A Framework for Mining Signatures from Event Sequences and Its Applications in Healthcare Data

Fei Wang, *Member*, *IEEE*, Noah Lee, Jianying Hu, *Senior Member*, *IEEE*, Jimeng Sun, Shahram Ebadollahi, *Member*, *IEEE*, and Andrew F. Laine, *Fellow*, *IEEE*

Abstract—This paper proposes a novel temporal knowledge representation and learning framework to perform large-scale temporal signature mining of longitudinal heterogeneous event data. The framework enables the representation, extraction, and mining of high-order latent event structure and relationships within single and multiple event sequences. The proposed knowledge representation maps the heterogeneous event sequences to a geometric image by encoding events as a structured spatial-temporal shape process. We present a doubly constrained convolutional sparse coding framework that learns interpretable and shift-invariant latent temporal event signatures. We show how to cope with the sparsity in the data as well as in the latent factor model by inducing a double sparsity constraint on the β-divergence to learn an overcomplete sparse latent factor model. A novel stochastic optimization scheme performs large-scale incremental learning of group-specific temporal event signatures. We validate the framework on synthetic data and on an electronic health record dataset.

Index Terms—Temporal signature mining, sparse coding, dictionary learning, nonnegative matrix factorization, stochastic gradient descent, beta-divergence

1 Introduction

Temporal event data are ubiquitous in nature and all aspects of our everyday life. Examples are daily traces of our activities, behaviors, and decisions, recording a complex network of interactions that form part of our society. Other examples include the

1. neural firing pattern of individual neurons in our brains [19],
2. business transactions in the financial sector [10],
3. external event stimuli a robot interacts with [18], or
4. other event-related data from sensor measurements for scientific, engineering, and business applications [21], [4].

Finding latent temporal signatures is important in many domains as they encode temporal concepts such as event trends, episodes, cycles, and abnormalities. For example, in the medical domain latent event signatures facilitate decision support for patient diagnosis, prognosis, and management. In the surveillance domain temporal event signatures aid in detection of suspicious events at specific locations. Of particular interest is the temporal aspect of information hidden in event data that may be used to perform intelligent reasoning and inference about the latent relationships between event entities over time. An event entity can be a person, an object, or a location in time. For instance, in the medical domain a patient would be considered as an event entity, where visits to the doctor's office would be considered as events.

Temporal event signature mining for knowledge discovery is a difficult problem. The vast amounts of complex event data pose challenges not only for humans, but also for data and information analysis by machines. Two fundamental questions in addressing this challenge are: What is an appropriate knowledge representation for mining longitudinal event data and how can we learn such representation from large complex datasets? An event knowledge representation (EKR) should be commensurate with human capabilities so complex event data can quickly be absorbed, understood, and efficiently transformed into actionable knowledge. In this regard, several problems need to be addressed:

1. the EKR should handle the time-invariant representation of multiple event entities as two event entities can be considered similar if they contain the same temporal signatures at different time intervals or locations,
2. EKR should be flexible to jointly represent different types of event structure such as single multivariate events and event intervals to allow a rich representation of complex event relationships,
3. EKR should be scalable to support analysis and inference on large-scale databases, and

- *F. Wang, J. Hu, J. Sun, and S. Ebadollahi are with the IBM T.J. Watson Research Center, 19 Skyline Dr., Hawthorne, NY 10532. E-mail: {fwang, jyhu, jimeng, ebad}@us.ibm.com.*
- *N. Lee is with 1010data, 230 Park Avenue, 27th Floor, New York, NY 10169. E-mail: nl2168@gmail.com.*
- *A.F. Laine is with the Department of Biomedical Engineering, Columbia University, 1210 Amsterdam Avenue, New York, NY 10027. E-mail: al418@columbia.edu.*

Manuscript received 15 Feb. 2011; revised 30 Nov. 2011; accepted 28 Apr. 2012; published online 9 May 2012.
Recommended for acceptance by V. Pavlovic.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2011-02-0102.
Digital Object Identifier no. 10.1109/TPAMI.2012.111.

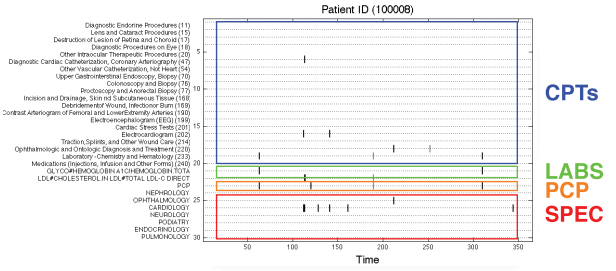


Fig. 1. An example of a diabetic patient's electronic record over one year. The x-axis corresponds to the day index, the y-axis represents different types of recorded events, which can be categorized into four groups including procedures (CPTs), lab results (LABs), visits to primary care physician (PCP), and visits to specialists (SPEC). The dots in the figure indicate the corresponding events happening at corresponding dates.

4. EKR should be sparse to enable interpretability of the learned signatures by humans.

This paper proposes a novel *Temporal Event Matrix Representation* (TEMR) and learning framework to perform temporal signature mining for large-scale longitudinal and heterogeneous event data. Basically, our TEMR framework represents the event data as a spatial-temporal matrix, where one dimension of the matrix corresponds to the type of the events and the other dimension represents the time information. In this case, if event i happened at time j with value k, then the (i, j)th element of the matrix is k. This is a very flexible and intuitive framework for encoding the temporal knowledge information contained in the event sequences. Fig. 1 illustrates a simple example on representing the longitudinal medical record of a diabetic patient over one year using our TEMR approach, where the vertical axis corresponds to the different events (including procedures, lab tests, primary care physician visits, and specialist visits), the horizontal axis represents the time information associated with these events. There is a dot in the matrix if the corresponding event happened at the corresponding time. Because of the analogy between matrix and image, TEMR offers a flexible and intuitive way of encoding comprehensive temporal knowledge, including event ordering, duration, and heterogeneity. With this representation, we develop a matrix approximation-based technology to detect the hidden signatures from the event sequences. We prove theoretically the convergence of the proposed algorithm. To improve the scalability of the proposed approach, we further developed an online updating technology. Finally, the effectiveness of the proposed algorithm is validated on a real-world healthcare dataset.

It is worthwhile to outline the advantages of the proposed approach.

First, on the knowledge representation level, TEMR provides a visual matrix-based representation of complicated event data composed of different types of events as well as event intervals, which supports the joint representation of both continuous and discrete valued data.

Second, on the algorithmic level, we propose a doubly sparse convolutional matrix approximation-based formulation for detecting the latent signatures contained in the datasets. Moreover, we derive a multiplicative updates procedure to solve the problem and proved theoretically its convergence. We further propose a novel stochastic optimization scheme for large-scale longitudinal event signature mining of multiple event entities in a group. We demonstrate that appropriate normalization constraints on the sparse latent factor model allow for automatic rank determination.

Third, on the experimental level, we have validated our approach using both synthetic data and a real-world Electronic Health Records (EHRs) dataset which contains the longitudinal medical records of over 20k patients over one year period. We report the results on the detected signatures, convergence behavior of the algorithm, and the final matrix reconstruction errors.

The rest of this paper is organized as follows: In Section 2 we outline some related work. Section 3 describes the proposed TEMR as well as the optimization approach in detail. Section 4 presents the experimental validation results on both synthetic datasets. Section 5 introduces a case study on real world dataset, followed by the conclusions and future work in Section 6.

2 Related Work

This section reviews some previous work related to this paper, which is divided into two parts. The first part reviews work on the topic of knowledge representations for temporal data mining. The second part outlines related work on nonnegative matrix factorization (NMF) and its various extensions.

2.1 Temporal Knowledge Representations

There are two types of temporal data, continuous and discrete. For knowledge representation of continuous time data, one of the most popular approaches is to transform the multivariate continuous time series into discrete symbolic representations (string, nominal, categorical, and item sets). For example, Lin et al. [12] summarized existing time series representations as data adaptive, such as Piecewise Linear Approximation (PLA), Adaptive Piecewise Constant Approximation (APCA), the Singular Value Decomposition (SVD), and Symbolic Aggregate approXimation (SAX), and non-data adaptive, such as the standard Discrete Fourier transform (DFT), Discrete Wavelet Transform (DWT), and Piecewise Aggregate Approximation (PAA).

For knowledge representation of discrete time series data, Moerchen et al. [14], [16], [15] proposed a novel *Time Series Knowledge Representation (TSKR)* as a pattern language (grammar) for temporal knowledge discovery from multivariate time series and symbolic interval data, where the temporal knowledge representation is in the form of symbolic languages and grammars that have been formulated as a means to perform intelligent reasoning and inference from time-dependent event sequences.

The TEMR framework we propose in this paper provides another alternative way to represent the temporal knowledges contained in discrete time data. Compared to the existing symbolic and grammar-based representations, our approach is more intuitive and easy to understand. Because we can always illustrate a matrix as an image, the relationships among all different types of events can clearly be observed using TEMR.

2.2 Nonnegative Matrix Factorization + Extensions

One key application of the TEMR framework in this paper is detecting latent event signatures using doubly sparse

convolutional matrix approximation technologies, which is closely related to *Nonnegative Matrix Factorization* techniques. NMF is a popular method for extracting the latent factors from nonnegative data matrix. One of the seminal works to make NMF so popular is Lee and Seung [11], where NMF was used to discover the part-based representation of facial images. Since then, many extensions have been proposed. Hoyer [8], [9] and Eggert [5] introduced sparse NMF by adding a sparsity inducing regularizer to the standard NMF objective, where the sparsity regularization further improves the model interpretability for efficient data representation. To address the dynamic nature of the data, *convolutional NMF* (cNMF) models have been proposed in Smaragdis [20] and O'Grady and Pearlmutter [17] to extract the latent sound objects from acoustic signals. Recently, in order to improve the scalability of NMF, several online optimization strategies have been proposed, such as Cao et al. [1], Mairal et al. [13], and Wang et al. [6].

The algorithm we propose in this paper is closely related to those NMF works as we also detect temporal signatures from a nonnegative event matrix using cNMF techniques. However, it is different from the existing works in the following aspects:

1. We apply cNMF on discrete time event sequences, while traditionally cNMF is used for detecting patterns from acoustic continuous signals.
2. We add sparsity regularizations on both the mined signatures and the combination coefficients because we detect those signatures from sparse discrete event sequences, while the traditional cNMF does not have such constraints.
3. We use a more general β-divergence to measure the matrix reconstruction loss, while most traditional cNMF works use Frobenius norm loss, which is a special case of the β-divergence loss.
4. We derived an efficient online optimization scheme to make our algorithm more scalable.

3 Temporal Event Signature Mining

In this section, we will introduce the details of how to detect temporal event signatures with our TEMR representation. First, we will introduce some preliminaries.

3.1 Preliminaries

Suppose we have a event matrix $\mathbf{X} \in \mathbb{R}^{n \times t}$, where n is the number of different event types, t is the length of the event sequence. As mentioned in Section 3.2, we assume $\mathbf{X}$ is the superposition of the one-side convolution of a set of hidden patterns $\mathcal{F} = \{\mathbf{F}^{(r)}\}_{r=1}^{R}$ across the time axis. We define the one-side convolutional operator $\star$ as follows:

Definition 1 (One-Sided Convolution). *The one-sided convolution of $\mathbf{F} \in \mathbb{R}^{n \times m}$ and $\mathbf{g} \in \mathbb{R}^{t \times 1}$ is an $n \times t$ matrix with*

$$(\mathbf{F} \star \mathbf{g})_{ij} = \sum_{k=1}^{t} g_{j-k+1} F_{ik}. \tag{3.1}$$

Note that $g_j = 0$ if $j \leq 0$ or $j > t$, and $F_{ik} = 0$ if $k > m$.

Thus, we can see that one-side convolution is the operation between a matrix and a vector. This operator is specially designed for detecting signatures composed of all events; thus there is no convolution on the vertical axis. Fig. 2 gives us an intuitive graphical illustration of the procedure of one-side convolution, where the bottom image is obtained through the one-side convolution of such signature on top-left and the time vector on top-right.

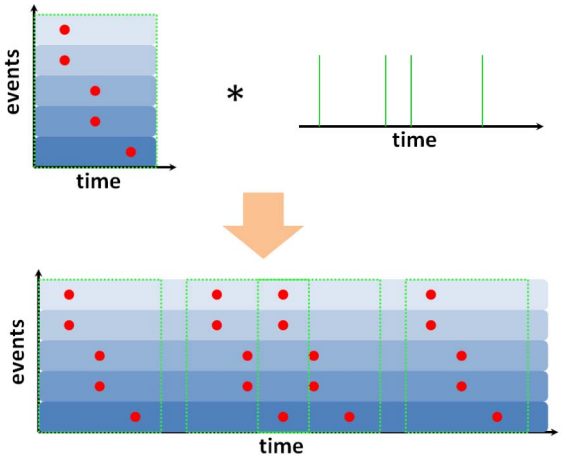


Fig. 2. A graphical illustration of one-side convolution. The top left figure shows the temporal signatures, and the top right figure is the time axis, where we use green bars to represent the position where the pattern appears. The bottom figure is the one-side convolution result, where each dotted line rectangle corresponds to a pattern.

Another important definition is the matrix β-divergence.

Definition 2 (β-divergence [7]). *The β-divergence between two matrices $\mathbf{A}$ and $\mathbf{B}$ with the same size is*

$$d_\beta(\mathbf{A}, \mathbf{B}) = \frac{1}{\beta(\beta-1)} \sum_{ij} \left(A_{ij}^\beta + (\beta-1) B_{ij}^\beta - \beta A_{ij} B_{ij}^{\beta-1}\right), \tag{3.2}$$

where $\beta \geq 0$ is a constant.

For completeness, by making use of the limit theory, we define $d_\beta(\mathbf{A}, \mathbf{B})$ for $\beta = 0$ and $\beta = 1$ as follows:

$$\begin{aligned} d_0(\mathbf{A}, \mathbf{B}) &= \lim_{\beta \to 0} \sum_{ij} \left(A_{ij} \frac{B_{ij}^{\beta-1}}{1-\beta} - \frac{A_{ij}^\beta - B_{ij}^\beta}{\beta} \right) + \frac{A_{ij}^\beta}{\beta - 1} \\ &= \sum_{ij} (A_{ij}/B_{ij} + (\log B_{ij} - \log A_{ij}) - 1), \end{aligned} \tag{3.3}$$

$$\begin{aligned} d_1(\mathbf{A}, \mathbf{B}) &= \lim_{\beta \to 1} \sum_{ij} \left(A_{ij} \frac{A_{ij}^{\beta-1} - B_{ij}^{\beta-1}}{\beta - 1} + \frac{B_{ij}^\beta - A_{ij}^\beta}{\beta} \right) \\ &= \sum_{ij} A_{ij} (\log A_{ij} - \log B_{ij}) + (B_{ij} - A_{ij}). \end{aligned} \tag{3.4}$$

β-divergence is a very general divergence: $d_0(\mathbf{A}, \mathbf{B})$, $d_1(\mathbf{A}, \mathbf{B})$, $d_2(\mathbf{A}, \mathbf{B})$ correspond to the *Itakura-Saito distance, generalized Kullback-Leighbler divergence, euclidean distance*, respectively.

3.2 Mining Signatures from a Single Event Sequence

Now coming back to our problem, we have the TEMR representation of the event matrix; the goal is to detect the latent temporal signatures from this event matrix using matrix approximation techniques.

Recall that we suppose that the event matrix $\mathbf{X}$ is constructed by the superposition of the one-side convolution of a set of hidden signatures $\mathcal{F} = \{\mathbf{F}^{(r)}\}_{r=1}^{R}$ across the time axis. Then, we propose to detect those patterns by minimizing the following objective:

$$\mathcal{J} = d_\beta\left(\mathbf{X}, \sum_{r=1}^{R} \mathbf{F}^{(r)} \star \mathbf{g}^{(r)}\right), \tag{3.5}$$

where $\mathcal{G} = \{\mathbf{g}^{(r)}\}_{r=1}^{R}$ is the set of convolutional coefficients. The problem our algorithm aims to solve is

$$\begin{aligned} \min_{\mathcal{F},\mathcal{G}} \quad & \mathcal{J} \\ s.t. \quad & \forall r = 1, 2, \ldots, R, \mathbf{F}^{(r)} \geq 0, \mathbf{g}^{(r)} \geq 0, \end{aligned} \tag{3.6}$$

where $\mathbf{g}^{(r)} \in \mathbb{R}^t$ is the coding matrix for pattern $\mathbf{F}^{(r)}$. In this paper, we consider a nonnegative matrix $\mathbf{X}$, and we also require $\{\mathbf{F}^{(r)}, \mathbf{g}^{(r)}\}_{r=1}^{R}$ to be nonnegative.[1] With the definition of β divergence (3.2), we have

$$\frac{\partial \mathcal{J}}{\partial F_{ik}^{(r)}} = \sum_{j=1}^{t} \left(Y_{ij}^{\beta-1} - X_{ij} Y_{ij}^{\beta-2}\right) \frac{\partial Y_{ij}}{\partial F_{ik}^{(r)}}, \tag{3.7}$$

where we define

$$\mathbf{Y} = \sum_{r=1}^{R} \mathbf{F}^{(r)} \star \mathbf{g}^{(r)}. \tag{3.8}$$

Combining (3.1) and (3.8), we have $\partial Y_{ij} / \partial F_{ik}^{(r)} = g_{j-k+1}^{(r)}$. Thus, we can update $F_{ik}^{(r)}$ by

$$F_{ik}^{(r)} \leftarrow F_{ik}^{(r)} \left(\frac{\sum_{j=1}^{t} X_{ij} Y_{ij}^{\beta-2} g_{j-k+1}^{(r)}}{\sum_{j=1}^{t} Y_{ij}^{\beta-1} g_{j-k+1}^{(r)}}\right)^{\eta(\beta)}, \tag{3.9}$$

where $\eta(\beta)$ is the learning rate defined as

$$\eta(\beta) = \begin{cases} \dfrac{1}{2-\beta}, & \beta < 1 \\ 1, & 1 \leq \beta \leq 2 \\ \dfrac{1}{\beta - 1}, & \beta > 2. \end{cases} \tag{3.10}$$

On the other hand, we have $\frac{\partial \mathcal{J}}{\partial g_k^{(r)}} = \sum_{i=1}^{n} \sum_{j=1}^{t} (Y_{ij}^{\beta-1} - X_{ij} Y_{ij}^{\beta-2}) F_{i,j-k+1}^{(r)}$; therefore

$$g_k^{(r)} \leftarrow g_k^{(r)} \left(\frac{\sum_{i=1}^{n} \sum_{j=1}^{t} X_{ij} Y_{ij}^{\beta-2} F_{i,j-k+1}^{(r)}}{\sum_{i=1}^{n} \sum_{j=1}^{t} Y_{ij}^{\beta-1} F_{i,j-k+1}^{(r)}}\right)^{\eta(\beta)}. \tag{3.11}$$

We have the following theorem (which is proven in the Appendix, which can be found in the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2012.111) to guarantee the convergence of the updates.

Theorem 1. *Starting from some initial guess on $\{\mathbf{F}^{(r)}, \mathbf{g}^{(r)}\}_{r=1}^{R}$ and iteratively updating them with (3.9) and (3.11) will finally converge to a stationary point.*

Proof. See the Appendix, available in the online supplemental material. □

1. It is not difficult to get the signatures when there are negative values in $\mathbf{X}$. We can just drop the nonnegativity constraints on $\mathcal{F}$, and this is similar as Semi-NMF [3] to NMF.

3.2.1 Complexity Analysis

For the storage complexity, during the iterations, it is conventional to hold $\mathbf{X}$ and $\mathbf{Y}$ in the memory, which costs $O(s_X + s_Y)$ space, where s_X and s_Y are the number of nonzero elements in $\mathbf{X}$ and $\mathbf{Y}$. We also need to hold $\mathbf{F}^{(r)}$ and $\mathbf{g}^{(r)}$ when updating themselves, which brings an additional $O(\bar{s}_F + \bar{s}_g)$ space. Here ,$\bar{s}_F$ and $\bar{s}_g$ are the averaged number of nonzero elements over $\{\mathbf{F}^{(r)}\}_{r=1}^{R}$ and $\{\mathbf{g}^{(r)}\}_{r=1}^{R}$. So, the total storage complexity is $O(s_X + s_Y + \bar{s}_F + \bar{s}_g)$.

For computational complexity, we need $O(\bar{s}_F \bar{s}_g)$ time to compute $\mathbf{Y}$, $O(2\bar{s}_F \bar{s}_g)$ time to update each $\mathbf{F}^{(r)}$ at every iteration, thus updating all $\mathcal{F} = \{\mathbf{F}^{(r)}\}_{r=1}^{R}$ over one step costs $O((2R+1)\bar{s}_F \bar{s}_g)$ time, and the complexity for updating all $\mathcal{G} = \{\mathbf{g}^{(r)}\}_{r=1}^{R}$ over one iteration is the same. Thus, the total computational complexity for OSC-NMF over T iterations is $O((4R+2)T\bar{s}_F \bar{s}_g)$.

3.2.2 Imposing the Sparsity Constraints

As shown in Fig. 1, the patient EHR matrices are very sparse. Therefore, it is natural to assume that the learned temporal pattern matrices and the convolutional coefficients are also sparse. Similarly to [5] and [8], we can enforce the sparsity constraints by adding ℓ_1 regularization terms to the objective in (3.5). As a consequence, we can solve for the optimal patterns and codes by minimizing the following objective:

$$\mathcal{J}_1 = d_\beta\left(\mathbf{X}, \sum_{r=1}^{R} \mathbf{F}^{(r)} \star \mathbf{g}^{(r)}\right) + \lambda_1 \sum_{r=1}^{R} \|\mathbf{F}^{(r)}\|_1 + \lambda_2 \sum_{r=1}^{R} \|\mathbf{g}^{(r)}\|_1, \tag{3.12}$$

where $\lambda_1 > 0$ and $\lambda_2 > 0$ are the regularization parameters, and

$$\|\mathbf{F}^{(r)}\|_1 = \sum_{ij} \left|F_{ij}^{(r)}\right|, \tag{3.13}$$

$$\|\mathbf{g}^{(r)}\|_1 = \sum_{i} \left|g_i^{(r)}\right|. \tag{3.14}$$

Then, the problem we want to solve becomes

$$\begin{aligned} \min_{\mathcal{F},\mathcal{G}} \quad & \mathcal{J}_1 \\ s.t. \quad & \forall r = 1, 2, \ldots, R, \mathbf{F}^{(r)} \geq 0, \mathbf{g}^{(r)} \geq 0. \end{aligned} \tag{3.15}$$

Similarly to the previous section, we can get the update rules for $\mathbf{F}$ and $\mathbf{g}$ as follows:

$$F_{ik}^{(r)} \leftarrow F_{ik}^{(r)} \left(\frac{\sum_{j=1}^{t} X_{ij} Y_{ij}^{\beta-2} g_{j-k+1}^{(r)}}{\sum_{j=1}^{t} Y_{ij}^{\beta-1} g_{j-k+1}^{(r)} + \lambda_1}\right)^{\eta(\beta)}, \tag{3.16}$$

$$g_k^{(r)} \leftarrow g_k^{(r)} \left(\frac{\sum_{i=1}^{n} \sum_{j=1}^{t} X_{ij} Y_{ij}^{\beta-2} F_{i,j-k+1}^{(r)}}{\sum_{i=1}^{n} \sum_{j=1}^{t} Y_{ij}^{\beta-1} F_{i,j-k+1}^{(r)} + \lambda_2}\right)^{\eta(\beta)}. \tag{3.17}$$

We can also observe that the storage and computational complexities of OSC-NMF after imposing those sparsity constraints remains the same as simple OSC-NMF.

However, as pointed out by Eggert and Korner [5], purely solving problem (3.15) may cause a scaling problem as we can always scale $\mathbf{F}$ and $\mathbf{G}$ to get the same cost

function value. To avoid this, we propose a normalization invariant formulation of problem (3.15) in the following.

3.2.3 Normalization Invariant Formulation

For the normalization invariant sparse OSC-NMF, we need to minimize the following objective with nonnegativity constraints:

$$\mathcal{J}_1^n = d_\beta\left(\mathbf{X}, \sum_{r=1}^{R} \widehat{\mathbf{F}}^{(r)} \star \mathbf{g}^{(r)}\right) + \lambda_1 \sum_{r=1}^{R} \|\widehat{\mathbf{F}}^{(r)}\|_1 + \lambda_2 \sum_{r=1}^{R} \|\mathbf{g}^{(r)}\|_1, \tag{3.18}$$

where $\widehat{\mathbf{F}}^{(r)}$ is the rth normalized signature matrix. In this paper, we will consider two types of normalization.

- *Individual normalization*. Each signature matrix is normalized to unit Frobenius norm, i.e.,

$$\widehat{F}_{ij}^{(r)} = F_{ij}^{(r)} \Big/ \sqrt{\sum_{ij} {F_{ij}^{(r)}}^2}. \tag{3.19}$$

- *Total normalization*. Each pattern matrix is normalized by the total Frobenius norm of all the signature matrices, i.e.,

$$\widehat{F}_{ij}^{(r)} = F_{ij}^{(r)} \Big/ \sqrt{\sum_{r}\sum_{ij} {F_{ij}^{(r)}}^2}. \tag{3.20}$$

Using the same trick as in [5], we can update $\mathcal{F}$ and $\mathcal{G}$ by

$$F_{ik}^{(r)} \leftarrow F_{ik}^{(r)} \left(\frac{\sum_{j=1}^{t} \left(X_{ij} + \widehat{Y}_{ij}\widehat{F}_{ik}^{(r)^2}\right)\widehat{Y}_{ij}^{\beta-2} g_{j-k+1}^{(r)} + \lambda_1 \widehat{F}_{ik}^{(r)^2}}{\sum_{j=1}^{t} \left(\widehat{Y}_{ij} + X_{ij}\widehat{F}_{ik}^{(r)^2}\right)\widehat{Y}_{ij}^{\beta-2} g_{j-k+1}^{(r)} + \lambda_1} \right)^{\eta(\beta)}, \tag{3.21}$$

$$g_k^{(r)} \leftarrow g_k^{(r)} \left(\frac{\sum_{i=1}^{n}\sum_{j=1}^{t} X_{ij}\widehat{Y}_{ij}^{\beta-2}\widehat{F}_{i,j-k+1}^{(r)}}{\sum_{i=1}^{n}\sum_{j=1}^{t} \widehat{Y}_{ij}^{\beta-1}\widehat{F}_{i,j-k+1}^{(r)} + \lambda_2} \right)^{\eta(\beta)}, \tag{3.22}$$

where

$$\widehat{Y}_{ij} = \sum_{r=1}^{R}\sum_{k=1}^{t} g_{j-k+1}\widehat{F}_{ik}^{(r)} = \left[\sum_{r=1}^{R} \widehat{\mathbf{F}}^{(r)} \star \mathbf{g}^{(r)}\right]_{ij}. \tag{3.23}$$

We can see that this normalization invariant formulation does not bring any extra storage burden, but brings an extra $O(2Rs\bar{s}_F)$ computational overhead at each iteration.

Algorithm 1 summarizes the whole procedure of the proposed OSC-NMF. Note that on line 5 the criterion we used for checking algorithm convergence is to examine the absolute difference of the objective function losses between two consecutive steps is less than a certain convergence threshold.

Algorithm 1. OSC-NMF (Individual)
Require: $\mathbf{X}, \mathcal{F}, \mathcal{G}, r, T, \beta, \lambda$
Ensure: $\mathcal{F} \geq 0, \mathcal{G} \geq 0$
1: Initialize $\mathcal{F}, \mathcal{G}$
2: **for** $i = 1$ **to** T **do**
3: Update $\mathcal{F}$ via Eq. (3.21)
4: Update $\mathcal{G}$ via Eq.(3.22)
5: **if** (converged) **then**
6: break
7: **end if**
8: **end for**
9: **return** $\mathcal{R}_\Theta = \{\mathcal{W}, \mathbf{H}\}$

3.3 Mining Signatures from Multiple Event Sequences

In many real-world scenarios, we are not only interested in discovering the signatures within a single event sequence, but also in detecting signatures from multiple event sequences. For example, in the medical domain, the event sequence of a single patient is usually very sparse. In this case, it makes more sense to detect signatures from a group of patients with similar disease conditions rather than a single patient.

More formally, we consider the case where the event matrices are composed of n event sequences. We use $\mathcal{X} = [\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n]$ to represent the event sequence group, with $\mathbf{X}_l$ representing the lth event sequence in this group. In the following we will extend our one-side convolutional NMF to this scenarios.

If we still denote the latent event signature set as $\mathcal{F} = \{\mathbf{F}^{(r)}\}_{r=1}^{R}$, then the problem we want to solve becomes[2]

$$\begin{aligned} \min_{\mathcal{F}, \{\mathcal{G}_c\}_{c=1}^{C}} \quad & \mathcal{J}_3 \\ s.t. \quad & \forall r = 1, \ldots, R; l = 1, \ldots, n_1, \\ & \mathbf{F}^{(r)} \geq 0, \mathbf{g}_l^{(r)} \geq 0, \end{aligned} \tag{3.24}$$

where $\mathcal{G} = \{\mathbf{g}_l^{(r)}\}_{i=1}^{n_1}$ is the convolution coefficients for the data, n_1 is the size of the group. Then, the objective we want to minimize is

$$\begin{aligned} \mathcal{J}_3 = & \sum_{l=1}^{n_1} d_\beta\left(\mathbf{X}_l, \sum_{r=1}^{R} \mathbf{F}^{(r)} \star \mathbf{g}_l^{(r)}\right) + \lambda_1 \sum_{r=1}^{R} \|\mathbf{F}^{(r)}\|_1 \\ & + \lambda_2 \sum_{l=1}^{n_1}\sum_{r=1}^{R} \left\|\mathbf{g}_l^{(r)}\right\|_1. \end{aligned} \tag{3.25}$$

By defining $\mathbf{Y}_l = \sum_{r=1}^{R} \mathbf{F}^{(r)} \star \mathbf{g}_l^{(r)}$, we can obtain the update rules for $\mathcal{F}$ and $\mathcal{G}$ as follows:

$$F_{ik}^{(r)} \leftarrow F_{ik}^{(r)} \left(\frac{\sum_{l=1}^{n_1}\sum_{j=1}^{t} X_{l_{ij}} Y_{l_{ij}}^{\beta-2} g_{l_{j-k+1}}^{(r)}}{\sum_{l=1}^{n_1}\sum_{j=1}^{t} Y_{l_{ij}}^{\beta-1} g_{l_{j-k+1}}^{(r)} + \lambda_1} \right)^{\eta(\beta)}, \tag{3.26}$$

$$g_{l_k}^{(r)} \leftarrow g_{l_k}^{(r)} \left(\frac{\sum_{i=1}^{n}\sum_{j=1}^{t} X_{c_{l_{ij}}} Y_{c_{l_{ij}}}^{\beta-2} F_{i,j-k+1}^{(r)}}{\sum_{i=1}^{n}\sum_{j=1}^{t} Y_{c_{l_{ij}}}^{\beta-1} F_{i,j-k+1}^{(r)} + \lambda_2} \right)^{\eta(\beta)}. \tag{3.27}$$

If we want to find normalized signatures, we can use the same trick as in [5] and derive the following update rules:

$$F_{ik}^{(r)} \leftarrow F_{ik}^{(r)} \left(\frac{\sum_{l,j} \left(X_{l_{ij}} + \widehat{Y}_{l_{ij}}\widehat{F}_{ik}^{(r)^2}\right)\widehat{Y}_{l_{ij}}^{\beta-2} g_{l_{j-k+1}}^{(r)} + \lambda_1 \widehat{F}_{ik}^{(r)^2}}{\sum_{l,j} \left(\widehat{Y}_{l_{ij}} + X_{l_{ij}}\widehat{F}_{ik}^{(r)^2}\right)\widehat{Y}_{l_{ij}}^{\beta-2} g_{l_{j-k+1}}^{(r)} + \lambda_1} \right)^{\eta(\beta)}, \tag{3.28}$$

2. Here, we directly give the sparsity constrained objective as the nonsparse case just corresponds to $\lambda_1 = \lambda_2 = 0$.

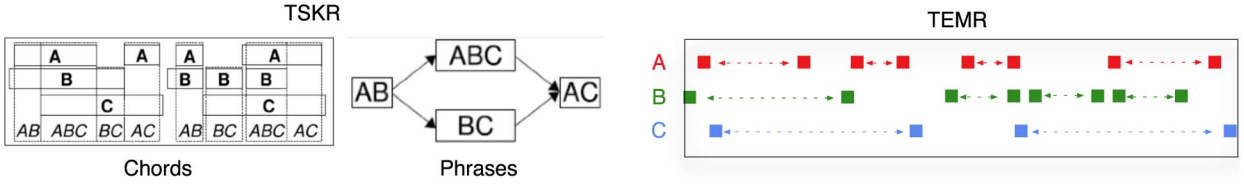


Fig. 3. TSKR and TEMR examples of Moerchen's event interval test pattern. Left: TSKR enables to distinguish between the partial ordering of so-called *Chords*. Such partial orderings form *Phrases*. Two *Chord* configurations are contained (i.e., AB-ABC-AC and AB-BC-AC). Right: TEMR can be used to emulate the same test pattern by representing an event interval with two consecutive events. The dotted arrows indicate the event interval that is marked by the colored solid squares, which denote the start and end of the interval.

$$g_{l_k}^{(r)} \leftarrow g_{l_k}^{(r)} \left(\frac{\sum_{i=1}^{n} \sum_{j=1}^{t} X_{l_{ij}} \widehat{Y}_{l_{ij}}^{\beta-2} \widehat{F}_{i,j-k+1}^{(r)}}{\sum_{i=1}^{n} \sum_{j=1}^{t} \widehat{Y}_{l_{ij}}^{\beta-1} \widehat{F}_{i,j-k+1}^{(r)} + \lambda_2} \right)^{\eta(\beta)}, \qquad (3.29)$$

where $\widehat{\mathbf{Y}}_l = \sum_{r=1}^{R} \widehat{\mathbf{F}}^{(r)} \star \mathbf{g}_l^{(r)}$.

3.3.1 Complexity Analysis

Similarly to OSC-NMF in the single event sequence case, we can analyze that the storage complexity of group OSC-NMF is $O(n(\bar{s}_X + \bar{s}_Y + \bar{\bar{s}}_g) + \bar{s}_F)$, where n is the size of the group, $\bar{s}_X, \bar{s}_Y$ are the averaged number of nonzero elements in $\{\mathbf{X}_l\}_{l=1}^{n}$ and $\{\mathbf{Y}_l\}_{l=1}^{n}$, and $\bar{\bar{s}}_g$ is the averaged number of nonzero elements of all $\{\mathbf{g}_l^{(r)}\}_{r=1,l=1}^{R,n}$. The total computational complexity is $O((4R+2)Tn\bar{s}_F\bar{\bar{s}}_g)$. For normalized cases, we just need an extra $O(2R\bar{s}_F)$ time for signature normalization.

Algorithm 2 summarizes the procedure of the group OSC-NMF algorithm.

Algorithm 2. OSC-NMF (Group)
Require: $\mathcal{X}, \mathcal{F}, \mathcal{G}, r, T, \beta, \lambda$
Ensure: $\mathcal{F} \geq 0, \mathcal{G} \geq 0$
1: Initialize $\mathcal{F}, \mathcal{G}$
2: **for** $i = 1$ **to** T **do**
3: Update $\mathcal{F}$ via Eq. (3.28)
4: Update $\mathcal{G}$ via Eq. (3.29)
5: **if** (converged) **then**
6: break
7: **end if**
8: **end for**
9: **return** $\mathcal{R}_\Theta = \{\mathcal{W}, \mathbf{H}\}$

3.3.2 A Stochastic Learning Scheme

It can be seen that group OSC-NMF is storage and time consuming if the group size n is very large. In this case, we can adopt the stochastic (online) learning scheme in [6], i.e., at each time t, the algorithm only (randomly) receives one or a small number of matrices $\mathcal{X}_t$ from the data pool, then proceeds with the following steps:

- Estimate the convolution coefficients $\mathcal{G}_t$ for $\mathcal{X}_t$ based on the current $\mathcal{F}_t$. This can be done by starting from some random initialization of $\mathcal{G}_t$, then iterating with (3.27) (or its normalized version).
- Integrating $\mathcal{X}_t$ and $\mathcal{G}_t$ with the previously received data and their estimated convolution coeffients to update $\mathcal{F}$ with (3.26) (or its normalized version).

With this scheme, when estimating $\mathcal{G}_t$ at step t, we need $O(n_t(\bar{s}_X + \bar{s}_Y) + \bar{s}_F)$ space, with n_t being the size of $\mathcal{X}_t$ and usually $n_t \ll n_1$. We also need $O(n_t(2R+1)\bar{s}_F\bar{\bar{s}}_g)$ computational time. For updating $\mathcal{F}$ from (3.26) (or (3.28)), we need to sum over all received data matrices for both numerator and denominator, thus we can save the summation results on the nominator and denominator in the previous step. Therefore, we just need to compute the corresponding summation terms on $\mathcal{X}_t$. For each round of updating $\mathbf{F}$, we need $O(n_t(\bar{s}_X + \bar{s}_Y + \bar{\bar{s}}_g) + 2\bar{s}_F)$ space and $O(n_t(2R+1)\bar{s}_F\bar{\bar{s}}_g)$ time. To conclude, the total storage complexity for this online scheme is $O(n_t(\bar{s}_X + \bar{s}_Y + \bar{\bar{s}}_g) + 3\bar{s}_F)$ and the total computational complexity is $O((4R+2)Tn_t\bar{s}_F\bar{\bar{s}}_g)$. For normalized cases, we just need to add additional $O(2R\bar{s}_F)$ computational time for pattern normalization.

4 Experiments on Synthetic Data

In this section, we will present the experiments of our proposed algorithm on several synthetic datasets.

4.1 Data Sets

We have created four sets of synthetic datasets. Each set consists of data matrices encoded with our proposed TEMR framework. All synthetic data matrices have 30 rows and 120 columns. The data matrices encode events as binary activation units in the form of a single 1-or-0 valued pixel, where a value of 1 (black) denoted an event realization and 0 (white) no event activity. Each row of the matrix refers to a particular event-type-level category and each column to a single time unit scale (e.g., days).

The first set of data is constructed to test the effectiveness of our individual OSC-NMF approach, which consists of Moerchens's TSKR event-interval-test-pattern [14], [15], [16] that has been abstracted from a tutorial figure. The pattern comprises a trivariate interval event sequence, where *Tones* (e.g., A, B, C) represent different event interval *durations*, *Chords* represent *coincidences* of *Tones*, and *Phrases* represent a *partial ordering* of the *Chords*. The pattern is shown in Fig. 3.

We have converted Moerchen's TSKR test pattern to our geometric TEMR, where an event interval is encoded with a start and end event. Moerchen's test pattern in TEMR form is shown in Fig. 3 (see right box, TEMR). In Fig. 4 (see left and middle boxes) we show two example scenarios of Moerchen's event-interval-test-pattern. The red box corresponds to the partially ordered *Phrase* (AB-ABC-AC) and the green box to (AB-BC-AC) accordingly.

The right figure in Fig. 4 shows a dataset consisted of various temporal concepts and operators including

1. synchronicity (red box),
2. trend of decreasing coincidences (green box),
3. trend of increasing coincidences (blue box),
4. concurrency (orange box).

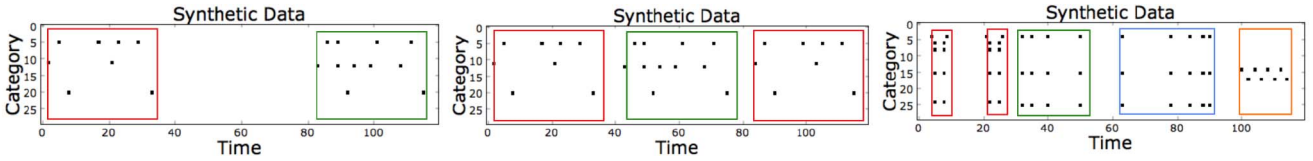


Fig. 4. Synthetic Dataset A. Left: Moerchen's event interval test pattern as outlined in Fig. 3 and separated with a large window of no event activity. Middle: Moerchen's event interval test pattern showing an alternation between Chord configurations 1 (red box) and 2 (green box). Right: From left to right, the red box shows a synthetic test pattern of synchronicity, the green box shows an event pattern trend of decreasing coincidence activity, the blue box shows an event pattern trend of increasing coincidence activity, and the last orange box shows the event pattern of concurrency. Other temporal operators and concepts such as *order*, *duration*, and *periodicity* are implicitly represented within the red, green, blue and orange enclosed patterns. Note that the synthetic datasets do not have a labeled event category specified.

The second dataset is designed for testing the *robustness* of individual OSC-NMF in the scenario where there are noisy events and different pattern elasticities contained in the data. The dataset is shown in Fig. 5, where the data in left figure contain a event sequence with one temporal signature surrounded with noisy events, and the data in right figure contain one signature with varying elasticity and noisy events surrounded.

The third set of data is constructed for testing the effectiveness of our group OSC-NMF approach, which is consisted of three data matrices that are shown in Fig. 5. The group dataset contained common and individual temporal event patterns. The red box shows temporal event pattern 1, which occurs in all three data samples. The green box shows temporal event pattern 2, which also occurs in all three data samples with multiple occurrences. The blue box shows temporal event pattern 3 that only occurs in the left and middle data samples. The orange box shows temporal event pattern 4 that only occurs in the left data sample. All patterns span a time window of seven days and an event-type dimension of 30.

The fourth dataset is created for examining the robustness of group OSC-NMF in the cases where there are noisy events and varying pattern elasticities. We have two data categories. One contains pattern I and II shown in Figs. 7a and 7b, the other contains pattern III and IV shown in Figs. 7c and 7d. We constructed 1,000 samples for each category, and the two patterns randomly appear 10 times each for every sample. For the datasets of testing noise tolerance, we randomly add different levels of events to each data matrix. For the datasets of testing pattern elasticity tolerance, we randomly add 0.3 percent noisy events, and then randomly change the levels of pattern elasticities.

4.2 Experimental Results

We conduct three sets of experiments to examine 1) the effectiveness of *individual OSC-NMF* shown in Algorithm 1, 2) the robustness of *individual OSC-NMF* shown in Algorithm 2, 3) the effectiveness of *group OSC-NMF* in Algorithm 2 as well as the stochastic training strategy.

4.2.1 The Effectiveness of Individual OSC-NMF

We examine the effectiveness of individual OSC-NMF by analyzing the reconstruction performance of the learned representation on **Synthetic Dataset A** shown in Fig. 4. Our intention for the experiments is to examine two questions: 1) Can our TEMR framework learn shift invariant interpretable latent temporal signatures? 2) Is the model sensitive to an optimally chosen rank? For this set of experiments we used the following parameter settings: $\lambda = 0.5, \beta = 0.5$. The number of iterations were set to $T = 100$ and the convergence threshold to 10^{-9}. For the Synthetic Datasets I and II we have used $m = 35$ and a rank of $R = 2$ and $R = 10$ to account for the number of true signatures in the data and their durations. For Synthetic Dataset III we have used a $m = 3$ and a rank of $R = 4$ and $R = 11$ accordingly, where $R = 11$ is an overcomplete specified rank. For each case, the first rank (i.e., $R = 2, R = 4$) was chosen based on the known number of distinct temporal signatures in the data. The second rank (i.e., $R = 10, R = 11$) was chosen as an overcomplete rank where the prespecified number of basis elements exceeds the number of true latent factors in the data. We ran 25 trials to evaluate the mean performance and standard error.

Fig. 8 shows the results of this set of experiments. One can observe that the algorithm successfully learned the correct bases set even though the rank was specified to be overcomplete. The sparse code ($\mathbf{H}$) and the sparse bases ($\mathbf{W}$) that were learned from Synthetic Datasets I, II, and III showed interpretable shift invariant sparse activation patterns. By looking at the activation codes one knows exactly when a particular latent temporal signature occurred in the data. Also the induced sparsity constraints on our model in conjunction with the nonnegativity constraints enable easy interpretation of the model. The experimental results demonstrate that our framework is able to learn shift invariant latent event signatures of different complexity.

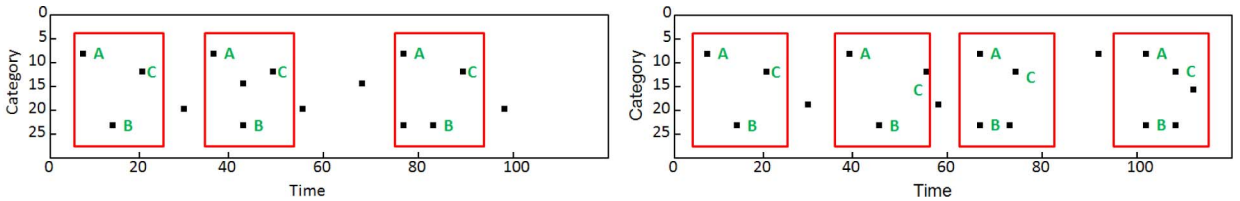


Fig. 5. Synthetic Dataset B. This dataset has two samples. The first sample (on the left) contains one repeating signature ABC but is corrupted with some noise events. The second sample (on the right) contains one signature ABC but is elasted to different length, and there are also some noisy events within or around.

TABLE 1
Reconstruction Performance for the Synthetic Datasets for Overcomplete Latent Factor Model

	D_c	$\beta = 2$	$\beta = 0.5$	I_{conv}
I	0.99(±.0288)	.47(±1.04)	.000211(±.00041)	36(±19)
II	0.96(±.0539)	.78(±1.14)	.000311(±.00041)	43(±31)
III	1.00(±.0087)	.70(±1.67)	.000236(±.00059)	54(±31)

Shown are the mean Dice coefficient (D_c), Frobenius norm ($\|\cdot\|_F$), β-divergence loss for $\beta = 0.5$, number of iterations until convergence (I_{conv}), and their standard deviations (SD).

Note that the signatures implicitly encode missing event values as no event activity is simply encoded with zeros within TEMR. Table 1 shows quantitative results of the reconstruction performance for Datasets I, II, and III, where we use three measures to evaluate the algorithm performance: the averaged iteration steps for the algorithm to converge I_{conv}, the averaged reconstruction error R_{err}, and the averaged Dice coefficient $\bar{D}_c$:

$$R_{err} = \frac{1}{T}\sum_{t=1}^{T} d_\beta(\mathbf{X}_t, \mathbf{R}_t), \tag{4.1}$$

$$\bar{D}_c = \frac{1}{T}\sum_{t=1}^{T} D_c(\mathbf{X}_t, \mathbf{R}_t). \tag{4.2}$$

The Dice coefficient is defined as:[3]

$$D_c(\mathbf{A}, \mathbf{B}) = \frac{2|\mathbf{A} \cap \mathbf{B}|}{|\mathbf{A}| + |\mathbf{B}|}, \tag{4.3}$$

where $\mathbf{A}$ and $\mathbf{B}$ are two binary matrices with the same size. $|\mathbf{A} \cap \mathbf{B}|$ counts the number of same entries in $\mathbf{A}$ and $\mathbf{B}$, $|\mathbf{A}|$ and $|\mathbf{B}|$ are the total number of entries in $\mathbf{A}$ and $\mathbf{B}$. Thus, D_c measures the set agreement between the original temporal event matrix and the reconstruction. The D_c score ranges from $[0, 1]$, where 1 means perfect agreement. For R_{err}, we report the results with $\beta = 2$ and $\beta = 0.5$. For $\bar{D}_c$, we first binarize each $\mathbf{R}_t$ with threshold 0.5. From the table we can observe that the mean Dice coefficient D_c for the overcomplete bases set for all three synthetic datasets were close to 1, which shows that the learned overcomplete representation adheres to the original data.

4.2.2 The Robustness of Individual OSC-NMF

In the second set of experiments, we examined the robustness of individual OSC-NMF in the scenario 1) when there are many noisy events; 2) when the latent temporal signatures have the same event ordering but different elasticities. We use **Synthetic Dataset B** in Fig. 5 to achieve this goal. Through the whole experimental process, we still set $\lambda = 0.5$, $\beta = 0.5$, $T = 100$, $m = 20$, $R = 5$ and the convergence threshold to 10^{-9}.

Fig. 9 (left) shows the detected signatures from the data shown in the left figure of Fig. 5, where one temporal signature ACB appeared three times. Each time this ACB appears there are some noisy events surrounding it. On the detected signature images in Fig. 9, the colors of those squares indicate the values in the corresponding signature. The figure illustrates that OSC-NMF detected three signatures in this case, where the first signature is the correct one with two gray event points brought by noisy events. The second and third signatures are generated by noisy events, which also appeared twice within the dataset. This experiment suggests that OSC-NMF can successfully detect the latent temporal signature with the existence of noisy events; however, those detected signatures may not be that "clear," i.e., with some faded background noisy events.

Fig. 9 (right) shows the detected signatures from the data shown in the right figure of Fig. 5, where the same temporal signature ACB appeared three times but with different elasticities. From the figure we can see that OSC-NMF failed to detect a correct signature in this case. The detected signature that is closest to the correct one is the first one, where there are multiple events C appearing with different values. The second and third signatures are produced by noisy events, but they also appeared multiple times within the data. This experiment suggests the signatures that OSC-NMF detected will encode all the information they demonstrated in the data. Therefore, if there is a huge variation on the elasticities of the temporal signatures, the results from OSC-NMF could be messed up, i.e., OSC-NMF is more appropriate for detecting the temporal signatures with fixed event positions or small elastic variations.

4.2.3 The Effectiveness of Group OSC-NMF

We also examined the effectiveness of the group OSC-NMF approach for detecting temporal signatures from multiple event sequences. The dataset we used in this set of experiments is **Synthetic Dataset C** shown in Fig. 6, where four different patterns appeared in three data samples. We run **Algorithm 2** with $\beta = 0.5$, $\lambda = 0.5$, $T = 100$, $R = 5, 10$, $m = 7$, and convergence threshold 10^{-9}. Fig. 10 shows the detected temporal signatures, from which we can see that although we set R to be a value large than the genuine number of underlying signatures, our algorithm can still detect the correct number of signatures contained in the dataset.

We also tested the convergence of the proposed stochastic learning scheme for group OSC-NMF introduced in Section 3.3. The result is shown in Fig. 11, which is a plot of reconstruction error with $\beta = 0.5$ versus the number of iterations. From the figure we can observe a clear convergence trend of the objective function value, with some fluctuations. This is in accordance with the previous observations on stochastic learning approaches.[4]

4.2.4 The Robustness of Group OSC-NMF

Finally, we tested the robustness of the group OSC-NMF approach introduced in Section 3.3 using Synthetic Dataset D shown in Fig. 7. We use the *Area Under the Curve* (AUC)[5] on the classification of two categories of data to measure the performance of our algorithm. During evaluation, we first partition the data into 10 folds, nine folds for training and one fold for testing. We apply our algorithm to first extract 10 patterns for each category, and then combine them to obtain a pattern dictionary of size 20. Then, each data sample will be represented as a 20D *Bag-of-Pattern* (BoP) vector, where the value on each dimension indicates the frequency of the corresponding pattern appearing in the corresponding sample. We use nearest neighbor classifier with euclidean

3. http://en.wikipedia.org/wiki/Dice_coefficient.

4. http://en.wikipedia.org/wiki/Stochastic_gradient_descent.

5. http://en.wikipedia.org/wiki/Receiver_operating_characteristic#Area_Under_Curve.

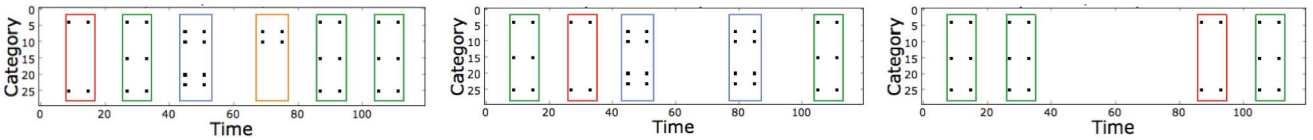


Fig. 6. Synthetic Dataset C. This is a data group containing three data samples. The patterns in red and green boxes appear in all three data samples. The pattern in the blue box appears in the left and middle data samples, while the pattern in the orange box only appears in the left data sample.

distance evaluated on BoP vectors to perform classification and report the average value and standard deviation of the 10-fold cross-validation AUC results.

We first test the noise tolerance of group OSC-NMF. Different levels of noise events (0.3, 0.5, 1, and 2 percent) are added to the datasets. The results are shown in Fig. 12. According to the dataset reconstruction, without any noise the event density of any data matrix would be $(3+3)*10/(30*120) = 1.67\%$. Therefore, if we add 2 percent noise events, the patterns will be completely corrupted and the algorithm will confuse. That is why only around 0.55 AUC is achieved. However, with less than 1 percent noise events, our algorithm can always get an AUC value of above 0.8, which suggests it can resist the noise events quite well.

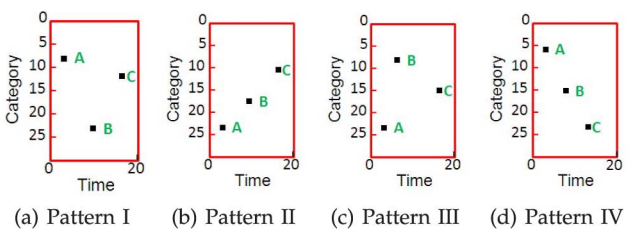
Second, we test the robustness of group OSC-NMF under different pattern elasticities. We denote the maximum pattern duration (time between A and C) to be d, and we set $d = 5, 8, 11, 14$. The position of B is also randomly chosen between A and C, and we assume events A, B, C happen in three different days. For each appearance of every pattern, we randomly sample an integer from three to d as the pattern duration. The results are demonstrated in Fig. 13, from which we can see that different pattern elasticities will not affect the algorithm performance significantly. This is because although the elasticity changes, the pattern structure (i.e., the relative position of the events) does not change. As long as the pattern duration does not exceed the pattern window length $m = 15$, our algorithm can still find them.

5 A Real-World Case Study

In this section, we will introduce a set of experiments conducted on a real-world healthcare dataset to demonstrate the effectiveness of the proposed approach.

5.1 The Dataset

The real-world dataset consisted of an Electronic Healthcare Record (EHR) data model. In conjunction with medical experts we have selected a diabetic patient pool ($\mathrm{n} = 21{,}384$) that was stratified into three groups A, B, and C. Group A consists of patients ($\mathrm{n} = 16{,}205$) with no disease complications, group B consists of patients ($\mathrm{n} = 4{,}925$) with chronic disease complications, and group C consists of patients ($\mathrm{n} = 254$) with acute complications. For all three groups we generated TEMRs for each patient using the clinical conditions defined on general outpatient encounters specific to diabetes care (see Tables 2 and 3). The chosen criteria consists of 30 different conditions that were grouped into four groups over a time period of 365 days: medical procedures ($G_1 = CPTs$), lab results ($G_2 = LABS$), primary care physician visits ($G_3 = PCP$), and specialty visits ($G_4 = SPEC$). Fig. 1 shows an example of a temporal event matrix from a patient in the diabetic patient pool of group B.



(a) Pattern I (b) Pattern II (c) Pattern III (d) Pattern IV

Fig. 7. Synthetic Dataset D. This dataset contains 2,000 samples from two categories, 1,000 from each. The samples from the first category contain patterns I and II. The samples from the second category contain patterns III and IV.

5.2 Experiment Results

In this section, we will present the results of two set of experiments: 1) investigating the performance of the proposed algorithms on this real-world dataset; 2) investigating the clinic values of the proposed algorithms.

5.2.1 Algorithm Performance Investigation

The first set of experiments was performed to analyze the reconstruction performance and convergence behavior of the learned representation for a single TEMR data matrix. A representative data sample selected from the patient pool is shown in Fig. 14, which includes multiple repeating temporal signatures. we perform cross-validation on 1,225 data samples over 25 independent trials. We examined the approximation error of group OSC-NMF as a function of different parameterizations of the β-divergence $\beta = \{0, 0.1, 0.25, 0.5, 1, 1.5, 2\}$ loss function, the degree of sparsity $\alpha = \{0, 0.5, 1, 2, 10\}$, the temporal window size $w = \{7, 14, 30, 60, 90\}$, and the rank of the factorization $R = \{1, 5, 10, 15, 20, 25, 50\}$. From this pool we examined the optimal model with respect to the rank, window size, sparsity, and parameterized loss function by computing the three performance metrics I_{conv}, R_{err}, and D_c.

Fig. 15 shows the results for the first set of experiments. One can observe that the algorithm converges within 50 iterations for all different model parameters. The approximation error measured in terms of the Dice coefficient and the Frobenius norm exponentially increased and decreased as the rank was increased. For $k > 10$, different rank sizes had an overall approximation error above $D_c > 0.9$ and $\|\cdot\|_F < 2.5$. The reconstruction performance showed that the algorithm is robust against varying window sizes and the sparsity parameter for $\lambda > 0$. The effect on different sparsity constraints showed that the mean convergence is not indicative of a low approximation error. The best model was achieved with a sparsity constraint of $\lambda = 0.5$ and a $\beta = 0.5$. Setting the sparsity constraint to $\lambda = 0$ led to a very low Dice coefficient. Also setting $\beta = 2$ gave the lowest Dice coefficient, showing that the Frobenius loss is not able to cope with double sparsity.

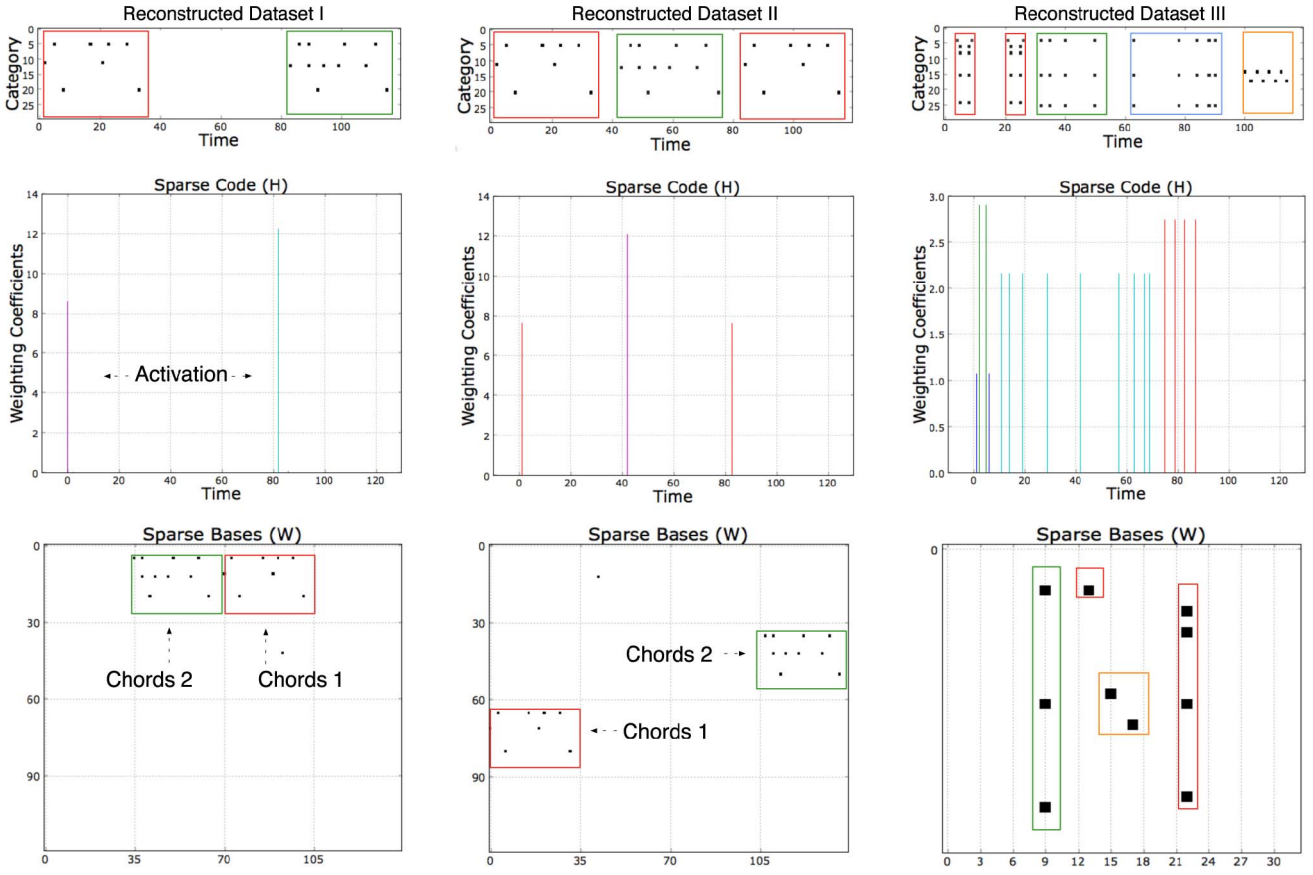


Fig. 8. Reconstruction performance for the synthetic dataset. Left: Synthetic Dataset I. The first row shows the reconstructed dataset based on the learned model, the second row shows the sparse code $\mathbf{H}$, and the last row shows the sparse bases $\mathbf{W}$. Middle: Corresponding figures of Synthetic Dataset II as specified in the left column. Right: Corresponding figures of Synthetic Dataset III as specified in the left column. The weighting coefficients in $\mathbf{H}$ were colored based on an arbitrary random color map. What is important is that one color corresponds to one basis element. The temporal patterns of interest in the first row are color-coded with the latent temporal patterns found in $\mathbf{W}$.

We summarize the optimal model parameters with respect to the computed performance metrics in Table 4. From Table 4 one can see that the convergence criterion should not be considered as a cross-validation measure. The optimal mean Dice coefficient and mean ℓ_2-norm both gave the same optimal model parameters, whereas the parameters for the convergence criterion disagreed. In general, the framework shows robustness with respect to the chosen window size and the sparsity parameter. This is encouraging since learning patterns of different window sizes is important for extracting a rich event structure within TEMR. Also, the optimal parameterization of the β-divergence with $\beta = 0.5$ shows that it outperforms the Itakura-Saito and generalized KL divergence.

In the second set of experiments, we examined the approximation error (mean Dice coefficient) of the stochastic gradient descent scheme as a function of the factorization rank R and window size m for two different settings of the stochastic learning scheme for group OSC-NMF in Algorithm 2. The motivation was to investigate the reconstruction performance of the stochastic optimization scheme on real data for different ranks and window sizes for all three population groups A, B, and C. We implemented the stochastic learning method with two different settings (type I and type II). In algorithm type I we update $\mathbf{W}$ and $\mathbf{H}$ over 1 iterations at each updating step, whereas in algorithm type II (green) we update $\mathbf{W}$ and $\mathbf{H}$ over 100 iterations at each updating step. We adopted similar parameter settings as in the first set of experiments, i.e., $\beta = 0.5$, $\lambda = 0.5$, $T = 100$, and the convergence threshold 10^{-9}. We varied $R = \{1, 5, 10, 50, 100, 200, 500, 1{,}000, 5{,}000, 10{,}000\}$ and for the window size $m = \{3, 7, 14, 30\}$.

Fig. 16 shows the experimental results. We only report graphical results for group A since the mean Dice coefficient plots for group B and C showed similar trends. For all three

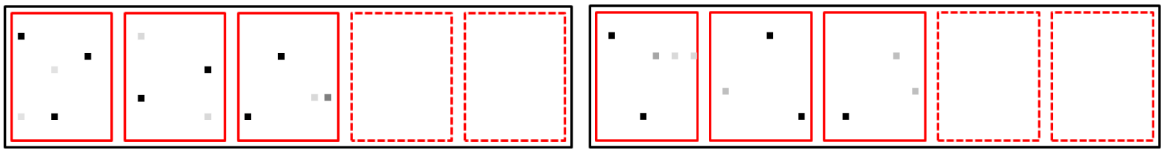
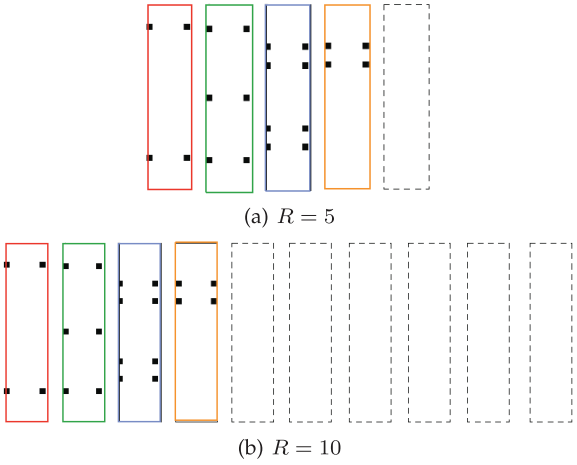


Fig. 9. Detected temporal signatures from Synthetic Dataset B. The left figure shows the temporal signatures detected from the event data in the left figure of Fig. 5, and the right figure shows the temporal signatures detected from the event data in the right figure of Fig. 5. The intensities of the black squares indicate the values of the corresponding signature, where white is 0 and black is 1.



(a) $R = 5$

(b) $R = 10$

Fig. 10. Detected temporal signatures from Synthetic Dataset C. (a) The results when we set $R = 5$. (b) The results when we set $R = 10$.

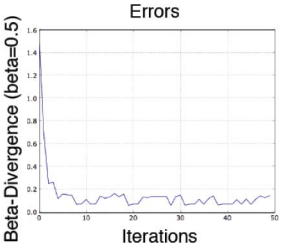


Fig. 11. Convergence curve of the Stochastic Learning scheme for Group OSC-NMF. The x-axis represents the number of iterations, and the y-axis corresponds to the reconstruction loss measured by β-divergence with $\beta = 0.5$.

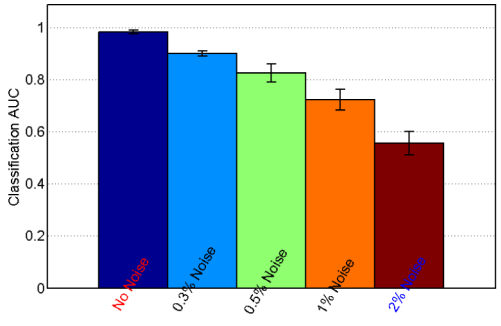


Fig. 12. Classification AUC with different noise levels.

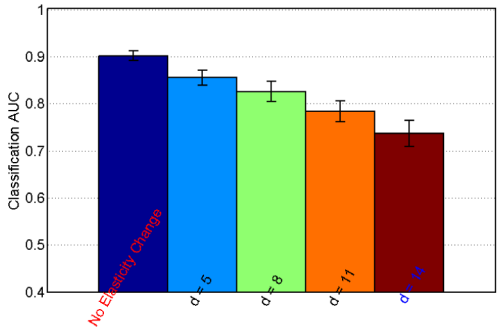


Fig. 13. Classification AUC with varying pattern elasticities.

groups A, B, and C, algorithm type II outperformed algorithm type I. Algorithm type I showed a linear increase and algorithm type II an exponential increase of the Dice coefficient as the rank increased. The reconstruction performance is robust against the window size and the number of basis elements (rank). For all three groups the stochastic

TABLE 2
Clinical Conditions for Diabetic Patient Encounters

CPT Code	G_1 Description
11	Diagnostic Endorine Procedures
15	Lens and Cataract Procedures
17	Destruction of Lesion of Retina and Choroid
18	Diagnostic Procedures on Eye
20	Other Intraocular Therapeutic Procedures
47	Diagnostic Cardiac Catheterization, Coronary Arteriography
54	Other Vascular Catheterization, Not Heart
70	Upper Gastrointerstinal Endoscopy, Biopsy
76	Colonoscopy and Biopsy
77	Proctoscopy and Anorectal Biopsy
168	Incision and Drainage, Skin nd Subcutaneous Tissue
169	Debridementof Wound, Infectionor Burn
190	Contrast Arteriogram of Femoral and LowerExtremity Arteries
199	Electroencephalogram (EEG)
201	Cardiac Stress Tests
202	Electrocardiogram
214	Traction,Splints, and Other Wound Care
220	Ophthalmologic and Ontologic Diagnosis and Treatment
233	Laboratory -Chemistry and Hematology
240	Medications (Injections, Infusion and Other Forms)

The table shows one of the four event-group level categories G_1 and their respective event-type levels.

TABLE 3
Clinical Conditions for Diabetic Patient Encounters

LABS	G_2 Description
	GLYCO and HEMOGLOBIN A1C/HEMOGLOBIN.TOTA
	LDL, CHOLESTEROL.IN LDL, and TOTAL LDL-C DIRECT
PCP	G_3 Description
	General Primary Care Physician Visits
SPECIALTY	G_4 Description
	NEPHROLOGY
	OPHTHALMOLOGY
	CARDIOLOGY
	NEUROLOGY
	PODIATRY
	ENDOCRINOLOGY
	PULMONOLOGY

The table shows the last three out of four event-group level categories G_2, G_3, and G_4 and their respective event-type levels.

optimization scheme could learn the latent temporal signature representation that leads to a mean Dice coefficient close to 1. The 95 percent confidence interval also showed that the computed means were representative of the three population groups. Visual examination of the learned patterns also confirmed that the algorithm could learn interpretable latent event patterns for all three groups.

5.2.2 Clinic Value Investigation

The objective of this set of experiments is to determine whether the number and severity of diabetes complications are associated with increased risk of mortality and hospitalizations. We use the DCSI, a discrete 13-point scale proposed by Young et al. [22], to stratify the three groups of our diabetic patient pool and to use the obtained severity score as group labels to correlate against HRU patterns. The DCSI score was derived from diagnostic, pharmacy, and

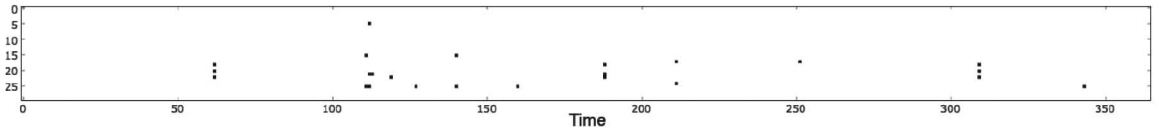


Fig. 14. A representative data sample with repeating temporal signatures.

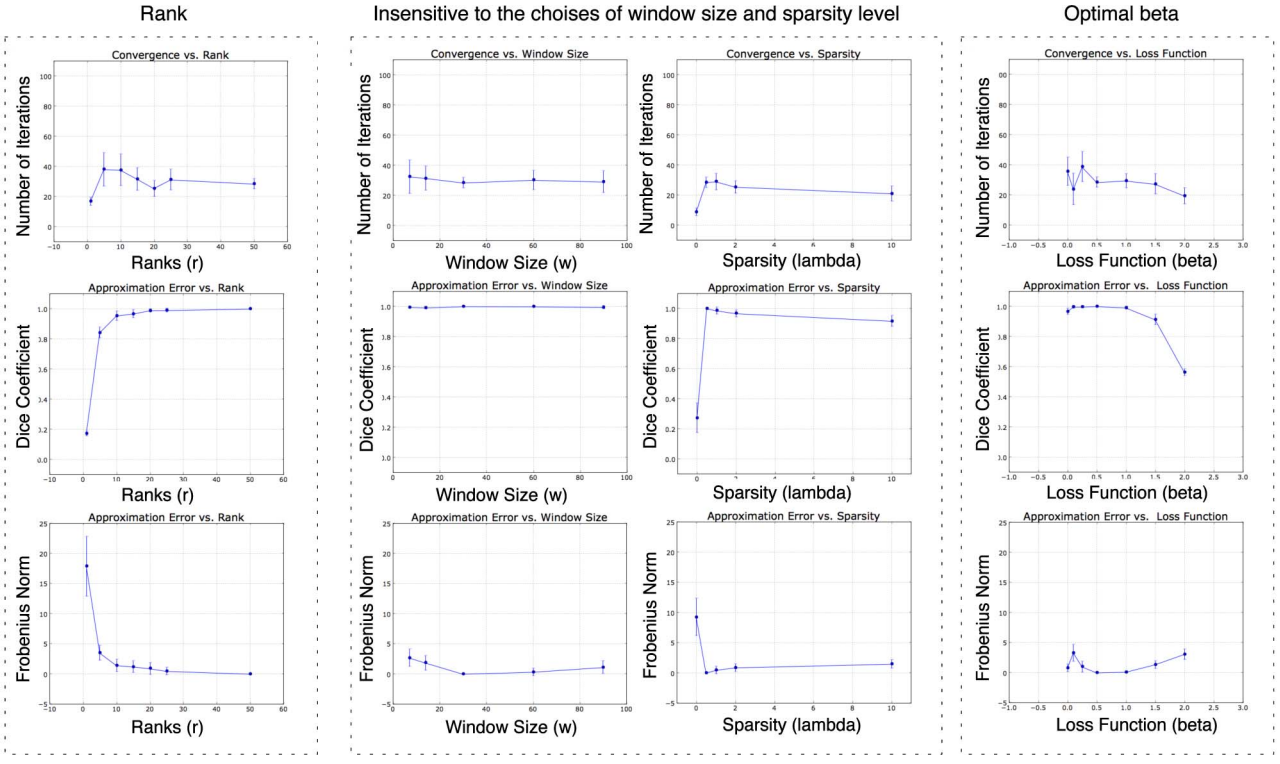


Fig. 15. Cross-validation test on the real-world dataset. Shown are mean performance measures with 95 percent confidence intervals. Row 1: Mean convergence versus the model parameters (rank, window size, sparsity, and different loss functions of the β-divergence). Row 2: Mean Dice coefficient versus the model parameters. Row 3: Mean Frobenius norm versus the model parameters.

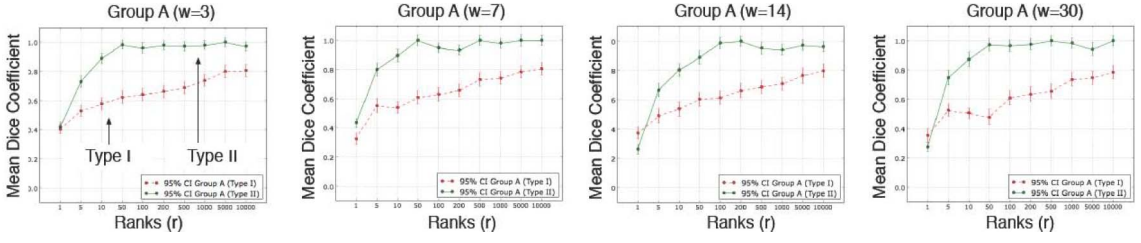


Fig. 16. Performance comparison of the Stochastic Learning Scheme for group OSC-NMF. We use the mean Dice coefficients for Group A as a function of different ranks (1-10,000) and window sizes (3-30) and their 95 percent confidence intervals. The green curve corresponds to setting where we update $\mathbf{W}$ and $\mathbf{H}$ over 100 iterations at each updating step (type II), while the red curve corresponds to the setting where we update $\mathbf{W}$ and $\mathbf{H}$ over only one iteration at each updating step (type I).

laboratory data to quantify the severity of complications and to potentially better predict the risk of adverse outcomes.

We performed an exploratory analysis of the diabetic patient pool by aggregating all the patients from group A, B, and C to assess how the detected temporal signatures relate to the severity of diabetic complications. We learned 30 weekly, biweekly, monthly, and quarterly temporal signatures for all patients using group OSC-NMF. The most frequent temporal signatures include *repeated high Hemoglobin A1C value, repeated cardiac disease related procedure, repeated lab test (CPT code 233) and co-occurrence of high Hemoglobin A1C value and high Cholesterol*, where cardiac disease is a common comorbidity of diabetes. Then, we represent each patient using a 30D BoP vector in the same way as in Section 4.2.4. Those vectors will be further normalized to unit norm and the cosine distance between pairwise normalized BoP vectors will be used as pairwise patient distances. Finally, we computed a kNN graph to examine the latent cluster structure of the mined latent patterns by looking at the Fiedler vector [2], which is the eigenvector

TABLE 4
Permutation Test for Cross Validation on a Real-World Dataset

Performance Criteria	λ	tw	β	r
Optimal mean convergence	2.0	7	0.5	1
Optimal mean Dice coefficient	0.5	30	0.5	50
Optimal mean ℓ_2-norm	0.5	30	0.5	50

Cross-validation results with respect to mean convergence (number of iterations), the reconstruction error (ℓ_2)-norm, and the Dice coefficient.

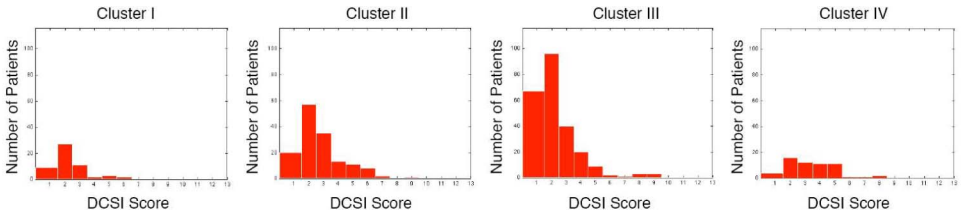


Fig. 17. Temporal signature groups versus diabetic disease severity level. Histograms that show the number of patients within each computed cluster versus their DCSI severity score.

corresponding to the second smallest eigenvalue of the Laplacian matrix of the kNN graph. Different cluster groups were computed, together with the DCSI score for each patient. We generated a histogram that captured the patient distribution in each cluster. We performed visual examination of the patient distribution based on their severity level to look for group specific differences.

Fig. 17 shows an example of a four cluster partitioning of a random subset of our diabetic patient population. One can infer that the identified patterns in cluster IV mostly occur in groups of patients with a high DCSI score. Taking a closer look to cluster IV one can see the low number of patients with a low severity score (i.e., 1) in contrast to the overall histogram shape. The majority of patients in cluster IV exhibit a higher DCSI score and thus have higher risk of hospitalization and mortality. Clusters II and III show similar shapes of the overall histogram, indicating that the learned patterns within these patient groups mainly consist of common temporal signatures that are not indicative of disease severity. The longer right tail of the histogram can be explained by the rarity of patients who have a very high DCSI score. We note that one can go back to the individual patterns to investigate what kind of care the patients received.

6 Conclusion

In this paper, we have presented a novel temporal event matrix representation and learning framework in conjunction with an in-depth validation on both synthetic and real world datasets. The framework has wide applicability to a variety of data and application domains that involve large-scale longitudinal event data. We have demonstrated that our proposed framework is able to cope with the double sparsity problem and that the induced double sparsity constraint on the β-divergence enables automatic relevance determination for solving the optimal rank selection problem via an overcomplete sparse latent factor model. Further, the framework is able to learn shift invariant high-order latent event patterns in large-scale data. We empirically showed that our stochastic optimization scheme converges to a fixed point and we have demonstrated that our framework can learn the latent event patterns within a group. Future work will be devoted to a thorough clinical assessment for visual interactive knowledge discovery in large electronic health record databases.

Acknowledgments

The authors would like to thank Robert Sorrentino (MD), Martin Kohn (MD), and Arno Klein (PhD). Noah Lee performed this work during a summer internship at IBM T.J. Watson Research Center and was supported by the IBM PhD Fellowship Award 2010-2011.

References

[1] B. Cao, D. Shen, J.T. Sun, X. Wang, Q. Yang, and Z. Chen, "Detect and Track Latent Factors with Online Nonnegative Matrix Factorization," *Proc. 20th Int'l Joint Conf. Artificial Intelligence,* pp. 2689-2694, 2007.

[2] F.R.K. Chung, *Spectral Graph Theory.* Am. Math. Soc., 1997.

[3] C. Ding, T. Li, and M.I. Jordan, "Convex and Semi-Nonnegative Matrix Factorizations," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 32, no. 1, pp. 45-55, Jan. 2010.

[4] M. Dong, "A Tutorial on Nonlinear Time-Series Data Mining in Engineering Asset Health and Reliability Prediction: Concepts, Models, and Algorithms," *Math. Problems in Eng.,* vol. 2010, pp. 1-23, 2010.

[5] J. Eggert and E. Korner, "Sparse Coding and NMF," *Proc. IEEE Int'l Joint Conf. Neural Networks,* vol. 2, pp. 2529-2533, 2004.

[6] W. Fei, L. Ping, and K. Christian, "Online Nonnegative Matrix Factorization for Document Clustering," *Proc. 11th SIAM Int'l Conf. Data Mining,* 2011.

[7] C. Févotte and J. Idier, *Algorithms for Nonnegative Matrix Factorization with the Beta-Divergence,* arXiv:1010.1763, 2010.

[8] P.O. Hoyer, "Non-Negative Matrix Factorization with Sparseness Constraints," *J. Machine Learning Research,* vol. 5, pp. 1457-1469, 2004.

[9] P.O. Hoyer, "Non-Negative Sparse Coding," *Proc. 12th IEEE Workshop Neural Networks for Signal Processing,* 2002.

[10] Y.R. Ramesh Kumar and P.A. Govardhan, "Stock Market Predictions—Integrating User Perception for Extracting Better Prediction a Framework," *Int'l J. Eng. Science,* vol. 2, no. 7, pp. 3305-3310, 2010.

[11] D.D. Lee and H.S. Seung, "Learning the Parts of Objects by Non-Negative Matrix Factorization," *Nature,* vol. 401, no. 6755, pp. 788-91, 1999.

[12] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, "A Symbolic Representation of Time Series, with Implications for Streaming Algorithms," *Proc. Eighth ACM SIGMOD Workshop Research Issues in Data Mining and Knowledge Discovery,* pp. 2-11, 2003.

[13] J. Mairal, F. Bach Inria Willow Project-Team, and G. Sapiro, "Online Learning for Matrix Factorization and Sparse Coding," *J. Machine Learning Research,* vol. 11, pp. 19-60, 2010.

[14] F. Moerchen, "Time Series Knowledge Mining Fabian," PhD thesis, 2006.

[15] F. Moerchen and D. Fradkin, "Robust Mining of Time Intervals with Semi-Interval Partial Order Patterns," *Proc. SIAM Conf. Data Mining,* pp. 315-326, 2010.

[16] F. Mörchen and A. Ultsch, "Efficient Mining of Understandable Patterns from Multivariate Interval Time Series," *Data Mining and Knowledge Discovery,* vol. 15, no. 2, pp. 181-215, 2007.

[17] P. OGrady and B. Pearlmutter, "Discovering Convolutive Speech Phones Using Sparseness and Non-Negativity," *Proc. Seventh Int'l Conf. Independent Component Analysis and Signal Separation,* pp. 520-527, 2007.

[18] R. Andrew Russell, "Mobile Robot Learning by Self-Observation," *Autonomous Robots,* vol. 16, no. 1, pp. 81-93, (2004).

[19] J. Shlens, G.D. Field, J.L. Gauthier, M. Greschner, A. Sher, A.M. Litke, and E.J. Chichilnisky, "The Structure of Large-Scale Synchronized Firing in Primate Retina," *J. Neuroscience: The Official J. Soc. for Neuroscience,* vol. 29, no. 15, pp. 5022-5031, 2009.

[20] P. Smaragdis, "Non-Negative Matrix Factor Deconvolution; Extraction of Multiple Sound Sources from Monophonic Inputs," *Proc. Fifth Int'l Conf. Independent Component Analysis and Blind Signal Separation,* 2004.
[21] L. Xie, H. Sundaram, and M. Campbell, "Event Mining in Multimedia Streams," *Proc. IEEE,* vol. 96, no. 4, pp. 623-647, Apr. 2008.
[22] B.A. Young, E. Lin, M. Von Korff, G. Simon, P. Ciechanowski, E.J. Ludman, S. Everson-Stewart, L. Kinder, M. Oliver, E.J. Boyko, and W.J. Katon, "Diabetes Complications Severity Index and Risk of Mortality, Hospitalization, and Healthcareutilization," *The Am. J. Managed Care,* vol. 14, no. 1, pp. 15-23, 2008.

Fei Wang received the PhD degree from the Department of Automation, Tsinghua University in 2008. He is currently a postdoctoral research scientist in the Healthcare Analytics Research Group at the IBM T.J. Watson Research Center. His major research interests include data mining, machine learning, social informatics, and healthcare informatics. He has published more than 70 papers on the top venues of the relevant field such as the *IEEE Transactions on Pattern Anallysis and Machine Intelligence*, *IEEE Transactions on Neural Networks, IEEE Transactions on Knowledge and Data Engineering*, and *IEEE Transactions on Multimedia*. He is a member of the IEEE.

Noah Lee received the PhD degree from Columbia University Medical Center in 2011. He is currently a quantitative big data analyst at 1010data. His interest lies in taking the latest computing technologies to help humans better reason about, learn, and understand big data.

Jianying Hu received the PhD degree in computer science from the State University of New York at Stony Brook in 1993. She is a research staff member at the IBM T.J. Watson Research Center, New York. Prior to joining IBM she was with Bell Labs from 1993 to 2000, and Avaya Labs Research from 2001 to 2003. Her main research interests include statistical pattern recognition, machine learning, and data mining, with applications to healthcare informatics, business analytics, document analysis, and multimedia content analysis and retrieval. She served as an associate editor of the *IEEE Transactions on Image Processing* from 2001 to 2005, an associate editor of the *IEEE Transactions on Pattern Analysis and Machine Intelligence* from 2006 to 2010, and chair of the Technical Committee on Reading Systems of the International Association for Pattern Recognition from 2004 to 2008. She is currently on the editorial board of the journals *Pattern Recognition* and *International Journal on Document Analysis and Recognition*. She is a fellow of the IAPR and a senior member of the IEEE and the American Physical Society.

Jimeng Sun received the MS and PhD degrees in computer science from Carnegie Mellon University in 2006 and 2007, respectively. He is a research staff member in the Healthcare Analytics Research Group at the T.J. Watson Research Center. His research interests include data mining for health care applications, medical informatics, social network analysis, visual analytics, and data streams. He received the best research paper award at ICDM 2008, the KDD 2007 dissertation award (runner-up), the best research paper award at SDM 2007. He has published more than 40 refereed articles and two book chapters. He has served as a program committee member for SIGKDD, ICDM, SDM, and CIKM and as a reviewer for AMIA, TKDE, VLDB, and ICDE. He cochaired the workshops on large-scale data mining: theory and applications at KDD '10 and ICDM '09, the workshop on large-scale analytics for complex instrumented systems at ICDM '10, and the workshop on visual analytics in health care at VisWeek '10. He also coedited the journal special issue on large-scale data mining fort *TKDD*.

Shahram Ebadollahi received the PhD degree in electrical engineering from Columbia University and has conducted research in the areas of medical imaging, multimedia content analysis for medical applications, and computer vision. He is the manager of the Healthcare Informatics Research group at the IBM T.J. Watson research center, New York, where his group is conducting research in the area of healthcare analytics. He is also the program manager for IBM's initiative in healthcare transformation. He is also an adjunct assistant professor at Columbia University. He is a member of the IEEE.

Andrew F. Laine received the DSc degree from the Washington University School of Engineering and Applied Science, St. Louis, Missouri, in computer science in 1989 and the BS degree from Cornell University, Ithaca, New York. He is the director of the Heffner Biomedical Imaging Laboratory in the Department of Biomedical Engineering at Columbia University in New York City and is the Percy K. and Vida L. W. Hudson Professor of Biomedical Engineering and Radiology. He served as chair of the Technical Committee (TC-BIIP) on Biomedical Imaging and Image Processing for the EMBS (2006-2009), and on the IEEE International Symposium on Biomedical Imaging (ISBI) steering committee (2006-2009) and as program chair for the IEEE EMBS annual conference in 2006 (New York City) and EMBC 2011 (Boston), program cochair for IEEE ISBI in 2008 (Paris, France); vice president of publications for IEEE EMBS since 2008. His research interests include quantitative image analysis, cardiac functional imaging, ultrasound and MRI, retinal imaging, intravascular imaging, and biosignal processing. He has graduated 23 doctoral students over the past 21 years and has more than 150 peer reviewed manuscripts. He is a fellow of the AIMBE and IEEE.