Personalized Disease Prediction Using a CNN-Based Similarity Learning Method

Qiuling Suo*, Fenglong Ma*, Ye Yuan†, Mengdi Huai*, Weida Zhong*, Aidong Zhang* and Jing Gao*
*Department of Computer Science and Engineering
State University of New York at Buffalo, NY, USA
Email: {qiulings, fenglong, mengdihu, weidazho, azhang}@buffalo.edu
†College of Information and Communication Engineering
Beijing University of Technology, Beijing, China
Email: yuanye91@emails.bjut.edu.cn

***Abstract*—Predicting patients' risk of developing certain diseases is an important research topic in healthcare. Personalized predictive modeling, which focuses on building specific models for individual patients, has shown its advantages on utilizing heterogeneous health data compared to global models trained on the entire population. Personalized predictive models use information from similar patient cohorts, in order to capture the specific characteristics. Accurately identifying and ranking the similarity among patients based on their historical records is a key step in personalized modeling. The electric health records (EHRs), which are irregular sampled and have varied patient visit lengths, cannot be directly used to measure patient similarity due to lack of an appropriate vector representation. In this paper, we build a novel time fusion CNN framework to simultaneously learn patient representations and measure pairwise similarity. Compared to a traditional CNN, our time fusion CNN can learn not only the local temporal relationships but also the contributions from each time interval. Along with the similarity learning process, the output information which is the probability distribution is used to rank similar patients. Utilizing the similarity scores, we perform personalized disease predictions, and compare the effect of different vector representations and similarity learning metrics.**

I. Introduction

Accurately predicting diseases plays a significant role in public health, especially at the early stage which allows patients to take prevention treatments in time. With the growing volume and availability of electronic health records (EHRs), predictive modeling tasks for disease progression and analysis have obtained increasing interest from researchers. The EHR data are temporally sequenced by patient visits with each visit represented as a set of high dimensional clinical events. Mining EHRs is especially challenging compared to standard data mining tasks, due to its noisy, irregular and heterogeneous nature. A conventional approach of disease prediction is the *one-size-fit-all* model [1]. That is, using all available training data to build a global model, and then with this model, predicting the risk of diseases for each patient. The benefit of applying a one-size-fit-all model is that it captures the overall information of the entire training population. However, patients may have different phenotypes, different medical conditions, etc. Using a global model may miss some specific information that is important for individual patients. Thus, building a targeted, patient-specific model for each individual patient is urgent and important for personalized medicine.

Recent studies [2–5] show that personalized models can improve predictive performance over global models. A general framework for personalized prediction contains two stages: (1) measuring the similarity among patients, and (2) building a separate model for each patient using his/her similar cohorts. 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 improve a lot. Many similarity learning methods have been proposed [6–10] 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.

Recently, deep learning approaches have been widely adopted and rapidly developed in patient representation learning [11–18] such as autoencoder, recurrent neural networks (RNNs) and convolution neural networks (CNNs). In [19], CNN has shown it superior ability on the task of measuring patient similarity. However, one drawback of the traditional CNN architecture is that it could not fully utilize the temporal and contextual information of EHRs for disease prediction. Consequently, simultaneously modeling temporality and content of EHR data is more challenging.

To tackle the aforementioned challenges and issues, in this paper, we aim to solve the following key problems in personalized prediction: how to build a model to accurately measure patients' similarity from their historical records, and how to build an accurate personalized prediction model with the learned similarities. To achieve these goals, we first design

978-1-5090-3050-7/17/$31.00 ©2017 IEEE

a novel time-fusion CNN based framework to account for the temporality across different time intervals. With the proposed framework, we can generate the vector representation for each patient. Based on the learned patient representations, a matching metric is then introduced to obtain a similarity representation. Considering the practical meaning, we add a firm symmetric constraint to the framework structure. This similarity learning framework is end-to-end, which learns patient representations and pairwise similarity simultaneously. 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. Finally, we build a personalized model for each patient using his/her similar cohorts. In summary, our contributions are as follows:

- We build a framework to jointly learn patient EHR representations and pairwise similarity, without the hand-crafted feature aggregations. With the framework, parameters of representation and similarity learning can be optimized simultaneously, yielding higher accuracy.
- We develop a time-fusion CNN model that not only preserves the local temporality across adjacent visits but also considers the global contributions from different time intervals.
- Our experimental results show that our similarity learning framework can learn better representation vectors for patients' historical information and improve the disease prediction accuracy. The personalized model based on weighted sampling improves personalized prediction accuracy compared to other commonly used strategies.

II. Method

Our model follows the two-stage manner: measuring patient similarity and performing personalized prediction. In this section, we first provide our end-to-end framework of similarity learning based on pairwise training CNN. Next, we talk about personalized predictive models.

A. Similarity 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 $c_1, c_2, ..., c_{|\mathcal{C}|} \in \mathcal{C}$, 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. Therefore, each patient record can be viewed as a matrix, where the horizontal dimension corresponds to medical events and the vertical dimension corresponds to visits. The (i, j)-th entry of a matrix is 1 if code c_j is observed at time stamp V_i for the corresponding patient. 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 $t = \max\{V_i\}_{i=1}^{T_n}$, for the sake of CNN operations.

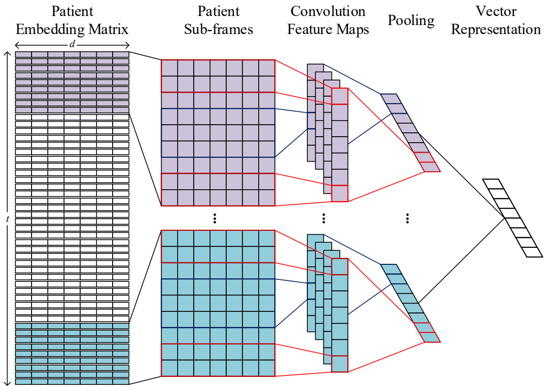

Fig. 1: Time-fusion CNN for vector representation learning. The patient embedding matrix is segmented into several sub-frames. For each sub-frame, one-side convolution and pooling are applied to obtain a vector representation. The set of vectors are weighted to form a comprehensive vector representation.

2) Visit Embedding: The original one-hot representation 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: $\boldsymbol{x_i} = \mathrm{ReLU}(\boldsymbol{W}_v \boldsymbol{v}_i + \boldsymbol{b}_v)$, where d is the embedding dimension, $\boldsymbol{W}_v$ and $\boldsymbol{b}_v$ are the weight matrix and bias vector to be learned, and the activation function ReLU is defined as $\mathrm{ReLU}(\boldsymbol{x}) = \max(\boldsymbol{x}, \boldsymbol{0})$. The adoption of ReLU ensures non-negative representation, which enables the learned vector to be interpretable [20]. After the embedding operation, we can obtain an embedding matrix $\boldsymbol{X} \in \mathbb{R}^{t \times d}$ for each patient.

3) Convolutional Neural Network: Different from images with spatial relations across pixels, the positions of medical codes have no spatial/temporal meaning. 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 $\boldsymbol{w}_c \in \mathbb{R}^{h \times d}$, 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 feature c_i is generated using $c_i = \mathrm{ReLU}(\boldsymbol{W}_c \cdot \boldsymbol{x}_{i:i+h-1} + b_c)$. This filter is applied to each possible window of timestamps $\{\boldsymbol{x}_{1:h}, \boldsymbol{x}_{2:h+1}, ..., \boldsymbol{x}_{t-h+1:t}\}$ with a stride equal to 1, to produce a feature map $\boldsymbol{c} = \{c_1, c_2, ..., c_{t-h+1}\}$, 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 feature 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

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

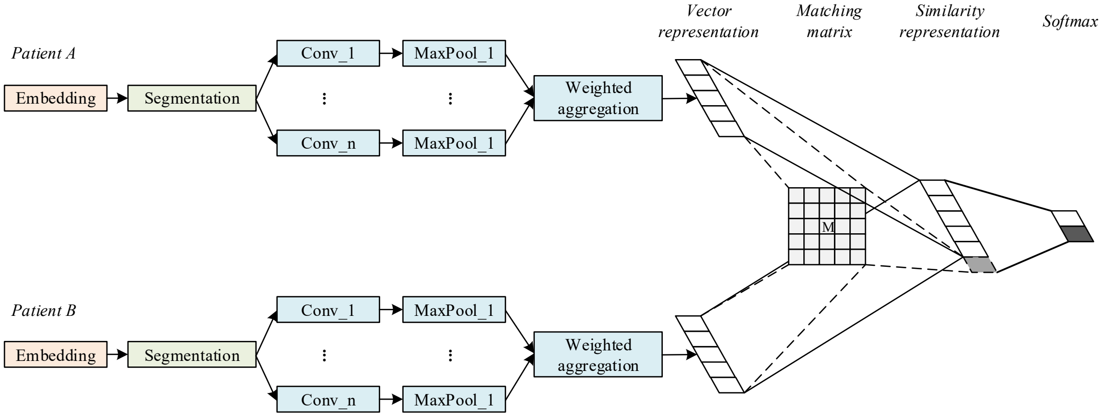

Fig. 2: The overall framework of pairwise patient similarity learning. The one-hot EHR matrix of patient A is first mapped into an embedding matrix with a lower feature dimension, and then segmented into several sub-frames. One-side convolution and max pooling are applied on each sub-frame. The sub-frame vectors are then aggregated to form a comprehensive vector for patient A. Patient B shares the same embedding and CNN parameters. The patient representations then pass through a matching matrix 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.

outputs from all the filters are concatenated to form a vector representation $\boldsymbol{h} \in \mathbb{R}^m$. $\boldsymbol{h}$ is the vector representation of the original embedding matrix $\boldsymbol{X}$.

4) Time Fusion: Although the convolution operation applied across time considers the relations across adjacent visits, it does not fully utilize the temporal information, and treats the importance of each time stamp equally. In practice, a symptom near the onset of a disease is usually more important than the same one appearing far away from the onset date. To account for the temporal information, we borrow the idea of time fusion in [12, 21] and further modify the CNN model. We segment every data sample as a set of short, fixed-sized sub-frames. Each sub-frame contains several contiguous visits, covering a certain time interval. Our model considers how much each sub-frame contributes to the final decision.

The framework of our time fusion CNN is shown in Fig. 1. The patient embedding matrix is first segmented into sub-frames with each sub-frame a fixed length of k visits. Since the segmentation operation may split the connection between border time stamps of two adjacent sub-frames, we use a sliding window to account for the overlap between sub-frames. For each sub-frame, we apply the convolution and pooling operations described in Section II-A3 to obtain a vector representation. Then we use weighted average of these vectors to obtain a comprehensive vector representation for the original embedding matrix. The aggregation weight of each sub-frame is learned using the formula: $\alpha_i = \tanh(\boldsymbol{W}_a^T \boldsymbol{h}_i + b_a)$. The weights are normalized as $\boldsymbol{\alpha} = \mathrm{Softmax}([\alpha_1, \alpha_2, ..., \alpha_t])$. The weights do not depend on the number of sub-frames. Therefore, this weighted average can reduce the effect from visits padded by zeros. The overall representation for a patient is $\tilde{\boldsymbol{h}} = \sum \alpha_i \boldsymbol{h}_i$.

5) Similarity Learning: The similarity between a pair of vectors can be measured by [8]: $S = \tilde{\boldsymbol{h}}_A \boldsymbol{M} \tilde{\boldsymbol{h}}_B$, 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 $\boldsymbol{L} \in \mathbb{R}^{g \times m}$ with $g < m$ to ensure a low rank characteristic. Different from [19] which directly concatenates the vector representations $\tilde{\boldsymbol{h}}_A$, $\tilde{\boldsymbol{h}}_B$ and $\boldsymbol{S}$, we consider the symmetric constraint 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 $\tilde{\boldsymbol{h}}_A$ and $\tilde{\boldsymbol{h}}_B$ into a single vector with their dimension holds using the formula: $\boldsymbol{H} = \boldsymbol{W}_h \tilde{\boldsymbol{h}}_A \oplus \boldsymbol{W}_h \tilde{\boldsymbol{h}}_B$, 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 $\hat{y}$, indicating the similarity degree between two patients. Here we set the ground truth y as 1 if two patients has the risk of developing the same disease. We use cross-entropy between y and $\hat{y}$ to calculate the loss for patient pairs:

$$\mathcal{L} = \frac{1}{\tilde{N}} \sum_{i=1}^{\tilde{N}} (y_i \log(\hat{y_i}) + (1 - y_i) \log(1 - \hat{y_i})),$$

where $\tilde{N}$ is the total number of patient pairs. Since there are N patients, $\tilde{N}$ would be $N(N-1)/2$. The overall framework of supervised patient similarity learning is shown in Figure 2. Our model learns vector representations and similarity scores simultaneously. This similarity learning process is end-to-end, and all the parameters are updated through back-propagation.

B. Personalized Prediction

The learned similarity can be used for personalized prediction. The similarity score from Section II-A5 can be used to measure the similarity degree between a pair of patients. For each test patient, we first calculate his/her similarity probability with each of the training patients, and then rank the training patients according to their similarity scores. We then build a specific personalized model for each patient based on his/her similarity cohorts. In this paper, we use three ways to build the personalized model:

1) K Nearest Neighbors: We first select k training patients with the top$k$ similarity probabilities for a test patient, and then use the most common disease appearing among them as the prediction. Intuitively, since the patients have similar health records/symptoms, it is highly possible that they have the risk of developing the same disease.

2) Discriminate Classification: In this way, predictive models are trained on patients with k highest and lowest scores. Patients who are dissimilar with the testing patient are discriminative for the classification. Here we use a multi-class logistic regression as the predictive model. The model parameters are trained separately to fit each specific patient.

3) Weighted Sampling: The above two methods need to optimize k, since the prediction performance highly depends on the learned similarity score and the number of selected patients. Therefore, we propose to use weighted sampling to select training patient cohorts. Specifically, this procedure is implemented by sampling patients with replacement following the distribution of similarity scores. Therefore, similar patients are more likely to be chosen, while those below the similarity threshold could also be sampled. After that, prediction models can be built using the sampled data. The above procedures are repeated several times, and majority voting among the results is used to get the most likely prediction.

III. 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 [22], 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 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 dataset is randomly divided into training and testing sets in a 0.8:0.2 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.

TABLE I: Statistics of dataset.

Cohorts	Diabetes	Obesity	COPD	Total
# Patients	3,214	3,441	2,328	8,983
Total # events	160,920	217,583	136,886	515,389
Avg.# of visits	23.03	30.62	26.91	26.25
Avg.# event per patient	50.44	63.76	58.42	57.37

B. Experimental Setup

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

1) Model Implementation: Our task is to predict the risk of developing a certain disease for each patient. We first train the similarity model described in Section II-A5 to obtain the optimized parameters of CNNs and the matching metric. Then, using the similarity framework, we calculate and rank the matching degree of each testing instance and all the training data. After that, personalized prediction methods in II-B are used for multi-label prediction.

The similarity framework is implemented with Tensorflow [23]. Adam [24] 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.

2) Baseline Approaches: To validate the performance of the proposed model for personalized prediction, we consider the following aspects: whether the learned vector representations can better represent the original data; whether the supervised similarity framework is effective; and whether the personalized modeling can distinct heterogeneous groups.

We consider three ways to map the EHR matrix into vector representations: aggregated vectors (aggr. vecs), CNN learned vectors (CNN vecs), and vectors by our time fusion CNN (CNN_t vecs). For the aggregated vectors, we count the number of medical codes for each patient based on all his/her visits, so that each element indicates the frequency of a corresponding code. The CNN vectors are learned through a traditional CNN model, while the CNN_t vectors are learned using our time fusion CNN variant.

Since the proposed architecture is for personalized disease prediction, we first compare it with a global model to show the necessary of designing personalized predictive models.

• Global model (GM) with multi-label prediction. This is a one-size-fit-all model which uses the entire training data to build a global model for the prediction of all the testing data. This model does not consider the inherent groupings of different patients. We use logistic regression (LR) as the classifier, and apply it on the three vector representations.

TABLE II: Disease prediction performance using different vector representations, similarity learning methods and final prediction strategies.

Method	Global Model	Personalized Predictive Models								
		KNN			Weighted sampling			Personalized LR		
		Euclidean	Cosine	Sup.sim	Euclidean	Cosine	Sup.sim	Euclidean	Cosine	Sup.sim
Aggr. vecs	0.7180	0.5669	0.5940	0.7558	0.5812	0.6279	0.7556	0.6073	0.6574	0.7614
CNN vecs	0.7508	0.7280	0.7402	0.7836	0.7274	0.7469	0.8109	0.7235	0.7525	**0.8153**
CNN_t vecs	**0.7736**	0.7530	0.7547	**0.8025**	0.7731	0.7736	**0.8164**	0.7686	0.7758	0.8115

The proposed model consists of two parts: similarity learning and final prediction. We use the following ways to perform the final disease prediction:

- Personalized prediction strategies. To find out an appropriate way to build personalized models, we compare the three strategies as discussed in Section II-B: KNN, discriminative classification and weighted sampling. For the discriminative classification, we apply *personalized LR* for each patient using his/her similarity training samples.

We can use several methods to measure patient similarity. In order to fairly and clearly show the performance of all the personalized prediction approaches, in our experiments, we employ the following similarity strategies on each personalized prediction model:

- Traditional similarity learning methods: Euclidean distance and cosine distance. The two types of distances are calculated on the three vector representations. We then build personalized models based on the learned similarity distances.
- Supervised similarity (Sup. sim) learning methods. This group of methods use the same similarity metric in Section II-A5, and can be regarded as reduced baselines of our method. Vector representations, together with the matching matrix are fed into a softmax layer to get the similarity probability as the scores. We then build personalized models based on the learned similarity scores.

C. Experimental Results

We compare both the performance of personalized disease prediction and similarity learning. The main task is disease prediction, while similarity learning is the key step to achieve high prediction accuracy.

1) Disease Prediction Results: The experimental results are shown in Table II. It contains the comparison among three vector representations (Aggr. vecs, CNN vecs, CNN_t vecs), four disease prediction methods (GM, KNN, weighted sampling, personalized LR), and three similarity learning methods (Euclidean, cosine, Sup.sim). Among them, GM is a one-size-fit-all model; KNN, weighted sampling and personalized LR are methods for the final personalized prediction; and Euclidean, cosine and Sup.sim are similarity learning methods. With the similarity ranking from the three distance metrics, we use KNN ($k = 10$) for personalized prediction, as the results change very little within a certain range of k. For fair comparison, we also experiment on weighted sampling and personalized LR with a sample size 500 to perform prediction. In total, there are 27 combinations of the vector representations, similarity learning methods and final personalized predictions. We use the overall accuracy as the measurement criteria, which is the ratio of the number of correctly predicted labels and total number of testing samples.

From Table II, we can see that under each the personalized prediction task, the vector representations learned by CNNs can notably improve the performance of different distance metrics. Since the static high-dimensional aggregated vector representation completely ignores the temporal relationship across timestamps, it could not exhibit the essence of original patient EHR matrix. CNN learns the local temporal relationships across visits and captures the most important information of a visit sequence, so that it can notably improve the classification performance. CNN_t not only learns local temporal relationships, but also measures the contribution from each smaller time interval, so that it can better represent the temporal information of the original visit sequences.

Fixing a vector representation, we compare the similarity learning methods. Euclidean and cosine distances, which are widely used similarity metrics, are directly applied on the three vector representations between each pair of patients, with closer distance for higher similarity. Compared with these two distance metrics, our supervised similarity method leads to a better performance. This improvement owes to the utilization of the label information, as the parameters for similarity learning are optimized with the guidance of similarity labels.

TABLE III: Confusion matrix of global model using aggregated vectors.

	Diabetes	Obesity	COPD	Accuracy
Diabetes	440	159	44	0.6842
Obesity	68	579	42	0.8403
COPD	82	111	273	0.5858

TABLE IV: Confusion matrix of KNN on Sup. sim using CNN_t vectors.

	Diabetes	Obesity	COPD	Accuracy
Diabetes	**515**	74	54	0.8009
Obesity	44	**602**	43	0.8737
COPD	56	31	**379**	0.8133

Compared with the global models, personalized predictions based on supervised similarity can achieve higher accuracy. This is due to the fact that although the global model considers the characteristics of the entire training population, it ignores the inherent cohort nature of individual patients. Personalized model can better fit each specific, targeted patient. Table III and Table IV respectively show the confusion matrix of a global classification model and our proposed model. The accuracy indicates the true positive rate for each disease prediction. We can see that our model can better distinct the three diseases. In fact, the three diseases do display several relationships with each other,and share some symptoms and complications, especially diabetes and obesity, making them hard to be discriminated. Compared to the global model, our method can better identify the three disease cohorts, especially the diabetes and COPD cohorts. Having more detailed subgroup information within each cohort may help to better discriminate the heterogeneous nature of EHRs.

TABLE V: Accuracy of similarity learning.

	K-means	Spectral	Sup.sim
Aggr. vecs	0.4840	0.5050	0.6519
CNN vecs	0.5743	0.4195	0.7281
CNN_t vecs	0.5751	0.4492	**0.7359**

2) Similarity Learning results: Since similarity learning is the main step of personalized prediction, we also evaluate the performance of our similarity framework. Similar to Section III-C1, we consider the performance on three levels of vector representations. Here we focus on only similarity measurement without going through the disease prediction step. For the testing patients, we predict whether each pair of patients are diagnosed with the same disease or not. The measuring criteria is the ratio of correctly grouped patient pairs and the total number of patient pairs. We compare our similarity framework with two popular clustering methods: k-means and spectral clustering. As in Table V, we can see that under each method, CNN-based models can learn much better vector representations. Moreover, supervised similarity framework significantly outperforms clustering methods on the task of grouping patient cohorts.

IV. Conclusion

Personalized predictive modeling in healthcare aims to find unique characteristics of individual patients, and build targeted, patient specific predictions. In this paper, we propose a time-fusion CNN based framework to pairwise measure patient similarity, and use three ways to perform personalized disease prediction. Experimental results show that our time fusion CNN can better represent the longitudinal EHR sequences, and our end-to-end similarity framework outperforms widely-used distance metrics. Having the similarity ranks, we then perform three ways for personalized prediction and show that weighted sampling can give a stable and high accuracy.

Acknowledgement

This work was supported in part by the US National Science Foundation under grants NSF IIS-1218393 and 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.

References

[1] K. Ng, J. Sun, J. Hu, and F. Wang, “Personalized predictive modeling and risk factor identification using patient similarity,” *AMIA Summits on Translational Science Proceedings*, 2015.

[2] N. Kasabov, “Global, local and personalised modeling and pattern discovery in bioinformatics: An integrated approach,” *Pattern Recognition Letters*, 2007.

[3] J. Lee, D. M. Maslove, and J. A. Dubin, “Personalized mortality prediction driven by electronic medical data and a patient similarity metric,” *PloS one’15*.

[4] J. Xu, J. Zhou, and P.-N. Tan, “Formula: Fac orized multi-task learning for task discovery in personalized medical models,” in *SDM’15*.

[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 *SDM’17*. SIAM.

[6] J. Sun, F. Wang, J. Hu, and S. Edabollahi, “Supervised patient similarity measure of heterogeneous patient records,” *ACM SIGKDD Explorations Newsletter*, 2012.

[7] A. Gottlieb, G. Y. Stein, E. Ruppin, R. B. Altman, and R. Sharan, “A method for inferring medical diagnoses from patient similarities,” *BMC medicine*, 2013.

[8] M. Zhan, S. Cao, B. Qian, S. Chang, and J. Wei, “Low-rank sparse feature selection for patient similarity learning,” in *ICDM’16*.

[9] A. Sharafoddini, J. A. Dubin, and J. Lee, “Patient similarity in prediction models based on health data: a scoping review,” *JMIR medical informatics*, 2017.

[10] Y. Sha, J. Venugopalan, and M. D. Wang, “A novel temporal similarity measure for patients based on irregularly measured data in electronic health records,” *Platelets*, 2016.

[11] Q. Suo, H. Xue, J. Gao, and A. Zhang, “Risk factor analysis based on deep learning models,” in *ACM-BCB’16*.

[12] Y. Cheng, F. Wang, P. Zhang, and J. Hu, “Risk prediction with electronic health records: A deep learning approach,” in *SDM’16*.

[13] 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 *SIGKDD’17*.

[14] Q. Suo, F. Ma, G. Canino, J. Gao, A. Zhang, P. Veltri, and A. Gnasso, “A multi-task framework for monitoring health conditions via attention-based recurrent neural networks,” in *AMIA annual symposium proceedings*, 2017.

[15] 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 *ACM-BCB’17*.

[16] F. Ma, C. Meng, H. Xiao, Q. Li, J. Gao, L. Su, and A. Zhang, “Unsupervised discovery of drug side-effects from heterogeneous data sources,” in *SIGKDD’17*.

[17] Y. Yuan, G. Xun, Q. Suo, K. Jia, and A. Zhang, “Wave2vec: Learning deep representations for biosignals,” in *ICDM’17*.

[18] Y. Yuan, G. Xun, K. Jia, and A. Zhang, “A novel wavelet-based model for eeg epileptic seizure detection using multi-context learning,” in *BIBM’17*.

[19] 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 *ICDM’16*.

[20] E. Choi, M. T. Bahadori, E. Searles, C. Coffey, M. Thompson, J. Bost, J. Tejedor-Sojo, and J. Sun, “Multi-layer representation learning for medical concepts,” in *SIGKDD’16*.

[21] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, “Large-scale video classification with convolutional neural networks,” in *CVPR’14*.

[22] 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 *NIPS’16*.

[23] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, “Tensorflow: Large-scale machine learning on heterogeneous distributed systems,” *arXiv preprint arXiv:1603.04467*, 2016.

[24] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.