boldsymbol{x}$) are vectors and lowercase letters (x) are scalars.

2.1 Tensor decomposition from temporal EHR data

An EHR dataset has a timed event-based structure described at least by three dimensions: patient identifiers, care events (procedures, lab tests, drug administrations, etc) and time. We begin by introducing the tensor-based representation of EHR data. Following that, we provide a comprehensive review of various techniques designed to tackle the challenge of patient phenotyping using this data representation.

2.1.1 Tensor based representation of temporal EHR data

Considering that time is discrete (*e.g.*, events are associated with a specific day during the patient's stay), each patient $k \in [K]$ is represented by a matrix $\boldsymbol{X}^{(k)}$ where the first dimension represents the type of events and the second one represents time. If patient k received a care event i at time t, then $x_{i,t}^{(k)}$ is a non-zero value. In the majority of cases, values are categorical, typically represented as boolean values (0 or 1). However, there are cases where values may be integers or real numbers, such as the count of drugs administered or a measurement of a biophysiological parameter.

Additionally, if we consider that all patients have the same length of stay, the set of patients is a regular three-dimensional tensor $\mathcal{X}$, *i.e.* a data-cube. Nevertheless, in practice, patients' stays do not have the same duration. It ensues that each matrix $\boldsymbol{X}^{(k)}$ has its own temporal size, noted T_k. In this case, the collection of matrices $\{\boldsymbol{X}^{(k)}\}_{k\in[K]}$ can not be stacked as a regular third-order tensor. Such a collection is termed an *irregular tensor* and we use the same notation $\mathcal{X}$. Figure 1 depicts an irregular tensor that represents the typical structure of the input of a patient phenotyping problem. In this figure, we assume all features are categorical, *i.e.* matrices are boolean valued. A black cell represents a 1 (the presence of a given event at a given time instant) and a white cell represents a 0 (the absence of a given event at a given time instant).

We will see that some tensor decomposition approaches handle irregular tensors while some others require regular ones. Padding the shortest care pathways with zeros to conform to a regular standardized tensor structure is an appealing option. However, two primary drawbacks come with this approach: (1) It would artificially inflate dataset sizes when accommodating a single lengthy care pathway; (2) It would blur the distinction between the absence of an event during a hospital stay and non-hospital stays, potentially undermining result accuracy and interpretability. For these reasons, irregular tensors are more suitable to represent care pathways.

2.1.2 Third-order tensor decomposition with PARAFAC2

PARAFAC2 (Kiers et al., 1999) is a seminal decomposition designed to handle irregular tensors. It extends the Canonical Polyadic (CP) factorization (Kiers et al., 1999) which is a foundational approach for decomposing regular tensors into a sum of rank-one components. PARAFAC2 introduces flexibility in accommodating tensors with different temporal lengths, making it particularly useful in scenarios where temporal information is crucial, such as in the analysis of longitudinal data (Fanaee-T and Gama, 2015) and specifically electronic health records (Perros et al., 2019).

Recent enhancements to PARAFAC2 have significantly strengthened the capabilities of this model to address specific challenges. In particular, SPARTan, proposed by Perros et al. (2017), stands out for its scalability and parallelizability on large and sparse datasets, while Dpar2 (Jang and Kang, 2022) was designed to handle effectively irregular dense tensors. Nonetheless, these interesting computational properties do not ensure that the solution given by PARAFAC2 identifies meaningful phenotypes in practice. For instance, factor matrices can contain negative values and this does not make sense in the context of patient phenotyping.

Alternative formulations of PARAFAC2 have been proposed to incorporate additional constraints. Cohen and Bro (2018) introduced a non-negativity constraint on the varying mode to enhance the interpretability of the resulting factors. Roald et al. (2022) extended the constraints to all-modes. On the other hand, COPA (Constrained PARAFAC2) (Afshar et al., 2018) took a step further by introducing various meaningful constraints in PARAFAC2 modeling, including latent components that change smoothly over time and sparse phenotypes to ease the interpretation.

Finally, a practical limitation of PARAFAC2 pertains to the rank value. In the context of patient phenotyping, the rank value represents the number of phenotypes. The decomposition technique requires that this value must not exceed the dimension of any mode, including the time dimension. Consequently, the number of phenotypes cannot exceed the minimum duration observed in an irregular third-order tensor, which implies the exclusion of pathways with duration shorter than the specified rank from the dataset.

In conclusion, PARAFAC2 proves to be an interesting model for patient phenotyping using temporal EHR data. It is especially suitable to handle large datasets of irregular tensors such as care pathways. However, its formulation may lead to extract meaningless phenotypes. To address this limitation, some extensions of PARAFAC2 include additional constraints, but they remain simple. The extraction of more meaningful and clinically relevant phenotypes requires flexible constraints.

2.1.3 Extracting relevant phenotypes with tensor decomposition

Recently, several approaches have been proposed to enrich tensor decomposition with expert constraints designed to yield more meaningful and clinically relevant phenotypes. We identified three specific questions that have been addressed in the literature:

Dealing with additional static information Real EHR data contain both temporal (*e.g.*, longitudinal clinical visits) and static information (*e.g.*, age, body mass index (BMI), smoking status, main reason for hospitalisation, etc.). It is expected that the static information impacts the temporal phenotypes, *i.e.* the distribution of care deliveries along the patient visits. This question has been addressed by TASTE (Afshar et al., 2020) and TedPar (Yin et al., 2021). TASTE takes as input an irregular tensor $\mathcal{X}$ and an additional matrix representing the static features of each patient. The decomposition maps input data into a set of phenotypes and patients' temporal evolution. Phenotypes are defined by two factor matrices: one for temporal features and and the other for static features.

Dealing with correlations between diagnoses and medication events In general, the set of medical events contains diagnostic events (e.g., lab tests) and care events (e.g., medications or medical procedures). The occurrences of these two types of events may be correlated: it is likely that establishing a diagnosis leads to the delivery of a specific care. For instance, an elevated blood glucose concentration, measured by a lab test, often leads to the administration of insulin, which constitutes a care event. In patient data, this implies that the likelihood of a blood glucose measurement and an insulin injection co-occurring is higher than, for example, a blood glucose measurement and mechanical ventilation. HITF (Yin et al., 2018) leverages these correlations to enhance tensor decomposition by splitting the dimension of feature types into two dimensions: one for medications and the other for diagnoses. This model has also been used within CNTF (Yin et al., 2019), which proposes modeling patient data as a tensor with four dimensions: patients identifier, lab tests, medications, and time.

Supervision of tensor decomposition Another improvement of the tensor decomposition methods involves extending them to supervised fashions. Henderson et al. (2018) proposed a semi-supervised tensor factorization method that introduces a cannot-link matrix on the patient factor matrix to encourage separation in the patient subgroups.

Predictive Task Guided Tensor Decomposition (TaGiTeD) (Yang et al., 2017) is another framework conceived to overcome the limitations of existing unsupervised approaches, such as the requirement for a large dataset to achieve meaningful results. TaGiTeD guide the decomposition by specific prediction tasks. This is done by learning representations that lead to best prediction performances. Lastly, Rubik (Wang et al., 2015) and SNTF (Anderson et al., 2017) are other tensor factorization models incorporating guidance constraints to align with existing medical knowledge.

It is worth noting that these advanced models benefit from the recent progress in machine learning, and more specifically automatic differentiation (Baydin et al., 2018). Automatic differentiation does not necessitate explicit gradient computation to evaluate efficiently the derivatives of a function. Thus, it eases the design of efficient optimization algorithms for various tensor decomposition tasks. The flexibility of these machine learning frameworks fosters the conception of complex models that produce more meaningful phenotypes.

2.1.4 Temporal dimension in tensor decomposition

While it appears crucial to manage the dynamics in a patient's evolution, most tensor decompositions do not explicitly model the temporal dependencies within the patient data. The temporal aspect is particularly significant when constructing phenotypes for typical care profiles.

First of all, it is worst noting that the seminal decomposition model, PARAFAC2, does not capture temporal information in their phenotypes. Let $\mathcal{X} = \left(\boldsymbol{X}^{(k)}\right)_{k\in[K]}$ be an irregular tensor of K patients, and $\mathcal{Y} = \left(\boldsymbol{Y}^{(k)}\right)_{k\in[K]}$ another irregular tensor such that $y^{(k)}_{:,t} = x^{(k)}_{:,\rho_k(t)}$ where ρ_k is a random permutation of daily vectors of the patient k. The decomposition of these two tensors leads to the exact same phenotypes. This illustrates that the phenotypes extracted by PARAFAC2 are insensitive to the temporal dimension of the data.

Some variants of PARAFAC2 have targeted this limitation by the introduction of temporal regularisation terms. According to Afshar et al. (2018), learning temporal factors that change smoothly over time is often desirable to improve the interpretability and alleviate the over-fitting. COPA, as introduced by Afshar et al. (2018), includes a smoothness regularization to account for irregularities in the temporal gaps between two visits. LogPar Yin et al. (2020) expands this regularization to binary and incomplete irregular tensors. However, both of these techniques have limitations as they rely solely on local information to smooth pathways, neglecting the temporal history needed to construct meaningful phenotypes.

To address long term dependencies, Temporally Dependent PARAFAC2 Factorization (TedPar) (Yin et al., 2021) was developed to model the gradual progression of chronic diseases over an extended period. TedPar introduces the concept of temporal transitions from one phenotype to another to capture temporal dependencies. Additionally, Ahn et al. (2022) proposed Time-Aware Tensor Decomposition (TATD), a tensor decomposition method that incorporates time dependency through a smoothing regularization with a Gaussian kernel. For CNTF (Yin et al., 2019), a recurrent neural network (RNN) was used to take into account the ordering of the clinical events. Given the sequence $w^k_{p,1}, \dots, w^k_{p,t-1}$ describing the progression of a phenotype p of a given patient k, an LSTM (Long Short-Term Memory) network is used to predict $w^k_{p,t}$ such that the Mean Square Error (MSE) between the real and predicted value is minimized. The idea behind this is to penalize a reconstruction model that does not allow to accurately predict the next sequence of events. It enforces to discover a decomposition that is easily predictable.

It is worth noting that all these models do not discover temporal patterns. The temporal dimension is used to constraint the extraction of daily phenotypes by taking into account the temporal dependencies. Nonetheless, these temporal dependencies are not explicit for a physician analyzing the care pathways. A daily phenotype shown to physicians only represent co-occurring events. The method presented in this article extracts phenotypes that describe a temporal pattern. The phenotype itself encapsulates information about temporal dependencies in an easily interpretable manner.

2.2 Alternative approaches for extracting temporal phenotypes

While tensor decomposition techniques have not yet tackled the issue of extracting temporal patterns from care pathways, similar challenges have been addressed using alternative approaches. In this section, we highlight three of them:

Temporal Extensions of Topic Models Originally, topic modeling (or latent block models) is a statistical technique for discovering the latent semantic structures in textual document. It can estimate, at the same time, the mixture of words that is associated with each topic, and the mixture of topics that describes each document. Pivovarov et al. (2015) and Ahuja et al. (2022) proposed to consider the patients' data as a collection of documents. The topic modeling of these documents results in a set of topics representing the phenotypes. Temporal extensions of topic modeling could then be used to extract temporal phenotypes. For instance, Temporal Analysis of Motif Mixtures (TAMM) (Emonet et al., 2014) is a probabilistic graphical model designed for unsupervised discovery of recurrent temporal patterns in time series. It uses non-parametric

Bayesian methods fitted using Gibbs sampling to describe both motifs and their temporal occurrences in documents. It is important to mention that the extracted motifs include a temporal dimension. This modeling capability seems to be very interesting in patient phenotyping to derive temporal phenotypes. TAMM relies on an improved version of the Probabilistic Latent Sequential Motif model (Varadarajan et al., 2010) which explains how the set of all observations is supposed to be generated. Variable Length Temporal Analysis of Motif Mixtures is a generalization of TAMM that allows motifs to have different lengths and infers the length of each motif automatically. The primary limitation is that they do not scale as effectively as the optimization techniques employed in tensor decomposition (Kolda and Bader, 2009). Furthermore, the models are highly inflexible, and making modifications requires developing new samplers. These limitations prevented us from using these topic models.

Phenotypes as Embeddings In the context of neural networks, embeddings are low-dimensional, learned continuous vector representations of discrete variables. Neural network embeddings are useful because they can reduce the dimensionality of categorical variables and meaningfully represent categories in the transformed space. The primary purposes of using embeddings are finding nearest neighbours in the embedding space and visualizing relations between categories. They can also be used as input to a machine learning model for supervised tasks. Hettige et al. (2020) introduces MedGraph, a supervised embedding framework for medical visits and diagnosis or medication codes taken from pre-defined standards in healthcare such as International Classification of Diseases (ICD). MedGraph leverages both structural and temporal information to improve the embeddings quality.

Pattern Mining Methods Sequential pattern mining (Fournier-Viger et al., 2017) addresses the problem of discovering hidden temporal patterns. A sequential pattern would represent a phenotype by a sequence of events. The well-known problem of pattern mining, which tensor decomposition does not suffer from, is pattern deluge. This problem makes it unsuitable for practical use. Nonetheless, tensor decomposition methods are close to pattern mining approaches based on compression. GoKrimp (Lam et al., 2014), SQS (Tatti and Vreeken, 2012) and more recently SQUISH (Bhattacharyya and Vreeken, 2017) proposed sequential pattern mining approaches that optimize a Minimum Description Length (MDL) criteria (Galbrun, 2022). Unfortunately, only GoKrimp is able to handle sequences of item-sets (*i.e.* with parallel events), but it extracts only sequences of items and does not allow interleaving occurrences of patterns. As representing the parallel events is a crucial aspect of phenotypes, these techniques can not answer the problem of temporal phenotyping.

3 Temporal phenotyping: new problem formulation

This section formalizes the problem of temporal phenotyping that is addressed in the remainder of the article. In short, temporal phenotyping is a tensor decomposition of a third-order temporal tensor discovering phenotypes that are manifested as temporal patterns.

Let $\mathcal{X}$ be an irregular third-order tensor, also viewed as a collection of K matrices of dimension $n \times T_k$, where K is the number of individuals (patients), n is the number of features (care events), and T_k is the duration of the k-th individual's observations.

Given $R \in \mathbb{N}^*$, a number of phenotypes and $\omega \in \mathbb{N}^*$ the duration of phenotypes (also termed as *phenotype size*), temporal phenotyping aims to build:

- $\mathcal{P} \in \mathbb{R}_+^{R\times n\times\omega}$: a third-order tensor representing the R temporal phenotypes shared among all individuals. Each temporal phenotype is a matrix of size $n \times \omega$. A phenotype represents the presence of an event at a relative time τ, $0 \le \tau < \omega$. ω is the same for all phenotypes. $\boldsymbol{p}_\tau^{(r)}$ denotes the vector representing the co-occurring events in the r-th phenotype at the relative temporal position τ.
- $\mathcal{W} = \left\{ \boldsymbol{W}^{(k)} \in \mathbb{R}_+^{R\times T'_k} \right\}_{k\in[K]}$: a collection of K assignment matrices of dimension $R \times T'_k$ where $T'_k = T_k - \omega + 1$ is the size for the k-th individual along the temporal dimension. A non-zero value at position (r, t) in $\boldsymbol{W}^{(k)}$ describes the start of the phenotype r at time t for the k-th individual. A matrix $\boldsymbol{W}^{(k)}$ is also named the *pathway* of the k-th individual as it describes his/her history as a sequence of temporal phenotypes.

These phenotypes and pathways are built to accurately reconstruct the input tensor, *i.e.* $\mathcal{X}$. The reconstruction we propose is based on a convolution operator that takes into account the time dimension of $\mathcal{P}$ to reconstruct the input tensor from $\mathcal{P}$ and $\mathcal{W}$. The convolution operator, denoted $\circledast$, is such that $\boldsymbol{X}^{(k)} \approx \widehat{\boldsymbol{X}}^{(k)} = \mathcal{P} \circledast \boldsymbol{W}^{(k)}$ for all $k \in [K]$ (we remind the reader that $\boldsymbol{X}^{(k)}$ is the matrix for the k-th patient, see Sect. 2.1.1 and Fig. 1). Formally, this operator reconstructs each vector of the matrix $\widehat{\boldsymbol{X}^{(k)}}$ at time t, denoted $\widehat{\boldsymbol{x}}_{\cdot,t}^{(k)}$, as follows:

$$\widehat{\boldsymbol{x}}_{\cdot,t}^{(k)} = \sum_{r=1}^{R} \sum_{\tau=1}^{\min(\omega,t-1)} \boldsymbol{w}_{r,t-\tau}^{(k)} \boldsymbol{p}_\tau^{(r)}. \tag{1}$$

Intuitively, $\widehat{\boldsymbol{x}}_{\cdot,t}^{(k)}$ is a mixture of phenotype columns that occurred at most ω time units ago, except at the beginning. At one time instant, the observed events are the sum of the τ-th day of the R phenotypes weighted by the $\boldsymbol{W}^{(k)}$ matrix.

Figure 2 depicts the reconstruction of one matrix $\boldsymbol{X}^{(k)}$ of an input tensor. This matrix is of length $T_k = 14$ with $n = 4$ features. Its decomposition is made of $R = 3$ phenotypes of size 4×2 each ($\omega = 2$ and $n = 4$) and a pathway of length $T'_k = 14 - 2 + 1 = 13$. A colored square in $\boldsymbol{W}^{(k)}$ indicates the start of phenotype occurrences, which can overlap in $\boldsymbol{X}^{(k)}$. For instance, the column $\boldsymbol{x}_{\cdot,5}^{(k)}$ combines the occurrence of the second day of the second phenotype (in green) and the first day of the third phenotype (in blue). Each patient is given a pathway matrix $\boldsymbol{W}^{(k)}$ based on the same phenotypes $\mathcal{P}$ and according to his input matrix $\boldsymbol{X}^{(k)}$. This means that a phenotype represents a typical pattern that might occur in the pathways of multiple patients.

The problem of temporal phenotyping consists in discovering both $\mathcal{P}$ and $\mathcal{W}$ tensors that reconstruct accurately the input tensor.

4 SWoTTeD model

SWoTTeD is a tensor decomposition model for temporal phenotyping. The generic problem of temporal phenotyping presented above is complemented by some additional hypotheses to guide the solving toward practically interesting solutions. These hypotheses are implemented through the definition of a reconstruction loss and regularization terms. This section presents the detail of the **S**liding **W**ind**o**w for **T**emporal **Te**nsor **D**ecomposition model.

4.1 Temporal phenotyping as a minimization problem

As in the case of the classic tensor decomposition problem, temporal phenotyping is a problem of minimizing the error between the input tensor and its reconstruction.

SWoTTeD considers the decomposition of binary tensors, *i.e.* $\mathcal{X} \in \{0, 1\}$. It corresponds to data that describe the presence/absence of events. In this case, we assume the input tensor $\mathcal{X}$ follows a Bernoulli distribution and we use the loss function for binary data proposed by Hong et al. (2020).[2] In the previous section, Eq. 1 details the reconstruction of a patient matrix. The resulting reconstruction loss $\mathcal{L}^{\circledast}$ is defined as follows:

$$\mathcal{L}^{\circledast}(\hat{\mathcal{X}}, \mathcal{X}) = \sum_{k=1}^{K} \sum_{t=1}^{T_k} \sum_{i=1}^{n} \log(\hat{x}_{i,t}^{(k)} + 1) - x_{i,t}^{(k)} \log(\hat{x}_{i,t}^{(k)}). \tag{2}$$

This reconstruction loss is super-scripted by $\circledast$ to remind that it is based on the convolution operator described in Eq. 1.

SWoTTeD also includes two regularization terms: sparsity and non-succession regularization. Sparsity regularization on $\mathcal{P}$ aims to enforce feature selection and improve the interpretability of phenotypes. It is implemented through an L_1 term. We chose this popular regularization technique among several others, as it has shown its practical effectiveness.

We also propose a phenotype non-succession regularization to prevent undesirable decomposition, as illustrated in Fig. 3. The described situation is a successive occurrence of the same event. This situation is often encountered in care pathways as a treatment might be a care delivery over several days. In this case, there are two opposite alternatives to decompose the matrix with equal reconstruction errors ($\mathcal{L}^{\circledast}$): the first alternative (at the top) is to describe the treatment as a daily care delivery and to assume that a patient received the same treatment three days in a row; the second alternative (at the bottom) is to describe the treatment as a succession of three care deliveries, but that is received only once. SWoTTeD implements the second solution as one of our objective is to unveil *temporal* patterns, *i.e.* phenotypes that correlate temporally some events.

To guide the decomposition toward our preferred one, we add a term to penalize a reconstruction that uses the same phenotype on successive days. If a phenotype occurs on one day, its recurrence within the following ω days will incur a cost. Formally, the non-succession regularization is defined as follows and depends only on the patient pathway $\boldsymbol{W}^{(k)}$:

$$\mathcal{S}(\boldsymbol{W}^{(k)}) = \sum_{r=1}^{R} \sum_{t=1}^{T'_k} \max\left(0, w_{r,t}^{(k)} \log\left(\sum_{\tau=t-\omega}^{t+\omega} w_{r,\tau}^{(k)}\right)\right). \tag{3}$$

This equation can be seen as a weighted logged convolution where the weight is $w_{r,t}$. Intuitively, as the prevalence of the phenotype grows, the cost of a new occurrence within the same time window also rises.

The inner log term sums all possible undesirable occurrences of the same phenotype r at time t. The log function is used to attenuate the effect of this term and to have a zero value

[2] Hong et al. (2020) discuss in detail the choice of the loss function regarding the distribution of a variable: Gaussian, Poisson (positive counts), Gamma (positive continuous data) or Bernoulli (binary data).

when $w_{r,t}^{(k)} = 1$ surrounded with 0 (the ideal case, depicted in the second decomposition of Fig. 3).

The final loss function of SWoTTeD is given by the weighted sum of the reconstruction error, the sparsity and the non-succession regularization:

$$\ell = \mathcal{L}^{\circledast}(\mathcal{P} \circledast \mathcal{W}, \mathcal{X}) + \alpha \; ||\mathcal{P}||_1 + \beta \sum_{k=1}^{K} \mathcal{S}\big(\boldsymbol{W}^{(k)}\big) \tag{4}$$

where α and β are two positive real-valued hyperparameters. Note that the two regularization terms have opposite effects: sparsity encourages phenotypes with many zeros, while non-succession favors the use of non-zero values in phenotypes rather than in pathways. The choice of hyperparameters may impact the quality of the discovered phenotypes.

4.2 Optimization framework

We aim to uncover temporal phenotypes by minimizing the overall loss function ℓ. The specification of our minimization problem is as follows:

$$\begin{aligned} \underset{\{\boldsymbol{W}^{(k)}\},\mathcal{P}}{\arg\min} \quad & \mathcal{L}^{\circledast}(\mathcal{P} \circledast \mathcal{W}, \mathcal{X}) + \alpha \; ||\mathcal{P}||_1 + \beta \sum_{k=1}^{K} \mathcal{S}\big(\boldsymbol{W}^{(k)}\big), \\ \text{subject to} \quad & \forall k,\ 0 \leq \boldsymbol{W}^{(k)} \leq 1, \quad 0 \leq \mathcal{P} \leq 1. \end{aligned} \tag{5}$$

The minimization problem presented in Eq. 5 restricts the values' range to [0, 1] in order to interpret $\mathcal{P}$ (resp. $\mathcal{W}$) as the probability of having an event (resp. a phenotype) at a given time. These constraints align with our assumption of Bernoulli distribution. Another motivation to normalize the pathways is related to the non-succession regularization, $\mathcal{S}(\boldsymbol{W}^{(k)})$. If the values of $\boldsymbol{W}^{(k)}$ are higher than 1, the non-succession regularization penalizes each presence of a phenotype which is not desired.

To optimize the overall loss function ℓ (see Eq. 4), we use an alternating minimization strategy and projected gradient descent (PGD). Alternating Gradient Descent algorithm optimizes one variable at a time, individually, using a gradient descent step, with all other variables fixed. Alternating the process of minimization guarantees reduction of the cost function, until convergence. PGD handles the non-negativity and the normalization constraints. It works by clipping the values after each iteration.

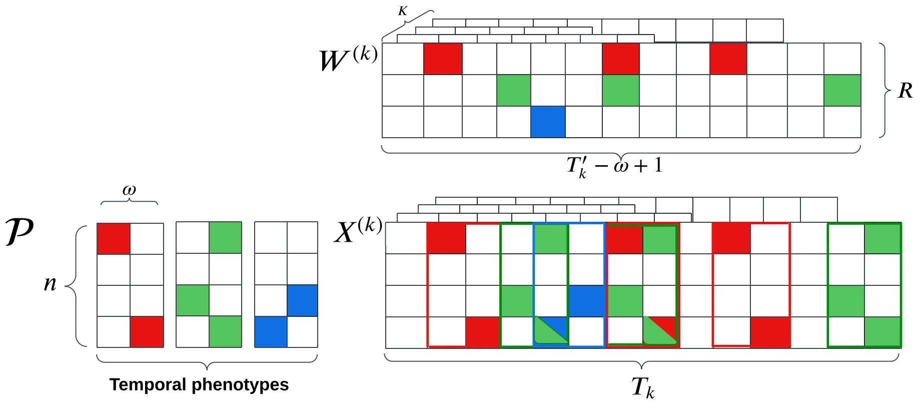


Fig. 2 Illustration of a matrix reconstruction ($\boldsymbol{X}^{(k)}$) from $R = 3$ phenotypes of size $\omega = 2$ on the left and a care pathway ($\boldsymbol{W}^{(k)}$) on the top. Each phenotype has a specific color. Each colored cell in $\boldsymbol{W}^{(k)}$ designates the start of a phenotype occurrence in the reconstruction (surrounded with a colored rectangle in $\boldsymbol{X}^{(k)}$). A cell with two colors received the contribution of two occurrences of different phenotypes

Algorithm 1 Optimization Framework for SWoTTeD

```
Data: $\mathcal{X} = \{\boldsymbol{X}^{(\boldsymbol{k})} \mid k \in [K]\}$: patient stays, $R$: the number of temporal
      phenotypes, $\omega$: the size of the temporal window, $\alpha$, $\beta$: loss
      hyper-parameters.
Result: $\mathcal{P}$: phenotype tensor, $\{\boldsymbol{W}^{(\boldsymbol{k})} \mid k \in [K]\}$: phenotype occurrences in
        patient stays
Definition: clip$(x, 0, 1) = \min(\max(x, 0), 1)$
$\mathcal{P} \leftarrow \mathcal{U}(0, 1)$      // Random initialization of optimization variables
$\mathcal{W} \leftarrow \mathcal{U}(0, 1)$
for each epoch do
    for each mini-batch $\mathcal{B} \subset \mathcal{X}$ do
        Update $\mathcal{P}$ by descending along the gradient direction
        $\mathcal{P} \leftarrow$ clip$(\mathcal{P}, 0, 1)$
        for $k \in \mathcal{B}$ do
            Update $\boldsymbol{W}^{(k)}$ by descending along the gradient direction
            $\boldsymbol{W}^{(k)} \leftarrow$ clip$(\boldsymbol{W}^{(k)}, 0, 1)$
        end
    end
end
```

The optimization framework of SWoTTeD is illustrated in Algorithm 1. $\mathcal{W}$ and $\mathcal{P}$ are initialized with random values drawn from a uniform distribution between 0 and 1. In each mini-batch, we first sample a collection of patient matrices $\{\boldsymbol{X}^{(k)} \mid k \in \mathcal{B}\}$ with $\mathcal{B}$ being the patient's indices of a batch. The phenotype tensor $\mathcal{P}$ is firstly optimized given $\{\boldsymbol{W}^{(k)} \mid k \in \mathcal{B}\}$ values, then $\{\boldsymbol{W}^{(k)} \mid k \in \mathcal{B}\}$ is optimized given $\mathcal{P}$ values. Note that the gradients are not explicitly computed, but evaluated by automatic differentiation (Baydin et al.,

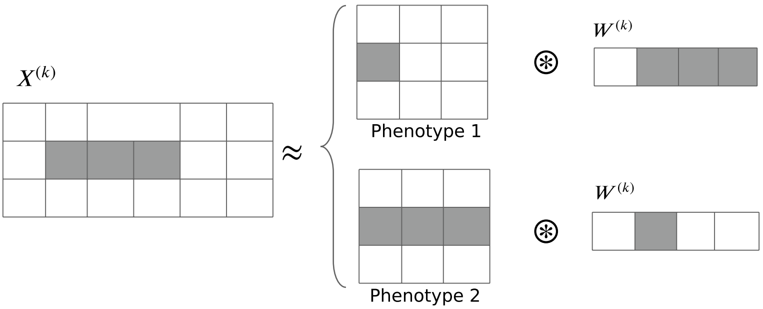


Fig. 3 Example of alternative decompositions of a sequence of similar events with the same $\mathcal{L}^{\circledast}$ value. Phenotype 1 does not capture the sequence of events, whereas phenotype 2 does. The information is reported in the pathway in the case of phenotype 1

2018). The algorithm stops after a fixed number of epochs, with the number of epochs being a predefined hyper-parameter.[3]

Among the conventional optimization hyper-parameters, including learning rate, batch size, and the number of epochs, the primary hyperparameters for SWoTTeD encompass R (number of temporal phenotypes), ω (temporal size of phenotypes), and α, β (loss weights). In contrast to recent deep neural network architectures, SWoTTeD features a limited set of interpretable hyperparameters.

4.3 Applying SWoTTeD on test sets

The tensor decomposition presented in the previous section corresponds to the training of SWoTTeD on an irregular tensor $\mathcal{X}$. This training provides a set of temporal phenotypes $\mathcal{P}$. In the minimization problem presented in Eq. 5, the assignment tensor $\mathcal{W}$ contains a set of free parameters that are to be discovered during the learning procedure but are not kept in the model because they are specific to the train set.

The results of the decomposition is evaluated on a different test set, $\mathcal{X}'$. The objective is to assess whether the unveiled phenotypes are useful for decomposing new care pathways. In this case, we can conclude that it captures generalizable phenotypes, otherwise it discovers too specific ones (overfitting).

Applying a tensor decomposition on a test set, $\mathcal{X}'$, consists in finding the optimal assignment given a fixed set of temporal phenotypes. $\mathcal{X}'$ is a third-order tensor with K' individuals, each having their duration, but sharing the same n features defined in the training dataset. The optimal assignment is obtained by solving the following optimization problem that is similar to Eq. 5, but with a fixed $\widehat{\mathcal{P}}$ (the optimal phenotypes obtained from the decomposition of a train set):

$$\begin{aligned} \underset{\mathcal{W}'}{\arg\min} \quad & \mathcal{L}^{\circledast}\left(\widehat{\mathcal{P}} \circledast \mathcal{W}', \mathcal{X}'\right) + \beta \sum_{k=1}^{K} \mathcal{S}\left(\boldsymbol{W}'^{(k)}\right) \\ \text{subject to} \quad & 0 \leq \mathcal{W} \leq 1. \end{aligned} \tag{6}$$

[3] An early stopping based on a convergence criteria would be possible to reduce the computing time of simplest decompositions.

This optimization problem can be solved by a classical gradient descent algorithm. Similarly to the training, we use PGD for the normalization constraint.

5 Experimental setup

SWoTTeD is implemented in Python using the PyTorch framework (Paszke et al., 2019), along with PyTorch Lightning[4] for easy integration into other deep learning architectures. The model is available in the following repository: https://gitlab.inria.fr/hsebia/swotted. In the experiments, we used two equivalent implementations of SWoTTeD: a classic version that handles irregular tensor and a fast-version that handles only regular tensor benefiting from improved vectorial optimization.[5] Additionally, we provide the repository which includes all the materials needed to reproduce the experiments except for the case study from Sect. 7: https://gitlab.inria.fr/tguyet/swotted_experiments. All experiments have been conducted with desktop computers,[6] without the use of GPU acceleration.

We trained the model with an Adam optimizer to update both tensors $\mathcal{P}$ and $\mathcal{W}$. The learning rate is set to 10^{-3} with a batch size of 50 patients. We fine-tuned the hyperparameters α and β by testing different values and selecting the ones that yielded the best reconstruction measures (see experiments in Sect. 6.1). The tensors $\mathcal{P}$ and $\mathcal{W}$ are initialized randomly using a uniform distribution ($\mathcal{U}(0, 1)$).

The quality measures reported in the results have been computed on test sets. For each experiment, 70% of the dataset is used for training, and 30% is used for testing. The test set patients are drawn uniformly.

5.1 Datasets

In this section, we present the open-access datasets used for quantitative evaluations of SWoTTeD, including comparisons with competitors. We conducted experiments on both synthetic and real-world datasets to evaluate the reconstruction accuracy of SWoTTeD against its competitors. Synthetic datasets are used to quantitatively assess the quality of the hidden patterns as they are known in this specific case.

5.1.1 Synthetic data

The generation of synthetic data involves the reverse process of the decomposition. Generating a dataset follows three steps:

1. A third-order tensor of phenotypes $\mathcal{P}$ is generated by randomly selecting a subset of medical events for each instant of the temporal window of each phenotype.
2. The patient pathways $\mathcal{W}$ are generated by randomly selecting the days of occurrence for each phenotype along the patient's stay, ensuring that the same phenotype cannot occur on successive days. Bernoulli distributions with $p = 0.3$ are used for this purpose.

[4] https://www.pytorchlightning.ai

[5] "Comparison of time efficiency of SWoTTeD versus FastSWoTTeD" section of "Appendix 2" provides comparison between the two implementations.

[6] Processor Intel i7-1180G7, 4.60 GHz, 16 Gb RAM, without graphical acceleration, running Ubuntu 22.04 system.

3. The patient matrices of $\mathcal{X}$ are then computed using the reconstruction formulation proposed in Eq. 1.

However, this reconstruction can result in values greater than 1 when multiple occurrences of phenotypes sharing the same drug or procedure deliveries accumulate. To fit our hypothesis of binary tensors, we binarize the tensor resulting from the process above by projecting non-zero values to 1.

The default characteristics of the synthetic datasets subsequently generated for various experiments are as follows: $K = 500$ patients, $n = 20$ care events, $R = 4$ phenotypes of length $\omega = 3$, and stays of $T_k = 10$ days for all k.

5.1.2 Real-world datasets

The experiments conducted on synthetic datasets are complemented by experiments on three real-world datasets, which are publicly accessible. We selected one classical dataset in the field of patient phenotyping (namely the MIMIC database) and two sequential datasets coming from very different contexts. Table 1 summarizes the main characteristics of these datasets.

- MIMIC dataset[7]: MIMIC-IV is a large-scale, open-source and deidentified database providing critical care data for over 40, 000 patients admitted to intensive care units at the Beth Israel Deaconess Medical Center (BIDMC) (Johnson et al., 2020). We used the version 0.4 of MIMIC-IV. The dataset has been created from the database by selecting a collection of patients and gathering their medical events during their stay. For the sake of reproducibility, the constitution of the dataset is detailed in "Appendix 1". In addition, the code used to generate our final dataset is provided in the repository of experiments.
- E-Shop dataset[8]: This dataset contains information on clickstream from one online store offering clothing for pregnant women. Data are from five months of 2008 and include, among others, product category, location of the photo on the page, country origin of the IP address and product price.
- Bike dataset[9]: This contains sequences of locations where shared bikes where parked in a city. Each item represents a bike sharing station and each sequence indicate the different locations of a bike over time. The specificity of this dataset is to contain only one location per date.

For each of these datasets, we also created a "regular" version, which contains individuals' pathways sharing the same length. This dataset is utilized with our fast-SWoTTeD implementation that benefits from better vectorization. The creation of these datasets involves two steps: (1) selecting individuals with pathway durations greater than or equal to T, and (2) truncating the end of the pathways if they exceed T in length. The selection of the values for T is a balance between maintaining the maximum length and retaining the maximum number of individuals.

[7] Original dataset: https://physionet.org/content/mimiciv/0.4/

[8] Original data: https://archive.ics.uci.edu/dataset/553/clickstream+data+for+online+shopping. We used the prepared version from SPMF repository.

[9] Original data: https://www.kaggle.com/cityofLA/los-angeles-metro-bike-share-trip-data. We used the prepared version from SPMF repository.

We remind that our case study presents another dataset of patients staying in the Greater Paris University Hospitals for qualitative analysis of phenotypes. This dataset will be detailed in Sect. 7.

5.2 Competitors

We compare the performance of SWoTTeD against four state-of-the-art tensor decompostion models. These models were selected based on the following criteria: (1) their motivation to analyze EHR datasets, (2) their competitiveness in terms of accuracy compared to other approaches, (3) the availability of their implementations, and (4) their handling of temporality.

We remind that SWoTTeD is the only tensor decomposition technique able to extract temporal patterns. Our competitors extract daily phenotypes.

The four competing models are the followings:

- *CNTF*—Collective Non-negative Tensor Factorization (Yin et al., 2019) a tensor decomposition model factorizing tensors with varying temporal size, assuming the input tensor to follow a Poisson distribution, but it has shown its effectiveness on binary data; CNTF is our primary competitor since it incorporates temporal regularization, aiming to capture data dynamics.
- *PARAFAC2* (Kiers et al., 1999), original decomposition model with non-negative constraint. This decomposition is based on Frobenius norm. We use the Tensorly implementation (Kossaifi et al., 2019).
- *LogPar* (Yin et al., 2020), a logistic PARAFAC2 for learning low-rank decomposition with temporal smoothness regularization. We choose to include LogPar in the competitors' list because, like SWoTTeD, it is designed for binary tensors and assumes a Bernoulli distribution. LogPar can handle only regular tensors.
- *SWIFT*—Scalable Wasserstein Factorization for sparse non-negative Tensors (Afshar et al., 2021), a tensor decomposition model minimizing the Wasserstein distance between the input tensor and its reconstruction. SWIFT does not assume any explicit distribution, thus it can model complicated and unknown distributions.

For each experiment, we manually configure their hyper-parameters to ensure the fairest possible comparisons.

5.3 Evaluation metrics

In tensor decomposition, a primary objective is to reconstruct accurately an input tensor. We adopt the $FIT \in (-\infty, 1]$ (Bro et al., 1999) to measure the quality of a model's reconstruction:

$$FIT_X = 1 - \frac{\sum_{k=1}^{K} ||\boldsymbol{X}^{(k)} - \widehat{\boldsymbol{X}}^{(k)}||_F}{\sum_{k=1}^{K} ||\boldsymbol{X}^{(k)}||_F} \tag{7}$$

Table 1 Real-world dataset characteristics: n number of features, K number of individuals, $\bar{T}$ mean duration

Dataset	n	K	$\bar{T}$
MIMIC-IV	2717	200	9.62 ± 12.7
MIMIC-IV-reg	825	200	10
Bike	21, 078	67	11.43 ± 5.93
Bike-reg	5243	67	10
E-Shop	24, 026	317	13.79 ± 11.19
E-Shop-reg	5212	317	10

where the input tensor $\mathcal{X}$ serves as the ground truth, the resulting tensor is denoted $\widehat{\mathcal{X}}$ and $|| \cdot ||_F$ is the Frobenius norm. The higher the value of *FIT*, the better. The *FIT* measure is also used to compare phenotypes and patient pathways when hidden patterns are known a priori, i.e., for synthetic datasets. Thus, FIT_P (resp. FIT_W) denotes the reconstruction quality of $\mathcal{P}$ (resp. $\mathcal{W}$).

It is worth noting that FIT_X measure is computed on a test set except for SWIFT and PARAFAC2. Evaluation on test sets requires the model to be able to project a test set on existing phenotypes (see Sect. 4.3), but SWIFT and PARAFAC2 do not have this capability.

We also introduce a similarity measure between two sets of phenotypes to evaluate empirically the uniqueness of solutions and a diversity measure of a set of phenotypes, adapted from similarity measures introduced by Yin et al. (2019).

Let $\mathcal{P} = \{\boldsymbol{P}_1, \dots, \boldsymbol{P}_R\}$ and $\mathcal{P}' = \{\boldsymbol{P}'_1, \dots, \boldsymbol{P}'_R\}$ be two sets of phenotypes defined over a temporal window size ω. The principle of our similarity measure is to find the optimal matching between the phenotypes of the two sets, and to compute the mean of the (dis) similarities between the matching pairs of phenotypes. More formally, in the case of cosine similarity, we compute:

$$\mathrm{sim}(\mathcal{P}, \mathcal{P}') = \max_{\pi} \begin{pmatrix} R-1 \\ 2 \end{pmatrix}^{-1} \sum_{(i,j)\in\pi} \cos(\boldsymbol{P}_i, \boldsymbol{P}'_j)$$

where π denotes an isomorphism between $\mathcal{P}$ and $\mathcal{P}'$, and $\cos(\cdot, \cdot)$ is a cosine distance between two temporal phenotypes. It is computed as the mean of the cosine similarity between each time slice of the phenotype:

$$\cos(\boldsymbol{P}, \boldsymbol{P}') = \frac{1}{\omega} \sum_{i=1}^{\omega} \frac{\langle \boldsymbol{p}_{:,i}, \boldsymbol{p}'_{:,i} \rangle}{||\boldsymbol{p}_{:,i}|| \, ||\boldsymbol{p}'_{:,i}||}$$

where $\langle \cdot, \cdot \rangle$ is the Euclidean inner product.

In practice, we first compute a matrix of costs and use the Hungarian algorithm (Kuhn, 1955) to find the optimal matching (π). Finally, we compute the measure with π.

The diversity measure of the set of phenotypes aims to quantify the redundancy among the phenotypes. In this case, we expect to have low similarities between the phenotype.

Let $\mathcal{P} = \{\boldsymbol{P}_1, \dots, \boldsymbol{P}_R\}$ be a set of R phenotypes defined over a temporal window of size ω, the cosine diversity is defined by:

$$\mathrm{div}(\mathcal{P}) = \left(\frac{R}{2}\right)^{-1} \sum_{1\le i<j\le R} 1 - \cos(\boldsymbol{P}_i, \boldsymbol{P}_j).$$

6 Experiments and results

In this section, we present the experiments we conducted to evaluate SWoTTeD.

6.1 Loss hyper-parameters exploration and ablation study

This section focuses on the loss hyper-parameters of SWoTTeD.[10] The objectives of these experiments are to evaluate the usefulness of each term, to provide a comprehensive review of the effect of the loss parameters and to assess whether the model behaves as expected. We start by presenting some experiments on synthetic datasets and then confirm the results on the three real-world datasets.

In this first experiment, we generated synthetic datasets using R hidden phenotypes ($R = 4, 12$ or 36) with a window size $\omega = 3$. SWoTTeD is run 10 times with different parameter values:

- $\alpha = \{0, 0.5, 1, 2, 4, 8, 16\}$ representing the weight of the sparsity term in the loss; $\alpha = 0$ disables this term.
- $\beta = \{0, 0.5, 1, 2, 4, 8, 16\}$ representing the weight of the phenotype non-succession term in the loss; $\beta = 0$ disables this term.

We collected the FIT_X and FIT_P metric values on a test set. The first metric assesses the quality of the reconstruction, while the second assesses whether the discovered phenotypes match the expected ones.

Figure 4 depicts the FIT measures with respect to the parameters α and β. The normalization is considered in this experiment. One general result is that the FIT_X values are high. Values exceeding 0.5 indicate significantly good reconstructions, and those surpassing 0.8 imply that the differences between two matrices become imperceptible. Furthermore, we observe that a good FIT_P implies a good FIT_X. This illustrates that in tensor decomposition, an accurate discovery of hidden phenotypes is beneficial for achieving a high-quality tensor reconstruction. However, as we see with $R = 32$, a good FIT_X does not necessarily mean that the method discovered the correct phenotypes.

A second general observation is that the same evolution of the FIT with respect to β is observed in most of the settings: the FIT increases between $\beta = 0$ and $\beta = 1$ and then it decreases quickly for higher values. This demonstrates two key points: (1) the inclusion of non-succession term improves the reconstruction accuracy, and (2) on average, a value of $\beta = 0.5$ yields the best results. Regarding the parameter α, we notice a slight improvement in FIT measures as α increases. This is more obvious with $R = 36$. When we focus on results with $\beta = 0.5$, we observe that $\alpha = 1$ is, on average, the best compromise for the sparsity term.

One last observation is that as R increases, FIT measures decrease. This may be counter-intuitive, as the expected results is that the FIT_X decreases with increasing R (as we will discuss with real-world dataset results later on). In this experiment, R corresponds to both the rank of the decomposition and the number of hidden phenotypes we used to generate the dataset. As R increases, the dataset contains more variability and denser events, making the reconstruction task challenging and leading to a slightly decrease of the FIT measure. FIT_X maintains a high value even with $R = 36$ but FIT_P has low values in this case. We explain this observation by colinearities between phenotypes.

[10] Additional results on the effect of the normalization are presented in Annex B.1.

Finally, we complemented our analysis of parameters by specifically investigating the non-succession term introduced in SWoTTeD. To assess its efficiency, we generated synthetic datasets with 4 random phenotypes and 6 phenotypes that have been designed to contain successions of similar events (see Fig. 3).

Figure 5 depicts the FIT_X and FIT_p values with respect to β (α is set to 2 in this experiment). We observe clearly that FIT_P are higher when β is not zero, *i.e.* when we use the non-succession term in SWoTTeD ($FIT_P = 0.75$ instead of 0.50 when $\beta = 0$). The best median FIT_X is close to 0.8 and occurs when $\beta = 0.5$. This confirms that adding the non-succession regularization disambiguates the situation illustrated in Fig. 3 and helps the model to correctly reconstruct the latent variables. We conclude that the use of the non-succession regularization increases the decomposition quality, whether there are event repetitions in phenotypes or not (see Fig. 4).

6.2 SWoTTeD against competitors

In this section, we compare SWoTTeD against competitors based on the ability to achieve accurate reconstructions, to extract hidden phenotype effectively and to efficiently handle large scale datasets. To address these various dimensions, we use both synthetic and real-world datasets.

For the real-world datasets, we use truncated versions because LogPar requires regular tensors. Additionally, we remind that *FIT* values are evaluated on test sets, except for SWIFT and PARAFAC2, for which we utilize the error on the training set since these approaches can not be applied on test sets.

6.2.1 SWoTTeD accuracy on daily phenotypes

This experiment compares the accuracy of SWoTTeD against competitors on 20 synthetic datasets. For the sake of fairness, the datasets are generated with daily hidden phenotypes ($\omega = 1$). Our goal is to evaluate the accuracy of phenotypes extracted by SWoTTeD compared to the ones of state-of-the-art models.

The results, depicted in Fig. 6, show that SWoTTeD achieves the best performance in terms of FIT_X and FIT_P metrics. The second-best model is PARAFAC2. It achieves good tensor reconstructions but it fails to identify the hidden phenotypes (low FIT_P values). The analysis of the phenotypes shows that PARAFAC creates mixtures of phenotypes. Note that PARAFAC2 has no value for $R = 36$ because R must be lower than the maximum of every dimension.

The reconstructions of CNTF are also satisfying but the phenotypes are different from the expected one. Our intuition is that assuming Poisson distribution is not effective for these data following a Bernoulli distribution. The two other competitors are assumed to be adapted to these data but in practice, we observe that they have reconstructed the tensors with lower accuracy, and the extracted phenotypes are less similar to the hidden ones compared to SWoTTeD.

To confirm these results, Fig. 7 depicts critical difference diagrams[11]. It ranks the methods based on the Wilcoxon signed-rank test on FIT_X or FIT_P metrics. The diagram shows

[11] Critical diagram (Demšar, 2006) is a visual representation of the results of the Wilcoxon signed-rank test (see figure legend for reading instructions).

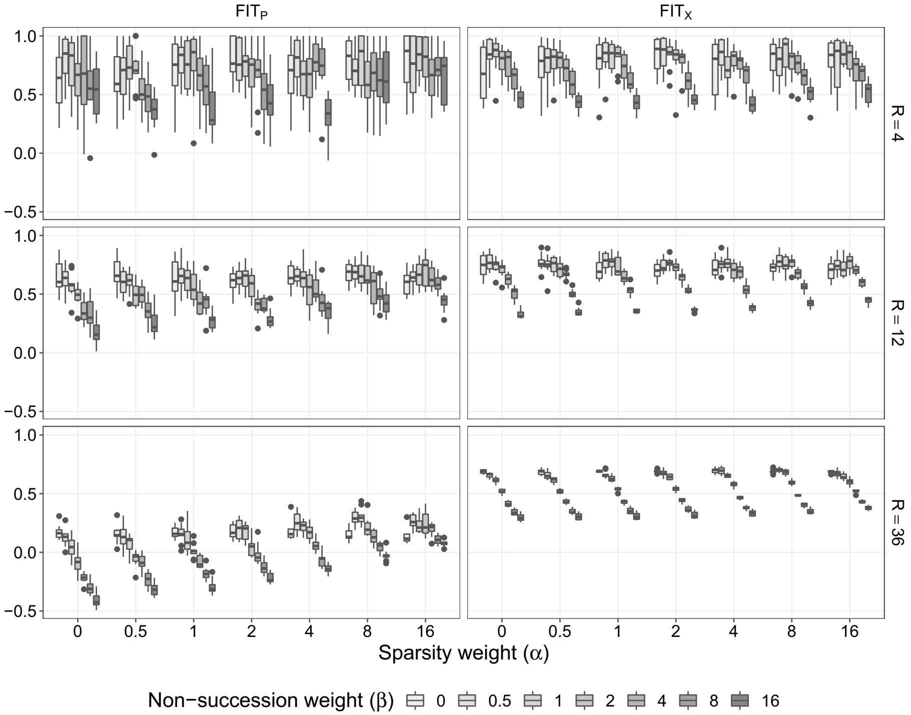


Fig. 4 FIT_P (left) and FIT_X (right) of SWoTTeD with $\omega = 3$ on synthetic data. Each graph represents a box plot for 10 runs

that SWoTTeD is ranked first, and the difference in FIT_P compared to other methods is statistically significant.

The high-quality performance of SWoTTeD can be attributed to two main factors. Firstly, SWoTTeD offers greater flexibility in reconstructing input data by allowing the overlap of different phenotypes and their arrival with a time lag. Secondly, SWoTTeD employs a loss function that assumes a Bernoulli distribution, which fits better binary data.

6.2.2 SWoTTeD against competitors on real-world datasets

In this section, we evaluate SWoTTeD against its competitors on the three real-world datasets to confirm that previous results applies on real-world data. We vary R from 4 to 36, ω from 1 to 5 for SWoTTeD, and we compare FIT_X values. It is worth noting that FIT_P can not be evaluated in this case as the hidden phenotypes are unknown. Each setting is ran 10 times with different train and test sets.

Figure 8 summarizes the results. Across all datasets, SWoTTeD achieves an average FIT_X of 0.21. PARAFAC2 achieves the second best reconstructions but we remind you that FIT_X is evaluated on the train set. CNTF has good results except for the bike dataset. Specifically, SWoTTeD outperforms all the competitors with comparable R on bike and

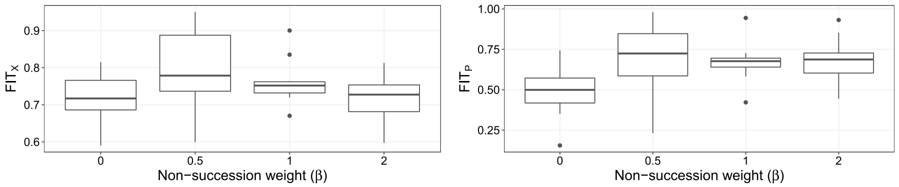


Fig. 5 Comparison of FIT_X (left) and FIT_P (right) with respect to the β hyper-parameter on a synthetic dataset with hidden phenotypes having repeated successive events

MIMIC datasets regardless of the phenotype size. On the E-shop dataset, CNTF exhibits good performances with $R = 36$ and outperforms SWoTTeD when $\omega > 1$. Nonetheless, SWoTTeD with $\omega = 3$ becomes better than CNTF. LogPar and SWIFT have the worse FIT_X on average. For the sake of graphic clarity, SWIFT is not represented in this figure. Its FIT_X values are below −0.5. The complete Figure is provided in "Appendix" (see Fig. 14, page 35).

The figure presents results for three different values of R. As expected, FIT_X of SWoTTeD increases with R. Intuitively, real-world datasets contain a diversity and a large number of hidden profiles. With more phenotypes, the model becomes more flexible and can capture the diversity in the data more effectively, resulting in more accurate tensor reconstructions.

We also observe that while FIT_X decreases slightly as the phenotype size (ω) increases, all values remain higher than those of CNTF, except for bike with $R = 36$. This suggests that SWoTTeD with $\omega > 1$ discovers phenotypes that are both complex and accurate. It may seem counter-intuitive that the FIT does not decrease. In fact, the number of model parameters increases with the size of the phenotype. However, these parameters are not completely free. Larger phenotypes also add more constraints to the reconstruction due to temporal relations, which can introduce errors when a phenotype is only partially identified. The more complex is the phenotype, the more likely there is a difference between the mean description and its instances.

The most important result of this experiment is that the reconstruction with temporal phenotypes competes with the reconstruction with daily phenotypes. This means that SWoTTeD strikes a balance between a good reconstruction of the input data and an extraction of rich phenotypes. Furthermore, the temporal phenotypes—with ω strictly higher than 1—convey a rich information to users by describing complex temporal arrangements of events.

6.2.3 Time efficiency

Figure 9 illustrates the computing times of the training process on real-world datasets. It compares the computing time of SWoTTeD with its competitors under different setting (varying values of R and ω). We have excluded SWIFT from this figure as its computing time is several orders of magnitude slower than the other competitors due to the computation of Wasserstein distances.

This figure shows that our implementation of SWoTTeD is one order of magnitude faster than CNTF or LogPar. This efficiency is attained through a vectorized implementation of

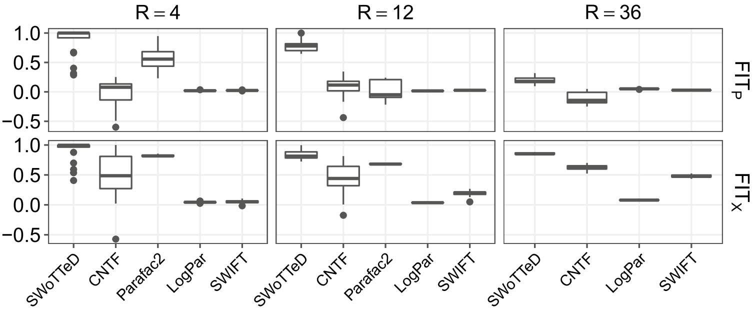


Fig. 6 FIT_P values (at the top) and FIT_X values (at the bottom) of SWoTTeD vs competitors on synthetic data with $\omega = 1$

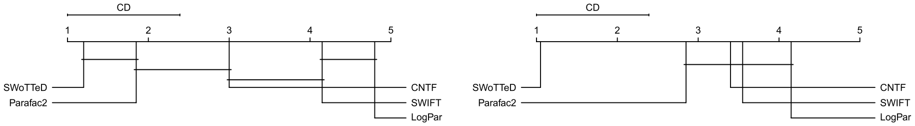


Fig. 7 Critical difference diagrams between SWoTTeD and its competitors (based on FIT_X metric on the left, and on FIT_P metric on the right). The lower the rank, the better. Horizontal bars indicate statistical non-significant difference between models. The significancy is obtained with a Wilcoxon signed-rank test ($\alpha = 0.05$)

the model. The mean computing times on a standard desktop computer[12] for SWoTTeD are $70.42s \pm 18.37$, $102.83s \pm 209.34$ and $14.34s \pm 1.03$ for the Bike, E-shop, and MIMIC datasets respectively.

Regarding the parameters of SWoTTeD, we observe that the computing time grows linearly with the number of phenotypes. More surprisingly, the size of the phenotypes has only a minor impact on computing time. This can be attributed to the efficient implementation of convolution in the PyTorch framework.

Despite the relatively high theoretical complexity of the reconstruction procedure (see Annex C), this experiment demonstrates that SWoTTeD has low computing times and can scale to handle large datasets.

6.3 SWoTTeD robustness to data noise

Noisy data are a common challenge encountered in the analysis of medical data. Physicians may make errors during data collection. Some exams may not be recorded in electronic health records and the data collection instruments themselves may be unreliable, resulting in inaccuracies within datasets. These inaccuracies are commonly referred to as noise. Noise can introduce complications as machine learning algorithms may interpret it as a valid pattern and attempt to generalize from it. Therefore, we conducted an assessment of the robustness of our model against simulated noisy data.

[12] Intel i7-1180G7, 4.60 GHz, 16 Gb RAM, without graphical acceleration.

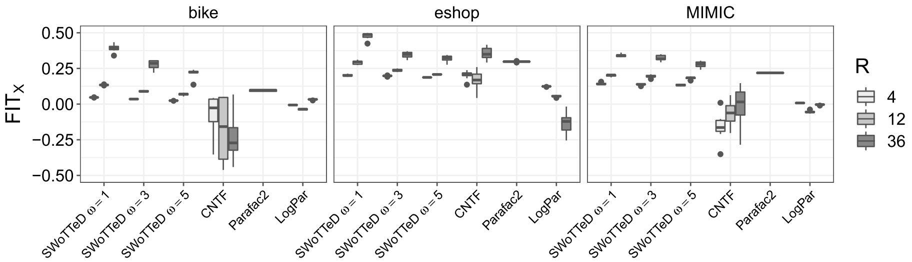


Fig. 8 Reconstruction error FIT_X of SWoTTeD ($\omega = 1, 3, 5$) and its competitors on real-world datasets and for different values of R

We considered two types of common noise that are due to data entry errors: (1) the additive noise due to occurrences of additional events in patient's hospital stays, and (2) the destructive noise, due to important events that have not been reported. We start by generating 20 synthetic datasets that are divided into training and test sets. Only the training set is disturbed, and the *FIT* value is measured over the test set. The idea behind disturbing only the training set is to assess SWoTTeD 's ability to capture meaningful phenotypes in the presence of noise that can be generalized over a non-disturbed test set. For additive noise, we inject additional events positioned randomly into the $\mathcal{X}$ tensor. The number of added events per patient is determined according to a Poisson distribution with a parameter λ. We vary λ from 2 to 25 with a stepsize of 5, except for the first step that has a value of 3.[13] The noise level is normalized by the number of ones in the dataset (*i.e.* the number of events). For instance, $\lambda = 0.3$ means that 30% of additional events have been injected into the data. Values greater than 1.0 for noise addition indicates than more than half of the events are random. For the destructive noise, we iterate over all the events of all patients in $\mathcal{X}$, and delete them based on a Bernoulli distribution with a parameter p. We vary p from 0 to 0.7 with a stepsize of 0.1.

Our focus was primarily on SWoTTeD 's ability to derive correct phenotypes from noisy data, as measured by the FIT_P metric.

Figure 10 displays the values of FIT_P obtained with various noise ratios. In the case of added events, we notice that FIT_P decreases as the average number of added events per patient increases. However, the quality of reconstruction remains above zero even when the average number of added events per patient reaches 10. In the case of deleting events, we notice that FIT_P starts to decrease when the ratio of missing events exceeds 0.3. In an extreme case where we have 70% of missing events, SWoTTeD still manages to have a positive phenotype reconstruction quality.

Consequently, we can conclude that our model exhibits robustness to noisy data, particularly in the case of missing data. This experiment further confirms the interest of tensor decomposition when data are noisy. Interestingly, adding some random noise even resulted in improved accuracy. We explain this by the relatively low number of epochs (200): some randomness in the data fasten the convergence of optimization algorithms. With fewer epochs, the model discovered better phenotypes in the presence of noise. Being robust to destructive noise is more promising. In real-world dataset, especially in care pathways that

[13] This mean that the values of λ are successively 3, 5, 10, 15, 20 and 25.

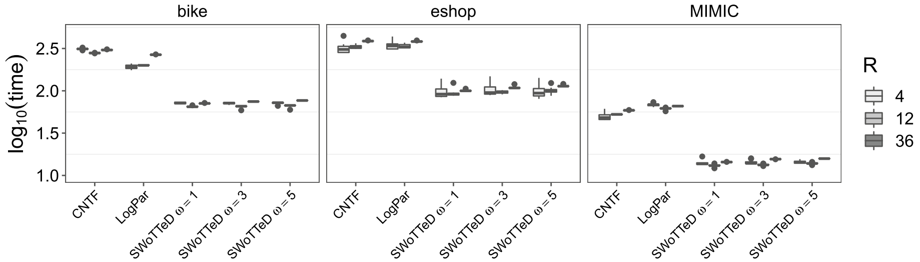


Fig. 9 Computing time in seconds (base-10 log-scale) of one run of SWoTTeD and its competitors

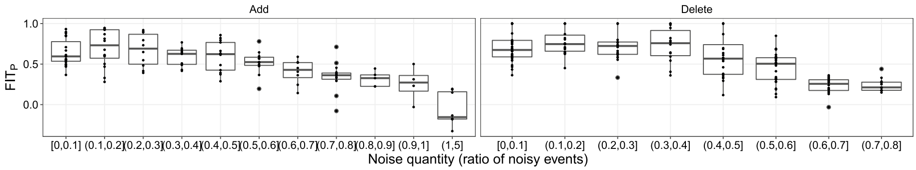


Fig. 10 FIT_P values of SWoTTeD on synthetic data as a function of normalized noise (% of new or deleted events)

is our primary application, missing events might be numerous. The results show that our model discovers the hidden phenotypes with high accuracy even with a lot of (random) missing events.

6.4 Uniqueness and diversity in SWoTTeD results

Solving tensor decomposition problems with an alternating optimization algorithm does not guarantee a convergence toward a global minimum or even a stationary point, but only to a solution where the objective function stops decreasing (Kolda and Bader, 2009). The final solution can also be highly dependent on the initialization and of the training set. Similarly, SWoTTeD does not come with convergence guarantees, but we can empirically evaluate the diversity of solutions obtained across different runs.

The experiments conducted on synthetic datasets illustrated that different runs of SWoTTeD converge toward the expected phenotypes (as detailed in Sect. 6.1). However, it can not conclusively determine the uniqueness of solutions, as it heavily relies on the random phenotypes that have been generated. We exclusively employ real-world datasets in this section.

In this experiment, we delve into the sets of phenotypes in our real-world datasets. For each dataset, we run SWoTTeD 10 times and compare the sets of phenotypes using average cosine dissimilarity.

Figure 11 depicts the cosine dissimilarity obtained with $R = 4$, 12 and 36 for SWoTTeD (with varying phenotype sizes) and CNTF. Lower dissimilarity values indicate greater similarity between phenotypes from one run to another, which is preferable.

With $R = 36$, the cosine dissimilarity is below 0.5 for all datasets. In the case of SWoTTeD, it generally exceeds 0.3. This observation suggests that there may be multiple local

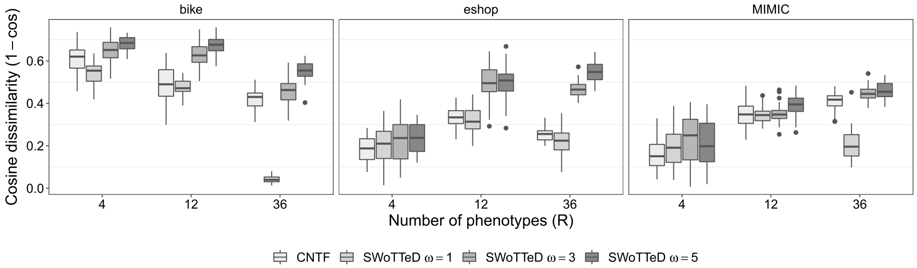


Fig. 11 Cosine dissimilarity between pairs of sets of phenotypes with respect to the phenotypes' size for SWoTTeD, and CNTF. The lower the dissimilarity, the more similar are the sets of phenotypes between each run

optima, where the optimization procedure must make choices among the phenotypes to represent. Consequently, the convergence location may depend on the initial state. The dissimilarity values show both high and low values, that correspond to have "clusters" of similar solutions.

On average, CNTF exhibits slightly better than SWoTTeD, but the difference with SWoTTeD ($\omega = 1$) is not significant across the dataset.

For $\omega = 3$ or 5, we observe higher dissimilarity between the runs. Part of this increase can be explained by the metric used: cosine similarity tends to be higher for high-dimensional vectors (*i.e.* larger phenotype sizes). This is because small differences in one dimension can lead to a significant decrease in cosine similarity, and the probability of such differences increases with the number of dimensions.

We conducted a qualitative analysis of the differences between the sets of phenotypes and found them to be almost the same. However, we observed some discrepancies with a few extra or missing events. These events are recurrent in the data, but not necessarily related to a pathway. As the number of phenotypes is limited, it is better to include such events in a phenotype to improve reconstruction accuracy. Their weak association with other events can lead to variations between runs.

Continuing our investigation of the similarities between phenotypes, we also evaluate the diversity of phenotypes within the sets of R phenotypes. We computed the pairwise cosine dissimilarity between phenotypes within each extracted set by the different runs of SWoTTeD. The objective is to evaluate each method's ability to extract a diverse set of phenotypes. It is worth noting that no orthogonality constraint, proposed by Kolda (2001) for instance, is directly implemented in SWoTTeD (nor in CNTF). The diversity is expected as a side-effect of the reconstruction loss with a small set of phenotypes. For datasets with numerous latent behaviors, a diverse set of phenotypes ensures a better coverage of the data.

Figure 12 presents the distributions of cosine dissimilarity values. In this experiment, higher cosine dissimilarity values indicate greater diversity, which is desirable. We can notice the results are correlated to the analysis of uniqueness. Diverse sets of phenotypes corresponds to robust settings. This may be explained by the fact that the diverse sets extracted the complete set.

To summarize this section, we conclude that SWoTTeD consistently converges toward sets of similar phenotypes on the real-world datasets for different runs. These sets contain diverse phenotypes, highlighting SWoTTeD 's ability to discover non-redundant latent behaviors in temporal data. Despite these promising results, we recommend running SWoTTeD multiple times on new datasets to enhance the confidence in the results.

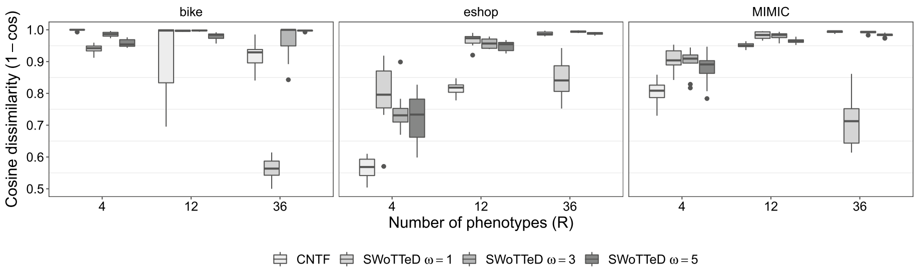


Fig. 12 Cosine dissimilarity between pairs of phenotypes with respect to the phenotypes' size for SWoTTeD, and CNTF. The higher the dissimilarity, the more diverse

7 Case study

The previous experiments demonstrated that SWoTTeD is able to accurately and robustly identify hidden phenotypes in synthetic data and to accurately reconstruct real-world data. In this section, we illustrate that the extracted temporal phenotypes are meaningful. For this purpose, we used SWoTTeD on an EHR dataset from the Greater Paris University Hospitals and showed the outputted phenotypes to clinicians for interpretation.

The objective of this case study is to describe typical pathways of patients that have been admitted into intensive care units (ICU) during the first waves of COVID-19 in the Greater Paris region, France. The typical pathways are representative of treatment protocols that have actually been implemented. Describing them may help hospitals to gain insight into their management of treatments during a crisis. In the context of COVID-19, we know that the most critical cases are patients with comorbidities (diabetes, hypertension, etc.). This complicates the analysis of these patients' care pathways because they blend multiple independent treatments. In such a situation, cutting edge tools for pathway analysis are helpful to disentangle the different treatments that have been delivered.

Care pathways of COVID-19 patients have been obtained from the data warehouse of the Greater Paris University Hospitals. We create one dataset per epidemic wave of COVID-19 for the first four waves. The periods of these waves are those officially defined by the French government. The patients selected for this study are adults (over 18 years old) with at least a positive PCR. For each patient, we create a binary matrix that represents the patient's care events (drugs deliveries and procedures) during the first 10 days of his/her stay in the Intensive Care Unit (ICU). Epidemiologists selected 85 types of care events (58 types of drugs and 27 types of procedures) based on their frequency and relevance for COVID-19. Drugs are coded using the third level of ATC[14] codes and procedures are coded using the third level of CCAM[15] codes.

In the following, we present the results obtained for the fourth wave (from 2021-07-05 to 2021-09-06) which holds 2, 593 patients and 21, 325 care events.[16] We run SWoTTeD to extract $R = 8$ phenotypes of length $\omega = 3$. We run 1, 000 epochs with a learning rate of 10^{-3}.

[14] ATC: Anatomical, Therapeutic and Chemical.

[15] CCAM is the French classification of medical procedures.

[16] Details of the dataset preparation and all phenotypes for all the waves are available on "Synthetic Dataset" and "MIMIC-IV Dataset" section of "Appendix 1".

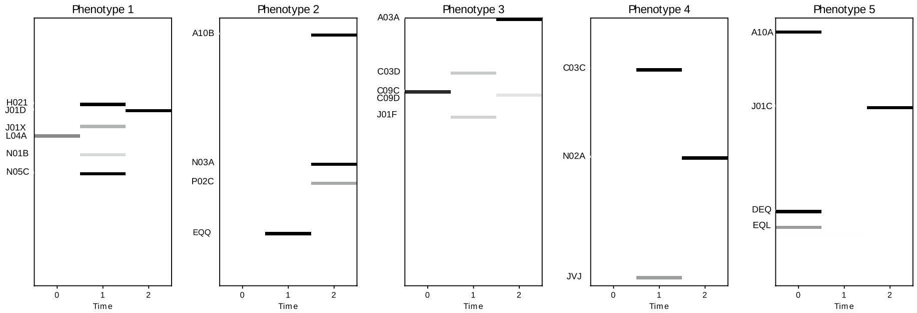


Fig. 13 Five phenotypes discovered for the 4th epidemic wave. Each gray cell represents the presence of a drug (in rows) at a relative time instant (in columns). The darker the cell, the higher the value. Cell values lie in the range [0, 1]

Figure 13 illustrates five of the eight phenotypes extracted from the fourth wave. At first glance, we can see that these phenotypes are sparse. This makes them almost easy to interpret: a phenotype is an arrangement of at most 7 care events, all with high weights. In addition, they make use of the time dimension: each phenotype describes the presence of care events on at least two different time instants. Thus, it demonstrates that the time dimension of a phenotype is meaningful in the decomposition. These phenotypes have been shown to a clinician for interpretation. It was confirmed that they reveal two relevant care combinations: some combinations of cares sketch the disease background of patients (hypertension, liver failure, etc.) while others are representative of treatment protocols. The phenotype 1 has been interpreted as a typical protocol for COVID-19. Indeed, L04A code referring to *Tocilizumab* has become a standard drug to help patients with acute respiratory problems avoid mechanical ventilation. In this phenotype, clinicians detect a switch from the prophylactic delivery of *Tocilizumab* (the first day) to a mechanical ventilation identified through the use of typical sedative drugs (N01B: *Lidocaine*, J01X: *Metronidazole* and N05C: *Midazolam*). This switch, including the discontinuation of *Tocilizumab* treatment, is a typical protocol. Nevertheless, further investigations are required to explain the presence of antibiotics (H02A: *Prednisone* and J01D: *Cefotaxime*). Phenotype 5 illustrates a severe septic shock: a patient in this situation will be monitored (DEQ), injected with *dopamine* (EQL) to induce cardiac activity, and with *insulin* (A10A) to manage the patient's glycaemia. This protocol is commonly encountered in ICU, and is applied for COVID-19 patients in critical condition.

The previously detailed phenotypes illustrate that SWoTTeD disentangles generic ICU protocols and specific treatments for COVID-19. Other phenotypes have also been readily identified by clinicians as corresponding to treatments of patients having specific medical backgrounds. The details can be found in "Appendix 4". Their overall conclusion is that SWoTTeD extracts relevant phenotypes that uncover some real practices.

8 Conclusion and perspectives

The state-of-the-art tensor decomposition methods are limited to the extraction of phenotypes that only describe a combination of correlated features occurring the same day. In this article, we proposed a new tensor decomposition task that extracts temporal phenotypes, *i.e.*, phenotypes that describe a temporal arrangement of events. We also propose SWoTTeD, a tensor decomposition method dedicated to the extraction of temporal phenotypes.

SWoTTeD has been intensively tested on synthetic and real-world datasets. The results show that it outperforms the current state-of-the-art tensor decomposition techniques on synthetic data by achieving the best reconstruction accuracies of both the input data and the hidden phenotypes. The results on real-world data show that the reconstruction competes with state-of-the art methods, and extracts information through temporal phenotypes that is not captured by other approaches.

In addition, we proposed a case study on COVID-19 patients to demonstrate the effectiveness of SWoTTeD to extract meaningful phenotypes. This experiment illustrates the relevance of the temporal dimension to describe typical care protocols.

The results of SWoTTeD are very promising and open new research lines in machine learning, temporal phenotyping and care pathway analytics. For future work, we plan to extend SWoTTeD to extract temporal phenotypes described over a variable window size. It would also be interesting to make the reconstruction more flexible for alternative applications. In particular, our temporal phenotypes are rigid sequential patterns: they describe the strict succession of days. This was expected to describe treatments in ICU, but some other applications might expect two consecutive days of a phenotype to match two days that are not strictly consecutive (with a gap in between). This is an interesting but challenging modification of the reconstruction which can be computationally expensive. Finally, another possible improvement would be to use an AO-ADMM solver (Huang et al., 2016; Roald et al., 2022), which is known to increase the stability of tensor decomposition (Becker et al., 2023).

Appendix 1: Details about experiments

Synthetic dataset

The code to generate synthetic data is available in the experiment repository: https://gitlab.inria.fr/tguyet/swotted_experiments. The file "`gen_data.py`" implements the generation of synthetic data. We also provide a jupyter notebook "`run_SWoTTeD_synthetic_data.ipynb`" that allows the reader to run readily SWoTTeD on a synthetic dataset.

MIMIC-IV dataset

The database can be accessed via this link https://mimic.mit.edu/. Once the database is set up in a Postgres database, the Jupyter notebook "`CohortConstruction.ipynb`" can be run to prepare the dataset: it builds the cohort of patients and the input tensors used in the experiments. We also provide a notebook "`run_SWoTTeD_mimic.ipynb`" to guide the reader to execute SWoTTeD on MIMIC-IV.

The availability of this data preparation enables anyone to create the exact dataset we used for our experiments and to compare his/her own results with ours.

Appendix 2: Additional results

Impact of normalization

Similar experiments were conducted without normalization. The Fig. 15 depicts the ratio of FIT_X with and without normalization. We observe that the values are very close to 1. On average, normalization appears to increase the FIT measure (ratio above 1), but it is not significant. We conclude from this experiment that normalization does not clearly improves the reconstruction accuracy neither the quality of the extracted phenotypes. Therefore, we recommend keeping the normalization constraint because it guarantees that $\mathcal{W}$'s values are limited to the range [0, 1] and they can be interpreted as probabilities.

Comparison of FIT computed on train or test datasets

The objective of this experiment is to show that the phenotypes discovered does not overfit the train dataset. To demonstrate this, we computed the FIT_X values on test and train sets.

Figure 16 presents the results obtained for our four real-world datasets. We observe that the FIT_X computed on train and test datasets are on average similar both for SWoTTeD and CNTF. It shows that neither SWoTTeD nor CNTF overfits the train dataset. This result contributes to show that the extracted phenotypes are meaningful.

Comparison of time efficiency of SWoTTeD versus FastSWoTTeD

We remind that FastSWoTTeD is an implementation of SWoTTeD for datasets having all sequences with the same length. In this case, it is possible to increase the computational efficiency of the implementation with vectorization techniques. We also remind that we did not use GPUs to accelerate the training.

In this experiment we generated 200 synthetic sequences with all same lengths ($T = 20$) and 40 features. We run the two models on the same datasets and we measure the training time, and the FIT_X on a test set.

Figure 17 depicts the computing times with respect to the number of hidden phenotypes (R). The boxplot averages the result for different model settings. We clearly see that FastSWoTTeD is more than one order of magnitude faster than SWoTTeD.

Figure 18 depicts the FIT_X obtained with the two implementations of SWoTTeD. We observe that the two implements have achieves the same FIT_X. The difference can be explained by the randomness in the training process, combined with small floating point errors that propagate differences between the decompositions. Statistically, the differences are not significant between these two models.

Appendix 3: Computational complexity of the reconstruction loss

Let $\mathcal{X}$ be a dataset with K patients, n medical features, and a maximum hospital stay length of $\bar{T}$; let R be the desired number of phenotypes described over a temporal window of size ω. The complexity to compute the reconstruction $\mathcal{P} \circledast \mathbf{W}^{(k)}$ is given by:

$$O\big(K \times \bar{T} \times R \times n \times \omega\big).$$

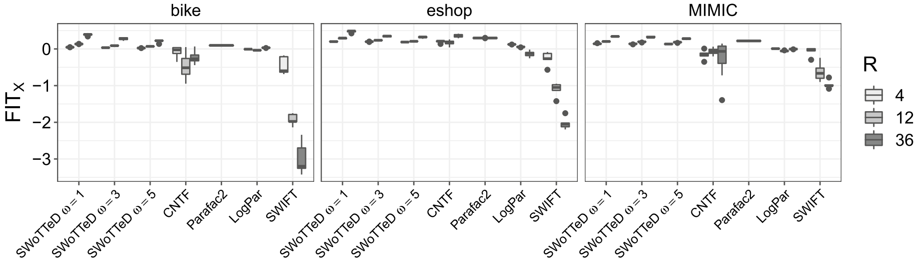


Fig. 14 Reconstruction error FIT_X of SWoTTeD ($\omega = 1, 3, 5$) and its competitor on real-world datasets and for different values of R

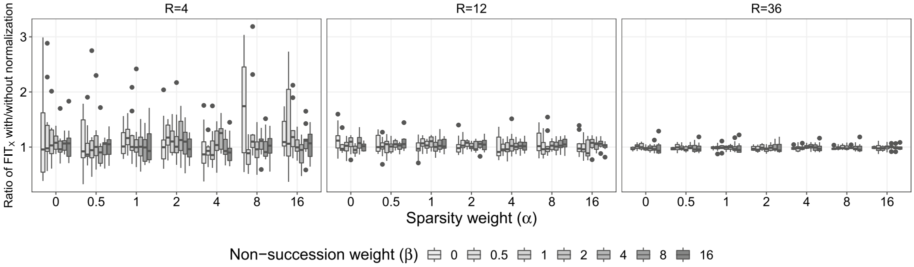


Fig. 15 Ratio of the FIT_X metric with and without normalization of SWoTTeD on synthetic datasets. A ratio above 1 indicates a benefit of using normalization on the reconstruction accuracy

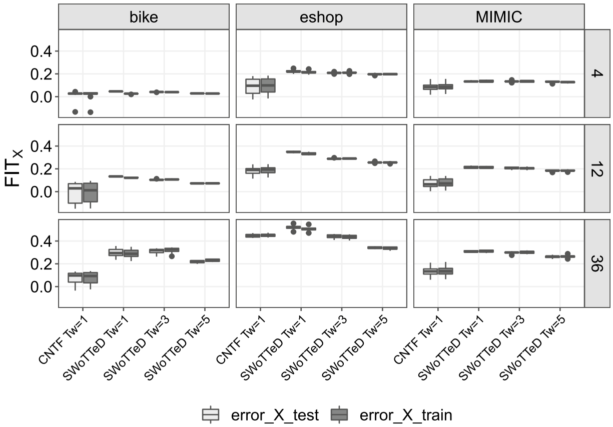
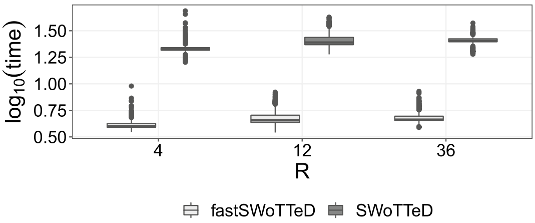


Fig. 16 FIT_X metric computed on test and train datasets for realworld datasets

Fig. 17 $\log_{10}(time)$ (in seconds) for training SWoTTeD and FastSWoTTeD models



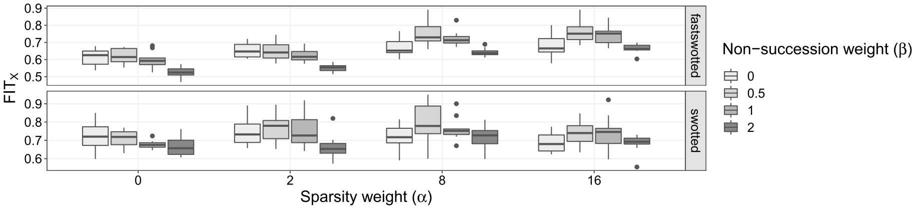


Fig. 18 FIT_X on synthetic datasets obtained with SWoTTeD and FastSWoTTeD models

It follows that the complexity to compute the error $\mathcal{L}^{\circledast}$ (see Eq. 2) is as follows:

$$O\big((R \times \omega + 1) \times K \times \bar{T} \times n\big).$$

The complexity of the loss of SWoTTeD (see Eq. 4) includes the complexity of the regularization terms, as outlined below:

- The complexity of the L_1 norm is $O(n \times \omega \times R)$ (i.e., the size of $\mathcal{P}$).
- The complexity of the non-succession regularization term $\mathcal{S}(\mathbf{W}^{(k)})$ is $O\big(K \times R \times \bar{T} \times \omega\big)$.

In comparison to non-temporal tensor decomposition techniques (e.g., CNTF), the complexity of SWoTTeD considers the ω factor that corresponds to the temporal width of phenotypes. While SWoTTeD is more expressive, it requires more computing resources than CNTF.[17] On the other hand, the reconstruction and non-succession term are based on convolution operators that benefit from material optimization. Thus, we anticipate that the efficiency of the optimization process will be preserved.

SWoTTeD is based on an optimization procedure for training the model. Each epoch comprises the computation of the loss ℓ, and its derivatives. Considering that SWoTTeD is based on automatic differentiation, the time complexity to evaluate the derivative is proportional to the computation of the loss (Verma, 2000).

[17] Here, we consider CNTF model without temporal regularization. Indeed, temporal regularization of CNTF is based on an internal LSTM architecture that is time consuming to train.

Appendix 4: Case study details

Dataset preparation

In our case study, we used data from the Greater Paris University Hospitals. For the sake of health data security and privacy, these data have been pseudo-anonymized and it is not possible to export them out of the hospital infrastructure. Then, all experiments on the AP-HP dataset were conducted on hospital servers through secured access.

In addition, all the experiments have been approved by the Ethical committee of AP-HP which is in charge to inform patients about their data usage for research purposes, to obtain their informed consent and to validate the appropriateness of data access by researchers.

In accordance with epidemiologists, we created one dataset per epidemic wave of COVID-19 for the first four waves. The periods of these waves are those officially defined by the French government[18]:

- wave 1: *2020-03-01 - 2020-07-06*, 8561 patients, 81, 517 care events
- wave 2: *2020-07-07 - 2021-01-04*, 10, 444 patients, 91, 824 care events
- wave 3: *2021-01-05 - 2021-07-05*, 14, 667 patients, 139, 045 care events
- wave 4: *2021-07-05 - 2021-09-06*, 2593 patients, 21, 325 care events

The patients selected for this study are:

- adults (over 18 years old)
- at least a positive PCR[19] test to witness the COVID-19 status of the patient
- admitted in an Intensive Care Unit (ICU). ICU includes here the historical and *de novo* units. The latter have been by set up to respond urgently to the massive arrival of patients.

For each patient we create a binary matrix that represents its care pathway.

We collected medical procedures and drugs delivered during the 10 first days in an Intensive Care Unit (ICU). All care pathways start from the entrance in an ICU. This means that all cares received at hospital before the entrance in ICU are ignored. If a patient is admitted several times in an ICU, each visit is independent of the others.

All care pathways of length strictly larger than 10 are truncated. Indeed, no further information is expected beyond the tenth day. In the final dataset, 60% of the care pathways are truncated. In addition, stays shorter than 2 days are discarded.

Drugs are coded using ATC[20] codes and procedures are coded using CCAM[21] codes. CCAM is the French Common Classification of medical procedures. Each code is a type of medical event in the input tensor. The date of occurrence of an event is determined by the number of days before the beginning of mechanical ventilation. More specifically, it means that the sampling rate is one day. In the data warehouse, procedure and drug deliveries are timestamped but, according to clinicians, these timestamps are not reliable (for instance,

[18] https://www.insee.fr/fr/statistiques/5432509?sommaire=5435421.

[19] Polymerase Chain Reaction.

[20] ATC: Anatomical, Therapeutical and Chemical.

[21] CCAM is the French classification of medical procedures.

care deliveries are entered in the system at once at the end of the day). If the length of stay is lower than 10 days (whatever the discharge reason), then the temporal dimension of its matrix is its length of stay. More specifically, all matrices do not have the same temporal dimensions.

We first selected the 500 most frequent drugs and the 200 most frequent procedures over a very large space. Then, drugs have been clustered at the 3rd ATC level, and procedures at the 3th level of CCAM coding hierarchy. Indeed, different drug names may correspond to the same kind of cares. Thus, we group drugs using the coding hierarchy of the ATC and of the CCAM. A second manual selection has been done by physicians to keep only the most relevant features in the context of COVID-19. In the end, 85 types of care events (58 types of drugs and 27 types of procedures) have been selected by physicians based on their frequency and relevance for COVID-19.

It is worth noticing that we do not have access to a reliable information about the duration of a drug prescription. For instance, we noticed that antibiotic drugs are delivered to patients on a single day each time, but not several days in a row. Nonetheless, clinicians indicates that antibiotic are always delivered for several days to be effective. This is a limitation to identify temporal phenotypes, but we prefer to not pre-process the data to prevent from having too questionable results.

The following table presents the parameters used for the results presented in Sect. D.2.

Parameter	Value
R	8 or 20
ω	3 or 5
α	0.5
β	0.5
λ	10^{-3}
epochs	$1,000$
batch size	64

Additional results on case study

In this section, we present the results obtained for the four first waves of the COVID-19 pandemic. We start by providing results about the reconstruction accuracy for several settings of SWoTTeD. Then, we illustrate the reconstruction of some patient matrices. Finally, we present the graphical and detailed descriptions of the phenotypes obtained with $R = 8$ phenotypes and $\omega = 3$.

Note that these results need additional investigations by clinicians to provide reliable interpretations. For this reason, we do not provide more detailed interpretations of phenotypes in this document.

Reconstruction accuracy

Table 2 gives the FIT_X values obtained on the four AP-HP datasets (four first waves of the pandemic). The values are close to the ones obtained on the MIMIC-IV dataset. These values are positive, indicating that the reconstruction is reasonably good on average.

Table 2 Reconstruction accuracy measured by FIT_X of the AP-HP datasets (one per wave) with different settings of SWoTTeD

Wave	$R = 8$ $\omega = 3$	$R = 20$ $\omega = 3$	$R = 20$ $\omega = 5$
w1	0.17	**0.27**	0.22
w2	0.15	**0.25**	0.23
w3	0.15	**0.23**	0.23
w4	0.15	0.25	**0.27**

Bold values are the bests

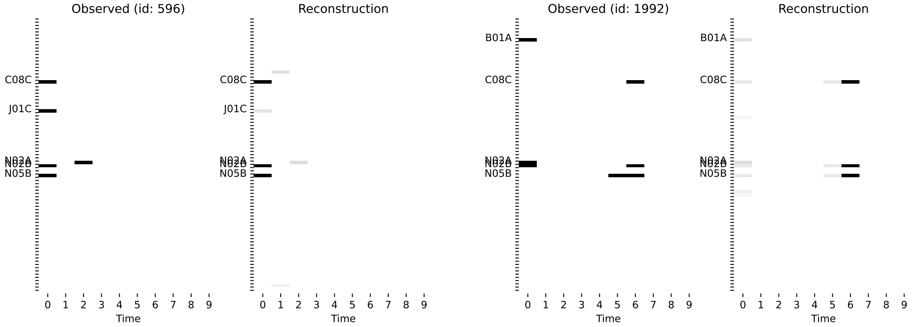


Fig. 19 Examples of patient matrix reconstruction (with $R = 8$ and $\omega = 3$). For each patient, the original matrix is on the left, the reconstructed matrix is on the right

We can notice that SWoTTeD achieves better reconstruction with $R = 20$ than with $R = 8$. Indeed, with more phenotypes, the decomposition becomes easier. The drawback of having more phenotypes is to discover simpler patterns, *i.e.* involving fewer events.

Finally, increasing the size of the temporal phenotypes does not improve the reconstruction. Larger temporal phenotypes means a more constrained reconstruction.

Reconstructions

Figure 19 illustrates the reconstructions of two patient's matrices. The patients are in the dataset of the fourth wave, and the reconstruction use $R = 8$ and $\omega = 3$.

These reconstructions illustrate the behaviour we observed: 1) a reconstruction contains all the events of the input matrix, 2) some additional events appear in the reconstruction. For the patient on the left (id : 596), the reconstruction contains, with significant weights, the five care events of the initial matrix. But, an additional event appears in the

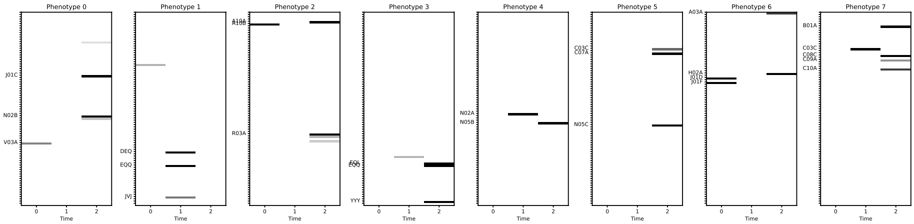


Fig. 20 Graphical representation of the phenotypes of the first wave

Table 3 Detailed description of phenotypes of the first wave

Phenotype 1				
Event code	Description	Time		
C02C	*Urapidil, hypertension*	0.00	0.00	0.17
J01C	*Amoxicilline*	0.00	0.00	1.00
N02B	*Paracetamol*	0.00	0.00	1.00
N03A	*Pregabaline*	0.00	0.00	0.33
V03A	*Oxygen*	0.61	0.00	0.00
Phenotype 2				
Event code	Description	Days		
C09C	*Irbesartan, ARB*	0.39	0.00	0.00
DEQ	*Electrocardiogram*	0.00	1.00	0.00
EQQ	*Cardiac monitoring*	0.00	1.00	0.00
JVJ	*Dialysis*	0.00	0.66	0.00
Phenotype 3				
Event code	Description	Days		
A10A	*Insuline*	0.00	0.00	1.00
A10B	*Metformine*	1.00	0.00	0.00
R03A	*Adrenergics, inhalants*	0.00	0.00	1.00
R03B	*Drugs for obstructive airway diseases*	0.00	0.00	0.39
R06A	*Antihistamines*	0.00	0.00	0.28
Phenotype 4				
Event code	Description	Days		
ENL	*Monitoring of intra-arterial pressure*	0.00	0.38	0.00
EQL	*Dopamine*	0.00	0.00	1.00
EQQ	*Cardiac monitoring*	0.00	0.00	1.00
YYY	*Rescucitation procedures*	0.00	0.00	1.00
Phenotype 5				
Event code	Description	Days		
N02A	*Morphines*	0.00	1.00	0.00
N05B	*Hydroxizine, sedation*	0.00	0.00	1.00
Phenotype 6				
Event code	Description	Days		
C03C	*Furosemide*	0.00	0.00	0.73
C07A	*Bisoprolol*	0.00	0.00	1.00
N05C	*Midazolam, sedation*	0.00	0.00	1.00
Phenotype 7				
Event code	Description	Days		
A03A	*Phloroglucinol*	0.00	0.00	0.88

Table 3 (continued)

Phenotype 7

Event code	Description	Days		
H02A	*Prednisone, antibiotic*	0.00	0.00	1.00
J01D	*Cefotaxime*	1.00	0.00	0.00
J01F	*Azithromicine*	1.00	0.00	0.00

Phenotype 8

Event code	Description	Days		
B01A	*Antithrombotic agents*	0.00	0.00	1.00
C03C	*Furosemide*	0.00	1.00	0.00
C08C	*Amlodipine*	0.00	0.00	1.00
C09A	*Ramipril*	0.00	0.00	0.54
C10A	*Statine*	0.00	0.00	0.89

reconstruction (with a low weight). For the patient on the right (*id* : 1992), the first day of the reconstruction contains a lot of lightly weighted events, but it reconstructs well the complex combination at time 6.

A possible explanation of this behaviour can be found in the weak quality of the data. As we notice, there may miss some events. The additional events in the reconstruction may witness such misinformed care deliveries. This hypothesis remains to be investigated.

Phenotypes of the first wave

Figure 20 illustrates the phenotypes that have been extracted. These phenotypes are detailed in Table 3.

Phenotypes of the second wave

Figure 21 illustrates the phenotypes that have been extracted. These phenotypes are detailed in Table 4.

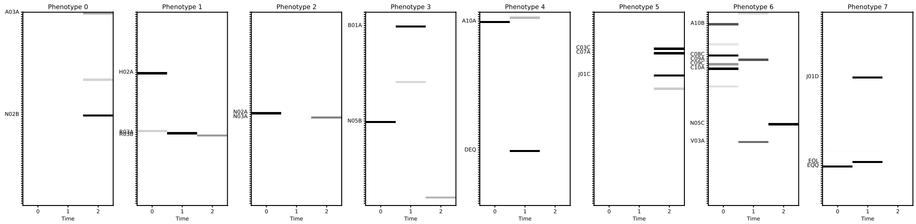


Fig. 21 Graphical representation of the phenotypes of the second wave

Table 4 Detailed description of phenotypes of the second wave

Event code	Description	Days		
Phenotype 1				
A03A	*Phloroglucinol*	0.00	0.00	0.42
J01E	*Sulfonamides and trimethoprim*	0.00	0.00	0.24
N02B	*Paracetamol*	0.00	0.00	1.00
Phenotype 2				
H02A	*Prednisone, antibiotic*	1.00	0.00	0.00
P02C	*Ivermectine*	0.26	0.00	0.00
R03A	*Adrenergics, inhalants*	0.00	1.00	0.00
R03B	*Drugs for obstructive airway diseases*	0.00	0.00	0.50
Phenotype 3				
N02A	*Morphines*	1.00	0.00	0.00
N03A	*Pregabaline*	0.00	0.00	0.63
Phenotype 4				
B01A	*Antithrombotic agents*	0.00	1.00	0.00
J01F	*Azithromicine*	0.00	0.22	0.00
N05B	*Hydroxizine, sedation*	1.00	0.00	0.00
JVJ	*Dialysis*	0.00	0.00	0.38
Phenotype 5				
A06A	*Constipation*	0.00	0.35	0.00
A10A	*Insuline*	1.00	0.00	0.00
DEQ	*Electrocardiogram*	0.00	1.00	0.00
Phenotype 6				
C03C	*Furosemide*	0.00	0.00	1.00
C07A	*Bisoprolol*	0.00	0.00	1.00
J01C	*Amoxicilline*	0.00	0.00	1.00
J01X	*Metronidazole*	0.00	0.00	0.27
Phenotype 7				
A03A	*Phloroglucinol*	0.00	0.19	0.00
A10B	*Metformine*	0.80	0.00	0.00
C03A	*Hydrochlorothiazide*	0.14	0.00	0.00
C08C	*Amlodipine*	1.00	0.00	0.00
C09A	*Ramipril*	0.00	0.80	0.00
C09C	*Irbesartan, ARB*	0.52	0.00	0.00
C10A	*Statine*	1.00	0.00	0.00
J01M	*Ciprofloxacine*	0.16	0.00	0.00
N05C	*Midazolam, sedation*	0.00	0.00	1.00
V03A	*Oxygen*	0.00	0.70	0.00
Phenotype 8				
J01D	*Cefotaxime*	0.00	1.00	0.00
DEQ	*Electrocardiogram*	0.00	0.01	0.00
EQL	*Dopamine*	0.00	1.00	0.00
EQQ	*Cardiac monitoring*	1.00	0.00	0.00

Phenotypes of the third wave

Figure 22 illustrates the phenotypes that have been extracted. These phenotypes are detailed in Table 5.

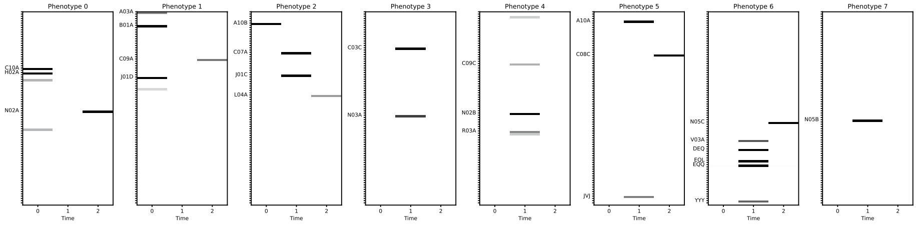


Fig. 22 Graphical representation of the phenotypes of the third wave

Table 5 Detailed description of phenotypes of the third wave

Vent code	Description	Days		
Phenotype 1				
C10A	*Statine*	1.00	0.00	0.00
H02A	*Prednisone, antibiotic*	1.00	0.00	0.00
J01E	*Sulfonamides and trimethoprim*	0.38	0.00	0.00
N02A	*Morphines*	0.00	0.00	1.00
P02C	*Ivermectine*	0.37	0.00	0.00
Phenotype 2				
A03A	*Phloroglucinol*	0.61	0.00	0.00
B01A	*Antithrombotic agents*	1.00	0.00	0.00
C09A	*Ramipril*	0.00	0.00	0.66
J01D	*Cefotaxime*	1.00	0.00	0.00
J01X	*Metronidazole*	0.21	0.00	0.00
Phenotype 3				
A10B	*Metformine*	1.00	0.00	0.00
C07A	*Bisoprolol*	0.00	1.00	0.00
J01C	*Amoxicilline*	0.00	1.00	0.00
L04A	*Tocilizumab*	0.00	0.00	0.51
Phenotype 4				
C03C	*Furosemide*	0.00	1.00	0.00
N03A	*Pregabaline*	0.00	0.88	0.00
Phenotype 5				
A06A	*Constipation*	0.00	0.26	0.00
C09C	*Irbesartan, ARB*	0.00	0.40	0.00
N02B	*Paracetamol*	0.00	1.00	0.00
R03A	*Adrenergics, inhalants*	0.00	0.60	0.00
R03B	*Drugs for obstructive airway diseases*	0.00	0.27	0.00

Table 5 (Continued)

Vent code	Description	Days		
Phenotype 6				
A10A	*Insuline*	0.00	1.00	0.00
C08C	*Amlodipine*	0.00	0.00	1.00
JVJ	*Dialysis*	0.00	0.63	0.00
Phenotype 7				
N05C	*Midazolam, sedation*	0.00	0.00	1.00
V03A	*Oxygen*	0.00	0.79	0.00
DEQ	*Electrocardiogram*	0.00	1.00	0.00
EQL	*Dopamine*	0.00	1.00	0.00
EQQ	*Cardiac monitoring*	0.01	1.00	0.01
YYY	*Rescucitation procedures*	0.00	0.74	0.00
Phenotype 8				
N05B	*Hydroxizine, sedation*	0.00	1.00	0.00

Phenotypes of the fourth wave

Figure 23 illustrates the phenotypes that have been extracted. These phenotypes are detailed in Table 6.

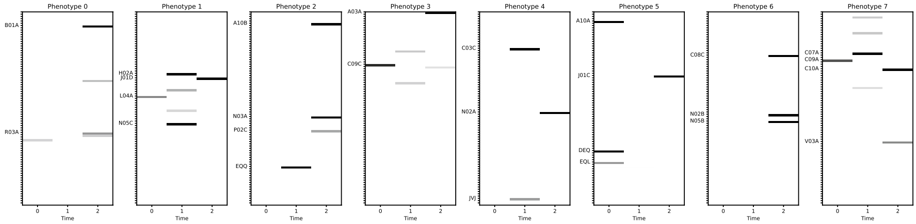


Fig. 23 Graphical representation of the phenotypes of the fourth wave

Table 6 Detailed description of phenotypes of the fourth wave

Event code	Description	Days		
Phenotype 1				
B01A	*Antithrombotic agents*	0.00	0.00	1.00
J01E	*Sulfonamides and trimethoprim*	0.00	0.00	0.33
R03A	*Adrenergics, inhalants*	0.00	0.00	0.53
R03B	*Drugs for obstructive airway diseases*	0.00	0.00	0.26
R06A	*Antihistamines*	0.23	0.00	0.00
Phenotype 2				
H02A	*Prednisone, antibiotic*	0.00	1.00	0.00
J01D	*Cefotaxime*	0.00	0.00	1.00
J01X	*Metronidazole*	0.00	0.40	0.00
L04A	*Tocilizumab*	0.59	0.00	0.00
N01B	*Lidocaine*	0.00	0.21	0.00
N05C	*Midazolam, sedation*	0.00	1.00	0.00

Table 6 (continued)

Event code	Description	Days		
Phenotype 3				
A10B	*Metformine*	0.00	0.00	1.00
N03A	*Pregabaline*	0.00	0.00	1.00
P02C	*Ivermectine*	0.00	0.00	0.45
EQQ	*Cardiac monitoring*	0.00	1.00	0.00
Phenotype 4				
A03A	*Phloroglucinol*	0.00	0.00	1.00
C03D	*Spironolactone, cardiac failure*	0.00	0.28	0.00
C09C	*Irbesartan, ARB*	0.94	0.00	0.00
C09D	*Anti-hypertensor ARB*	0.00	0.00	0.16
J01F	*Azithromicine*	0.00	0.24	0.00
Phenotype 5				
C03C	*Furosemide*	0.00	1.00	0.00
N02A	*Morphines*	0.00	0.00	1.00
JVJ	*Dialysis*	0.00	0.49	0.00
Phenotype 6				
A10A	*Insuline*	1.00	0.00	0.00
J01C	*Amoxicilline*	0.00	0.00	1.00
DEQ	*Electrocardiogram*	1.00	0.00	0.00
EQL	*Dopamine*	0.50	0.00	0.00
Phenotype 7				
C08C	*Amlodipine*	0.00	0.00	1.00
N02B	*Paracetamol*	0.00	0.00	1.00
N05B	*Hydroxizine, sedation*	0.00	0.00	1.00
Phenotype 8				
A06A	*Constipation*	0.00	0.27	0.00
C01B	*Amiodarone, cardiac failure*	0.00	0.29	0.00
C07A	*Bisoprolol*	0.00	1.00	0.00
C09A	*Ramipril*	0.81	0.00	0.00
C10A	*Statine*	0.00	0.00	1.00
J01M	*Ciprofloxacine*	0.00	0.18	0.00
V03A	*Oxygen*	0.00	0.00	0.58

Acknowledgements The authors thank the reviewers for their remarks and suggestions that greatly contributed to the improvement of this article. This work has been partly funded by the AI-RACLES Chair. The authors thank all contributors from the Clinical Data Warehouse of Greater Paris University Hospitals who work on making data available for research. The case study on data from Greater Paris Hospital has an agreement from the Scientific and Ethical Committee of AP-HP (CSE-20-11 COVIPREDS, 27/03/2020).

References

Afshar, A., Perros, I., Papalexakis, E. E., Searles, E., Ho, J., & Sun, J. (2018). COPA: Constrained PARAFAC2 for sparse and large datasets. In *Proceedings of the international conference on information and knowledge management (CIKM)* (pp. 793–802).

Afshar, A., Perros, I., Park, H., Defilippi, C., Yan, X., Stewart, W., Ho, J., & Sun, J. (2020). TASTE: Temporal and static tensor factorization for phenotyping electronic health records. In *Proceedings of the conference on health, inference, and learning (CHIL)* (pp. 193–203).

Afshar, A., Yin, K., Yan, S., Qian, C., Ho, J. C., Park, H., & Sun, J. (2021). SWIFT: Scalable Wasserstein factorization for sparse nonnegative tensors. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 6548–6556).

Ahn, D., Jang, J., & Kang, U. (2022). Time-aware tensor decomposition for sparse tensors. *Journal of Machine Learning, 111*(4), 1409–1430.

Ahuja, Y., Zou, Y., Verma, A., Buckeridge, D., & Li, Y. (2022). MixEHR-Guided: A guided multi-modal topic modeling approach for large-scale automatic phenotyping using the electronic health record. *Journal of Biomedical Informatics, 134*, 104190.

Anandkumar, A., Ge, R., Hsu, D., Kakade, S. M., & Telgarsky, M. (2014). Tensor decompositions for learning latent variable models. *Journal of Machine Learning Research, 15*, 2773–2832.

Anderson, D., Bapst, A., Coon, J., Pung, A., & Kudenov, M. (2017). Supervised non-negative tensor factorization for automatic hyperspectral feature extraction and target discrimination. In *Proceedings of SPIE—algorithms and technologies for multispectral, hyperspectral, and ultraspectral imagery* (p. 101980Q).

Baydin, A. G., Pearlmutter, B. A., Radul, A. A., & Siskind, J. M. (2018). Automatic differentiation in machine learning: A survey. *Journal of Machine Learning Research, 18*, 1–43.

Becker, F., Smilde, A. K., & Acar, E. (2023). Unsupervised EHR-based phenotyping via matrix and tensor decompositions. *WIREs Data Mining and Knowledge Discovery, 13*(4), e1494.

Bhattacharyya, A., & Vreeken, J. (2017). Efficiently summarising event sequences with rich interleaving patterns. In *Proceedings of the SIAM international conference on data mining (SDM)* (pp. 795–803).

Bro, R., Andersson, C., & Kiers, H. (1999). PARAFAC2-part II. Modeling chromatographic data with retention time shifts. *Journal of Chemometrics, 13*(3–4), 295–309.

Cohen, J. E., & Bro, R. (2018). Nonnegative PARAFAC2: A flexible coupling approach. In *International conference on latent variable analysis and signal separation* (pp. 89–98).

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research, 7*, 1–30.

Emonet, R., Varadarajan, J., & Odobez, J.-M. (2014). Temporal analysis of motif mixtures using Dirichlet processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 36*(1), 140–156.

Fanaee-T, H., & Gama, J. (2015). Eigenevent: An algorithm for event detection from complex data streams in syndromic surveillance. *Intelligent Data Analysis (IDA), 19*(3), 597–616.

Fanaee-T, H., & Gama, J. (2016). Tensor-based anomaly detection: An interdisciplinary survey. *Knowledge-Based Systems, 98*, 130–147.

Fournier-Viger, P., Lin, J.C.-W., Kiran, R. U., Koh, Y. S., & Thomas, R. (2017). A survey of sequential pattern mining. *Data Science and Pattern Recognition, 1*(1), 54–77.

Galbrun, E. (2022). The minimum description length principle for pattern mining: A survey. *Data Mining and Knowledge Discovery, 36*(5), 1679–1727.

Henderson, J., He, H., Malin, B. A., Denny, J. C., Kho, A. N., Ghosh, J., & Ho, J. C. (2018). Phenotyping through semi-supervised tensor factorization (PSST). In *Proceedings of the annual symposium of AMIA* (pp. 564–573).

Hettige, B., Wang, W., Li, Y., Le, S., & Buntine, W. L. (2020). MedGraph: Structural and temporal representation learning of electronic medical records. In *Proceedings of the European conference on artificial intelligence (ECAI)* (Vol. 325, pp. 1810–1817).

Hong, D., Kolda, T. G., & Duersch, J. A. (2020). Generalized canonical polyadic tensor decomposition. *SIAM Review, 62*(1), 133–163.

Huang, K., Sidiropoulos, N. D., & Liavas, A. P. (2016). A flexible and efficient algorithmic framework for constrained matrix and tensor factorization. *IEEE Transactions on Signal Processing, 64*(19), 5052–5065.

Jang, J.-G., & Kang, U. (2022). Dpar2: Fast and scalable PARAFAC2 decomposition for irregular dense tensors. In *Proceedings of the international conference on data engineering (ICDE)* (pp. 2454–2467).

Johnson, A., Bulgarelli, L., Pollard, T., Horng, S., Celi, L. A., & Mark, R. (2020). *MIMIC-IV* (version 0.4). PhysioNet.

Kiers, H. A., Ten Berge, J. M., & Bro, R. (1999). PARAFAC2-part I. A direct fitting algorithm for the PARAFAC2 model. *Journal of Chemometrics, 13*(3–4), 275–294.

Kolda, T. G. (2001). Orthogonal tensor decompositions. *SIAM Journal on Matrix Analysis and Applications, 23*(1), 243–255.

Kolda, T. G., & Bader, B. W. (2009). Tensor decompositions and applications. *SIAM Review, 51*(3), 455–500.

Kossaifi, J., Panagakis, Y., Anandkumar, A., & Pantic, M. (2019). TensorLy: Tensor learning in python. *Journal of Machine Learning Research, 20*(26), 1–6.

Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly, 2*(1–2), 83–97.

Lam, H. T., Mörchen, F., Fradkin, D., & Calders, T. (2014). Mining compressing sequential patterns. *Statistical Analysis and Data Mining: The ASA Data Science Journal, 7*(1), 34–52.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., et al. (2019). PyTorch: An imperative style, high-performance deep learning library. In *Proceedings of advances in neural information processing systems (NIPS)* (pp. 8024–8035).

Perros, I., Papalexakis, E. E., Wang, F., Vuduc, R., Searles, E., Thompson, M., & Sun, J. (2017). SPARTan: Scalable PARAFAC2 for large and sparse data. In *Proceedings of the international conference on knowledge discovery and data mining (SIGKDD)* (pp. 375–384).

Perros, I., Papalexakis, E. E., Vuduc, R., Searles, E., & Sun, J. (2019). Temporal phenotyping of medically complex children via PARAFAC2 tensor factorization. *Journal of Biomedical Informatics, 93*, 103125.

Pivovarov, R., Perotte, A. J., Grave, E., Angiolillo, J., Wiggins, C. H., & Elhadad, N. (2015). Learning probabilistic phenotypes from heterogeneous EHR data. *Journal of Biomedical Informatics, 58*, 156–165.

Roald, M., Schenker, C., Calhoun, V. D., Adali, T., Bro, R., Cohen, J. E., & Acar, E. (2022). An AO-ADMM approach to constraining PARAFAC2 on all modes. *SIAM Journal on Mathematics of Data Science, 4*(3), 1191–1222.

Tatti, N., & Vreeken, J. (2012). The long and the short of it: Summarising event sequences with serial episodes. In *Proceedings of the international conference on knowledge discovery and data mining (SIGKDD)* (pp. 462–470).

Varadarajan, J., Emonet, R., Odobez, J.-M. (2010). Probabilistic latent sequential motifs: Discovering temporal activity patterns in video scenes. In *Proceedings of the British machine vision conference (BMVA)* (pp. 117.1–117.11).

Verma, A. (2000). An introduction to automatic differentiation. *Current Science, 78*(7), 804–807.

Wang, Y., Chen, R., Ghosh, J., Denny, J.C., Kho, A., Chen, Y., et al. (2015). Rubik: Knowledge guided tensor factorization and completion for health data analytics. In *Proceedings of the international conference on knowledge discovery and data mining (SIGKDD)* (pp. 1265–1274).

Yang, K., Li, X., Liu, H., Mei, J., Xie, G., Zhao, J., et al. (2017). TaGiTeD: Predictive task guided tensor decomposition for representation learning from electronic health records. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 2824–2830).

Yin, K., Afshar, A., Ho, J. C., Cheung, W. K., Zhang, C., & Sun, J. (2020). LogPar: Logistic PARAFAC2 factorization for temporal binary data with missing values. In *Proceedings of the international conference on knowledge discovery and data mining (SIGKDD)* (pp. 1625–1635).

Yin, K., Cheung, W. K., Fung, B. C., & Poon, J. (2021). TedPar: Temporally dependent PARAFAC2 factorization for phenotype-based disease progression modeling. In *Proceedings of the SIAM international conference on data mining (SDM)* (pp. 594–602).

Yin, K., Cheung, W. K., Liu, Y., Fung, B. C., & Poon, J. (2018). Joint learning of phenotypes and diagnosis-medication correspondence via hidden interaction tensor factorization. In *Proceedings of the international joint conference on artificial intelligence (IJCAI)* (pp. 3627–3633).

Yin, K., Qian, D., Cheung, W. K., Fung, B. C. M., & Poon, J. (2019). Learning phenotypes and dynamic patient representations via RNN regularized collective non-negative tensor factorization. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 1246–1253).

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.