Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

TaGiTeD: Predictive Task Guided Tensor Decomposition for Representation Learning from Electronic Health Records

Kai Yang,[1,2] Xiang Li,[3] Haifeng Liu,[3] Jing Mei,[3] Guotong Xie,[3]
Junfeng Zhao,[1,2*] Bing Xie,[1,2] Fei Wang[4]
[1]Software Engineering Institute, Peking University, Beijing 100871, China
[2]Key Laboratory of High Confdence Software Technologies, Ministry of Education, Beijing 100871,China
[3]IBM Research-China, Beijing, China
[4]Department of Healthcare Policy and Research, Cornell University
{smileyk, zhaojf, xiebing} @pku.edu.cn, {lixiang, liuhf, meijing,XIEGUOT} @cn.ibm.com, feiwang03@gmail.com

Abstract

With the better availability of healthcare data, such as Electronic Health Records (EHR), more and more data analytics methodologies are developed aiming at digging insights from them to improve the quality of care delivery. There are many challenges on analyzing EHR, such as high dimensionality and event sparsity. Moreover, different from other application domains, the EHR analysis algorithms need to be highly interpretable to make them clinically useful. This makes representation learning from EHRs of key importance. In this paper, we propose an algorithm called *Predictive Task Guided Tensor Decomposition* (TaGiTeD), to analyze EHRs. Specifically, TaGiTeD learns event interaction patterns that are highly predictive for certain tasks from EHRs with supervised tensor decomposition. Compared with unsupervised methods, TaGiTeD can learn effective EHR representations in a more focused way. This is crucial because most of the medical problems have very limited patient samples, which are not enough for unsupervised algorithms to learn meaningful representations form. We apply TaGiTeD on real world EHR data warehouse and demonstrate that TaGiTeD can learn representations that are both interpretable and predictive.

Introduction

With the rapid development of computer technologies, more and more healthcare data, such as Electronic Health Records (EHR), are becoming readily available. In the current big data era, those healthcare data are invaluable resources that carry important insights for the various aspects of care delivery. Data analytics technologies, such as data mining and machine learning, are important tools for digging those insights out from the healthcare data.

There are many aspects that are unique for patient EHRs that make their analysis very challenging. For example, many heterogeneous medical events, such as diagnosis, procedures, medications and lab tests are all included. This makes the dimensionality of EHR very high (because of the large number of unique medical events). Also every patient only has a limited number of visits to hospitals or clinical facilities. Thus the patient EHRs are usually very sparse.

*Corresponding author

In order to overcome those challenges, researchers usually do not directly work with raw EHRs. Instead, they first project the medical events in a lower dimensional space and then build follow up analytical algorithms in that space. The process of obtaining such space is usually called representation learning, and the different dimensions in that space are usually referred to novel EHR representations. Many approaches have been adopted or developed for representation learning (Bengio, Courville, and Vincent 2013). With the new representations, not only the dimensionality of EHR is reduced, but also the entries are becoming denser (Macskassy et al. 2014).

Recently, Ho *et al*. (Ho, Ghosh, and Sun 2014) proposed a representation approach from EHR with tensor decomposition. The advantage of their method, comparing to traditional vector based approaches, is the capability of capturing high order event interactions through tensors. This is crucial because in reality, it is unlikely only single medical event contributes to a specific healthcare problem, high order event interactions always matter. The authors showed that the learned EHR representations (a.k.a. phenotypes) are not only leading to better prediction performance, but also highly interpretable.

However, despite the success, there are still limitations in those existing tensor based approaches.

- *They are unsupervised*. Typically unsupervised learning will need many more samples to achieve meaningful results comparing to supervised learning methods (Goodfellow, Courville, and Bengio 2012). Different from other application domains like image or speech analysis (Hinton et al. 2012), we always have limited patient samples in healthcare problems, which means we cannot obtain as many data as we want. This will make the results of completely unsupervised learning not reliable.
- *They are two-stage*. Existing algorithms typically work in a two-stage way. In the first stage they learn novel EHR representations. In the second stage they build analytical models, such as predictive models, on top of those representations. There is no guarantee that the representations learned from the first stage can lead to good performance in the second stage.

In this paper, we propose a novel EHR representation

learning algorithm called *Predictive Task Guided Tensor Decomposition* (TaGiTeD). TaGiTeD is based on tensor decomposition so it can still capture the high order event interactions as those existing methods, but it is guided by specific prediction tasks through learning specific representations that can lead to the best prediction performance. In this way, we achieve a nice balance between interpretability and performance. Also we learn such representations in an integrated way by solving an optimization problem with both representation learning and predictive modeling in the objective. Efficient solution strategies are developed with proximal gradient methods (Wang et al. 2015a). Experimental results on large scale real world EHR warehouse are presented to demonstrate the effectiveness of the proposed algorithm.

Predictive Task Guided Tensor Decomposition

In this section, we first describe the preliminaries of tensor decomposition and formulate the problem. Finally, we provide a general overview of the framework and design a concrete algorithm for solving the problem.

Notes and Preliminaries

Throughout, scalars are denoted by lowercase letters (a), vectors by boldface lowercase letters ($\mathbf{v}$), matrices by boldface capital letters ($\mathbf{A}$), and higher order tensors by boldface Euler script letters ($\mathcal{X}$). We use multi-index notation so that a boldface $\mathbf{i}$ represents the index $(i_1, \ldots, i_N)$.

The notation $\|.\|$ refers to the two-norm for vectors or Frobenius norm for matrices, i.e., the sum of the squares of the entries. Let $\mathbf{X}^T, \mathbf{X}^{-1}, \mathbf{X}^\dagger$ denote the transpose, inverse, Moore-Penrose pseudoinverse of X. Let $\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}$ represent a sequence of N factor matrices for N dimensions. And $*$ and $\oslash$ denote elementwise production and elementwise division.

Definition 1. *(Tensor Matricization): The mode-n matricization or unfolding of a tensor $\mathcal{X}$ is denoted by $\mathbf{X}_{(n)}$ and is of size $I_n \times J_n$, where $J_n \equiv \prod_{(m \neq n)} I_m$.*

Definition 2. *(KhatriRao Product): Give two matrices $\mathbf{A}$ and $\mathbf{B}$ of sizes $I_1 \times R$ and $I_2 \times R$; then $\mathbf{C} = \mathbf{A} \odot \mathbf{B}$ is a matrix of size $I_1 I_2 \times R$ such that*

$$\mathbf{C} = [\mathbf{a}_1 \otimes \mathbf{b}_1 \quad \mathbf{a}_2 \otimes \mathbf{b}_2 \quad \ldots \quad \mathbf{a}_\mathbf{R} \otimes \mathbf{b}_\mathbf{R}]$$

where the Kronecker product $\otimes$ of two vectors $\mathbf{a}, \mathbf{b}$ of size I_1 and I_2 is a vector of length $I_1 I_2$

Definition 3. *(CANDECOMP/PARAFAC Decomposition): CP decomposes a tensor into a sum of component rank-one tensors. For example, given a third-order tensor $\mathcal{X} \in \mathbb{R}^{(\mathbf{I} \times \mathbf{J} \times \mathbf{K})}$, we wish to write it as*

$$\mathcal{X} \approx \sum_{r=1}^{\mathbf{R}} (\mathbf{a}_r \cdot \mathbf{b}_r \cdot \mathbf{c}_r) \tag{1}$$

where R is a positive integer, and $\mathbf{a}_r \in \mathbb{R}^\mathbf{I}, \mathbf{b}_r \in \mathbb{R}^\mathbf{J}, \mathbf{c}_r \in \mathbb{R}^\mathbf{K}, for \quad r = 1, ..., R$. The factor matrices refer to the combination of the vectors from the rank-one components, i.e., $\mathbf{A} = [\mathbf{a}_1 \quad \mathbf{a}_2 \quad \ldots \quad \mathbf{a}_R]$ and likewise for $\mathbf{B}$ and $\mathbf{C}$. CP model can be concisely expressed as: $\mathcal{X} \approx [\vec{\lambda}; \mathbf{A}, \mathbf{B}, \mathbf{C}]$

Formulation

We formulate our model TaGiTeD as a constrained tensor optimization where the constraints are a kind of penalty terms for guidance. Thus our model is composed of two terms: Poisson Tensor Decomposition and Factor Matrix Constraints. The general formulation is:

$$\min_{\mathcal{M}} \mathcal{L}(\mathcal{X}, \mathcal{M}) = \min_{\mathcal{M}} \{\Psi(\mathcal{X}, \mathcal{M}) + \Upsilon(\mathcal{M})\} \tag{2}$$

where,

$$\Psi(\mathcal{X}, \mathcal{M}) = \sum m_\mathbf{i} - x_\mathbf{i} \log m_\mathbf{i} \tag{3}$$

$$\Upsilon(\mathcal{M}) = \frac{1}{|\zeta|} \sum loss\left(\mathbf{A}^{(n)}, \mathbf{Y}^{(n)} | \mathcal{H}^{(n)}\right) \tag{4}$$

$$s.t. \quad \mathcal{X} \in \mathbb{R}^{(I_1, \ldots, I_N)},$$
$$\mathcal{M} = [\vec{\lambda}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, ..., \mathbf{A}^{(N)}] \in \Omega_\lambda \times \Omega_1 \times \Omega_2 \times ... \times \Omega_N,$$
$$\Omega_n = \left\{\mathbf{A} \in [\gamma_n, 1]^{\mathbf{I}_n \times \mathbf{T}_R} \, | \|\mathbf{a}_r\| = 1 \; for \; r = 1, ..., \mathbf{T}_R\right\},$$
$$\mathbf{Y}^{(n)} \in \mathbf{I}_n \times 1, \mathcal{H} : \mathbf{A}^{(n)} \mapsto \phi\left(\mathbf{A}^{(n)}; \Theta\right),$$
$$\Omega_\lambda = [0, +\infty)^R, \mathbf{T}_R \in \mathbb{R}^+$$

$\mathcal{X}$ is the partially observed tensor with size $(I_1, ..., I_N)$ which is constructed by the count data of patient events from the EHR. $\mathcal{M}$ is the unknown estimated tensor with same size, which can be represented as the tensor itself or its constituent parts. The $m_\mathbf{i}$ and $x_\mathbf{i}$ represent the element of $\mathcal{X}$ and $\mathcal{M}$ with the multi-index notation $\mathbf{i}$. $\mathbf{Y}^{(n)}$ represents the n^{th} predictive task target. ζ is the total instances in training set where we have label $\mathbf{Y}^{(n)}$. That means, for example, we let $Y_i^{(n)}$ = 1 or - 1 whether a patient i will be hospitalized or not. With the explicit label information, we can define the regularization equation (4) directly with a discriminative model as penalty term: $loss\left(\mathbf{A}^{(n)}, \mathbf{Y}^{(n)} | \mathcal{H}^{(n)}\right)$. The $\mathcal{H}^{(n)}$ can be different for different prediction models, which sets the constraints on the n^{th} factor matrices $\mathbf{A}^{(n)}$. Therefore there may be N predictive tasks guiding the estimated tensor $\mathcal{M}$ in N dimensions. We set $\mathbf{T}_R$ as the possible rank of original tensor in advance.

Alternating Poisson Regression

In many applications such as chemometrics, the model can be fit to the data using a least squares (LS) criteria, implicitly assuming that the random variation in the tensor data follows a Gaussian distribution. In the case of count data, however, the random variation is better described via a Poisson distribution $x_i \sim Poisson(m_i)$ (Chi et al. 2011). Thus, for count data we should minimize the (generalized) Kullback-Leibler (KL) divergence like equation (3) which equals the negative log-likelihood of the observations up to an additive constant.

In this paper, we extend the CP-APR (Alternating Poisson Regression) algorithm (Chi et al. 2011) fitting the nonnegative Poisson tensor decomposition (CP-APR) to count data. We note that we can express the same $\mathcal{M}$ in different ways, i.e.

$$\mathcal{M} = [\mathbf{A}^{(1)}, ..., \mathbf{A}^{(i-1)}, \mathbf{B}^{(i)}, \mathbf{A}^{(i+1)}, ..., \mathbf{A}^{(N)}] \tag{5}$$

where

$$\mathbf{B}^{(i)} = \mathbf{A}^{(i)}\Lambda, \quad \Lambda = diag\,(\lambda) \tag{6}$$

The weights above are omitted because they are absorbed into the n^{th} mode. From equation (5), we can express $\mathcal{M}$ as $\mathcal{M}_{(i)} = B^{(i)}\Pi^{(i)}$, where $B^{(i)}$ is defined in equation (6) and we use the matrix $\Pi^{(i)}$ to denote the fixed parts as follows,

$$\Pi^{(i)} \equiv \left(\mathbf{A}^{(1)} \odot ... \odot \mathbf{A}^{(i-1)} \odot \mathbf{A}^{(i+1)} \odot ... \odot \mathbf{A}^{(N)}\right)^T \tag{7}$$

Thus, we can rewrite the objective function in equation (3) as

$$\Psi\left(\mathcal{X}, \mathcal{M}\right) = \mathbf{e}^T \left[\mathbf{B}^{(i)}\Pi^{(i)} - \mathbf{X}_{(i)} * \log\left(\mathbf{B}^{(i)}\Pi^{(i)}\right)\right]\mathbf{e} \tag{8}$$

where $\mathbf{e}$ is the vector of all ones, and the log function is applied element wise. $\mathbf{X}_{(i)}$ is the mode-i matricization of the observed tensor. The updates for $\vec{\lambda}$ and $\mathbf{A}^{(i)}$ come directly from $\mathbf{B}^{(i)}$.

Factor Matrix Constraints

The features obtained by tensor decomposition can be directly used for various prediction tasks (Classification and Regression). As a kind of unsupervised representation learning method, tensor decomposition do not contain any category (label of a class) information which is often useful for representation learning. To exploit such information, we make full use of the prediction target to guide our tensor decomposition. We discuss the cases of penalty terms from Classification and Regression respectively.

Classification Suppose the prediction task is to predict whether a patient P_n will be hospitalized in the next year. We let $Y_n = 1\ or\ -1$ represent hospitalization or not. With the label information, we can define the penalty term $\Upsilon(\mathcal{M})$:

$$\Upsilon(\mathcal{M}) = -\frac{1}{|\zeta|}\sum_{n\in\zeta}\log \Pr\left(\mathbf{A}_n^{(i)}, \mathbf{Y}_n^{(i)}|\mathcal{H}^{(i)}\right) \tag{9}$$

which is term as average log-loss. Formally we use superscript (i) to represent the i^{th} factor matrix and subscript n for the n^{th} instance of dataset. Here, ζ is the total instances in training set where we have label $\mathbf{Y}^{(i)}$. Consider the classifier is Logistic Regression, we can re-define the cost function as $\Pr = 1/\left[1 + \exp\left(-\mathbf{Y}_n^{(i)} f(\mathbf{A}_n^{(i)})\right)\right]$, where the linear model $\mathcal{H} : \mathbf{A}^{(i)} \mapsto f\left(\mathbf{A}^{(i)}\right) = \mathbf{A}^{(i)}\Theta + \theta$ and the (Θ, θ) is parameters in the model $\mathcal{H}$.

In addition to the log-loss for the representation model, hinge loss can also be used with the linear model $\mathcal{H}$.

$$\Upsilon(\mathcal{M}) = \frac{1}{|\zeta|}\sum_{n\in\zeta}\max\left\{0, 1 - \mathbf{Y}_n^{(i)} f(\mathbf{A}_n^{(i)})\right\} \tag{10}$$

which is corresponded to the SVM classifier. Both regularizations are served as penalty term which can utilize the target information to guide the overall decomposition.

Regression Suppose the regression problem is to predict how much a patient P_n will spend on medical services in the next year. We let Y_n represent the medical expense which is a continuous value. Thus, we use the loss function of linear regression which is the least-squares function as the penalty term:

$$\Upsilon(\mathcal{M}) = \frac{1}{2|\zeta|}\sum_{n\in\zeta}\|\mathbf{Y}_n^{(i)} - f(\mathbf{A}_n^{(i)})\|^2 \tag{11}$$

Unified Optimization

The unified framework can be obtained by adding the two components: Poisson Tensor Decomposition and Factor Matrix Constraints. We solve equation (2) via an alternating approach which is similar to CP-APR, holding all factor matrices constants except the one being updated, which is presented in **Algorithm 1** TaGiTeD Algorithm .

Log-loss Regularization: Considering the problem for the i^{th} factor matrix, if we use the Logistic Regression as the classifier, the penalty term $\Upsilon(\mathcal{M})$ can be same to equation (9) and the objective function is :

$$\begin{aligned}\mathcal{L} = \min_{\mathcal{M},\Theta,\theta}\Bigg\{&\mathbf{e}^T\left[\mathbf{B}^{(i)}\Pi^{(i)} - \mathbf{X}_{(i)} * \log\left(\mathbf{B}^{(i)}\Pi^{(i)}\right)\right]\mathbf{e} \\ &- \frac{\lambda_0}{|\zeta|}\sum_{n\in\zeta}\log\left(1 + \exp\left[-\mathbf{Y}_n\left(\mathbf{B}_n^{(i)}\Theta + \theta\right)\right]\right) \\ &+ \frac{\lambda_1}{2}\left(\|\Theta\|^2 + \theta^2\right)\Bigg\}\end{aligned} \tag{12}$$

Note that the size of $\Pi^{(i)}$ is $\mathbf{T}_R \times \prod_{j=1,j\neq n}^{N} I_j$ and the size of vector Θ is $\mathbf{T}_R \times 1$. The multiplicative update derivation for the weight factor matrix $\mathbf{B}^{(i)}$ with size of $I_i \times \mathbf{T}_R$ can be computed by taking the partial derivative of the objective function respect to a single element $b_{pq}^{(i)}$:

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial b_{pq}^{(i)}} = &\sum_{l=1}^{\Pi_{j=1,j\neq n}^{N} I_j}\Pi_{ql}^{(i)}\left(1 - \frac{x_{pl}^{(i)}}{\sum_{r=1}^{R} b_{pr}^{(i)}\Pi_{ql}^{(i)}}\right) \\ &- \frac{\lambda_0}{|\zeta|}\frac{Y_p\Theta_q}{1 + \exp\left(Y_p\sum_{k=1}^{R+1}\hat{b}_{pk}^{(i)}\hat{\Theta}_k\right)}\end{aligned}$$

Where, $b_{pq}^{(i)}$ represents the value of matrix $\mathbf{B}^{(i)}$ whose indices are (p,q) and likewise for $\Pi_{ql}^{(i)}$ and $x_{pl}^{(i)}$. $\hat{b}_{pk}^{(i)}$ is the (q,k) value of appended matrix $\hat{\mathbf{B}}^{(i)}$ with size of $I_i \times (\mathbf{T}_R + 1)$ which adds a column of 1. Setting the gradient descent step size to $\alpha = b_{pq}^{(i)}/\sum_{l=1}^{\Pi_{j=1,j\neq n}^{N} I_j}\Pi_{ql}^{(i)}$ yields the multiplicative update.The non-negative factor matrix constraints are also satisfied using the multiplicative update. Generalizing for the entire factor matrix, the update equation is as follows:

$$\mathbf{B}^{(i)} := \mathbf{B}^{(i)} * \left[\Phi^{(i)} - \frac{\lambda_0}{|\zeta|}\left(Y_n * \Theta\right) \oslash \left(\exp\left(\mathbf{B}^{(i)}\Theta * Y_n\right) + \mathbf{E}\right)\right] \tag{13}$$

where the matrix $\Phi^{(i)} \equiv \left[\mathbf{X}_{(i)} \oslash \left(\mathbf{B}^{(i)}\Pi^{(i)}\right)\right]\Pi^{(i)^T}$ will come up repeatedly in this paper. If we do not add any constraints, the update equation is :

$$\mathbf{B}^{(i)} := \mathbf{B}^{(i)} * \Phi^{(i)} \tag{14}$$

The added parameter Θ in equation (13) can be solved by just computing the Logistic Regression.

Hinge-loss Regularization: Combining the hinge loss regularization equation (10) with equation (8), we obtain the overall objective function which involves a sum of convex differentiable term and convex non-differentiable regularizations. To update $\mathbf{B}^{(i)}$, since the hinge loss is non-differentiable, we take the general proximal gradient optimization approach.

If the convex nondifferentiable function is equation (10), it follows that

$$\mathbf{B}^{(i)} := \mathbf{B}^{(i)} * \Phi^{(i)} + \text{prox}_{\frac{\lambda_1}{|\zeta|}\Upsilon}\left(\mathbf{B}^{(i)}\right) \tag{15}$$

where,

$$\text{prox}_{\tau\Upsilon}\left(\mathbf{B}_n^{(i)}\right) = \mathbf{B}_n^{(i)} + Y_n \,\text{proj}_{[0,\tau]}\left(\frac{1 - Y_n f\left(\mathbf{B}_n^{(i)}\right)}{\|\Theta\|^2}\right)\Theta'$$

The proj is defined as $\text{proj}_{[0,\tau]}(\alpha) = \begin{cases} \alpha & \alpha \in [0,\tau] \\ 0 & \alpha < 0 \\ \tau & \alpha > \tau \end{cases}$.

To update (Θ, θ) in the hinge loss regularization, the objective is to compute exactly the SVM on training data $\left\{\left(B_n^{(i)}, Y_n\right) | n \in N\right\}$

Quadratic-loss Regularization To solve the regression tasks, we combine the least-square loss function equation (11) and equation (8). Similar to the optimization of classification tasks, we can update $B^{(i)}$.

$$\mathbf{B}^{(i)} := \mathbf{B}^{(i)} * \Phi^{(i)} - \frac{\lambda_1}{|\zeta|}\left(Y_n - B^{(i)}\Theta\right)\Theta^T \tag{16}$$

And the Θ can be computed directly by solving the linear regression. $\Theta = \left(\mathbf{B}^{(i)^T}\mathbf{B}^{(i)} + CI\right)^{-1}\mathbf{B}^{(i)^T}Y_n$

Thus overall representation learning precess is presented in **Algorithm 1** TaGiTeD Algorithm .

Algorithm 1 TaGiTeD algorithm (general version).

Input: $\mathcal{X}, \mathbf{Y}, \mathbf{T}_R, OutIter, InerIter, TMode, \lambda_1, \lambda_2$
Output: $\mathcal{M}$ in shape of **Eq. 5**

Initialize $\mathcal{M}$, Θ of $\mathcal{H}$;
while *not converged* and $Iter_{Now} \leqslant OutIter$ **do**
  // For each mode n.
  for $n = 1$ to N **do**
    if $n == TMode$ **then**
      Compute Θ of $\mathcal{H}$ given $\mathbf{B}^{(n)}$ of $TMode$;
      while *not converged* and $iter \leqslant InerIter$ **do**
        Update $\mathbf{B}^{(n)}$ by **Eq. 13** | **Eq. 15** | **Eq. 16**;
      end while
    else
      while *not converged* and $iter \leqslant InerIter$ **do**
        Update $\mathbf{B}^{(n)}$ **by Eq. 14**;
      end while
    end if
  end for
end while

Tensor Projection

To avoid the overfitting, we only use the TaGiTeD to compute the training data and project the testing data by setting unconstrained factor matrices fixed. That means we project the testing data onto the feasible space which is already trained by TaGiTeD. Suppose the original tensor $\mathcal{X} = \|\vec{\lambda}; A^{(1)}, A^{(2)}, A^{(3)}\|$ is a mode-3 tensor. And we need to compute the factor matrix $A^{(1)}$ as the feature representation. We split the original tensor $\mathcal{X}$ along the mode-1 into training tensor $\mathcal{X}_{train}$ and testing tensor $\mathcal{X}_{test}$. First we compute the supervised tensor decomposition TaGiTeD on $\mathcal{X}_{train}$, and then use unsupervised tensor decomposition method to project $\mathcal{X}_{test}$ by only computing $A_{test}^{(1)}$ and reusing the $\left(A_{train}^{(2)}, A_{train}^{(3)}\right)$.

Experiment Evaluation

In this section, we showed several cases of TaGiTeD on two real-world EHR datasets which construct small dense tensor and big sparse tensor. Then, we demonstrated the performance of the proposed algorithms and interpretability of mined representations.

Datasets

We conducted experiments on two real-world EHR datasets. The data is actually a collection of medical events where each event can be described as a patient p is required to take a medicine m for a diagnosis d . The datasets also contain the hospitalization and medical expenses information of patients, which can be defined as the predictive targets.

Sparse Big Data focuses on a subset of patients whose records span over 3 years. It contains $34,395$ patients with a total of 758 diagnoses, $1,765$ medicines and 34 million medical events. The proportion of non-zero elements in this sparse tensor is 0.00747%

Dense Small Data is a subset of previous dataset. It contains $9,000$ patients with a total of 247 diagnoses and 816 medicines.The proportion of non-zero elements in this densor tensor is 0.1626%

We used the datasets above to construct 3-modes tensors which respectively represent patient, diagnosis and medicine. Each element of the constructed tensor was the number of the corresponding medical events happened during the observation time window (We set observation window 2 years and prediction window 1 year after observation window).We evaluated the performance on the tasks of classification and regression respectively.

- **Classification: One-Year Hospitalization Prediction**: we utilized the medical events in the observation window to predict whether a patient will be hospitalized or not in the prediction window. We evaluated the preformance in terms of Area under the ROC curve (AUC) .
- **Regression: One-Year Medical Expense Prediction**: We predicted how much a patient will spend in hospital by utilizing the medical events in the observation window, and we used two evaluation metrics: the R2 coefficient of determination and Root Mean Squared Error (RMSE).

Baseline Algorithm

We compared our proposed TaGiTeD algorithm with the following four methods, Chi-squared based feature selection (**FS**), principal components analysis (**PCA**), Gaussian mixture model (**GMM**), and nonnegative CP alternation Possion regression (**CP-APR**) (Chi et al. 2011) for tensor de-

Table 1: Regression Result on Small Dense Dataset

	Metrics	Vec+FS	Vec+PCA	Vec+GMM	CP-APR	TaGiTeD-Quad
LinearR	RMSE	25.67	37.64	19.41	10.750	7.79
	R2	0.1731	0.1675	0.2302	0.2653	0.2830

Table 2: AUC on Small Dense Dataset

Methods	LR	RF	SVM
Vec+FS	0.7230	0.7410	0.7048
Vec+PCA	0.7190	0.695	0.7118
Vec+GMM	0.7476	0.7331	0.7293
CP-APR	0.7593	0.7435	0.7488
TaGiTeD-Log	**0.7886**	**0.7627**	0.7665
TaGiTeD-Hinge	0.7771	0.7475	**0.7877**

Table 3: AUC on Big Sparse Dataset

Methods	LR	RF	SVM
Vec+FS	0.6211	0.6181	0.607
Vec+PCA	0.6121	0.6048	0.5898
Vec+GMM	0.6126	0.6296	0.6128
CP-APR	0.6325	0.6529	0.6266
TaGiTeD-Log	**0.653**	0.6465	0.6426
TaGiTeD-Hinge	0.6513	**0.655**	**0.6432**

Table 4: Representations of Three Phenotypes

Medicines	Diagnoses
Low Molecular Weight Heparin Sodium Recombinant Human Erythropoietin Anti-Renal Failure Tablets Levocarnitine Injection Recombinant Human Erythropoietin β ...	Chronic kidney Renal failure ...
Difenidol Hydrochloride Tablets Paracetamol and Caffeine Tablets Po Chai Pills Amoxicillin Capsules Norfloxacin Capsules ...	Acute nasopharyngitis Cough Acute laryngitis Conjunctivitis Dizziness & giddiness ...
Sufentanil Citrate Injection Tinidazole and Glucose Ampicillin Sodium Oxytocin Carboprost Methylate ...	Diabetes in pregnancy Person consulting ...

composition on count data. FS, PCA and GMM are vector-based traditional representation learning methods and CP-APR is tensor-based methods. In vector-based methods, we concatenated the diagnosis vector to medicine vector to obtain a long combined vector for each patient. Then we learn the principal components or clusters from training datasets and transformed the testing datasets based on the learned models. In CP-APR method, we costructed a original tensor. Each element $\mathcal{X}_{ijk}$ represented the count number of this event happened. Tensor-based baseline algorithms constructed training tensor and testing tensor seperately from training and testing datasets. And after computing training tensor decomposition, we projected the testing tensor onto it by fixing unconstrained factor matrices.

Experimental Details

We set the proportion of validation set, training set and testing set 10:7:3. We implemented the FS, PCA, and GMM based on scikit-learn 0.17. We let K be the number of cluster centers or principal components in the range of [10,200]. For classification task, we used Logistic Regression, Random Forests and SVM to train the tasks' models. In detail, we set 100 as the number of trees in Random Forests Classifier. For regression task, we used Linear Regression model. We also computed the 10-folds cross validation on the validation set for the parameters of models.

Experimental Results

In our first experiment, we evaluated the predictive performance of our method on small Dense Dataset. K value represents the length of learned representations. We first used the validation dataset to get the K value for each method, then computed the representation learning on the training dataset, Finally we used the learned representation of the testing dataset to calculate results. As shown in Table 2, The TaGiTeD-Log achieved the best performance using Logistic Regression and Random Forests classifiers. And TaGiTeD-Hinge could get the best performance using SVM classifier. This also indicated that using related loss function as penalty terms could help corresponding representation learning. Table 1 shows the regression tasks' results, and the TaGiTeD-Quad outperformed the other methods. Both metrics with good performance shows that TaGiTeD-Quad could extract directional and effective representations.

Figures 1$\sim$4 shows the detail results of prediction tasks given different K values in range of [10,100]. We compared the ability of feature reduction by comparing the results of representations with same length . LR^1 represents Logistic Regression model and LR^2 represents Linear Regression model. The baseline algorithms got poor performance at small K value and increased slowly. However, our methods could achieve the peak value of result when K value equals to 50, which means our methods have better information extraction ability.

When we computed *Predictive Task Guided Tensor Decomposition* on sparse big data, it was hard and time consuming to compute whole original tensor even when the internal memory was big enough. In our implementation, we constructed the tensor only using the non-zero elements' indices and values: (v^q, c), which adopted the sparse tensor implementation approaches presented in (Chi et al. 2011; Ho, Ghosh, and Sun 2014). Table 3 shows the preformance of this experiment, and we can find that the TaGiTeD-Log and TaGiTeD-Hinge always get better scores using correspond-

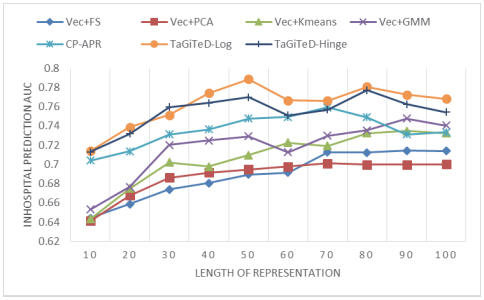

Figure 1: AUC of Classificatin Using LR^1

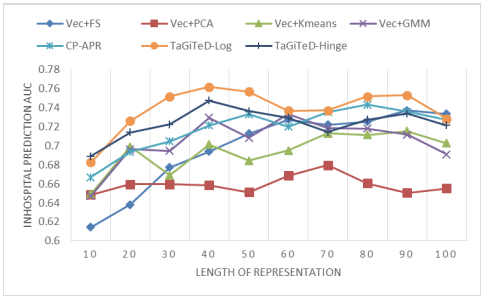

Figure 2: AUC of Classificatin Using RF

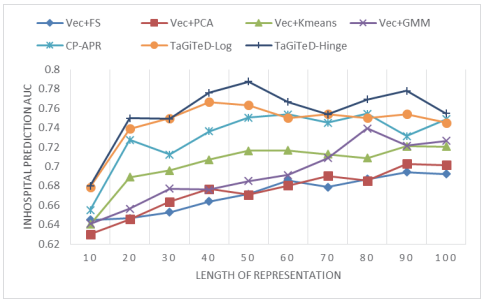

Figure 3: AUC of Classificatin Using SVM

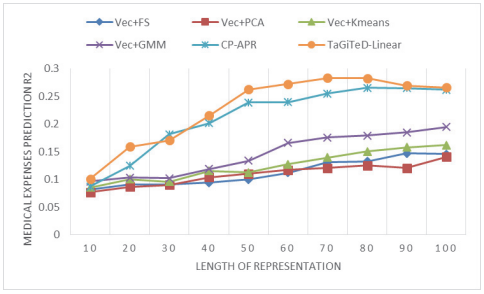

Figure 4: R2 of Regression Using LR^2

ing classifiers. In general, our method also get good performance on sparse big data. Figure 5 shows the runtime comparison between CP-APR algorithm with TaGiTeD methods and our methods needed acceptable time in each iteration.

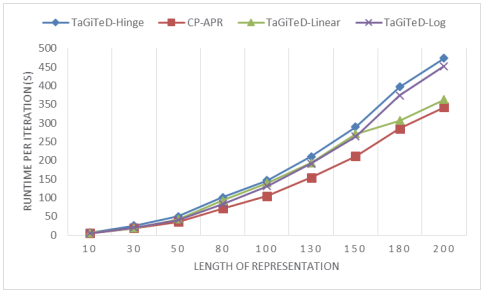

Figure 5: RunTime of Each Iteration Comparison

In addition, one key advantage of TaGiTeD is that it can lead to clinically meaningful representations. In the setting of solving hospitalization prediction using TaGiTeD-Log (with $K = 80$), we had 80 meaningful phenotypes after representation learning. Due to the space limitation, Table 4 shows the diagnoses and medicines of three phenotypes. Clinical experts verified that the diagnoses of first phenotype is Chronic kidney related diseases, and the medicines are required injections and preparations of Hemodialysis operation or something controlling Renal failure. The second phenotype represents that if a patient has a cold, he/she may have the symptoms and take the medicines listed in Table 4. And the last phenotype is mainly for diabetes patients in pregnancy.

Related Work

Representation Learning The performance of machine learning methods heavily depends on the choice of data representations (or features). Representation learning has become a field in the machine learning community especially in Speech Recognition (Hinton et al. 2012), Signal Processing (Jos et al. 2012) and Natural Language Processing (Bengio 2008). In this paper, we focused on the medical prediction tasks on EHR data. Since EHR data only provides the structured clinical data from which it is hard to learn medical concepts, we need an appropriate representation learning method for EHR data.

The quest for AI is motivating the design to be more powerful representation-learning algorithms. We first consider the traditional feature extraction methods, principal components analysis (PCA)(Roweis 1999) and Autoencoder (Hinton and Zemel 1994). There are numerous examples of their successful applications as feature representation schemes (Coates and Ng 2011). Contractive Auto-Encoders (CAE), proposed by (Rifai et al. 2011), follow up on Denoising Auto-Encoders (DAE) and share a similar motivation of learning robust representations. However, these state-of-the-art representation learning methods are either very limited to form deeper, more abstract representations or hard to understand the meaning of representations which is important for medical field.

Tensor Decomposition Therefore we use a natural description for EHR data to achieve good performance and meaningful representation (Kolda and Gibson 2006). There are many applications of tensor decomposition, like missing

data imputation (Gandy, Recht, and Yamada 2011), feature extraction (Phan and Cichocki 2010). Many works are also proposed on improving the optimization algorithm of tensor decomposition (Bhaskara et al. 2014; Kolda 2015). It is desirable to impose a non-negativity constraint on tensor factorizations in order to facilitate easier interpretation when analyzing non-negative data (Welling and Weber 2001). (Chi et al. 2011) proposed nonnegative CP alternation Possion regression (CP-APR) model for the case of count data. Based on this algorithm, (Ho, Ghosh, and Sun 2014) proposed constrained CP-APR model by adding bias tensor.(Wang et al. 2015b) proposed a constrained non-negative tensor factorization by adding prior knowledge.(Chi et al. 2011; Wang et al. 2015b) are both medical applications of tensor for phenotyping which indicates that decomposed result is interpretable in medical field. Based on(Chi et al. 2011), in this paper, we focused not only on the predictive accuracy but also the clinical meaning of learned representation.

Conclusion and Future Work

In this paper, we proposed a framework of supervised tensor decomposition (TaGiTeD) which is guided by the predictive tasks for representation learning. This framework is able to learn abstract representation for specific prediction tasks and achieve better preformance. Moreover, the learned representation can be seen as meaningful phenotypes that are helpful for clinicians. While EHR data also contains other auxiliary information, like demographic information and laboratory results, it is helpful in some scenarios if we utilize the auxiliary information in the process of tensor decompostion. Like the approach of adding prior knowledge in Rubik (Wang et al. 2015b), we can represent the information as extra matrics that sharing the same dimension with the tensor and constrain the corresponding factor result. Futhermore, our method is limited to a single prediction task by putting constraints upon one dimension, and multi-task learning is also a good avenue for addressing this limitation. We leave these generalizations to future work.

Acknowledgments

This research was sponsored by the National Natural Science Foundation of China under Grant No.61472007, China National Funds for Distinguished Young Scientists No. 61525201. Fei Wang is partially supported by National Science Foundation under Grant Number IIS-1650723.

References

Bengio, Y.; Courville, A.; and Vincent, P. 2013. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35(8):1798–1828.

Bengio, Y. 2008. Neural net language models. *Scholarpedia* 3(1).

Bhaskara, A.; Charikar, M.; Moitra, A.; and Vijayaraghavan, A. 2014. Smoothed analysis of tensor decompositions. In *ACM Symposium on Theory of Computing*, 594–603.

Chi; Eric, C.; Kolda; and Tamara, G. 2011. On tensors, sparsity, and nonnegative factorizations. *Siam Journal on Matrix Analysis and Applications* 33(33):1272–1299.

Coates, A., and Ng, A. Y. 2011. The importance of encoding versus training with sparse coding and vector quantization. In *International Conference on Machine Learning, ICML 2011, Bellevue, Washington, Usa, June 28 - July*, 921–928.

Gandy, S.; Recht, B.; and Yamada, I. 2011. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems* 27(2):25010–25028(19).

Goodfellow, I. J.; Courville, A.; and Bengio, Y. 2012. Spike-and-slab sparse coding for unsupervised feature discovery. *Computer Science*.

Hinton, G. E., and Zemel, R. S. 1994. Autoencoders, minimum description length and helmholtz free energy. *Advances in Neural Information Processing Systems* 6:3–10.

Hinton, G.; Deng, L.; Yu, D.; Dahl, G. E.; Mohamed, A.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; and Sainath, T. N. 2012. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal Processing Magazine* 29(6):82–97.

Ho, J. C.; Ghosh, J.; and Sun, J. 2014. Marble: High-throughput phenotyping from electronic health records via sparse nonnegative tensor factorization.

Jos, A.; eacute;Luis M.; N; eacute;Meth Bal; aacute;zs; Jean-Claude, G.; Peeter, B.; Aiko, A.; and Roos, N. E. 2012. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. *Chemistry A European Journal* 18(13):3981–3991.

Kolda, T. G., and Gibson, T. 2006. Multilinear operators for higher-order decompositions.

Kolda, T. G. 2015. Numerical optimization for symmetric tensor decomposition. *Mathematical Programming* 151(1):225–248.

Macskassy, S. A.; Perlich, C.; Leskovec, J.; Wang, W.; and Ghani, R., eds. 2014. *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*. ACM.

Phan, A. H., and Cichocki, A. 2010. Tensor decompositions for feature extraction and classification of high dimensional datasets. *Nonlinear Theory and Its Applications* 1(1):37–68.

Rifai, S.; Vincent, P.; Muller, X.; Glorot, X.; and Bengio, Y. 2011. Contractive auto-encoders: Explicit invariance during feature extraction. In *ICML*.

Roweis, S. 1999. Em algorithms for pca and sensible pca. In *California Institute of Technology, Computation and Neural Systems*.

Wang, F.; Liu, C.; Wang, Y.; Hu, J.; and Yu, G. 2015a. A graph based methodology for temporal signature identification from her. *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium* 2015.

Wang, Y.; Chen, R.; Ghosh, J.; Denny, J. C.; Kho, A.; Chen, Y.; Malin, B. A.; and Sun, J. 2015b. Rubik: Knowledge guided tensor factorization and completion for health data analytics. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1265–1274.

Welling, M., and Weber, M. 2001. Positive tensor factorization. *Pattern Recognition Letters* 22(12):1255–1261.