Towards Heterogeneous Temporal Clinical Event Pattern Discovery:A Convolutional Approach

Fei Wang[1], Noah Lee,[1,2], Jianying Hu[1], Jimeng Sun[1], Shahram Ebadollahi[1]
[1]Healthcare Analytics Research Group, IBM T.J. Watson Research Center
[2]Department of Biomedical Engineering, Columbia University
fwang,jyhu,jimeng,ebad@us.ibm.com,nl2168@columbia.edu

ABSTRACT

Large collections of electronic clinical records today provide us with a vast source of information on medical practice. However, the utilization of those data for exploratory analysis to support clinical decisions is still limited. Extracting useful patterns from such data is particularly challenging because it is *longitudinal*, *sparse* and *heterogeneous*. In this paper, we propose a *Nonnegative Matrix Factorization* (NMF) based framework using a convolutional approach for open-ended temporal pattern discovery over large collections of clinical records. We call the method *One-Sided Convolutional NMF* (OSC-NMF). Our framework can mine common as well as individual *shift-invariant* temporal patterns from heterogeneous events over different patient groups, and handle sparsity as well as scalability problems well. Furthermore, we use an event matrix based representation that can encode quantitatively all key temporal concepts including order, concurrency and synchronicity. We derive efficient *multiplicative update* rules for OSC-NMF, and also prove theoretically its convergence. Finally, the experimental results on both synthetic and real world electronic patient data are presented to demonstrate the effectiveness of the proposed method.

Categories and Subject Descriptors

J.3 [**Life and Medical Sciences**]: Health; I.5 [**Pattern Recognition**]: Design Technology—*Pattern analysis*

General Terms

Agorithms

Keywords

Pattern Discovery, NMF, Convolution

1. INTRODUCTION

Electronic Health Records (EHR) are systematic collections of longitudinal patient health information generated by one or more encounters in any care delivery setting. Included in this information are patient demographics, encounter records such as claims, progress notes, problems, medications, vital signs, immunizations, laboratory data and radiology reports, etc. Fig.1 illustrates an example of a temporal event record of a diabetic patient over one year, where 30 key event factors are recorded, including procedures (CPTs), lab results (LABs), visits to primary care physician (PCP) and various specialists (SPEC).

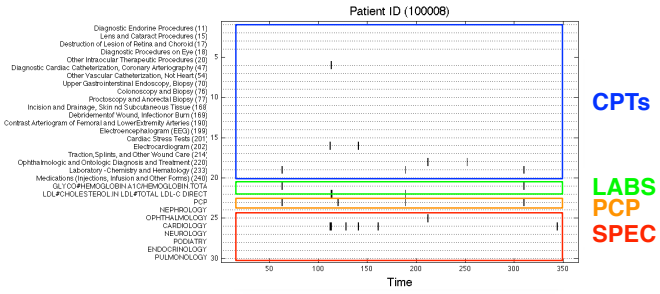


Figure 1: An example of a diabetic patient's electronic record over one year. The x-axis corresponds to the day index, the y-axis represents different types of recorded events, which can be categorized into 4 groups including procedures (CPTs), lab results (LABs), visits to primary care physician (PCP) and visits to specialists (SPEC). The dots in the figure indicate the corresponding events happening at corresponding dates.

In this paper, we study *Temporal Pattern Discovery* (TPD) for EHR data, which aims at finding temporal patterns of one or more groups of patients. TPD is an open-ended problem in the sense that the mined patterns can be utilized in various scenarios such as predictive modeling [1], information visualization [29] and comparative effectiveness research [25]. TPD is also an active research direction that has attracted a lot of interests from data mining related applications including financial marketing [6], video content analysis [4] and social network analysis [19]. Many challenges in TPD are shared by these applications, however some are particularly pronounced in the medical domain when performing TPD from medical data. These include

(1) *Shift-Invariance*. EHR for all patients are not temporally aligned. Moreover, due to various complexities such as comorbidities, the trajectories for different patients are very different over a long time period. However, it is possible to extract time-invariant patterns within a shorter timeframe across patients. Thus an appropriate TPD approach should not be affected by the absolute time stamps.

KDD'12, August 12–16, 2012, Beijing, China.

(2) *Heterogeneity*. The EHR data contain multiple types of events (e.g., diagnosis, medication, lab). The method needs to be able to mine patterns from the combination of all types of events and capture the relationship among them.

(2) *Sparsity and Irregularity*. The EHR data are usually very sparse with irregular intervals, as most of the patients do not have frequent events (e.g., the patient shown in Fig.1 only has around 20 events in a year) or with high regularity.

(4) *Quantitative Nature*. In clinical settings qualitative relations such are event A is followed by event B is important but typically insufficient. Quantitative measures of the duration of a given event and the interval between multiple events are often of key importance.

(5) *Scalability*. We may face a large patient population, and each patient may has a long longitudinal record (e.g., chronic disease) and many different event factors. The algorithm needs to be able to efficiently learn patterns from such a large volume of data.

We propose a novel *geometric* framework for EHR representation and perform the TPD task based on that. Specifically, we model the EHR record as an image matrix like Fig.1, where the x-axis corresponds to the time stamps and y-axis corresponds to the event values. Note that different events can have different value ranges and types (e.g., blood pressure has continuous values, while PCP visits has nonnegative integer values)[1]. Furthermore, each event could be either instantaneous (represented by a single pixel), or with a certain duration (represented by a line segment in the image). This provides a succinct representation that can effectively encode a large range of temporal information including event value, time and duration, relationships among different events, and intervals between pairwise events using an image. We call such a representation an *event matrix*.

Based on the event matrix representation, we propose a novel approach, *One-Sided Convolutional Nonnegative Matrix Factorization* (*OSC-NMF*), to detect temporal patterns from EHR. NMF is a powerful tool for identifying underline structures in a matrix through regularized decomposition and has been successfully applied to many applications including clustering, metric learning, and classification [13][14]. In order to apply this tool to our problem setting, where each patient is represented by an event matrix with fix number of rows (determined by the events of interested) but varying number of columns (determined by the length of the longitudinal record of the patient), we adapt a convolutional approach. Our approach assumes that each patient matrix is generated by the superposition and concatenation of a set of temporal pattern matrices over the time axis. The method is called *one-sided* because the convolution only occurs along the time axis but not on event side. Each pattern matrix essentially encodes a composite temporal pattern with the inherent order, concurrency and synchronicity relations among different events.

To carry out this convolutional decomposition approach we introduce a methodology to minimize the β-divergence [9] between the convoluted matrix and the original patient matrix to obtain the optimal pattern matrices under nonnegativity constraints. β-divergence is a general divergence measure that includes many common measures such as KL-divergence and Euclidean distance as special cases. We provide efficient multiplicative update rules for solving the optimal patterns based on this general measure, and rigorous proofs for the algorithm convergence. Experimental results on applying OSC-NMF to both synthetic and real world data are presented to demonstrate its effectiveness.

It is worthwhile to highlight the strength of OSC-NMF.

(1) *The mined patterns are shift invariant*. Because of the convolutional nature of OSC-NMF, the mined temporal patterns are independent of the absolute time stamps.

(2) *The mined patterns are comprehensive*. The mined pattern matrices represent relationships among all different types of events recorded in the patient matrices.

(3) *OSC-NMF can incorporate sparsity constraints*. OSC-NMF can easily handle the high sparsity in the data by adding sparsity regularization terms in the objective.

(4) *The mined patterns are quantitative*. The pattern matrices can naturally encode quantitative relationships among all different events, including duration of events as well as interval and overlap between events.

(5) *OSC-NMF can handle large scale data set*. We provide an efficient *stochastic* learning framework for OSC-NMF, which processes one or a small portion of the data matrices each time, thus results in a constant memory cost.

The rest of this paper is organized as follows. Section 2 introduces related work. Algorithm details and extensions are described in section 3. Section 4 presents the experimental results, is followed by the conclusion in section 5.

2. RELATED WORK

In this section we briefly review some previous work that are closely related to our work in this paper.

A lot of work has been done for temporal sequence representation and mining. For example, Keogh *et al.* proposed *symbolic aggregate approximation* (SAX) [15] to represent time series using symbolic sequences. However, SAX cannot take into account the relationships among heterogeneous time sequences. [17] proposed a temporal knowledge representation method with symbolic languages and grammars. [5] proposed a TPD approach based on first-order temporal logic under regular expression constraints. These methods is that they specify some explicit symbolic languages and temporal grammars according to prior knowledge. Frequent item set [22][11] or sub-sequence [30] mining are also closely related to the work in this paper. However, in these approaches, the time intervals between pairwise events are not considered. In our case, this pairwise event intervals are of key importance (e.g., two events happen in a week and a year are completely different disease condition signals).

In medical domain, [20] proposed a statistical approach for summarizing and visualizing the temporal associations between the prescription of a drug and the occurrence of a medical event. [24] proposed a *temporal abstraction* approach for medical TPD. Similar to [17], this method also requires predefined temporal grammar and logic with prior knowledge. [8] proposed a visual interface for finding temporal patterns in multivariate temporal clinical data. The interface was further used in [23] for searching temporal patterns in patient histories, but the user needs to specify the structure of the pattern on the interface.

On the methodology side, *Nonnegative Matrix Factorization* (NMF) [13][14], which aims at factorizing a nonnegative

[1]In this paper, we only consider binary event values, i.e., the (i, j)-th element of the patient matrix is 1 if the i-th event happens at time j, otherwise its value would be 0.

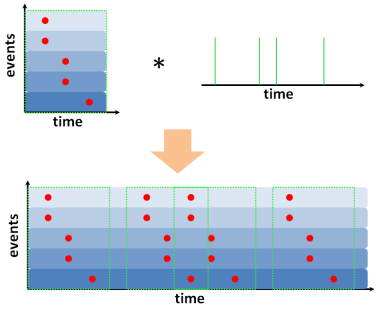


Figure 2: A graphical illustration of one-side convolution. The top left figure shows temporal pattern, and the top right figure is the time axis where we use green bars to represent the position where the pattern appears. The bottom figure is the one-side convolution result, where each dotted line rectangle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

matrix into the product of two low-rank nonnegative matrices, has attracted considerable interests from data mining in recent years. There are also a lot of NMF variants that are related to our work. For instance, [12][7] proposed to enforce sparsity regularizations on the decomposed matrices to obtain sparse solutions. [10], [21], [26] and [18] proposed *convolutional* sparse NMF to discover the shift-invariant temporal patterns from acoustic signals and images. However their approach is designed for time series data and is not applicable to heterogeneous event sequences. Furthermore, all prior methods measure the factorization quality using matrix Frobenius norm, and obtain a common set of patterns by batch learning methods. On the contrary, our method proposed in this paper (1) is based on a general β-divergence loss; (2) can detect common and individual patterns from different data groups, and (3) can be trained with efficient stochastic learning scheme.

3. ONE-SIDED CONVOLUTIONAL NMF

We now describe the *One-Sided Convolutional NMF* (OSC-NMF) algorithm in detail.

3.1 Preliminaries

Suppose we have a patient matrix $\mathbf{X} \in \mathbb{R}^{n \times t}$, where n is the number of event factors, t is the length of the patient clinical history. As mentioned in section 1, 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 $*$ 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*

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

Note that $g_j = 0$ if $j \leqslant 0$ 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 our scenario on TPD from electronic clinical records. As in our case, we are interested in patterns 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 the pattern top-left and the time vector top-right.

Another important definition is the matrix β-divergence.

Definition 2. (β-divergence [9]) *The β-divergence between two matrices $\mathbf{A}$ and $\mathbf{B}$ with the same size is*

$$d_\beta(\mathbf{A}, \mathbf{B}) = \frac{1}{\beta(\beta-1)} \sum_{ij} \left(A_{ij}^{\beta} + (\beta-1) B_{ij}^{\beta} - \beta A_{ij} B_{ij}^{\beta-1} \right) \tag{2}$$

where $\beta \geqslant 0$ 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.

$$\begin{aligned} d_0(\mathbf{A}, \mathbf{B}) &= \lim_{\beta \to 0} \sum_{ij} \left(A_{ij} \frac{B_{ij}^{\beta-1}}{1-\beta} - \frac{A_{ij}^{\beta} - B_{ij}^{\beta}}{\beta} \right) + \frac{A_{ij}^{\beta}}{\beta - 1} \\ &= \sum_{ij} A_{ij} (\log A_{ij} - \log B_{ij}) + (B_{ij} - A_{ij}) \end{aligned} \tag{3}$$

$$\begin{aligned} d_1(\mathbf{A}, \mathbf{B}) &= \lim_{\beta \to 1} \sum_{ij} \left(A_{ij} \frac{A_{ij}^{\beta-1} - B_{ij}^{\beta-1}}{\beta - 1} + \frac{B_{ij}^{\beta} - A_{ij}^{\beta}}{\beta} \right) \\ &= \sum_{ij} A_{ij} (\log A_{ij} - \log B_{ij}) + (B_{ij} - A_{ij}) \end{aligned} \tag{4}$$

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* and *Euclidean distance*.

3.2 The Algorithm

Now coming back to our problem, we first introduce how to make use of OSC-NMF to detect temporal patterns from a single patient's EHR. Second, we extend OSC-NMF to detect patterns from the EHR of multiple groups of patients.

Recall we suppose the patient EHR matrix $\mathbf{X}$ is constructed by the superposition of the one-side convolution of a set of patterns $\mathcal{F} = \{\mathbf{F}^{(r)}\}_{r=1}^{R}$ across the time axis. Then we propose to detect them by minimizing

$$\mathcal{J} = d_\beta \left(\mathbf{X}, \sum_{r=1}^{R} \mathbf{F}^{(r)} * \mathbf{g}^{(r)} \right) \tag{5}$$

where $\mathbf{g}^{(r)} \in \mathbb{R}^t$ is the coding matrix for pattern $\mathbf{F}^{(r)}$. The problem our algorithm aims to solve is

$$\min_{\mathbf{F}^{(r)} \geqslant 0,\ \mathbf{g}^{(r)} \geqslant 0} \mathcal{J} \ (\forall r = 1, 2, \cdots, R) \tag{6}$$

Since the patient matrix $\mathbf{X}$ is nonnegative, we also require $\{\mathbf{F}^{(r)}, \mathbf{g}^{(r)}\}_{r=1}^{R}$ to be nonnegative. With the definition of β divergence (Eq.(2)), we have

$$\frac{\partial \mathcal{J}}{\partial F_{ik}^{(r)}} = \sum_{j=1}^{t} \left(Y_{ij}^{\beta-1} - X_{ij} Y_{ij}^{\beta-2} \right) \frac{\partial Y_{ij}}{\partial F_{ik}^{(r)}} \tag{7}$$

where we define

$$\mathbf{Y} = \sum_{r=1}^{R} \mathbf{F}^{(r)} * \mathbf{g}^{(r)} \tag{8}$$

Combining Eq.(1) and Eq.(8), we have $\partial Y_{ij} / \partial F_{ik}^{(r)} = g_{j-k+1}^{(r)}$. Thus we can update $F_{ik}^{(r)}$ by

$$F_{ik}^{(r)} \leftarrow F_{ik}^{(r)} \left(\frac{\sum_{j=1}^{t} X_{ij} Y_{ij}^{\beta-2} g_{j-k+1}^{(r)}}{\sum_{j=1}^{t} Y_{ij}^{\beta-1} g_{j-k+1}^{(r)}} \right)^{\eta(\beta)} \tag{9}$$

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

$$\eta(\beta) = \begin{cases} \frac{1}{2-\beta}, & \beta < 1 \\ 1, & 1 \leqslant \beta \leqslant 2 \\ \frac{1}{\beta - 1}, & \beta > 2 \end{cases} \tag{10}$$

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

$$g_k^{(r)} \leftarrow g_k^{(r)} \left(\frac{\sum_{i=1}^n \sum_{j=1}^t X_{ij} Y_{ij}^{\beta-2} F_{i,j-k+1}^{(r)}}{\sum_{i=1}^n \sum_{j=1}^t Y_{ij}^{\beta-1} F_{i,j-k+1}^{(r)}} \right)^{\eta(\beta)} \tag{11}$$

We have the following theorem (which is proved in the Appendix) 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 update them with Eq.(9) and Eq.(11) will finally converge to a stationary point.*

Complexity Analysis

For the storage complexity, during the iterations, it is good 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 update 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 $O((4R+2)T\bar{s}_F \bar{s}_g)$.

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. Similar to [12] and [7], we can enforce the sparsity constraints by adding ℓ_1 regularization terms to the objective in Eq.(5). As a consequence, we can solve for the optimal patterns and codes by minimizing

$$\mathcal{J}_1 = d_\beta \left(\mathbf{X}, \sum_{r=1}^R \mathbf{F}^{(r)} * \mathbf{g}^{(r)} \right) + \lambda_1 \sum_{r=1}^R \|\mathbf{F}^{(r)}\|_1 + \lambda_2 \sum_{r=1}^R \|\mathbf{g}^{(r)}\|_1 \tag{12}$$

where $\lambda_1 > 0$ and $\lambda_2 > 0$ are the regularization parameters. Then the problem we want to solve becomes

$$\min_{\mathbf{F}^{(r)} \geqslant 0,\ \mathbf{g}^{(r)} \geqslant 0} \mathcal{J}_1 \ (\forall r = 1, 2, \cdots, R) \tag{13}$$

Similar to the previous subsection, we can get the update rules for $\mathbf{F}$ and $\mathbf{g}$ as follows.

$$F_{ik}^{(r)} \leftarrow F_{ik}^{(r)} \left(\frac{\sum_{j=1}^t X_{ij} Y_{ij}^{\beta-2} g_{j-k+1}^{(r)}}{\sum_{j=1}^t Y_{ij}^{\beta-1} g_{j-k+1}^{(r)} + \lambda_1} \right)^{\eta(\beta)} \tag{14}$$

$$g_k^{(r)} \leftarrow g_k^{(r)} \left(\frac{\sum_{i=1}^n \sum_{j=1}^t X_{ij} Y_{ij}^{\beta-2} F_{i,j-k+1}^{(r)}}{\sum_{i=1}^n \sum_{j=1}^t Y_{ij}^{\beta-1} F_{i,j-k+1}^{(r)} + \lambda_2} \right)^{\eta(\beta)} \tag{15}$$

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 [7], purely solving problem (13) 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 an normalization invariant formulation of problem (13) in the following.

Normalization Invariant Formulation

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

$$\mathcal{J}_1^n = d_\beta \left(\mathbf{X}, \sum_{r=1}^R \widehat{\mathbf{F}}^{(r)} * \mathbf{g}^{(r)} \right) + \lambda_1 \sum_{r=1}^R \|\widehat{\mathbf{F}}^{(r)}\|_1 + \lambda_2 \sum_{r=1}^R \|\mathbf{g}^{(r)}\|_1 \tag{16}$$

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

- *Individual Normalization.* Each pattern matrix is normalized to unit Frobenius norm, i.e., $\widehat{F}_{ij}^{(r)} = F_{ij}^{(r)} / \sqrt{\sum_{ij} {F_{ij}^{(r)}}^2}$
- *Total Normalization.* Each pattern matrix is normalized by the total Frobenius norm of all the pattern matrices, i.e., $\widehat{F}_{ij}^{(r)} = F_{ij}^{(r)} / \sqrt{\sum_r \sum_{ij} {F_{ij}^{(r)}}^2}$

Using the same trick as in [7], we can update $\mathcal{F}$ and $\mathcal{G}$ by

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

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

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

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

3.3 OSC-NMF on Data Groups

In the medical domain, patients are often characterized by multiple groups based on characteristics such as diagnosis or treatments. More formally we consider the case where the patient matrices are composed of C groups. We use $\mathcal{X}_c = [\mathbf{X}_{c_1}, \mathbf{X}_{c_2}, \cdots, \mathbf{X}_{c_{n_c}}]$ to represent the c-th patient group, with $\mathbf{X}_{c_l}$ representing the l-th data point in this group. n_c is the number of patient in the c-th group. In many real world applications we are also interested in finding patterns hidden in those groups. In this section we will extend our one-side convolutional NMF to these scenarios.

(1). TPD From One Group

We first consider the case of detecting common patterns from one group, i.e., $C = 1$, which is the same setting as in group sparse coding [2]. If we still denote the hidden pattern set as $\mathcal{F} = \{\mathbf{F}^{(r)}\}_{r=1}^R$, then the problem we want to solve becomes (here we directly give the sparsity constrained objective as the non-sparse case just correspond to $\lambda_1 = \lambda_2 = 0$)

$$\begin{aligned} \min_{\mathcal{F}, \{\mathcal{G}_c\}_{c=1}^C} \quad & \mathcal{J}_3 \\ s.t. \quad & \forall r = 1, \cdots, R; l = 1, \cdots, n_1, \\ & \mathbf{F}^{(r)} \geqslant 0, \mathbf{g}_l^{(r)} \geqslant 0 \end{aligned} \tag{17}$$

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

$$\mathcal{J}_3 = \sum_{l=1}^{n_1} d_\beta \left(\mathbf{X}_l, \sum_{r=1}^R \mathbf{F}^{(r)} * \mathbf{g}_l^{(r)} \right) + \lambda_1 \sum_{r=1}^R \|\mathbf{F}^{(r)}\|_1 + \lambda_2 \sum_{l=1}^{n_1} \sum_{r=1}^R \|\mathbf{g}_l^{(r)}\|_1 \tag{18}$$

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

$$F_{ik}^{(r)} \leftarrow F_{ik}^{(r)} \left(\frac{\sum_{l=1}^{n_1} \sum_{j=1}^{t} X_{l_{ij}} Y_{l_{ij}}^{\beta-2} g_{l_{j-k+1}}^{(r)}}{\sum_{l=1}^{n_1} \sum_{j=1}^{t} Y_{l_{ij}}^{\beta-1} g_{l_{j-k+1}}^{(r)} + \lambda_1} \right)^{\eta(\beta)} \quad (19)$$

$$g_{l_k}^{(r)} \leftarrow g_{l_k}^{(r)} \left(\frac{\sum_{i=1}^{n} \sum_{j=1}^{t} X_{c_{l_{ij}}} Y_{c_{l_{ij}}}^{\beta-2} F_{i,j-k+1}^{(r)}}{\sum_{i=1}^{n} \sum_{j=1}^{t} Y_{c_{l_{ij}}}^{\beta-1} F_{i,j-k+1}^{(r)} + \lambda_2} \right)^{\eta(\beta)} \quad (20)$$

If we want to find normalized patterns, we can use the same trick as in [7] and derive the following update rules

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

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

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

Complexity Analysis

Similar to the simple OSC-NMF case, we can analyze that the storage complexity of group OSC-NMF is $O(n_1(\bar{s}_X + \bar{s}_Y + \bar{\bar{s}}_g) + \bar{s}_F)$, where n_1 is the size of the group, $\bar{s}_X, \bar{s}_Y$ are the averaged number of nonzero elements in $\{\mathbf{X}_l\}_{l=1}^{n_1}$ and $\{\mathbf{Y}_l\}_{l=1}^{n_1}$, and $\bar{\bar{s}}_g$ is the averaged number of nonzero elements of all $\{\mathbf{g}_l^{(r)}\}_{r=1,l=1}^{R,n_1}$. The total computatioinal complexity is $O((4R+2)Tn_1\bar{s}_F\bar{\bar{s}}_g)$. For normalized cases, we just need an extra $O(2R\bar{s}_F)$ time for pattern normalization.

A Stochastic Learning Scheme

We can see that group OSC-NMF is storage and time consuming if the group size n_1 is very large. In this case, we can adopt the stochastic (online) learning scheme in [16][3][28], 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 the following steps:

(1) 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 of $\mathcal{G}_t$, then iterating with Eq.(20) (or its normalized version) several times.

(2) Integrating $\mathcal{X}_t$ and $\mathcal{G}_t$ with the previously received data and their estimated convolution coeffients to update $\mathcal{F}$ with Eq.(19) (or its normalized version) only *once*.

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 Eq.(19) (or Eq.(21)), 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. Therefor, 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.

(2). TPD From Multiple Data Groups

Now assume there are C ($C > 1$) data groups, and we want to find a common pattern dictionary $\mathcal{F}^S$ across all data groups, as well as an individual pattern dictionary $\{\mathcal{F}_c^I\}_{c=1}^{C}$ for each data group, then we need to solve

$$\min_{\mathbf{F}_S^{(r)} \geqslant 0, \mathbf{g}_{S_{cl}}^{(r)} \geqslant 0, \mathbf{F}_{I_c}^{(v)} \geqslant, \mathbf{g}_{I_{cl}}^{(v)} \geqslant 0} \quad \mathcal{J}_4 \quad (21)$$

$$s.t. \quad \forall r = 1, \cdots, R; c = 1, \cdots, C; l = 1, \cdots, n_c; v = 1, \cdots, V$$

where we use r to index the common patterns (whose total number is R), v to index the individual patterns in the c-th group (whose total number is V_c), and l to index the data within each group. The objective we want to minimize is

$$\mathcal{J}_4 = \sum_{c=1}^{C} \left[\sum_{l=1}^{n_c} d_\beta \left(\mathbf{X}_{cl}, \sum_{r=1}^{R} \mathbf{F}_S^{(r)} * \mathbf{g}_{S_{cl}}^{(r)} + \sum_{v=1}^{V} \mathbf{F}_{I_c}^{(v)} * \mathbf{g}_{I_{cl}}^{(v)} \right) \right] + \lambda_S \sum_{r=1}^{R} \left\| \mathbf{F}_S^{(r)} \right\|_1$$
$$+ \sum_{c=1}^{C} \sum_{l=1}^{n_c} \left(\gamma_S \sum_{r=1}^{R} \left\| \mathbf{g}_{S_{cl}}^{(r)} \right\|_1 + \gamma_I \sum_{v=1}^{V} \left\| \mathbf{g}_{I_{cl}}^{(v)} \right\|_1 \right) + \lambda_I \sum_{c=1}^{C} \sum_{v=1}^{V} \left\| \mathbf{F}_{I_c}^{(v)} \right\|_1$$

Let $\widetilde{\mathbf{Y}} = \sum_{r=1}^{R} \mathbf{F}_S^{(r)} * \mathbf{g}_{S_{c_l}}^{(r)} + \sum_{v=1}^{V} \mathbf{F}_{I_c}^{(v)} * \mathbf{g}_{I_{c_l}}^{(v)}$, then we can get the update rules for unnormalized case

$$F_{S_{ik}}^{(r)} \leftarrow F_{S_{ik}}^{(r)} \left(\frac{\sum_{c,l,j} X_{c_{l_{ij}}} \widetilde{Y}_{c_{l_{ij}}}^{\beta-2} g_{S_{cl\,j-k+1}}^{(r)}}{\sum_{c,l,j} \widetilde{Y}_{c_{l_{ij}}}^{\beta-1} g_{S_{cl\,j-k+1}}^{(r)} + \lambda_S} \right)^{\eta(\beta)} \quad (22)$$

$$F_{I_{cik}}^{(v)} \leftarrow F_{I_{cik}}^{(v)} \left(\frac{\sum_{l,j} X_{c_{l_{ij}}} \widetilde{Y}_{c_{l_{ij}}}^{\beta-2} g_{I_{cl\,j-k+1}}^{(v)}}{\sum_{l,j} \widetilde{Y}_{c_{l_{ij}}}^{\beta-1} g_{I_{cl\,j-k+1}}^{(v)} + \lambda_I} \right)^{\eta(\beta)} \quad (23)$$

$$g_{S_{cl\,k}}^{(r)} \leftarrow g_{S_{cl\,k}}^{(r)} \left(\frac{\sum_{i,j} X_{c_{l_{ij}}} \widetilde{Y}_{c_{l_{ij}}}^{\beta-2} F_{S_{i,j-k+1}}^{(r)}}{\sum_{i,l,j} \widetilde{Y}_{c_{l_{ij}}}^{\beta-1} F_{S_{i,j-k+1}}^{(r^S)}} + \gamma_S \right)^{\eta(\beta)} \quad (24)$$

$$g_{I_{cl\,k}}^{(v)} \leftarrow g_{I_{cl\,k}}^{(v)} \left(\frac{\sum_{i,j} X_{c_{l_{ij}}} \widetilde{Y}_{c_{l_{ij}}}^{\beta-2} F_{I_{ci,j-k+1}}^{(r_c^I)}}{\sum_{i,l,j} \widetilde{Y}_{c_{l_{ij}}}^{\beta-1} F_{I_{ci,j-k+1}}^{(v)}} + \gamma_I \right)^{\eta(\beta)} \quad (25)$$

Let $\widehat{\widetilde{\mathbf{Y}}} = \sum_{r=1}^{R} \widehat{\mathbf{F}}_S^{(r)} * \mathbf{g}_{S_{c_l}}^{(r)} + \sum_{v=1}^{V} \widehat{\mathbf{F}}_{I_c}^{(v)} * \mathbf{g}_{I_{c_l}}^{(v)}$, where $\widehat{\mathbf{F}}_S$ and $\{\widehat{\mathbf{F}}_{I_c}\}$ are normalized basis, then we have the following updating rules for normalized cases

$$F_{S_{ik}}^{(r)} \leftarrow F_{S_{ik}}^{(r)} \left(\frac{\sum_{c,l,j} \left(X_{c_{l_{ij}}} + \widehat{\widetilde{Y}}_{c_{l_{ij}}} \widehat{F}_{S_{ik}}^{(r)^2} \right) \widehat{\widetilde{Y}}_{c_{l_{ij}}}^{\beta-2} g_{S_{cl\,j-k+1}}^{(r)} + \lambda_S \widehat{F}_{S_{ik}}^{(r)^2}}{\sum_{c,l,j} \left(\widehat{\widetilde{Y}}_{c_{l_{ij}}} + X_{c_{l_{ij}}} \widehat{F}_{S_{ik}}^{(r)^2} \right) \widehat{\widetilde{Y}}_{c_{l_{ij}}}^{\beta-2} g_{S_{cl\,j-k+1}}^{(r)} + \lambda_S} \right)^{\eta(\beta)}$$

$$F_{I_{cik}}^{(v)} \leftarrow F_{I_{cik}}^{(v)} \left(\frac{\sum_{l,j} \left(X_{c_{l_{ij}}} + \widehat{\widetilde{Y}}_{c_{l_{ij}}} \widehat{F}_{I_{cik}}^{(r)^2} \right) \widehat{\widetilde{Y}}_{c_{l_{ij}}}^{\beta-2} g_{I_{cl\,j-k+1}}^{(v)} + \lambda_I \widehat{F}_{I_{cik}}^{(r)^2}}{\sum_{l,j} \left(\widehat{\widetilde{Y}}_{c_{l_{ij}}} + X_{c_{l_{ij}}} \widehat{F}_{I_{cik}}^{(r)^2} \right) \widehat{\widetilde{Y}}_{c_{l_{ij}}}^{\beta-2} g_{I_{cl\,j-k+1}}^{(v)} + \lambda_I} \right)^{\eta(\beta)}$$

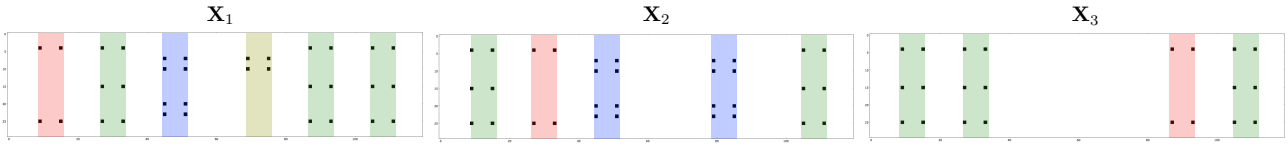
$$g_{S_{cl\,k}}^{(r)} \leftarrow g_{S_{cl\,k}}^{(r)} \left(\frac{\sum_{i,j} X_{c_{l_{ij}}} \widehat{\widetilde{Y}}_{c_{l_{ij}}}^{\beta-2} \widehat{F}_{S_{i,j-k+1}}^{(r)}}{\sum_{i,l,j} \widehat{\widetilde{Y}}_{c_{l_{ij}}}^{\beta-1} \widehat{F}_{S_{i,j-k+1}}^{(r)}} + \gamma_S \right)^{\eta(\beta)}$$

$$g_{I_{cl\,k}}^{(v)} \leftarrow g_{I_{cl\,k}}^{(v)} \left(\frac{\sum_{i,j} X_{c_{l_{ij}}} \widehat{\widetilde{Y}}_{c_{l_{ij}}}^{\beta-2} \widehat{F}_{I_{ci,j-k+1}}^{(v)}}{\sum_{i,l,j} \widehat{\widetilde{Y}}_{c_{l_{ij}}}^{\beta-1} \widehat{F}_{I_{ci,j-k+1}}^{(r_c^I)}} + \gamma_I \right)^{\eta(\beta)}$$

The complexity of this multi-group OSC-NMF can be analyzed similarly as our analysis in previous sections, thus we omit the details here. Note that when facing with large data groups, we can also adopt the stochastic learning scheme.

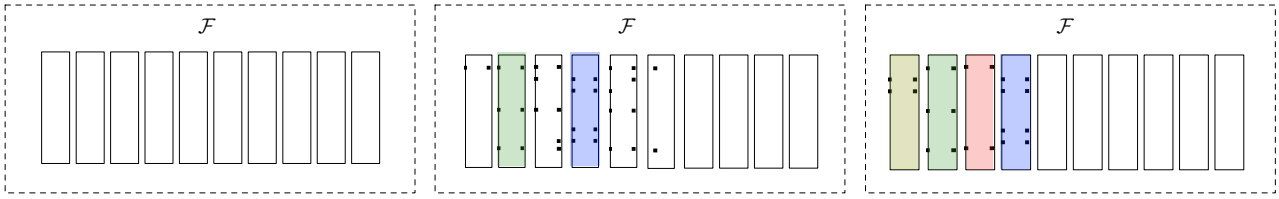
4. EXPERIMENTS

In this section we present the experimental evaluation results for OSC-NMF and its variants.



(a) Sample 1 (b) Sample 2 (c) Sample 3

Figure 3: Synthetic data set I. There are three samples $\{\mathbf{X}_i\}_{i=1}^3$ of size 30×120 with binary values. We use black dots to denote value 1, and the 4 types of temporal patterns are shaded with different colors.



(a) GOSC-NMF (b) GOSC-NMF-IN (c) GOSC-NMF-TN

Figure 4: Detected patterns. GOSC-NMF-TN algorithm can identify all four types successfully.

4.1 Synthetic Data

We generated two synthetic data sets to validate the effectiveness of the proposed methods. The first one is designed to test whether Group OSC-NMF series methods can detect common temporal patterns contained in the data samples. The data set is illustrated in Fig.3, which is composed of one group of three data samples with size 30×120. There are four types of common patterns present in the data samples indicated by windows of different color in Fig.3. These samples are binary, with black dots representing 1.

The following algorithms were applied to this first synthetic data set to evaluate their effectiveness.

(1) **Group OSC-NMF** (**GOSC-NMF**). The algorithm is introduced in section 3.3 with $\beta = 0.5$, $R = 11$, $\lambda_1 = \lambda_2 = 0$. All $\{\mathbf{F}^{(r)}, \mathbf{g}^{(r)}\}_{r=1}^R$ are randomly initialized.

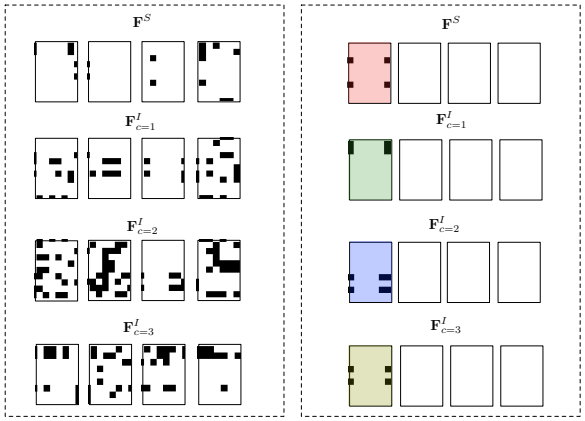
(2) **Group OSC-NMF with Individual Normalization** (**GOSC-NMF-IN**). The algorithm is the same as in GOSC-NMF except that we use normalization invariant updates with individual normalization introduced in section 3.3, and $\lambda_S = \lambda_I = 0.5$.

(3) **Group OSC-NMF with Total Normalization** (**GOSC-NMF-TN**). The algorithm is the same as in GOSC-NMF-IN except we use normalization invariant updates with total normalization introduced in section 3.3, and $\lambda_S = \lambda_I = 0.5$.

For all three algorithms, we set the window length to $m = 7$ and the number of iterations to $T = 100$.

Fig.4 shows the learned patterns by those algorithms. Fig.4(a) shows the patterns learned by simple GOSC-NMF, from which we can see that all of them are null patterns. This is because the three original data samples are all very sparse. By convolving those null patterns over the time axis we get a zero matrix, and the total β-divergence ($\beta = 0.5$) between the data samples to this zero matrix is very small. This makes GOSC-NMF trapped in this local optimum.

Fig.4(b) demonstrates that sparse GOSC-NMG with individual normalization correctly learns two repeating patterns appearing in the original data samples, while Fig.4(c) shows that sparse GOSC-NMG with total normalization correctly learns all temporal patterns. It can be observed that by adding the sparsity regularizations, the learned patterns are much better. More interestingly, we can see that GOSC-NMF-TN identify all 4 patterns.



(a) MGOSC-NMF-IN (b) MGOSC-NMF-TN

Figure 6: Group Patterns Discovery: MGOSC-NMF-TN can successfully identify all patterns including shared pattern (red, first row) and all individual patterns from each group (row 2-4).

The second data set is designed to test whether our multi-group OSC-NMF series algorithms introduced in section 3.3 can detect both common and individual temporal patterns contained in different data groups. The data is shown in Fig.5. We tested three algorithms: (1) **MGOSC-NMF**. Simple multi-group OSC-NMF with $\lambda_S = \lambda_I = 0$; (2) **MGOSC-NMF-IN**. MGOSC-NMF with individual normalization, and $\lambda_S = \lambda_I = 0.5$; (3) **MGOSC-NMF-TN**. MGOSC-NMF with total normalization, and $\lambda_S = \lambda_I = 0.5$. For all three methods, we set $\beta = 0.5$, $T = 100$, $R = V = 4$. All pattern images and convolution coefficients are randomly initialized. The results are shown in Fig.6, from which we can make similar observations as in Fig.4. We do not show the learned patterns from simple MGOSC-NMF because all of them are zero. MGOSC-NMF-IN can learn a rich set of pattern images, but not all of them are correct. Finally MGOSC-NMF-TN correctly learns all individual and common patterns.

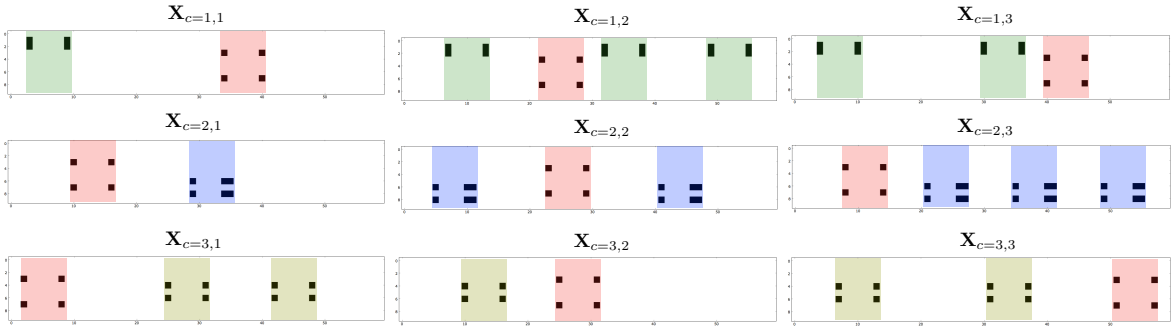


Figure 5: Synthetic data set II. Each row is a data group containing three data samples. All three data groups share a common temporal pattern shaded in red color, while the data within each group has one individual pattern shaded in different colors.

4.2 Real World Data

The real-world dataset consists of records from 21K diabetes patients collected over a period of up to one year. The patients are stratified into three groups A, B, and C based on their specific type of diabetes diagnosis using ICD9 code. Group A (with size 16K) consists of patients with no complications, group B (with size 4,925) consists of patients with chronic disease complications, and group C has patients (with size 254) with acute complications. To evaluate the clinical relevance of the temporal patterns mined by our algorithm, we treat the diagnoses as labels. We then use the mined temporal patterns as additional features for predicting the diagnoses, and compare the performance against a baseline classifier using the aggregate clinical features without consideration of any temporal relations. The hypothesis is that if the temporal patterns mined by our algorithm indeed contain useful clinical information, then their inclusion should improve the classification performance.

For all three groups, 30 different event conditions were selected as being relevant to the progress of diabetes based on consultations with physicians. The events fall into four different groups: medical procedures (CPTs), lab results (LABS), primary care physician visits (PCP), and visits to various specialists (SPEC). One typical patient EHR example is shown in Fig.1. We show some examples of repeating patterns (with one week window length) for each group in Fig.7, which have been identified manually by domain experts. Note that in this case, each patient is represented by a $30 \times T$ matrix, where $T = 365$, and the total patient population is represented by 21k such matrices, and we adopt the stochastic learning strategy for learning the patterns.

To quantitatively evaluate the performance, we randomly selected 70% of the data samples from each group to form the training set, and used the rest data for testing. In this experiment, we set $\beta = 0.5$ and $T = 10000$. The mini-batch size when performing stochastic learning was set to 20. We applied OSC-NMF series methods to construct different feature representations for the data and then used *Nearest Neighbor* (NN) classifier (with Euclidean distance) to classify the test samples. We repeated the experiments 50 times using random partitioning, and report the average classification accuracy compared against the baseline (which is the performance using aggregate clinical features alone, with no considerations of any temporal relations).

Both single group (GOSC-NFM-IN and GOSC-NMF-TN) and multi-group (MGOSC-NMF-IN and MGOSC-NMF-TN)

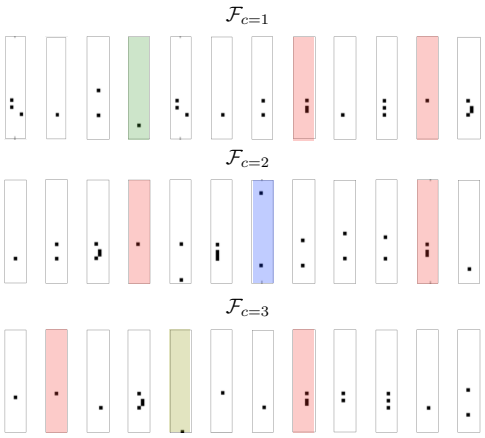


Figure 7: Patient samples from three groups. Shaded windows are manually identified clinical patterns. Red are common patterns across three groups. The other colors are group specific patterns.

series were tested. For GOSC-NMF-IN and GOSC-NMF-TN, the algorithms were applied to all training data each time and $R = 30$ patterns were learned (null patterns were discarded). Each sample was then represented by a 30 (or less if there are zero patterns) dimensional vector with the value on each dimension equal to the sum of the convolution coefficients $\mathbf{g}$ of the corresponding pattern on this sample. This is very similar to the *bag-of-words* representation for text data. For MGOSC-NMF-IN and NGOSC-NMF-TN, we learned both common and individual patterns for all groups, and discarded the common patterns (since they are not important for classification across groups), using only the individual patterns to construct the dictionary. We set the number of patterns $R = V = 30$ and discarded the null patterns. Thus each sample was represented by a 90 (or less) dimensional vector. For comparison purpose, we also implemented Group and Multi-Group PrefixScan [22] (G-PrefixScan and MG-PrefixScan), i.e., we use a sliding window (with the same length as we used for OSC-NMF methods) to segment the patient record sequence into a set of overlapping transactions, and then apply PrefixScan to mine frequent item sets from all these transactions. For G-PrefixScan, we mine 30 most frequent patterns from all training data, while for MG-PrefixScan, we mine 30 most frequent patterns for the training data in each class. Then we also construct the bag-of-pattern matrix for each patient

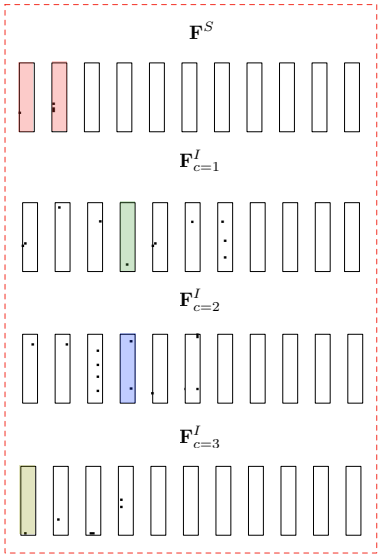


Figure 8: Common and individual patterns are correctly identified by MGOSC-NMF-TN with one week window length. Row 1 are two common patterns, and row 2-4 are group specific patterns. Besides the 5 known patterns, MGOSC-NMF-TN also reveals some unknown patterns, which can potentially lead to new clinical discovery.

(each patient is a 30 dimensional vector with the value on each dimension representing the number of times the corresponding pattern appears within the patient records) for classification.

Table 1 shows the averaged classification performance measured by *Areas Under the Curve* (AUC) with pattern window length set to one week, two weeks and one month. The results show that (1) the classification performances with the inclusion of temporal pattern based features are indeed much better compared to the baseline representation; (2) longer window patterns are more effective, which is likely due to the fact that the progression of diabetes is slow, making the patterns more salient when we increase the window length; (3) multi-group methods tend to perform better, as they extract more discriminative patterns for each group; (4) our matrix approximation based approaches perform better than traditional PrefixScan type methods.

5. CONCLUSION

In this paper we propose an One-Sided Convolutional Nonnegative Matrix Factorization (OSC-NMF) approach for temporal pattern discovery in longitudinal clinical records. We present how to adapt OSC-NMF to extract patterns from one data sample, a group of data samples and multiple groups of data samples. The experimental results on both synthetic and real world data sets are presented to demonstrate the effectiveness of the proposed approaches.

Appendix: Convergence Proof

With $\mathbf{Y}$ defined in Eq.(8), we can expand $d_\beta(\mathbf{X},\mathbf{Y})$ using Eq.(2). For a power function $f(x)=x^\beta$ with $\beta \geqslant 1$, $f(x)$ is convex; otherwise $f(x)$ is concave. Therefore, when $\beta \geqslant 1$, we have the following conclusion according to *Jensen's inequality*

$$Y_{ij}^\beta = \left(\sum_r \sum_k g^{(r)}_{j-k+1} F^{(r)}_{ik}\right)^\beta \leqslant \sum_r \sum_k \alpha^{(r)}_{ijk} \left(\frac{g^{(r)}_{j-k+1} F^{(r)}_{ik}}{\alpha^{(r)}_{ijk}}\right)^\beta$$

when $1 \leqslant \beta \leqslant 2$, where $\alpha_k \geqslant 0$ and $\sum_k \alpha_k = 1$. The equality holds when

$$\alpha^{(r)}_{ijk} = g^{(r)}_{j-k+1} F^{(r)}_{ik} / \sum_r \sum_k g^{(r)}_{j-k+1} F^{(r)}_{ik}$$

On the other hand, when $\beta < 1$, $f(x)=x^\beta$ is concave, and we have that [27] for any z, $f(x) \leqslant f'(z)(x-z)+f(z)$. Therefore in this case

$$Y_{ij}^\beta \leqslant \beta Z_{ij}^{\beta-1}\left(\sum_r \sum_k g^{(r)}_{j-k+1} F^{(r)}_{ik} - Z_{ij}\right) + Z_{ij}^\beta \tag{26}$$

The equality holds when$Z_{ij} = Y_{ij} = \sum_r \sum_k g^{(r)}_{j-k+1} F^{(r)}_{ik}$. Similarly, $f(x)=-x^\beta$ is convex, if $\beta < 1$; and it is concave, if $\beta \geqslant 1$. Hence

$$-Y_{ij}^\beta \leqslant \begin{cases} -\beta Z_{ij}^{\beta-1}\left(\sum_r \sum_k g^{(r)}_{j-k+1} F^{(r)}_{ik} - Z_{ij}\right) - Z_{ij}^\beta, & \text{if } \beta < 1 \\ -\sum_r \sum_k \alpha^{(r)}_{ijk} \left(\frac{g^{(r)}_{j-k+1} F^{(r)}_{ik}}{\alpha^{(r)}_{ijk}}\right)^\beta, & \text{if } \beta \geqslant 1 \end{cases}$$

Now let's define the following terms for notational convenience.

$$\mathcal{P}_{ij}^\beta(\mathbf{F},\alpha) = \sum_r \sum_k \alpha^{(r)}_{ijk} \left(g^{(r)}_{j-k+1} F^{(r)}_{ik} / \alpha^{(r)}_{ijk}\right)^\beta$$
$$\mathcal{Q}_{ij}^\beta(\mathbf{F},Z) = \beta Z_{ij}^{\beta-1}\left(\sum_r \sum_k g^{(r)}_{j-k+1} F^{(r)}_{ik} - Z_{ij}\right) + Z_{ij}^\beta$$

We designed the following function

$$\mathcal{W}^\beta(\mathbf{F},\alpha,Z) = \sum_{ij}\left(X_{ij}^\beta + \mathcal{I}_{ij}^\beta(\mathbf{F},\alpha,Z)\right)/\beta(\beta-1) \tag{27}$$

where
$$\mathcal{I}_{ij}^\beta(\mathbf{F},\alpha,Z) = \begin{cases} (\beta-1)\mathcal{Q}_{ij}^\beta(\mathbf{F},Z) - \beta X_{ij}\mathcal{P}_{ij}^{\beta-1}(\mathbf{F},\alpha), & \beta < 1 \\ (\beta-1)\mathcal{P}_{ij}^\beta(\mathbf{F},\alpha) - \beta X_{ij}\mathcal{P}_{ij}^{\beta-1}(\mathbf{F},\alpha), & 1 \leqslant \beta \leqslant 2 \\ (\beta-1)\mathcal{P}_{ij}^\beta(\mathbf{F},\alpha) - \beta X_{ij}\mathcal{Q}_{ij}^{\beta-1}(\mathbf{F},Z), & \beta > 2 \end{cases}$$

Consequently, if we treat $\mathcal{W}(\mathbf{F},\alpha,Z)$ as a function of $\mathbf{F}$, we have

$$\frac{\partial \mathcal{W}(\mathbf{F},\alpha,Z)}{\partial F^{(r)}_{ik}} = \begin{cases} \sum_j \left(Z_{ij}^{\beta-1}\bar{g}_{j-k+1} - \bar{F}_{ik}^{\beta-2}\bar{g}^{\beta-1}_{j-k+1}\bar{\alpha}^{2-\beta}_{ijk}X_{ij}\right), & \beta < 1 \\ \sum_j \left(\bar{F}_{ik}^{\beta-1}\bar{g}^{\beta}_{j-k+1}\bar{\alpha}^{1-\beta}_{ijk} - Z_{ij}^{\beta-2}\bar{g}_{j-k+1}X_{ij}\right), & \beta > 2 \\ \bar{F}_{ik}^{\beta-2}\sum_j \bar{g}^{\beta-1}_{j-k+1}\bar{\alpha}^{1-\beta}_{ijk}\left(\bar{F}_{ik}\bar{g}_{j-k+1} - \bar{\alpha}_{ijk}X_{ij}\right), & \text{else} \end{cases}$$

where to avoid the notational clutter, we use an overhead bar to replace the superscript (r), i.e., $\bar{F}_{ik} = F^{(r)}_{ik}, \bar{g}_k = g^{(r)}_k, \bar{\alpha}_{ijk} = \alpha^{(r)}_{ijk}$. In the following we will use these two symbols interchangablly. Then the *Hessian* matrix

$$\frac{\partial \mathcal{W}(\mathbf{F},\alpha,Z)}{\partial F^{(r)}_{ik}\partial F^{(r)}_{i'k'}} = \begin{cases} -\delta^{k,k'}_{i,i'}(\beta-2)\bar{F}_{ik}^{\beta-3}\sum_k \bar{g}^{\beta-1}_{j-k+1}\bar{\alpha}^{2-\beta}_{ijk}X_{ij}, & \beta < 1 \\ \delta^{k,k'}_{i,i'}\bar{F}_{ik}^{\beta-3}\sum_j \bar{g}^{\beta-1}_{j-k+1}\bar{\alpha}^{1-\beta}_{ijk}\bar{f}^{\beta}_{ijk}, & 1 \leqslant \beta \leqslant 2 \\ \delta^{k,k'}_{i,i'}(\beta-1)\sum_j \bar{F}_{ik}^{\beta-2}\bar{g}^{\beta}_{j-k+1}\bar{\alpha}^{1-\beta}_{ijk}, & \beta > 2 \end{cases}$$

where
$$\delta^{k,k'}_{i,i'} = \begin{cases} 1, & \text{if } i=i', k=k' \\ 0, & \text{otherwise} \end{cases} \tag{28}$$

and
$$f^\beta_{ijk} = (\beta-1)\bar{F}_{ik}\bar{g}_{j-k+1} - (\beta-2)\bar{\alpha}_{ijk}X_{ij} \tag{29}$$

Thus $\mathcal{W}(\mathbf{F},\alpha,Z)$ is convex with respect to $\mathbf{F}$, whose minumum value can be achieved at the point when $\partial\mathcal{W}(\alpha,Z,\mathbf{F})/\partial\mathbf{F} = \mathbf{O}$, from which we can get

$$F^{(r)*}_{ik} = \begin{cases} \left(\frac{\sum_j X_{ij}\bar{g}^{\beta-1}_{j-k+1}\bar{\alpha}^{2-\beta}_{ijk}}{\sum_j Z_{ij}^{\beta-1}\bar{g}_{j-k+1}}\right)^{\frac{1}{2-\beta}}, & \beta < 1 \\ \frac{\sum_j X_{ij}\bar{g}^{\beta-1}_{j-k+1}\bar{\alpha}^{2-\beta}_{ijk}}{\sum_j \bar{g}^{\beta}_{j-k+1}\bar{\alpha}^{1-\beta}_{ijk}}, & 1 \leqslant \beta \leqslant 2 \\ \left(\frac{\sum_j X_{ij}\bar{g}_{j-k+1}Z_{ij}^{\beta-2}}{\sum_j \bar{g}^{\beta}_{j-k+1}\bar{\alpha}^{1-\beta}_{ijk}}\right)^{\frac{1}{\beta-1}}, & \beta > 2 \end{cases} \tag{30}$$

Therefore for any specific α, Z, we have $\mathcal{W}(\mathbf{F}^*,\alpha,Z) \leqslant \mathcal{W}(\mathbf{F},\alpha,Z)$. Now let's treat $\mathcal{W}(\mathbf{F}^*,\alpha,Z)$ as a function of α, Z, we know that its minimum can be achieved with

$$\alpha^{(r)*}_{ijk} = \frac{g^{(r)}_{j-k+1}F^{(r)*}_{ik}}{\sum_r \sum_k g^{(r)}_{j-k+1}F^{(r)*}_{ik}},\ Z^*_{ij} = \sum_r \sum_k g^{(r)}_{j-k+1}F^{(r)*}_{ik} \tag{31}$$

Table 1: Averaged AUC Values

	1 Week	2 Weeks	1 Month	2 Months
Baseline	0.5703 ± 0.1236	0.5703 ± 0.1236	0.5703 ± 0.1236	0.5703 ± 0.1236
G-PrefixScan	0.5971 ± 0.0536	0.6018 ± 0.0479	0.6208 ± 0.0405	0.6227 ± 0.0317
MG-PrefixScan	0.6002 ± 0.0484	0.6021 ± 0.0369	0.6230 ± 0.0410	0.6231 ± 0.0420
GOSC-NMF-IN	0.5996 ± 0.0408	0.6045 ± 0.0458	0.6219 ± 0.0347	0.6223 ± 0.0258
GOSC-NMF-TN	0.6004 ± 0.0506	0.6152 ± 0.0352	0.6218 ± 0.0376	0.6315 ± 0.0308
MGOSC-NMF-IN	0.6147 ± 0.0374	$\mathbf{0.6436 \pm 0.0405}$	0.6386 ± 0.0350	0.6432 ± 0.0324
MGOSC-NMF-TN	$\mathbf{0.6208 \pm 0.0406}$	0.6377 ± 0.0370	$\mathbf{0.6417 \pm 0.0290}$	$\mathbf{0.6440 \pm 0.0312}$

Therefore $\mathcal{J}(\mathbf{F}^*, \mathbf{g}) = \mathcal{W}(\mathbf{F}^*, \alpha^*, Z^*) \leqslant \mathcal{W}(\mathbf{F}^*, \alpha, Z)$. Actually if we combine Eq.(30) and Eq.(31) together, we obtain

$$F_{ik}^{(r)} \leftarrow F_{ik}^{(r)} \left(\frac{\sum_{j=1}^{t} X_{ij} Y_{ij}^{\beta-2} g_{j-k+1}^{(r)}}{\sum_{j=1}^{t} Y_{ij}^{\beta-1} g_{j-k+1}^{(r)}} \right)^{\eta(\beta)} \tag{32}$$

where $\eta(\beta)$ is defined as in Eq.(10). In this way, assume the current iteration step is from t to $t+1$, then we have

$$\begin{aligned} &\mathcal{J}(\mathbf{F}(t+1), \mathbf{g}) = \mathcal{W}(\mathbf{F}(t+1), \alpha(t+1), Z(t+1)) \\ &\leqslant \mathcal{W}(\mathbf{F}(t+1), \alpha(t), Z(t)) \leqslant \mathcal{W}(\mathbf{F}(t), \alpha(t), Z(t)) = \mathcal{J}(\mathbf{F}(t), \mathbf{g}) \end{aligned}$$

Thus with the updating rules, the objective is monotonically decreasing, and clearly it is lower bounded by 0, thus the iterations will converge to a stationary point.

6. REFERENCES

[1] I. Batal, L. Sacchi, R. Bellazzi, and M. Hauskrecht. A temporal abstraction framework for classifying clinical temporal data. In *AMIA*, pages 29–33, 2009.

[2] S. Bengio, F. Pereira, Y. Singer, and D. Strelow. Group sparse coding. In *NIPS*, 2009.

[3] B. Cao, D. Shen, J.-T. Sun, X. Wang, Q. Yang, and Z. Chen. Detect and Track Latent Factors with Online Nonnegative Matrix Factorization. In *Proceedings of the 20th IJCAI*, pages 2689–2694, 2007.

[4] M. Cooper, T. Liu, and E. Rieffel. Video Segmentation via Temporal Pattern Classification. *IEEE TMM*, 9(3):610–618, 2007.

[5] S. de Amo and D. A. Furtado. First-order temporal pattern mining with regular expression constraints. *Data & Knowledge Engineering*, 62(3):401–420, 2007.

[6] X. Du, R. Jin, L. Ding, V. E. Lee, and J. H. T. Jr. Migration motif: a spatial - temporal pattern mining approach for financial markets. In *KDD*, pages 1135–1144, 2009.

[7] J. Eggert and E. Körner. Sparse coding and nmf. In *IJCNN*, pages 2529–2533, 2004.

[8] J. Fails, A. Karlson, L. Shahamat, and B. Shneiderman. A Visual Interface for Multivariate Temporal Data: Finding Patterns of Events across Multiple Histories. In *Visual Analytics Science And Technology, 2006 IEEE Symposium On*, pages 167–174, 2006.

[9] C. Févotte and J. Idier. Algorithms for nonnegative matrix factorization with the beta-divergence. *arXiv:1010.1763*, 2010.

[10] R. Grosse, R. Raina, H. Kwong, and A. Y. Ng. Shift-invariant Sparse Coding for Audio Classification. In *UAI*, 2007.

[11] J. Han, J. Pei, Y. Yin, and R. Mao. Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach. *DMKD*, 8(1):53–87, 2004.

[12] P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. *JMLR*, 5:1457–1469, 2004.

[13] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[14] D. D. Lee and H. S. Seung. Algorithms for Non-negative Matrix Factorization. In *NIPS 13*, pages 556–562, 2001.

[15] J. Lin, E. Keogh, S. Lonardi, and B. Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, pages 2–11, 2003.

[16] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online Learning for Matrix Factorization and Sparse Coding. *Journal of Machine Learning Research*, 11:10–60, 2010.

[17] F. Mörchen and A. Ultsch. Efficient mining of understandable patterns from multivariate interval time series. *DMKD*, 15(2):181–215, 2007.

[18] M. Mørup, M. N. Schmidt, and L. K. Hansen. Shift invariant sparse coding of image and music data. Technical report, 2008.

[19] P. N. E. Nohuddin, F. Coenen, R. Christley, and C. Setzkorn. Detecting temporal pattern and cluster changes in social networks: A study focusing uk cattle movement database. In *IFIP Advances in Information and Communication Technology*, pages 163–172, 2010.

[20] G. Norén, J. Hopstadius, A. Bate, K. Star, and I. Edwards. Temporal pattern discovery in longitudinal electronic patient records. *DMKD*, 20(3):361–387, 2010.

[21] P. D. O'Grady and B. A. Pearlmutter. Discovering convolutive speech phones using sparseness and non-negativity. In *ICA*, 2007.

[22] J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, and M. Hsu. Mining sequential patterns by pattern-growth: The prefixspan approach. *IEEE TKDE*, 16(11):1424–1440, 2004.

[23] C. Plaisant, S. Lam, B. Shneiderman, M. S. Smith, D. Roseman, G. Marchand, M. Gillam, C. Feied, J. Handler, and H. Rappaport. Searching electronic health records for temporal patterns in patient histories: a case study with microsoft amalga. In *AMIA*, pages 601–605, 2008.

[24] M. Robert Moskovitch and Y. Shahar. Medical temporal-knowledge discovery via temporal abstraction. In *AMIA*, pages 452–456, 2009.

[25] B. R. Shah, J. Drozda, and E. D. Peterson. Leveraging observational registries to inform comparative effectiveness research. *American Heart Journal*, 160(1):8–15, 2010.

[26] P. Smaragdis. Non-negative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs. In *ICA*, pages 494–499. 2004.

[27] H. A. Varian. *Microeconomic Analysis Third Edition*. W.W. Norton and Company, 1992.

[28] F. Wang, C. Tan, P. Li, and C. König. Efficient document clustering via online nonnegative matrix factorization. In *Proceedings of the 11th SDM*, 2011.

[29] T. Wang, C. Plaisant, A. Quinn, R. Stanchak, B. Shneiderman, and S. Murphy. Aligning temporal data by sentinel events: Discovering patterns in electronic health records. In *Proc. of ACM Conference on Human Factors in Computing Systems*, pages 457–466, 2008.

[30] M. J. Zaki. Spade: An efficient algorithm for mining frequent sequences. *Machine Learning*, 42:31–60, 2001.