Integrating Multimodal Electronic Health Records for Diagnosis Prediction

Rui Li[1], Fenglong Ma, PhD[2], Jing Gao, PhD[3]
[1]Department of Computer Science and Engineering, University at Buffalo, NY, USA
[2]College of Information Sciences and Technology, Pennsylvania State University, PA, USA
[3]School of Electrical and Computer Engineering, Purdue University, IN, USA

Abstract

Diagnosis prediction aims to predict the patient's future diagnosis based on their Electronic Health Records (EHRs). Most existing works adopt recurrent neural networks (RNNs) to model the sequential EHR data. However, they mainly utilize medical codes and ignore other useful information such as patients' clinical features and demographics. We proposed a new model called MDP to augment the prediction performance by integrating the multimodal clinical data. MDP learns the clinical feature representation by adjusting the weights of clinical features based on a patient's current health condition and demographics. Also, the clinical feature representation, diagnosis codes representation and the demographic embedding are integrated to perform the prediction task. Experiments on a real-world dataset demonstrate that MDP outperforms the state-of-the-art methods.

Introduction

Electronic Health Records (EHRs) are digital version of patient's medical charts, which consist of longitudinal multimodal data including demographics, diagnosis, clinical notes and clinical features. Among predictive modeling tasks utilizing EHRs, diagnosis prediction is one of the most challenging and widely explored tasks, which aims to predict future diagnosis from the patient's historical diagnosis record. The input to this task consists of a sequence of past patient visits with diagnosis codes (e.g., ICD9 codes) in each visit, and the output of the task should be the diagnosis codes at the next visit. The challenge of diagnosis prediction mainly exist in the following two aspects: (1) EHR data is heterogeneous and noisy. Various types of features (variables) are included in EHR, including categorical variables (e.g., medical codes), numerical variables (e.g., clinical measurements), and textual variables (e.g., clinical notes). Meanwhile, due to patients' irregular visits and incomplete recording, there may be a lot of missing data. (2) The potential prediction space is very large. The current ICD-9-CM system consists of more than 13,000 codes, and it is difficult to make predictions given the large target space.

Recently, deep learning techniques have been widely adopted for diagnosis prediction tasks. In order to model the sequential EHR data, most approaches use the recurrent neural networks (RNNs) due to its ability of keeping track of sequential information. The key idea is to project the diagnosis codes in the sequence into low-dimensional features (i.e., embeddings) that capture the information that is the most relevant to the prediction task. This embedding learning is enabled by the RNN, and various methods[1–5] differ in the strategies that are designed to learn such embeddings effectively. Among these methods, Retain[1] and Dipole[2] make predictions based on sequences of medical codes only, and GRAM[3], KAME[4] and HAP[5] also incorporates the hierarchical structure of the disease taxonomy. Despite these successes, existing diagnosis prediction methods still have the following limitations.

(1) Most of the existing methods only use medical codes as input and ignore other useful information such as clinical features and demographics. Clinical features include vital signs and lab test results, which contain plenty of details about the patient's symptoms and are considered as a significant complement of diagnosis codes. Similarly, patient demographics record the static information about the patient, including variables such as age, gender and ethnicity. Combining clinical features and demographics with diagnosis codes can greatly boost the prediction performance.

(2) Recently, a method called MHM[6] is developed to integrate the clinical features into the diagnosis prediction task. However, this approach simply concatenates the diagnosis codes representation and the clinical feature representation, and learns a representation for every layer of the disease taxonomy. In fact, the importance of clinical features varies enormously for different diseases. In addition, it is essential to assign different weights to clinical features based on current health condition of the patient, which can be derived based on patients' clinical features and demographics information. This weighted mechanism helps to learn the clinical feature representation that better depicts the clinical symptoms.

(3) The granularity of time stamps when clinical features are recorded fluctuates and there may be a lot of missing data. Clinical features are recorded during a hospital stay. For different visits, the length of hospital stay varies. Meanwhile, in a hospital stay, vital signs and lab test results may not be recorded regularly, and different clinical features may be missing at different time stamps.

Motivated by these observations, we propose a **m**ultimodal **d**iagnosis **p**rediction model (MDP) to address these challenges. MDP takes diagnosis codes, disease taxonomy, clinical features and demographics as input, which are all relevant to the prediction task, and then feeds the input to a deep neural networks consisting of the following integral components. As shown in Figure 1, the diagnosis code encoder utilizes the disease taxonomy to learn the diagnosis code representations, and the clinical feature encoder learns the clinical feature representation by integrating the weight adjustment mechanism and the attentive clinical feature aggregation mechanism. The final representation, which combines predictive information obtained from the patient's demographics, diagnosis codes and clinical features, is used to make the prediction. The proposed model is able to extract the relevant signals from all these valuable information sources, and effectively utilize them for challenging diagnosis prediction tasks with missing data and varied time granularity in the data.

Our main contributions are summarized as follows:

- We propose a novel and effective deep learning framework for diagnosis prediction, a very important task in health informatics. To the best of our knowledge, we are the first to integrate medical codes, clinical features and patient demographics in the diagnosis prediction task.
- We design a clinical feature weight adjustment mechanism which learns the correlation between the patient's diagnosis codes and the clinical features, and adjusts the weight of clinical features based on the patient's health condition and demographics.
- We incorporate an attentive clinical feature aggregation mechanism into the framework to deal with the missing data and the varying time granularity of clinical features. The aggregation mechanism assigns lower importance scores to the time stamps with missing data and captures the long-term dependencies by integrating the hidden states of clinical features at different time stamps.
- We empirically show that MDP outperforms existing methods on a real-world EHR data set. We also reveal some important properties of the proposed model that could explain the model superiority by experiments involving a quantitative analysis of clinical feature importance degrees and qualitative analysis based on case studies.

Problem Statement

EHR Data. For each patient, the clinical record can be viewed as a sequence of visits $V_1, \ldots, V_T$, where each visit record V_t contains diagnosis information $\mathbf{x}_t$ and clinical features $\mathbf{c_t}$. The diagnosis information $\mathbf{x_t} \in \{\mathbf{0}, \mathbf{1}\}^{|\mathcal{D}|}$ is a multi-hot binary vector, where $|\mathcal{D}|$ is the number of unique diagnosis codes, and $\mathcal{D} = \{d_1, d_2, \ldots, d_{|\mathcal{D}|}\}$. $x_{t,i} = 1$ indicates that the patient was diagnosed with disease d_i in the t-th visit; otherwise 0. For clinical features $\mathbf{c_t} \in \mathbb{R}^{N \times T_t}$, N is the number of clinical features that we are interested in, and T_t is the length of the ICU stay in the t-th visit. The clinical features include vital signs and some lab test results. Besides, there is a demographic vector $\mathbf{p} \in \{\mathbf{0}, \mathbf{1}\}^{\mathbf{r}}$ associated with each patient, which includes the patient's gender, ethnicity, age and other demographic information, and r denotes the number of such information.

Disease Taxonomy. Let $\mathcal{G}$ denote the disease taxonomy, which contains the hierarchy of disease concepts in the form of a $parent\text{-}child$ relationship, and the diagnosis codes in $\mathcal{D}$ are the leaf nodes. We define $\mathcal{D}' = \{d_{|\mathcal{D}|+1}, d_{|\mathcal{D}|+2}, \ldots, d_{|\mathcal{D}|+|\mathcal{D}'|}\}$ as the set containing the ancestor codes, and all nodes in $\mathcal{G}$ form the set $\mathcal{C} = \mathcal{D} + \mathcal{D}'$. We construct $\mathcal{G}$ using the multi-level diagnoses CCS categories[1].

Diagnosis Prediction Task. Based on the above notations, we define our task as follows. Given the patient's diagnosis information $\mathbf{x_t}$, clinical features $\mathbf{c_t}$, demographic data $\mathbf{p}$, and the disease ontology $\mathcal{G}$, the goal of this task is to predict diagnosis codes of the next visit denoted as $\hat{\mathbf{y}}_{t+1}$.

[1] https://www.hcup-us.ahrq.gov/toolssoftware/ccs/ccs.jsp

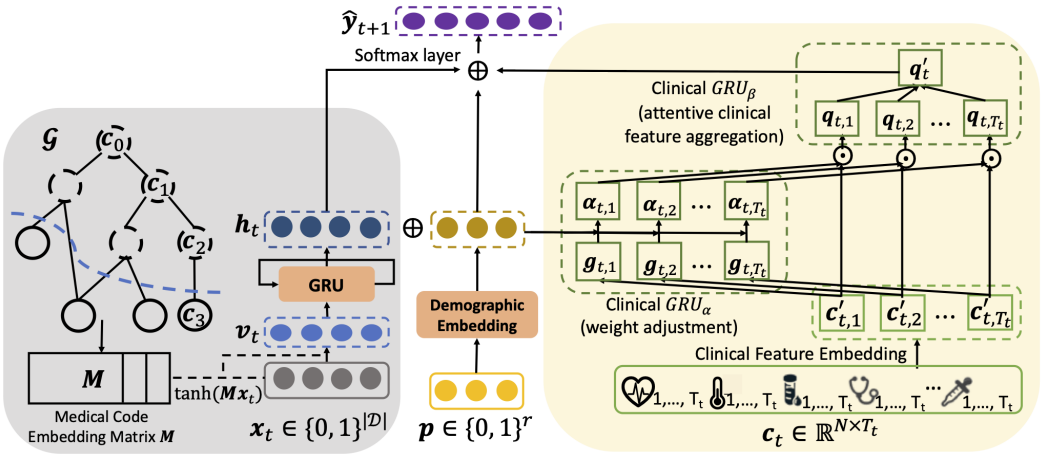


Figure 1: Framework for MDP.

Methods

Figure 1 shows the overview of the proposed MDP framework, which mainly contains two parts: (1) Diagnosis code encoder that utilizes the patient's diagnosis codes and Gated Recurrent Unit (GRU)[7] to learn the representations capturing the patient's health conditions, and (2) clinical feature encoder which adjusts the weights of clinical features based on the current health conditions and demographic information, and finally learns the clinical feature representation during the ICU stay. Next, we describe the two parts separately and show how they can be optimized jointly.

Diagnosis Code Encoder

Diagnosis Code Embedding. In order to learn the robust embeddings of the diagnosis codes in the disease taxonomy $\mathcal{G}$, we employ the graph embedding method GRAM[3]. In Figure 1, leaf nodes or solid circles in $\mathcal{G}$ represent diagnosis codes in set $\mathcal{D}$, while non-leaf nodes or dashed circles represent more general concepts in $\mathcal{D}'$. Every node in $\mathcal{G}$ has a basic learnable embedding $\mathbf{e}_i (1 \leq i \leq |\mathcal{D}| + |\mathcal{D}'|)$, where $|\mathcal{D}'|$ represents the number of intermediate nodes. GRAM learns the final embedding vector of the i-th diagnosis code $\mathbf{m}_i$ by combining the base embedding $\mathbf{e}_i$ and its ancestors' base embeddings via the graph-based attention mechanism. After concatenating the diagnosis embedding vectors $\mathbf{m}_1$, $\mathbf{m}_2$, $\cdots$, $\mathbf{m}_{|\mathcal{D}|}$ of all diagnosis codes, we obtain the diagnosis code embedding matrix $\mathbf{M} \in \mathbb{R}^{d_1 \times |\mathcal{D}|}$, where d_1 is the dimension size of the embedding vector. For the t-th visit, given the diagnosis information $\mathbf{x}_t$, the vector $\mathbf{v}_t$ is computed as $\mathbf{v}_t = tanh(\mathbf{M}\mathbf{x}_t)$.

Visit Representation Learning. After obtaining the vector $\mathbf{v}_t$ that contains all diagnosis information at the t-th visit, we use Recurrent Neural Networks (RNNs) to capture the dependencies among multiple visits. RNN is an efficient method to model sequential data, and it has been widely used on healthcare data[1,2,4,8,9]. RNNs come in multiple variants, including Long-Short Term Memory (LSTM)[10] and Gated Recurrent Unit (GRU)[7]. In this paper, we choose GRU. The hidden state $\mathbf{h}_t \in \mathbb{R}^{g_1}$ is calculated recurrently by $\mathbf{h}_t = GRU(\mathbf{v}_t, \mathbf{h}_{t-1}, \Theta_h)$, where g_1 is the dimension size of the hidden state, $\mathbf{v}_t$ is the diagnosis codes vector, $\mathbf{h}_{t-1}$ is the hidden state of the previous visit, and Θ_h represents all GRU parameters to be learned. We use $\mathbf{h}_t$ to denote the representation of the t-th visit.

Clinical Feature Encoder

Clinical features include vital signs and lab test results, and they are considered as significant complements of diagnosis codes. During the ICU stay in the t-th visit, the vital signs such as heart rate and blood pressure are recorded hourly, and the lab tests such as blood chemistry analysis and urine analysis are performed periodically. We represent the clinical features as $\mathbf{c}_t \in \mathbb{R}^{N \times T_t}$, which have two characteristics. (1) The importance of clinical features varies enormously for different diseases. For example, for patients with diabetes, glucose is more important comparing with body temperature, while for patients with fever, the opposite happens. (2) The timescale of clinical features fluctuates.

For different visits, the length of the ICU stay fluctuates from less than 24 hours to more than 500 hours. For visits with long ICU stay, RNNs may fail to capture the long term dependency. To take these unique characteristics into consideration, we design two corresponding modules in MDP, which are clinical feature weight adjustment and attentive clinical feature aggregation.

Clinical Feature Weight Adjustment. Given the patient's current health condition $\mathbf{h}_t$ and demographic data $\mathbf{p}$, the weight adjustment system adjusts the importance of clinical features dynamically, maintaining those features that are highly relevant to the diagnosis codes, and compressing others that are less relevant. Next, we provide the details of adjusting clinical feature weights.

Given the clinical features $\mathbf{c}_t$, we learn the embedding $\mathbf{c}'_t \in \mathbb{R}^{d_2 \times T_t}$ with $\mathbf{c}'_t = \mathbf{W}_c \mathbf{c}_t + \mathbf{b}_c$, where d_2 is the dimension size, $\mathbf{W}_c \in \mathbb{R}^{d_2 \times N}$ and $\mathbf{b}_c \in \mathbb{R}^{d_2}$ are the parameters to be learned. The i-th column of $\mathbf{c}'_t$, $\mathbf{c}'_{t,i} \in \mathbb{R}^{d_2}$ contains the clinical information for the i-th time stamp in the t-th visit. The clinical embedding $\mathbf{c}'_{t,i} (1 \leq i \leq T_t)$ is fed into the GRU denoted as clinical GRU_α. The hidden state $\mathbf{g}_{t,i} \in \mathbb{R}^{g_2}$ is calculated recurrently by $\mathbf{g}_{t,i} = GRU_\alpha(\mathbf{c}'_{t,i}, \mathbf{g}_{t,i-1}, \Theta_\alpha)$, where g_2 is the dimension size, and Θ_α is the corresponding parameter.

In order to adjust the importance of clinical features, we need to learn the weights of all clinical features for every time stamp by considering both current health conditions and demographic data. In particular, we first learn a vector $\mathbf{k}$ that encodes the current health condition and demographic data. Given the demographic information $\mathbf{p} \in \{0, 1\}^r$, the embedding $\mathbf{p}' \in \mathbb{R}^{r_1}$ is calculated by $\mathbf{p}' = \mathbf{W}_p \mathbf{p} + \mathbf{b}_p$, where r_1 is the dimension size, $\mathbf{W}_p \in \mathbb{R}^{r_1 \times r}$ and $\mathbf{b}_p \in \mathbb{R}^{r_1}$ are the parameters to be learned. The diagnosis codes representation $\mathbf{h}_t$ contains the health condition. Then we concatenate $\mathbf{h}_t$ and $\mathbf{p}'$, and compute $\mathbf{k} \in \mathbb{R}^{g_2}$ using $\mathbf{k} = \mathbf{W}_o[\mathbf{h}_t \oplus \mathbf{p}']$, where g_2 is the dimension size and $\mathbf{W}_o \in \mathbb{R}^{g_2 \times (r_1 + g_1)}$ is the parameter to be learned.

For every time stamp i, we learn a vector $\boldsymbol{\alpha}_{t,i} \in \mathbb{R}^{g_2}$ with $\boldsymbol{\alpha}_{t,i} = tanh(\mathbf{k} \odot \mathbf{g}_{t,i})$, where $\odot$ is the Hadamard product. $\boldsymbol{\alpha}_{t,i}$ contains the correlation between the clinical features and the patient's health condition at the i-th time stamp. Then we map the correlation vector $\boldsymbol{\alpha}_{t,i} \in \mathbb{R}^{g_2}$ into the clinical embedding space via $\boldsymbol{\alpha}'_{t,i} = tanh(\mathbf{W}_\alpha \boldsymbol{\alpha}_{t,i} + \mathbf{b}_\alpha)$, where $\mathbf{W}_\alpha \in \mathbb{R}^{d_2 \times g_2}$ and $\mathbf{b}_\alpha \in \mathbb{R}^{d_2}$ are the parameters to be learned, and $\boldsymbol{\alpha}'_{t,i} \in \mathbb{R}^{d_2}$. Here $\boldsymbol{\alpha}'_{t,i}$ is regarded as the weights of clinical features at the i-th time stamp. We compute the weighted clinical embedding $\mathbf{c}^w_{t,i} \in \mathbb{R}^{d_2}$ by multiplying the original clinical embedding $\mathbf{c}'_{t,i}$ by its corresponding weights, which is $\mathbf{c}^w_{t,i} = \boldsymbol{\alpha}'_{t,i} \odot \mathbf{c}'_{t,i}$. In such a way, $\mathbf{c}^w_{t,i}$ assigns more weights to clinical features that are highly related with the patient's current health condition.

Attentive Clinical Feature Aggregation. After we obtain the weighted clinical embedding $\mathbf{c}^w_{t,i} (1 \leq i \leq T_i)$, we use another GRU, clinical GRU_β, to capture the dependencies among the multiple time stamps. The hidden state $\mathbf{q}_{t,i} \in \mathbb{R}^{g_3}$ is computed with $\mathbf{q}_{t,i} = GRU_\beta(\mathbf{c}^w_{t,i}, \mathbf{q}_{t,i-1}, \Theta_\beta)$, where g_3 is the dimension size, Θ_β is the parameter to be learned.

Because some length of the ICU stay may be extremely long, GRU may fail to learn long-range dependencies. Instead of using the hidden state at the last time stamp $\mathbf{q}_{t,T_t}$ as the clinical feature representation, we import the attentive clinical feature aggregation mechanism which combines the hidden states among all time stamps. For time stamp i, $1 \leq i \leq T_i$, the aggregation mechanism computes the corresponding attention weight β'_i, which is a scalar, and then calculates the weighted sum of the hidden states as the final clinical feature representation $\mathbf{q}'_t \in \mathbb{R}^{g_3}$. The attention weight β'_i is computed as follows:

$$\beta_i = \mathbf{w}_\beta^\top \mathbf{q}_{t,i} + b_\beta, \qquad \beta'_1, \beta'_2, \cdots, \beta'_{T_t} = softmax(\beta_1, \beta_2, \cdots, \beta_{T_t}) \tag{1}$$

where $\mathbf{w}_\beta \in \mathbb{R}^{g_3}$ and $b_\beta \in \mathbb{R}$ are the parameters to be learned, β'_i and β_i are scalars. Finally, the clinical feature representation $\mathbf{q}_t$ is calculated by $\mathbf{q}'_t = \sum_{i=1}^{T_t} \beta'_i \mathbf{q}_{t,i}$.

Joint Optimization

We concatenate the diagnosis codes representation $\mathbf{h}_t$, the demographic embedding $\mathbf{p}'$, and the clinical feature representation $\mathbf{q}_t$ to obtain the vector $\mathbf{s}_t$ that encodes the overall patient's health status $\mathbf{s}_t = [\mathbf{h}_t \oplus \mathbf{p}' \oplus \mathbf{q}_t]$. $\mathbf{s}_t$ is then fed into a softmax layer to predict the diagnosis codes in the next visit, which is denoted as $\hat{\mathbf{y}}_{t+1} = Softmax(\mathbf{W}_s \mathbf{s}_t + \mathbf{b}_s)$, where $\mathbf{W}_s \in \mathbb{R}^{|\mathcal{D}| \times (g_1 + r' + g_3)}$ and $\mathbf{b}_s \in \mathbb{R}^{|\mathcal{D}|}$ are the parameters to be learned.

Table 1: Statistics of dataset

Statistics	MIMIC-III
Number of patients	5,033
Number of visits	13,096
Average number of visits per patient	2.60
Number of unique diagnosis codes	4,093
Average number of diagnosis codes per visit	13.10
Maximum number of diagnosis codes per visit	39
Number of unique CCS group codes	476
Average number of CCS group codes per visit	11.59
Maximum number of CCS group codes per visit	34
Number of clinical features	17
Average number of hours per visit	164.40
Maximum number of hours per visit	500

We compute the cross entropy loss between the ground truth $\mathbf{y}_t$ and the predicted $\hat{\mathbf{y}}_t$ using

$$\mathcal{L} = -\frac{1}{T-1}\sum_{t=1}^{T=1}(\mathbf{y}_t^T log(\hat{\mathbf{y}}_t) + (1-\mathbf{y}_t)^T log(1-\hat{\mathbf{y}}_t)). \tag{2}$$

This loss is calculated for a certain patient, and we compute the loss of all patients by averaging $\mathcal{L}$.

Experiments

In this section, we first introduce the experimental settings and then demonstrate the performance of the proposed algorithm on a public EHR dataset MIMIC-III. Moreover, We analyze the importance of clinical feature and visualize the clinical feature importance of three patients with different diseases. Finally, a case study is conducted to illustrate that MDP is able to correctly predict more diagnosis codes comparing with baselines.

Experimental Settings

Dataset. MIMIC-III[11] is a publicly available EHR dataset, which consists the admission records of ICU patients over 11 years. EHR data include diagnosis information such as diagnosis codes and procedure codes, clinical features such as vital signs and lab tests results, and clinical notes charted by care providers. We select the patients who made at least two visits. Following the previous work[12], we only select patients that are older than 18 and have single ICU stay per admission. Table 1 shows the details about the dataset. When performing the diagnosis prediction task, instead of predicting diagnosis categories like most of the previous research[3,4,6,13], we aim to predict the real diagnosis codes. This means that our task is more difficult since the target space is much larger.

Baselines. We select seven baselines that can be divided into three groups. Group 1 contains models that do not use the disease taxonomy information, which includes RNN and Dipole[2]. Group 2 contains models that use the disease taxonomy information, and these models only use patients' historical diagnosis codes as input, including GRAM[3], KAME[4], and HAP[5]. Models in Group 3 use the disease taxonomy information and use the multimodal data as input, including CAMP[13] and MHM[6].

Implementation Details. Following previous work[12], we extract 17 clinical features, including heart rate, mean blood pressure, glucose and other vital signs. For every clinical feature, we assign a binary variable indicating whether the clinical feature was observed at the current time stamp, and we fill in the missing clinical features with the values in the previous time stamp. For ICU stays with long length, we only use the records in the first 500 hours. The dimension size N of the clinical feature is set to 76. The demographic data is preprocessed as the method used in CAMP[13], and the dimension size r is 11, which contains 2 genders, 5 age groups and 4 admission types. The clinical embedding size d_2 is set to 76, and d_1, g_1, g_2 and g_3 are set to 128.

Table 2: Performance comparison on MIMIC-III

Method	Recall@K			MAP@K		
	$K = 20$	$K = 40$	$K = 60$	$K = 20$	$K = 40$	$K = 60$
RNN	0.350 ± 0.004	0.459 ± 0.005	0.529 ± 0.004	0.213 ± 0.003	0.238 ± 0.003	0.249 ± 0.003
Dipole	0.359 ± 0.003	0.474 ± 0.002	0.546 ± 0.003	0.215 ± 0.003	0.242 ± 0.003	0.253 ± 0.003
GRAM	0.355 ± 0.002	0.474 ± 0.002	0.547 ± 0.002	0.210 ± 0.003	0.237 ± 0.003	0.248 ± 0.003
KAME	0.363 ± 0.003	0.481 ± 0.003	0.554 ± 0.003	0.220 ± 0.003	0.246 ± 0.003	0.258 ± 0.003
HAP	0.370 ± 0.003	0.485 ± 0.003	0.555 ± 0.003	0.225 ± 0.001	0.252 ± 0.001	0.263 ± 0.001
CAMP	0.372 ± 0.002	0.490 ± 0.003	0.565 ± 0.002	0.225 ± 0.002	0.253 ± 0.002	0.265 ± 0.002
MHM	0.371 ± 0.003	0.487 ± 0.003	0.559 ± 0.003	0.224 ± 0.002	0.251 ± 0.003	0.263 ± 0.003
MDP	$\mathbf{0.383 \pm 0.003}$	$\mathbf{0.501 \pm 0.002}$	$\mathbf{0.572 \pm 0.002}$	$\mathbf{0.237 \pm 0.001}$	$\mathbf{0.265 \pm 0.001}$	$\mathbf{0.277 \pm 0.002}$

We randomly split the datasets into the training, validation and testing sets based on the number of patients in a 0.75:0.1:0.15 ratio. For fair comparison, all models are implemented with Pytorch, and we use the Adam optimizer with learning rate 0.001 and weight decay 0.001. For all models, the dimension of the diagnosis code embedding d_1 is set to 128, and the corresponding hidden state size g_1 is set to 128. We run every experiment ten times, and the average values and standard deviations are reported.

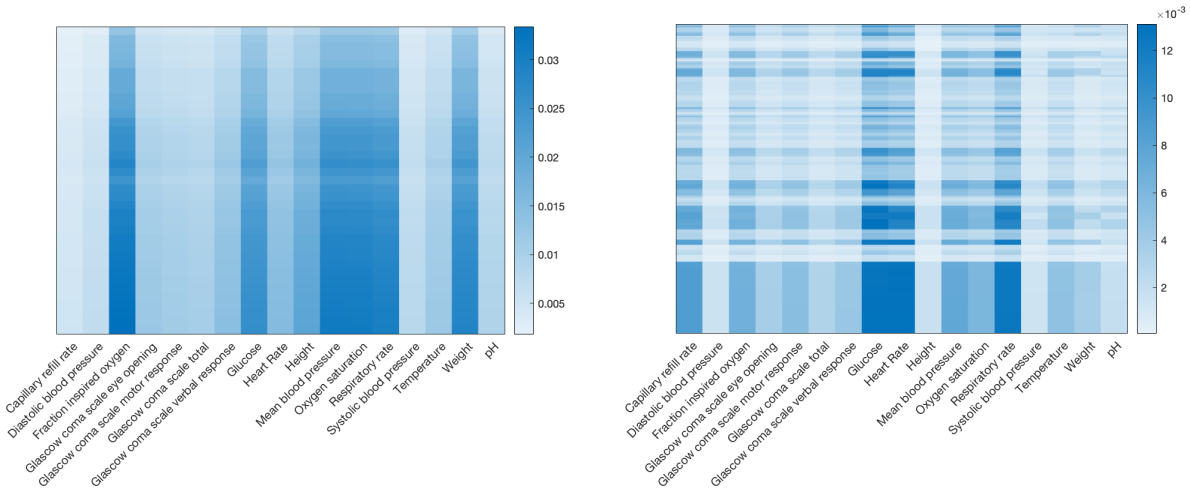
Evaluation Metric. Following CAMP[13], we use Recall@K and MAP@K as the evaluation criteria. For every visit V_t, we get a 1 if the target diagnosis code appears in the top k predictions and 0 otherwise. Recall@K is defined as the number of diagnosis codes that are predicted correctly in the top k of $\hat{\mathbf{y}}_t$ divided by the total number of diagnosis code in V_t. MAP@K refers to mean average precision, and it considers not only the precision and accuracy but also the order of diagnosis codes which are predicted correctly. We vary K from 20 to 60.

Performance Comparison

Table 2 shows the performance of MDP comparing with seven baselines. The proposed model MDP outperforms all baselines and achieves 1.2% higher Recall@5 and MAP@5 over the best baseline. This demonstrates the effectiveness of the clinical features in diagnosis prediction task. Meanwhile, MDP is able to adjust the weights of clinical features according to the patient's current health condition and the demographics. Among these baselines, methods in Group 1 do not use the disease taxonomy information, and they directly learn the diagnosis code embedding from the input data. The performance of RNN is worse than that of Dipole. It is because Dipole uses the bidirectional RNNs and apply the location-based attention mechanism when it makes predictions, which are able to capture visit dependency even for long sequences.

Three baselines in Group 2 all use the disease taxonomy information, and the inputs are the patient's diagnosis codes. The performance increases from GRAM to HAP. GRAM only updates the embedding of the leave nodes, and uses the diagnosis code representation to make predictions. KAME is built upon GRAM but learns a knowledge vector that contains the coarse-grained information of ancestor codes. KAME outperforms GRAM, and it suggests that general knowledge of the ancestors helps to represent the patient's health condition and further boosts the performance of the diagnosis prediction. However, both GRAM and KAME ignore the order among the ancestors. For example, in Figure 1, node $\mathbf{c_0}$, $\mathbf{c_1}$ and $\mathbf{c_2}$ are the ancestors of $\mathbf{c_3}$. When learning the diagnosis code embedding, GRAM and KAME treat $\mathbf{c_0}$, $\mathbf{c_1}$ and $\mathbf{c_2}$ equally, without considering their order. HAP fills the gap by designing a two-round attention propagation mechanism, and the embedding of diagnosis code is updated layer by layer. The performance of HAP exceeds GRAM and KAME, which indicates that using the full ontology hierarchy improves the models' expressibility.

Methods in Group 3 uses the multimodal data as input, and the performance of CAMP and MHM exceeds HAP. CAMP imports a memory network to save the fine-grained patient conditions, and the patient's demographic data cooperate with the READ/WRITE operations of the memory network. Instead of using the disease hierarchy to learn a more robust embedding, MHM uses the ontology hierarchy in a different way. It models the diagnosis prediction



(a) Patient 1 (b) Patient 2

Figure 2: Clinical Feature Importance Analysis.

task as a hierarchical multi-label classification problem, and clinical features and diagnosis codes are used as inputs. The performance of CAMP is better than MHM because the hierarchical multi-label model generated by MHM can not fully use the ontology hierarchy. The proposed model MDP uses the patient's diagnosis code, clinical features and demographic data as inputs, and the weights of clinical features are adjusted by the patient health condition and demographic data dynamically. The clinical feature representation learned by MDP is used as a complement of diagnosis code representation, and it contains plenty of details about the patient's symptoms. Comparing with the best baseline CAMP, MDP gets 1% mean improvement of Recall@K and 1.2% mean improvement of MAP@K.

Clinical Feature Importance Analysis

To demonstrate the benefit of clinical feature weight adjustment and attentive clinical feature aggregation mechanism in diagnosis prediction task, we define a criteria to evaluate the importance of clinical features for every time stamp. For the i-th time stamp in the t-th visit, the clinical feature importance $\boldsymbol{\omega}_i \in \mathbb{R}^N$ is calculated by $\boldsymbol{\omega}_i = \beta'_i \boldsymbol{\alpha}'_{t,i}$, where $\boldsymbol{\alpha}'_{t,i} \in \mathbb{R}^{d_2}$ is a vector learned in the clinical feature weight adjustment section, and β'_i is a scalar learned in attentive clinical feature aggregation section. Since $\boldsymbol{\omega}_i$ may contain negative numbers, we use the abs function to compute the absolute value, i.e., $\boldsymbol{\omega}_i = abs(\beta'_i \boldsymbol{\alpha}'_{t,i})$. Meanwhile, categorical features are encoded into multiple dimensions. In order to fairly evaluate the contributions of all features, we compute the average as the final importance. We concatenate $\boldsymbol{\omega}_i$ for $1 \leq i \leq T_i$ and obtain the importance matrix of clinical features $\boldsymbol{\omega} \in \mathbb{R}^{T_t \times N}$.

Table 3: Diagnosis Codes for Patients in Weight Analysis

Patient ID	Diagnosis Codes
1	Symptoms involving respiratory system and other lung symptoms(518.81, 482.42, 714.81, 519.3), Anemia(285.9), Hypothyroidism(244.9), Hypotension(458.9)
2	Symptoms involving heart disease(414.01, V45.82, V45.81, E87.90, 410.31), Hypertension(401.9) Symptoms involving cerebrovascular disease(438.6, 438.20), Diabetes(250.00)

Figure 2 shows the heatmap of the clinical feature importance matrix for two patients in a visit. In Figure 2, the x axis is the clinical feature, and the y axis is the time stamp. The color represents the feature importance, the darker color means the more feature importance. We first analyze the importance of different clinical features. Table 3 lists the corresponding diagnosis codes in the visit. We can observe that patient 1 mainly suffered *respiratory diseases*, and the heatmap shows that clinical features related with respiratory system obtain more weights. For example, fraction

inspired oxygen, oxygen saturation and respiratory rate have greater weights. Meanwhile, the patient also suffered *anemia* and *hypothyroidism*, and both diseases can result in hypoglycemia. We can observe that glucose also has higher weight. Patient 2 suffered *heart diseases* and *cerebrovascular diseases*, and MDP assigns high weights to respiratory rate and heart rate. Besides, this patient also suffered *diabetes*, and glucose is also highly weighted. The above observation shows that clinical feature weight adjustment mechanism can adjust the weights of clinical features based on the patient's health condition.

We then analyze the importance of different time stamps. We observe that the importance increases as time goes by. In general, MDP assigns more weights to the recent time stamps. It is because clinical features in the latter time stamps can better represent the patient's symptoms in the current visit, and these time stamps are more helpful to predict the diagnosis codes in the next visit. For patient 2, intermediate time stamps are assigned lower weights comparing with the before and after time stamps. This is because there are lots of missing clinical features in these intermediate time stamps. The above observation shows that the attentive clinical feature aggregation mechanism can assign different weights to different time stamps.

In conclusion, the heatmap of clinical feature importance illustrates that both clinical feature weight adjustment mechanism and the attentive clinical feature aggregation mechanism can coordinate to adjust the importance of clinical features and time stamps, which further helps to learn the clinical feature representations.

Case Study

Table 4 shows the correctly predicted diagnosis codes of another two patients when $K = 20$. For patient 1, the ground truth contains 14 diagnosis codes. The proposed model MDP correctly predicts 7 diagnosis codes in the top 20 predictions. Dipole achieves the same performance, while other baselines correctly predicted 4-5 diagnosis codes. We can observe that patient 1 mainly suffered from heart diseases, MDP correctly predicts not only the diagnosis codes directly related to heart disease but also the ones describing the details, such as V45.82, which is percutaneous transluminal coronary angioplasty status. This illustrates that the clinical feature can provide more details about the disease and is helpful to make the prediction. Similarly, patient 2 has 5 diagnosis codes, and MDP correctly predicts 3 diagnosis codes in the top 20 predictions, while most baselines only predict 1-2 diagnosis codes. These results show the superiority of the proposed MDP.

Table 4: Comparison of Correctly Predicted Diagnosis Code @20

Method	Predicted Diagnosis Code	
	Patient 1	Patient 2
Ground Truth	250.63, 414.01, V45.82, 536.3, 585.6, 790.7, 285.21, 428.0, 041.19, 337.1, 428.22, 403.01, 999.31, 414.11	276.8, 518.81, 070.70, 291.81, 070.30
RNN	428.0, 285.21, 585.6	518.81
Dipole	536.3, 428.0, 403.01, 585.6, 414.01, 250.63, 285.21	291.81, 070.70
GRAM	585.6, 285.21, 403.01, 536.3	291.81, 070.70
KAME	585.6, 536.3, 414.01, 250.63, 285.21	518.81, 291.81
HAP	536.3, 428.0, 250.63, 585.6, 414.01	070.70, 291.81, 518.81
CAMP	585.6, 285.21, 536.3, 428.0, 250.63	070.70, 291.81
MHM	414.01, 536.3, 585.6, 403.01	070.70, 518.81
MDP	536.3, 414.01, V45.82, 428.0, 250.63, 585.6, 285.21	291.81, 518.81, 070.70

Related Work

Diagnosis prediction, which aims to predict the patient's future health condition based on their historical EHRs, is an important task in health informatics, and thus has been widely studied. Most of the previous studiesinvestigate how to effectively utilize the patient's historical diagnosis codes for the prediction. Some representative methods are

discussed below. RETAIN[1] applies an RNN with reverse time ordered EHR sequences, and designs a two-level neural attention model to provide detailed interpretation of the prediction results. Dipole[2] designs three different attention mechanisms to learn the diagnosis code representations, and feeds the representations into a bidirectional GRU. The hidden state of the GRU is used to predict the potential diagnosis codes in the next visit. Note that diagnosis codes and medical concepts are naturally organized in a hierarchy, and the ancestor node information plays an important role in the prediction of the diagnosis code corresponding to the descendant node. Therefore, several studies exploit this hierarchical structure in diagnosis prediction. GRAM[3] incorporates a graph-based attention mechanism, and computes the embedding of the diagnosis code as the weighted sum of the basic embedding of itself and its ancestors. KAME[4] learns the ancestor representation and concatenates it with the diagnosis code representation. The ancestor representation contains the coarse-grained health condition. HAP[5] proposes a hierarchical attention mechanism in which the attention propagates across the entire hierarchy from layer to layer. The embedding of the node absorbs knowledge from not only its ancestors, but also its descendants, siblings and even some distant nodes. Different from the aforementioned methods that rely on visit sequences of diagnosis codes and their relations only, the proposed MDP model incorporates additional valuable information including patient demographics and clinical features to boost the performance of diagnosis prediction.

Recently, researchers also explore the use of multimodal data to perform the diagnosis prediction task. CAMP[13] incorporates patients' demographics information in the prediction model. Specifically, it imports an external memory network which saves the fine-grained health condition for every top level categories in the disease taxonomy, and combines the diagnosis codes information with the patient's demographics to help the READ/WRITE operation of the memory network. Clinical features, which also encodes predictive information for diagnosis prediction and serves as an input to the proposed MDP model, are not considered by the CAMP model. MHM[6] uses the diagnosis codes and clinical features as input. MHM models the diagnosis prediction task as a hierarchical multi-label classification problem, and the disease taxonomy is used to generate the hierarchical label. MHM learns a representation for every layer of the disease taxonomy, and the global representation is the weighted sum of the layer representation. However, MHM neglects the relation between the diagnosis codes and clinical features and ignores the variance in patient groups defined by demographics. The proposed model MDP overcomes this shortcoming by importing a clinical feature weight adjustment mechanism. By this mechanism, the importance of clinical features is adjusted according to the health condition and the demographics.

The proposed MDP and the aforementioned models all aim to tackle the diagnosis prediction task. Existing models utilize one or two sources of information, while the proposed model takes advantage of all the relevant information including diagnosis code sequences, hierarchy, clinical features and patient demographics. Another relevant research topic is the analysis of clinical features, but note that the task tackled by these methods is not diagnosis prediction. Concare[14] learns the inter-dependencies among clinical features, and it improves the multi-head self-attention via the cross-head decorrelation.

Conclusions

In this paper, we propose a novel model MDP to perform the diagnosis prediction task. MDP takes the diagnosis codes, clinical features and demographics as input and consists of integral components that effectively integrates these heterogeneous information sources. The diagnosis code encoder in MDP utilizes the disease taxonomy to learn the diagnosis code representations which capture the patient's current health condition. The clinical feature encoder learns the clinical feature representations by incorporating a weight adjustment mechanism and an attentive clinical feature aggregation mechanism. The weight adjustment mechanism adjusts the weights of clinical features based on the diagnosis information and demographics, and the attentive clinical feature aggregation mechanism enables the capture of the long term dependencies among the patient's conditions at multiple time stamps during an ICU stay. By integrating these components, the proposed MDP is able to extract meaningful signals that are relevant to diagnosis prediction from any available source. Experimental results on a real-world EHR dataset show the effectiveness of MDP for diagnosis prediction. To analyze the insights behind the proposed framework, we further demonstrate the assigned weights to various clinical features and time stamps via the clinical feature encoder as well as a comparison with baselines based on the correctly predicted diagnosis codes in the top 20 predictions for some case studies. These results explain how the proposed MDP achieves the superior performance.

Acknowledgements

This work is sponsored by NSF-IIS 1553411. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

1. Choi E, Bahadori MT, Sun J, Kulas J, Schuetz A, Stewart W. RETAIN: An Interpretable Predictive Model for Healthcare using Reverse Time Attention Mechanism. In: Lee D, Sugiyama M, Luxburg U, Guyon I, Garnett R, editors. Advances in Neural Information Processing Systems. vol. 29. Curran Associates, Inc.; 2016. .
2. Ma F, Chitta R, Zhou J, You Q, Sun T, Gao J. Dipole: Diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining; 2017. p. 1903–1911.
3. Choi E, Bahadori MT, Song L, Stewart WF, Sun J. GRAM: graph-based attention model for healthcare representation learning. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining; 2017. p. 787–795.
4. Ma F, You Q, Xiao H, Chitta R, Zhou J, Gao J. Kame: Knowledge-based attention model for diagnosis prediction in healthcare. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management; 2018. p. 743–752.
5. Zhang M, King CR, Avidan M, Chen Y. Hierarchical Attention Propagation for Healthcare Representation Learning. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining; 2020. p. 249–256.
6. Qiao Z, Zhang Z, Wu X, Ge S, Fan W. MHM: Multi-modal Clinical Data based Hierarchical Multi-label Diagnosis Prediction. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval; 2020. p. 1841–1844.
7. Chung J, Gulcehre C, Cho K, Bengio Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. NeurIPS 2014 Workshop on Deep Learning. 2014.
8. Choi E, Bahadori MT, Schuetz A, Stewart WF, Sun J. Doctor ai: Predicting clinical events via recurrent neural networks. In: Machine learning for healthcare conference. PMLR; 2016. p. 301–318.
9. Choi E, Schuetz A, Stewart WF, Sun J. Using recurrent neural network models for early detection of heart failure onset. Journal of the American Medical Informatics Association. 2017;24(2):361–370.
10. Hochreiter S, Schmidhuber J. Long short-term memory. Neural computation. 1997;9(8):1735–1780.
11. Johnson AE, Pollard TJ, Shen L, Li-Wei HL, Feng M, Ghassemi M, et al. MIMIC-III, a freely accessible critical care database. Scientific data. 2016;3(1):1–9.
12. Harutyunyan H, Khachatrian H, Kale DC, Ver Steeg G, Galstyan A. Multitask learning and benchmarking with clinical time series data. Scientific data. 2019;6(1):1–18.
13. Gao J, Wang X, Wang Y, Yang Z, Gao J, Wang J, et al. Camp: Co-attention memory networks for diagnosis prediction in healthcare. In: 2019 IEEE International Conference on Data Mining (ICDM). IEEE; 2019. p. 1036–1041.
14. Ma L, Zhang C, Wang Y, Ruan W, Wang J, Tang W, et al. Concare: Personalized clinical feature embedding via capturing the healthcare context. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34; 2020. p. 833–840.