Deep Patient Similarity Learning for Personalized Healthcare

Qiuling Suo, Fenglong Ma, Ye Yuan, Mengdi Huai, Weida Zhong, Jing Gao, *Member, IEEE*, and Aidong Zhang, *Fellow, IEEE*

***Abstract*— Predicting patients' risk of developing certain diseases is an important research topic in healthcare. Accurately identifying and ranking the similarity among patients based on their historical records is a key step in personalized healthcare. The electric health records (EHRs), which are irregularly sampled and have varied patient visit lengths, cannot be directly used to measure patient similarity due to the lack of an appropriate representation. Moreover, there needs an effective approach to measure patient similarity on EHRs. In this paper, we propose two novel deep similarity learning frameworks which simultaneously learn patient representations and measure pairwise similarity. We use a convolutional neural network (CNN) to capture local important information in EHRs and then feed the learned representation into triplet loss or softmax cross entropy loss. After training, we can obtain pairwise distances and similarity scores. Utilizing the similarity information, we then perform disease predictions and patient clustering. Experimental results show that CNN can better represent the longitudinal EHR sequences, and our proposed frameworks outperform state-of-the-art distance metric learning methods.**

***Index Terms*— Patient similarity, convolutional neural network, personalized healthcare.**

I. Introduction

PATIENT similarity learning is a fundamental and important task in healthcare domain, which helps to improve clinical decision making without incurring additional efforts from physicians. The goal of patient similarity is to learn a clinical meaningful metric which measures the relative similarities between patient pairs according to their health records. A proper similarity measure enables various downstream applications, such as personalized medicine [1], [2], medical diagnoses [3], trajectory analysis [4] and cohort study [5].

The prevalence and growing volume of electronic health records (EHRs) provides unprecedented opportunities to improve clinical decision support. The EHR data, which is a longitudinal electronic record of patient health information, is a valuable source for predictive modeling which can assist clinical and medical research. The EHRs are temporally sequenced by patient visits with each visit represented as a set of high dimensional clinical events (i.e. medical codes). Mining EHRs is especially challenging compared to standard data mining tasks, due to its noisy, irregular and heterogeneous nature.

Personalized healthcare has obtained increasing interest from researchers [2], [5]–[8]. A general framework for personalized prediction contains two stages: measuring the similarity among patients and grouping patients into cohorts, and analyzing the cohort to perform disease diagnosis, therapy prescription, etc. This framework is motived by the working process of human doctors, i.e., after reviewing or recalling the diagnosed patients with similar diseases or symptoms, the doctors then carefully make decision. If doctors can find similar patients, the probability of successfully curing this patient may be improved a lot. Therefore, how to accurately and precisely measure patient similarity is an important and challenging issue.

Many similarity learning methods have been proposed [3], [9]–[13] on healthcare datasets. However, these models are developed for handcrafted vector representations such as demographics or average numerical values, without considering the temporal information from different visits. For the longitudinal EHR data, the number of patient visits varies largely, due to patients' irregular visits and incomplete recordings. The aforementioned learning metrics cannot be directly applied to the longitudinal data, since the historical records of each patient do not naturally form a comparable vector. Therefore, one of the key challenges in measuring patient similarity is to derive an effective representation for each patient without loss of his/her historical information. A traditional vector based representation is to summarize data statistics (e.g., sum, average, max, etc) of corresponding events within a time period, and calculate similarity distance on top of those patient vectors. However, this removes temporal relations across adjacent visits.

Recently, deep learning approaches have been widely adopted and rapidly developed in patient representation learning [14]–[21] such as autoencoder, recurrent neural networks (RNNs) and convolution neural networks (CNNs). In this paper, we propose two deep metric learning frameworks on EHR to measure patient similarity. There are two parts in the model: representation learning and similarity learning.

Manuscript received May 10, 2018; accepted May 13, 2018. Date of publication May 16, 2018; date of current version July 31, 2018. This work was supported in part by the U.S. National Science Foundation under Grant NSF IIS-1218393 and Grant IIS-1514204. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. *(Corresponding author: Qiuling Suo.)*

Q. Suo, F. Ma, M. Huai, W. Zhong, J. Gao, and A. Zhang are with the Department of Computer Science, State University of New York at Buffalo, Buffalo, NY 14260 USA (e-mail: qiulings@buffalo.edu; fenglong@buffalo.edu; mengdihu@buffalo.edu; weidazho@buffalo.edu; jing@buffalo.edu; azhang@buffalo.edu).

Y. Yuan is with the College of Information and Communication Engineering, Beijing University of Technology, Beijing 100022, China (e-mail: yuanye91@emails.bjut.edu.cn).

Digital Object Identifier 10.1109/TNB.2018.2837622

In representation learning, we utilize the ability of CNN on representing longitudinal data, and obtain a vector representation which contains the local important information of the original data. In similarity learning, we use two ways to learn the similarity between patient pairs. The first is based on triplet loss function, which learns a margin to separate the distance of negative and positive samples. Therefore, we can get a distance value indicating the relative similarity of two patients. The second is to perform classification on the learned representations with positive label for similar pairs and negative label for dissimilar pairs. Since the similarity probability between a pair of patients indicates the risk level of the two patients developing the same disease, we use it as the score to rank the similarity among patients. After obtaining the similarity information, we perform two tasks: disease prediction and patient clustering which are application areas of personalized healthcare, in order to validate the learned metrics. In summary, our contributions are as follows:

- We propose two end-to-end frameworks to jointly learn patient EHR representations and pairwise similarity, without the handcrafted feature aggregations. With the framework, parameters of representation and similarity learning can be optimized simultaneously, yielding higher accuracy.
- In our proposed frameworks, CNN makes use of sequential structure and learns local important information, triplet loss ensures large margin to separate the samples in the same class and samples in different classes, and softmax cross entropy loss ensures pairwise labels to be correctly classified.
- Our experimental results show that our similarity learning framework can learn better representation vectors for patients' historical information and improve the similarity learning accuracy compared to other state-of-the-art baseline models.

II. Related Work

In this section, we review some related works on evaluating patient similarity and building personalized models.

A. Similarity Learning

For a new patient, identifying historical records of patients who are similar to him/her could help retrieve similar reference cases for predicting the clinical outcomes of this new patient. Reference [1] combined patient similarity and drug similarity analysis and proposed a heterogeneous label propagation method to identify which drug is likely to be effective for a given patient. In practice, different physicians have different understandings of patient similarity based on the specifics of the cases. Using physician feedback as the supervision, [9] presented a locally supervised metric learning (LSML) algorithm that learns a generalized Mahalanobis distance. Given that obtaining physicians' input is difficult and expensive in reality, Wang and Sun [22] proposed a weakly supervised patient similarity learning method which only uses a small amount of supervision information provided by the physicians. Due to the fact that patient similarity is highly context sensitive, Wang *et al.* [23] used both statistical and wavelet based features to capture the characteristics of patients, and then presented a patient similarity learning method that leverages localized supervised metric learning. Considering the high dimensionality and redundancy of medical data, Zhan *et al.* [10] proposed to perform feature selection and patient similarity learning at the same time.

B. Personalized Healthcare

Recently, personalized prediction in healthcare applications obtains increasing interest from researchers. It aims to find out the unique characteristics for individual patients, and perform targeted, patient specific predictions, recommendations and treatments [24], [25]. Most of the works perform personalized prediction by matching clinical similar patients. Reference [6] performed a comparative study of global, local, and personalized modeling, and found that personalized models can achieve better performance across different bioinformatics classification tasks. Reference [26] used a locally supervised metric learning for similarity measurement and logistic regression as the predictive model for diabetes onset prediction. Reference [2] used cosine distance to obtain patient's distance and built classifiers for mortality prediction. Reference [7] proposes to learn base models of the population and personalized model of each patient via a sparse multi-task learning method.

The aforementioned methods require the input of each patient as a vector. A traditional way is to manually obtain feature vectors by using the static information of patients such as demographic, and data statistics (e.g. sum, average, etc) within a certain time range, as the patient representation. However, these handcrafted feature vectors completely ignore the temporal relations across visit sequences. To account for the temporal information, [12] used a dynamic programming algorithm to find optimal local alignments of patient sequences; [27] developed two solutions for patient similarity learning, unsupervised and supervised, using a CNN-based similarity matching framework; and [5] developed a 2D-RNN for dynamic temporal matching of patient sequences to obtain the patient similarity ranking.

III. Method

In this section, we give the details of our proposed metric learning models on healthcare dataset. We first show how to learn an effective representation for the longitudinal EHR data, and then introduce two methods to measure the similarity between patient pairs. With the learned similarity information, we then perform two tasks for personalized healthcare: disease prediction and patient clustering.

A. Representation Learning

1) Basic Notations: A patient's health record contains a sequence of visit information, and in each visit, medical codes are recorded indicating the disease or treatment the patient suffered or received. The codes can be mapped to the International Classification of Disease (ICD-9).[1] We denote all the unique medical codes from the EHR data as $c_1, c_2, \ldots, c_{|\mathcal{C}|} \in \mathcal{C}$, where $|\mathcal{C}|$ is the number of unique medical codes. Assuming there are N patients, the n-th patient has a number of visits T_n. A patient $\boldsymbol{p}_n$ can be represented by a sequence of visits

[1] https://en.wikipedia.org/wiki/List of ICD-9 codes

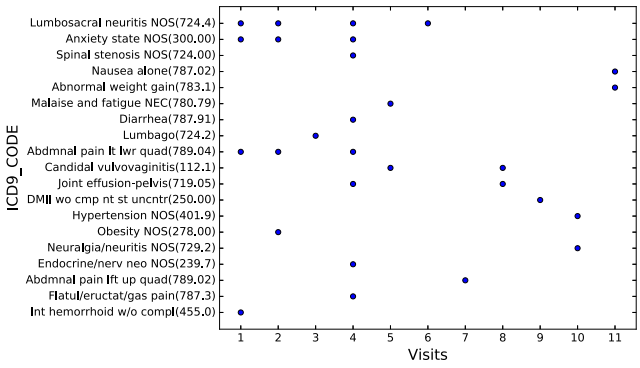

Fig. 1. An example of the data for one patient. *X*-axis is the visit sequences, and *y*-axis is the medical events with corresponding ICD9 codes. The blue dots indicate the patient has the events in certain visits.

denoted as $V_1, V_2, \ldots, V_{T_n}$. Each visit V_i is denoted by a high dimensional binary vector $\boldsymbol{v}_i \in \{0,1\}^{|\mathcal{C}|}$, indicating whether V_i has the code c_j or not.

Fig. 1 shows an example of the EHR data, which can be viewed as a matrix. The horizontal axis corresponds to visits, and the vertical axis is medical events (ICD9 codes). The (i, j)-th entry is 1 if code c_j is observed at time stamp V_i for the corresponding patient, otherwise 0. Since the number of visits of different patients varies, we pad zero to the visit dimension, making each patient have a fixed length of visits $t = \max\{T_i\}_{i=1}^{N}$, for the sake of CNN operations.

2) Visit Embedding: The original one-hot representation stated in Section III-A.1 ignores code relations, and makes the EHR matrix high dimensional and sparse. To reduce feature dimensions and learn relationships among codes, we use a fully connected network layer to embed each code into a vector space. As a result, each visit $\boldsymbol{v}_i$ is mapped into a vector $\boldsymbol{x}_i \in \mathbb{R}^d$ using the formula:

$$\boldsymbol{x_i} = \text{ReLU}(\boldsymbol{W}_v \boldsymbol{v}_i + \boldsymbol{b}_v), \tag{1}$$

where $d < |\mathcal{C}|$ is the embedding dimension, $\boldsymbol{W}_v \in \mathbb{R}^{d \times |\mathcal{C}|}$ and $\boldsymbol{b}_v \in \mathbb{R}^d$ is the weight matrix and bias vector to be learned, and the activation function ReLU is defined as $\text{ReLU}(\boldsymbol{x}) = \max(\boldsymbol{x}, \boldsymbol{0})$. The adoption of ReLU ensures non-negative representation, which enables the learned vector to be interpretable [28]. After the embedding operation, we can obtain an embedding matrix $\boldsymbol{X} \in \mathbb{R}^{t \times d}$ for each patient with lower feature dimension compared to the original one-hot matrix.

3) Convolutional Neural Network: A patient embedding matrix can be viewed as a 2D pixel matrix of an image. A convolution operation can be applied to capture the sequential relation across adjacent visits. However, different from images with spatial relations across pixels in two dimensions, the positions of medical codes have no spatial/temporal meaning, which makes the convolution operation [29] across feature dimension unreasonable. Therefore, a one-side convolution operation across the time dimension is applied to capture the sequential relation across adjacent visits instead of using a standard 2D CNN.

The convolutional layer has p different filter sizes and the number of filters per size is q, so that the total number of filters is $m = pq$. Each filter is defined as a matrix $\boldsymbol{w}_c \in \mathbb{R}^{h \times d}$, where h is a window size of visit length, meaning that the convolution operation is applied over h sequential timestamps. Suppose a filter is applied over a concatenation from visit vector $\boldsymbol{x}_i$ to $\boldsymbol{x}_{i+h-1}$, a scalar value c_i can be generated using the formula:

$$c_i = \text{ReLU}(\boldsymbol{W}_c \cdot \boldsymbol{x}_{i:i+h-1} + b_c), \tag{2}$$

where $b_c \in \mathbb{R}$ is a bias term, and $\cdot$ is the convolution operation. This filter is applied to each possible window of timestamps $\{\boldsymbol{x}_{1:h}, \boldsymbol{x}_{2:h+1}, \ldots, \boldsymbol{x}_{t-h+1:t}\}$ with a stride equal to 1, to produce a feature map $\boldsymbol{c} = \{c_1, c_2, \ldots, c_{t-h+1}\}$, where $\boldsymbol{c} \in \mathbb{R}^{t-h+1}$. Since we have totally m filters, we can obtain m feature maps.

The outputs from the convolutional layer are then passed into the pooling layer. A max pooling is applied over $\boldsymbol{c}$ as $\hat{c} = \max\{c\}$, where $\hat{c}$ is the maximum value corresponding to a particular filter. The key idea here is to capture the most important information for each feature map. It can naturally deal with variable visit lengths, since the padded visits have no contribution to the pooled outputs.

The pooled outputs from all the filters are concatenated to form a vector representation $\boldsymbol{h} \in \mathbb{R}^m$. The learned vector $\boldsymbol{h}$ is the vector representation of the original embedding matrix $\boldsymbol{X}$. It contains not only visit information of the patient, but also the relationship across adjacent time points.

B. Similarity Learning

Learning the relative similarity/distance among each pair of patients is the key step for personalized healthcare. We propose two methods to measure the similarity among patient vectors learned from Section III-A, softmax based framework and triplet loss based framework.

1) Predictive Similarity Learning: The similarity between a pair of vectors can be measured by a bilinear distance: $S = \boldsymbol{h}_i \boldsymbol{M} \boldsymbol{h}_j$, where the matching matrix $\boldsymbol{M} \in \mathbb{R}^{m \times m}$ is symmetric for the reason of practical meaning. To ensure the symmetric constraint of $\boldsymbol{M}$, it is decomposed as $\boldsymbol{M} = \boldsymbol{L}^T \boldsymbol{L}$, where $\boldsymbol{L} \in \mathbb{R}^{l \times m}$ with $l < m$ to ensure a low rank characteristic.

We consider a symmetric constraint for vector concatenation and convert patient vectors to get a similarity vector, as to ensure that the order of patients has no effect on the similarity score. We first convert $\boldsymbol{h}_i$ and $\boldsymbol{h}_j$ into a single vector with their dimension holds using the formula:

$$\boldsymbol{H} = \boldsymbol{W}_h \boldsymbol{h}_i \oplus \boldsymbol{W}_h \boldsymbol{h}_j, \tag{3}$$

where $\boldsymbol{W}_h \in \mathbb{R}^{m \times m}$ and $\oplus$ is a bitwise addition. After that, $\boldsymbol{H}$ and S are concatenated and then fed into a fully connected softmax layer, to get an output probability $\hat{y}$ which is a float value between 0 and 1. Here we set the ground truth y as 1 if two patients has the risk of developing the same disease, otherwise 0. We use cross-entropy between y and $\hat{y}$ to calculate the loss for patient pairs:

$$\mathcal{L} = \frac{1}{\tilde{N}} \sum_{k=1}^{\tilde{N}} (y_k \log(\hat{y_k}) + (1 - y_k) \log(1 - \hat{y}_k)), \tag{4}$$

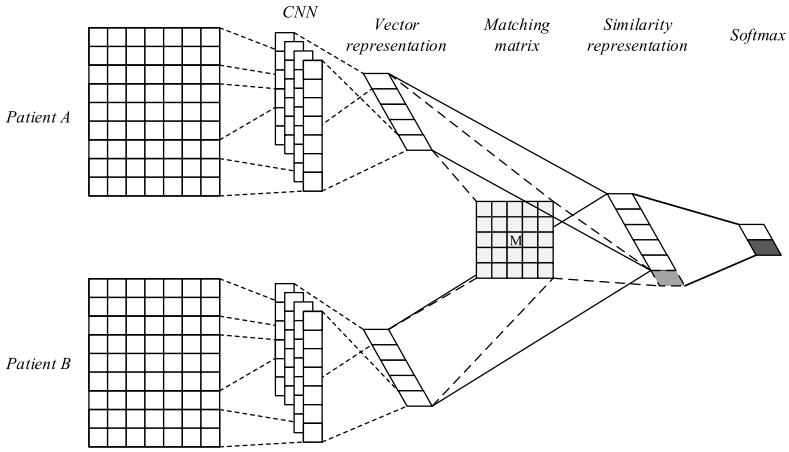

Fig. 2. The overall framework of pairwise patient similarity learning. The one-hot EHR matrix of patient *A* is first mapped into an embedding matrix with a lower feature dimension, and then feed into CNN to obtain a vector representation. Patient *B* shares the same embedding and CNN parameters. The patient representations then pass through a matching matrix *M* and a converting layer to get the similarity representation vector. Softmax layer is added after the similarity vector to utilize the label and update all the parameters.

where $\tilde{N}$ is the total number of patient pairs. Since there are N patients, $\tilde{N}$ would be $N(N-1)/2$. The probabilistic output $\hat{y}$ indicats the similarity degree between two patients. The higher value of $\hat{y}$ means the higher probability that $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ belonging to the same class, or, the two patients have smaller distance and are more similar to each other. The overall framework is shown in Fig. 2. The model can be trained end-to-end and all the parameters are updated simultaneously.

2) Triplet-Loss Metric Learning: Metric learning aims to learn a proper distance metric for a particular task, which is crucial to the performance of many algorithms. We utilize the idea of metric learning to learn the relative distance of patients.

In traditional metric learning, a linear transformation $\boldsymbol{L}$ is used to map the raw data into a new space. The new metric in the space can better measure the relative distance of input instances. The distance between instances x_i and x_j can be obtained by calculating the Euclidean distance in the new space, as shown in the formula,

$$d^2(\boldsymbol{x}_i, \boldsymbol{x}_j) = ||\boldsymbol{x}_i - \boldsymbol{x}_j||_L^2 = ||\boldsymbol{L}(\boldsymbol{x}_i - \boldsymbol{x}_j)||^2, \quad (5)$$

where $\boldsymbol{L}$ is the transformation matrix to be learned.

In deep metric learning, the linear transformation $\boldsymbol{L}$ is replaced by a neural network f to learn the complex nonlinear relations among raw features. In our problem of patient similarity learning, this nonlinear transformation is learned through the CNN operation described in Section III-A. Therefore, the distance between two patients $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ in the transformed space can be written as

$$d^2(\boldsymbol{p}_i, \boldsymbol{p}_j) = ||f(\boldsymbol{p}_i) - f(\boldsymbol{p}_j)||^2 = ||\boldsymbol{h}_i - \boldsymbol{h}_j||^2, \quad (6)$$

where f is the CNN operation in our problem setting, and $\boldsymbol{h}_i$ and $\boldsymbol{h}_j$ are the vector representations learned via the process described in Section III-A.

We use triplet loss [30] as the objective function. This contains a set of triplets, where each triplet has an anchor, a positive and a negative example. A positive sample has the same class label as the anchor, while the negative sample has the different class label. During the training, the positive should be moved closer to the anchor and the negative should be pushed far away, i.e. the distance between anchor $\boldsymbol{p}_i$ and positive sample $\boldsymbol{p}_i^+$ should be closer than the distance between $\boldsymbol{p}_i$ and negative sample $\boldsymbol{p}_i^-$ with some fixed margin. Therefore, the triplet loss can be written as,

$$\mathcal{L} = \frac{1}{\mathcal{T}} \sum_{(\boldsymbol{p}_i, \boldsymbol{p}_i^+, \boldsymbol{p}_i^-) \in \mathcal{T}} [d^2(\boldsymbol{p}_i, \boldsymbol{p}_i^+) + g - d^2(\boldsymbol{p}_i, \boldsymbol{p}_i^-)]_+, \quad (7)$$

where $\mathcal{T}$ is a set of triplets, the operator $[\cdot]_+ = max(\cdot, 0)$ denotes the hinge function which takes the positive components, and g denotes a predefined margin which is a constant.

This metric learning layer is added on top of CNN, which takes the learned vector representation as the input to calculate distance between patients. The objective function Eq. 7 is minimized through back propagation, and all the parameters are updated simultaneously. The learned distance metric indicates the similarity between patient pairs, with smaller distance values for higher similarity. The framework of triplet-loss based deep similarity learning is shown in Fig. 3, which is also an end-to-end learning framework.

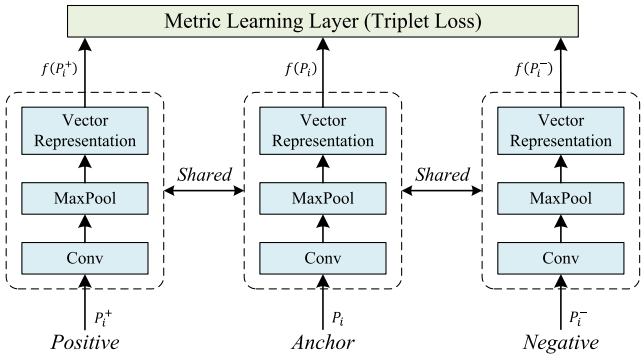

Fig. 3. The overall framework of triplet patient similarity learning. The one-hot EHR matrix of patient $\boldsymbol{p}_i$ is mapped into an embedding matrix, and then feed into CNN to obtain a vector representation. $\boldsymbol{p}_i^+$ and $\boldsymbol{p}_i^-$ share the same parameters as $\boldsymbol{p}_i$. Pairwise distances are then calculated based on the vector representations, and triplet loss is used to update all the parameters.

C. Personalized Healthcare

The learned similarity can be used for personalized prediction. The similarity score from Section III-B can be used to measure the similarity degree between a pair of patients. For each test patient, we first calculate the distance between him/her and each of the training patients, and then rank the training patients according to the distance values in an ascending order.

We use K Nearest Neighbor (KNN) classifier to predict patients' risk of developing certain diseases in the future. For each test patient, we select the closest k patients from the training set with the top$k$ smallest distance, and then use the most common class label appearing among the k training samples as the predicted label. Intuitively, since the patients have similar health records/symptoms, it is highly possible that they have the risk of developing the same disease.

The classification task can assign labels to samples, but it does not give the information of how close or how far the distances are. Therefore, to better evaluate the learned distance metrics/similarities, we also perform clustering method on the mapped spaces. This way can give doctors an intuitive visualization of the distance distribution of patients.

IV. Experiments

In this section, we evaluate our model on a real world EHR dataset, compare its performance with other state-of-the-art prediction models, and show that it yields better performance.

A. Data Description

We conduct experiments on a real world dataset, which consists of medical claims from more than 100,000 patients over two years. Each patient has a longitudinal visit sequence, represented by a set of high dimensional clinical events (i.e. ICD-9 codes). To perform disease prediction, we extract three patient cohorts from the dataset: diabetes, obesity, and chronic obstructive pulmonary disease (COPD). Following the disease selection criteria in [31], we identify the diseased patients who have 1) qualifying ICD-9 codes for a specific disease in the encounter records or medication orders, and 2) at least three clinical encounters with qualifying ICD-9 codes occur within 12 months. The date at which the first target diagnosis appears is denoted as the index date. We split the patient sequences at the index date into two parts, and use only the part before the index date which contains early symptoms and complications for similarity learning and disease prediction. To enable distinct cohorts, we remove overlapped patients so that each patient only suffers from one disease. Moreover, we remove the clinical events which appear more than 90% of patients or less than five patients to avoid biases and noise. Finally, there are totally 9,528 patients and 3,852 distinct codes, and the maximum visit length is cut to be 150. The statistics of the dataset is summarized in Table I. The experimental setting on dataset is shown in Fig. 4.

TABLE I
Statistics of Dataset

Cohorts	Diabetes	Obesity	COPD
# Patients	3,214	3,441	2,873
# unique codes in cohort	3,455	3,585	3,260
# unique codes per person	34.48	24.11	28.30
Total # events	160,920	217,583	136,886
Avg.# of visits	22.52	30.34	21.14
Avg.# event per patient	50.07	63.23	19.08

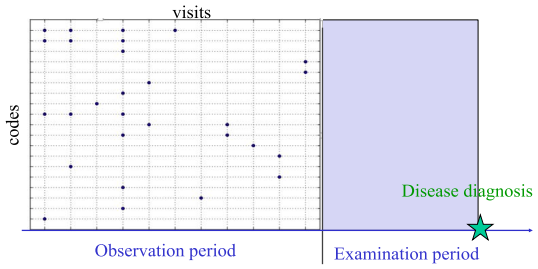

Fig. 4. Experimental setting for patients' risk evaluation.

B. Experimental Setup

Here we give some details of the model implementation, and the baseline approaches to compare with.

1) Model Implementation: We first train the two similarity models described in Section III-B to obtain the optimized parameters of CNNs and matching metric. Then, using the frameworks, we calculate and rank the distance/similarity of each testing instance and all the training data. After obtaining the similarity information, we perform two tasks: disease prediction and patient clustering.

The dataset is randomly divided into training, validation and testing sets in a 0.75:0.1:0.15 ratio. For the similarity training process, the ground truth is binary, as two patients having the same disease are considered as a positive sample pair while having different diseases are a negative sample pair. The prediction process is a multi-class classification problem corresponding to the three diseases.

The similarity learning frameworks are implemented with Tensorflow [32]. Adam [33] is used to optimize model parameters. Different from a normal CNN model with the input to be a mini-batch of patients, the similarity framework is trained on a batch of patient pairs to ensure that each of the patient pairs can be measured. With regard to the overfitting issue, we use the L-2 regularization and dropout strategy with a dropout rate 0.5.

2) Baseline Approaches: To validate the performance of proposed deep patient similarity approaches, we compare them with the following state-of-the-art baseline methods.

- Basic metrics. **Euclidean** and **Cosine** distances on raw inputs are calculated to measure the similarity between sample pairs. The two methods directly measure similarity on the original input space, without any mapping parameter to be learned.
- Distance metric learning methods. **LMNN** [34] is a classical metric learning method, which pulls the k-nearest neighbors belonging to the same class closer, and separates examples from different classes by a large margin. **ITML** [35] learns the Mahalanobis distance by minimizing the differential relative entropy under the pairwise constraints between two

multivariate Gaussians. **GMML** [36] formulates the learning process as an unconstrained smooth and convex optimization problem. **SCML** [45] leans a sparse combination of locally discriminative metrics, which regards the Mahalanobis matrix as a nonnegative weighted sum of k low-dimensional basis.

Since the historical health data for each patient forms a 2D matrix, the above metric learning approaches cannot be directly used on the raw data. Therefore, we use an aggregated vector representation: we count the number of medical codes for each patient based on all his/her visits, so that each element in the vector indicates the frequency of a corresponding code. The aggregated vector indicates the information of code frequency which is important for disease prediction, and has been adopted [5], [37] as a way to represent EHR data.

Some other metric learning methods such as NCA [38], MLKR [39] and R2ML [40] cannot scale well and are not applicable for large datasets, so they are not used as baselines under our problem setting.

3) Evaluation Measures: We perform two tasks using the learned distance metrics: disease prediction and patient clustering.

• **Disease prediction measure**. To evaluation the performance of all the patient similarity learning approaches on disease prediction, we calculate accuracy, precision, recall and F1 score as the measures. Since we perform multi-class classification, the measurements for binary classification cannot be directly used. Therefore, we use macro-averaging [41] to evaluate how the algorithms work overall across the sets of data. The measures are calculated as following,

$$\text{Accuracy} = \frac{1}{l}\sum_{i=1}^{l}\frac{tp_i + tn_i}{tp_i + fn_i + fp_i + tn_i},$$

$$\text{Precision} = \frac{1}{l}\sum_{i=1}^{l}\frac{tp_i}{tp_i + fp_i},$$

$$\text{Recall} = \frac{1}{l}\sum_{i=1}^{l}\frac{tp_i}{tp_i + fn_i},$$

$$\text{F1-score} = 2\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}},$$

where l is the total number of classes, and tp_i, tn_i, fp_i and fn_i are true positive, true negative, false positive and false negative for class i respectively.

• **Patient clustering measure**. We evaluate the clustering results via three widely used criteria, *Rand index* (RI), *purity*, and *normalized mutual information* (NMI).

RI, which measures the percentage of correct decisions, can be calculated via:

$$\text{RI} = \frac{a+b}{\binom{n}{2}},$$

where a is the number of pairs belonging to the same cohort who are grouped into one single cluster, b is the number of pairs coming from different cohorts that are grouped into distinct clusters, and n is the total number of patients. Usually, the higher the RI, the better the clustering result is.

TABLE II
MACRO-AVERAGING MEASURES OF DISEASE PREDICTION PERFORMANCE BASED ON THE DIFFERENT SIMILARITY LEARNING METHODS

Method	Accuracy	Recall	Precision	F1 score
Euclidean	0.5660	0.5490	0.6199	0.5347
Cosine	0.5981	0.5920	0.6032	0.5914
GMML	0.5877	0.5801	0.6062	0.5812
ITML	0.5751	0.5591	0.6202	0.5476
LMNN	0.6848	0.6789	0.6925	0.6808
SCML	0.7093	0.7062	0.7085	0.7064
CNN_triplet	0.7736	0.7731	0.7740	0.7730
CNN_softmax	**0.8442**	**0.8410**	**0.8519**	**0.8438**

Purity can be computed as below:

$$\text{Purity}(Cluster, Cohort) = \frac{1}{n}\sum_{i}\max_{j}|p_i \cap q_j|,$$

where $Cluster = \{p_1, p_2, \cdots, p_I\}$ is the set of clusters, and $Cohort = \{q_i, q_2, \cdots, q_J\}$ is the group of classes or cohorts in our case. The upper bound of *Purity* is 1, which indicates perfect match between the partitions.

NMI measures the information shared by two clusterings,

$$\text{NMI}(Cluster, Cohort) = \frac{I(Cluster, Cohort)}{[H(Cluster) + H(Cohort)]/2},$$

where I is mutual information between two random variables, and H is the information entropy of the given random variable. The value of *NMI* also can vary between 0 and 1. Here, it achieves its maximum value of 1 when grouping clusterings are the same to the real cohorts.

C. Experimental Results

We compare our proposed patient similarity frameworks with other state-of-the-art metric learning methods, and show that our proposed methods can significantly outperform the baseline methods. We denote the proposed framework in Section III-B.1 as CNN_softmax, and the framework in Section III-B.2 as CNN_triplet.

1) Disease Prediction Results: We first compare the performance on the task of disease prediction. We train the different metric learning models to learn the relative distance/similarity degree among patients, and then use KNN (k is set to 5) to perform classification based on the learned distance metrics. The results of macro accuracy, recall, precision and F1 score are shown in Table II.

In the table, Euclidean, cosine cannot perform well. This is because the three methods measure similarity on the original space which is high-dimensional, sparse and noisy. Distance metric learning methods make use of the similarity labels to optimize the mapping parameters, and can achieve better results compared to the basic metrics. Among traditional metric learning methods, SCML performs better than LMNN and ITML. This owes to its ability of sparse feature selection, which can deal with the sparsity characteristic of the EHR data.

TABLE III
CONFUSION MATRIX OF VARIOUS SIMILARITY LEARNING METHODS

(a) Euclidean

Predict / Truth	Diabetes	Obesity	COPD
Diabetes	250	211	22
Obesity	56	440	21
COPD	103	208	120
Accu.	.6112	.5122	.7362

(b) Cosine

Predict / Truth	Diabetes	Obesity	COPD
Diabetes	326	101	56
Obesity	122	333	62
COPD	136	98	197
Accu.	.5582	.6259	.6254

(c) PCA

Predict / Truth	Diabetes	Obesity	COPD
Diabetes	230	226	27
Obesity	104	386	27
COPD	127	239	65
Accu.	.4989	.4536	.5462

(d) LMNN

Predict / Truth	Diabetes	Obesity	COPD
Diabetes	333	106	44
Obesity	79	399	39
COPD	112	71	248
Accu.	.6355	.6927	.7492

(e) ITML

Predict / Truth	Diabetes	Obesity	COPD
Diabetes	262	197	24
Obesity	60	432	25
COPD	106	196	129
Accu.	.6121	.5236	.7247

(f) SCML

Predict / Truth	Diabetes	Obesity	COPD
Diabetes	321	95	67
Obesity	54	405	58
COPD	76	66	289
Accu.	.7118	.7155	.6981

(g) GMML

Predict / Truth	Diabetes	Obesity	COPD
Diabetes	299	147	37
Obesity	118	353	46
COPD	127	115	189
Accu.	.5496	.5740	.6949

(h) CNN_triplet

Predict / Truth	Diabetes	Obesity	COPD
Diabetes	372	54	57
Obesity	65	405	47
COPD	75	26	330
Accu.	.7266	.8351	.7604

(i) CNN_softmax

Predict / Truth	Diabetes	Obesity	COPD
Diabetes	419	48	16
Obesity	49	454	14
COPD	66	30	335
Accu.	.7846	.8534	.9178

Although these methods try to learn a proper transformation to obtain a new space with various constraints, they are not able to learn the sequential and contextual information existed in the longitudinal EHR. Moreover, they are not suitable for large amount of data. Our proposed similarity learning methods based on CNN significantly improve the disease prediction performance, as CNN captures the local important information across consecutive visits. In our CNN learning, the one-side convolution operation learns the local information across consecutive visits, and max-pooling operation enables the most important information to be captured while reduces the noisy information from EHR. Therefore, the learned vector representation based on CNN contains important local information for the prediction task. Among the two methods, CNN_triplet learns the margin between positive and negative pairs, while CNN_softmax utilizes the cross-entropy loss to classify pair labels. CNN_triplet makes the positive samples closer to anchor while pushes the negative samples by a fixed margin. CNN_softmax fully utilizes the bilinear similarity and patient vector representations to perform supervised classification, which can further improve the accuracy.

The confusion matrix of prediction results based on various similarity learning methods are shown in Table III. The accuracy indicates the true positive rate for each disease prediction. We can see that our proposed methods can better distinct the three diseases. In fact, the three diseases do display several relationships with each other [42]–[44], and share some come symptoms and complications, especially diabetes and obesity, making them hard to be discriminated. Compared with other baselines, our method can better identify three disease cohorts, especially diabetes and COPD cohorts. Having more detailed sub-group information may help to better discriminate the heterogeneous nature of EHRs.

2) Patient Clustering Results: Risk prediction can help the medical decision on identifying symptoms for early diagnosis, while patient clustering can help to analyze disease cohort distributions. The clustering performance based on learned distances/similarities is shown in Table IV. The choice of various clustering algorithms should not affect the relative performance comparison, and we adopt k-means here with k=3. GMML, LMNN, ITML and CNN_triplet map original data to a new space and then calculate the Euclidean distance between pairs, so that we can perform clustering on the learned new space. Cosine, SCML and CNN_softmax measure pairwise similarity information, but do not obtain the mapped coordinates of samples. Therefore, we do not perform clustering on these methods.

In Table IV, we use rand index, purity and NMI to measure the performance of clustering algorithm. The higher values mean more coherence between clustered groups and true labels, i.e., more similar samples are grouped together, indicating better clustering performance. we can see that CNN_triplet significantly outperforms baseline methods, which means that CNN_triplet can learn an appropriate distance metric which can be used to cluster similar patients to the same cohort.

We visualize the transformed testing samples for different metric learning methods in Fig. 5. The samples of Euclidean

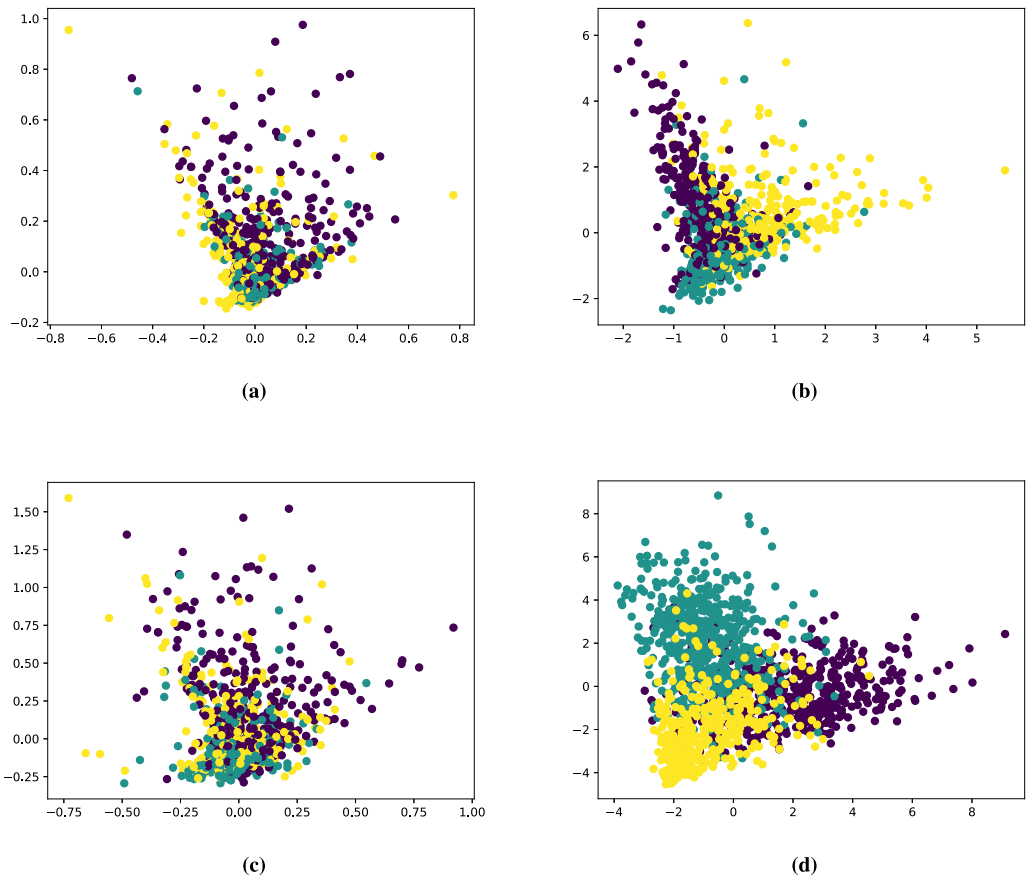

Fig. 5. Visualization of testing samples in the transformed space. (a) Euclidean. (b) LMNN. (c) ITML. (d) CNN_triplet.

TABLE IV
PATIENT CLUSTERING PERFORMANCE BASED ON THE LEARNED DISTANCES USING DISTANCE METRIC LEARNING METHODS

Method	Rand index	Purity	NMI
Euclidean	0.4743	0.4633	0.0593
GMML	0.4862	0.4654	0.0582
LMNN	0.5778	0.5374	0.1148
ITML	0.5024	0.4822	0.0698
CNN_triplet	0.7351	0.7561	0.3599

and ITML cannot be well separated, LMNN can learn relatively better clusters, and our proposed CNN_triplet is able to better distinct the three disease cohorts. This observation matches the performance results in Table II and Table IV. Other methods cannot be visualized because they only get the relative similarity values for sample pairs, but do not learn the transformation matrix for sample itself, which means that a sample has no absolute location in the new space. This visualization results can be used to further study of disease cohort distributions.

V. CONCLUSION

Patient similarity learning aims to find appropriate distance metrics to measure patient pairs for a specific task. To capture the historical information of patient' record, a proper way to represent longitudinal EHR is necessary. Moreover, we need a way to learn the similarity degree or distance between each pair of patients. In this paper, we propose two patient similarity learning frameworks on EHR dataset. The raw EHRs are feed into a CNN model which captures the consecutive sequential information to learn a vector representation. Then soft-max based supervised classification method and triplet loss based distance metric learning method are used to learn the similarity of patient pairs. Experimental results on disease prediction and patient clustering show that CNN can better represent the longitudinal EHR sequences, and our end-to-end similarity frameworks outperform state-of-the-art distance metric learning methods.

REFERENCES

[1] P. Zhang, F. Wang, J. Hu, and R. Sorrentino, "Towards personalized medicine: Leveraging patient similarity and drug similarity analytics," in *Proc. AMIA Summits Transl. Sci.*, 2014, pp. 132–136.
[2] J. Lee, D. M. Maslove, and J. A. Dubin, "Personalized mortality prediction driven by electronic medical data and a patient similarity metric," *PLoS ONE*, vol. 10, no. 5, p. 0127428, 2015.
[3] A. Gottlieb, G. Y. Stein, E. Ruppin, R. B. Altman, and R. Sharan, "A method for inferring medical diagnoses from patient similarities," *BMC Med.*, vol. 11, no. 1, p. 194, 2013.
[4] S. Ebadollahi, J. Sun, D. Gotz, J. Hu, D. Sow, and C. Neti, "Predicting patient's trajectory of physiological data using temporal trends in similar patients: A system for near-term prognostics," in *Proc. AMIA Annu. Symp.*, 2010, pp. 192–196.
[5] C. Che, C. Xiao, J. Liang, B. Jin, J. Zho, and F. Wang, "An RNN architecture with dynamic temporal matching for personalized predictions of Parkinson's disease," in *Proc. SIAM Int. Conf. Data Mining*, 2017, pp. 198–206.
[6] N. Kasabov, "Global, local and personalised modeling and pattern discovery in bioinformatics: An integrated approach," *Pattern Recognit. Lett.*, vol. 28, no. 6, pp. 673–685, 2007.

[7] J. Xu, J. Zhou, and P.-N. Tan, "FORMULA: FactORized MUlti-task LeArning for task discovery in personalized medical models," in *Proc. SIAM Int. Conf. Data Mining*. Philadelphia, PA, USA: SIAM, 2015, pp. 496–504.

[8] Q. Suo *et al.*, "Personalized disease prediction using a CNN-based similarity learning method," in *Proc. IEEE Int. Conf. Bioinformat. Biomed. (BIBM)*, Nov. 2017, pp. 811–816.

[9] J. Sun, F. Wang, J. Hu, and S. Edabollahi, "Supervised patient similarity measure of heterogeneous patient records," *ACM SIGKDD Explor. Newslett.*, vol. 14, no. 1, pp. 16–24, 2012.

[10] M. Zhan, S. Cao, B. Qian, S. Chang, and J. Wei, "Low-rank sparse feature selection for patient similarity learning," in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, Dec. 2016, pp. 1335–1340.

[11] A. Sharafoddini, J. A. Dubin, and J. Lee, "Patient similarity in prediction models based on health data: A scoping review," *JMIR Med. Inf.*, vol. 5, no. 1, p. e7, 2017.

[12] Y. Sha, J. Venugopalan, and M. D. Wang, "A novel temporal similarity measure for patients based on irregularly measured data in electronic health records," in *Proc. 7th ACM Int. Conf. Bioinformat., Comput. Biol., Health Informat.*, 2016, pp. 337–344.

[13] M. Huai, C. Miao, Q. Suo, Y. Li, J. Gao, and A. Zhang, "Uncorrelated patient similarity learning," in *Proc. SIAM Int. Conf. Data Mining*, 2018, pp. 270–278.

[14] Q. Suo, H. Xue, J. Gao, and A. Zhang, "Risk factor analysis based on deep learning models," in *Proc. 7th ACM Int. Conf. Bioinformat., Comput. Biol., Health Informat.*, 2016, pp. 394–403.

[15] Y. Cheng, F. Wang, P. Zhang, and J. Hu, "Risk prediction with electronic health records: A deep learning approach," in *Proc. SIAM Int. Conf. Data Mining*, 2016, pp. 432–440.

[16] F. Ma, R. Chitta, J. Zhou, Q. You, T. Sun, and J. Gao, "Dipole: Diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2017, pp. 1903–1911.

[17] Q. Suo *et al.*, "A multi-task framework for monitoring health conditions via attention-based recurrent neural networks," in *Proc. AMIA Annu. Symp.*, 2017.

[18] Y. Yuan, G. Xun, K. Jia, and A. Zhang, "A multi-view deep learning method for epileptic seizure detection using short-time Fourier transform," in *Proc. 8th ACM Int. Conf. Bioinformat., Comput. Biol., Health Informat.*, 2017, pp. 213–222.

[19] F. Ma *et al.*, "Unsupervised discovery of drug side-effects from heterogeneous data sources," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2017, pp. 967–976.

[20] Y. Yuan, G. Xun, Q. Suo, K. Jia, and A. Zhang, "Wave2Vec: Learning deep representations for biosignals," in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, Nov. 2017, pp. 1159–1164.

[21] Y. Yuan, G. Xun, K. Jia, and A. Zhang, "A novel wavelet-based model for eeg epileptic seizure detection using multi-context learning," in *Proc. IEEE Int. Conf. Bioinformat. Biomed. (BIBM)*, Nov. 2017, pp. 694–699.

[22] F. Wang and J. Sun, "PSF: A unified patient similarity evaluation framework through metric learning with weak supervision," *IEEE J. Biomed. Health Inform.*, vol. 19, no. 3, pp. 1053–1060, May 2015.

[23] F. Wang, J. Sun, and S. Ebadollahi, "Integrating distance metrics learned from multiple experts and its application in patient similarity assessment," in *Proc. SIAM Int. Conf. Data Mining*, 2011, pp. 59–70.

[24] R. Snyderman, "Personalized health care: From theory to practice," *Biotechnol. J.*, vol. 7, no. 8, pp. 973–979, 2012.

[25] C. L. Overby *et al.*, "A collaborative approach to developing an electronic health record phenotyping algorithm for drug-induced liver injury," *J. Amer. Med. Informat. Assoc.*, vol. 20, no. e2, pp. e243–e252, 2013.

[26] K. Ng, J. Sun, J. Hu, and F. Wang, "Personalized predictive modeling and risk factor identification using patient similarity," in *Proc. AMIA Summits Transl. Sci.*, 2015, pp. 132–136.

[27] Z. Zhu, C. Yin, B. Qian, Y. Cheng, J. Wei, and F. Wang, "Measuring patient similarities via a deep architecture with medical concept embedding," in *Proc. Data Mining (ICDM)*, 2016, pp. 749–758.

[28] E. Choi *et al.*, "Multi-layer representation learning for medical concepts," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 1495–1504.

[29] Y. Kim. (2014). "Convolutional neural networks for sentence classification." [Online]. Available: https://arxiv.org/abs/1408.5882

[30] H. O. Song, S. Jegelka, V. Rathod, and K. Murphy, "Deep metric learning via facility location," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 2206–2214.

[31] E. Choi, M. T. Bahadori, J. Sun, J. Kulas, A. Schuetz, and W. Stewart, "Retain: An interpretable predictive model for healthcare using reverse time attention mechanism," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016.

[32] M. Abadi *et al.* (2016). "TensorFlow: Large-scale machine learning on heterogeneous distributed systems." [Online]. Available: https://arxiv.org/abs/1603.04467

[33] D. P. Kingma and J. Ba. (2014). "Adam: A method for stochastic optimization." [Online]. Available: https://arxiv.org/abs/1412.6980

[34] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *J. Mach. Learn. Res.*, vol. 10, pp. 207–244, Feb. 2009.

[35] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in *Proc. 24th Int. Conf. Mach. Learn.*, 2007, pp. 209–216.

[36] P. Zadeh, R. Hosseini, and S. Sra, "Geometric mean metric learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 2464–2471.

[37] J. Ni, J. Liu, C. Zhang, D. Ye, and Z. Ma, "Fine-grained patient similarity measuring using deep metric learning," in *Proc. ACM Conf. Inf. Knowl. Manage.*, 2017, pp. 1189–1198.

[38] J. Goldberger, G. E. Hinton, S. T. Roweis, and R. R. Salakhutdinov, "Neighbourhood components analysis," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, pp. 513–520.

[39] K. Q. Weinberger and G. Tesauro, "Metric learning for kernel regression," in *Proc. Artif. Intell. Stat.*, 2007, pp. 612–619.

[40] Y. Huang, C. Li, M. Georgiopoulos, and G. C. Anagnostopoulos, "Reduced-rank local distance metric learning," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2013, pp. 224–239.

[41] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf. Process., Manage.*, vol. 45, no. 4, pp. 427–437, 2009.

[42] P. Rogliani, G. Lucà, and D. Lauro, "Chronic obstructive pulmonary disease and diabetes," *COPD Res. Pract.*, vol. 1, p. 3, Aug. 2015.

[43] S. E. Inzucchi *et al.*, "Management of hyperglycaemia in type 2 diabetes: A patient-centered approach. Position statement of the American diabetes association (ADA) and the European association for the study of diabetes (EASD)," *Diabetologia*, vol. 55, no. 6, pp. 1577–1596, 2012.

[44] C. Hanson, E. P. Rutten, E. F. M. Wouters, and S. Rennard, "Influence of diet and obesity on COPD development and outcomes," *Int. J. Chronic Obstructive Pulmonary Disease*, vol. 9, p. 723, Aug. 2014.

[45] Y. Shi, A. Bellet, and F. Sha, "Sparse compositional metric learning," in *Proc. AAAI*, Jul. 2014, pp. 2078–2084.