Doctor AI: Predicting Clinical Events via Recurrent Neural Networks

BMI Journal Club

Supreeth

7 June 2017

Overview

1. Quick Summary of paper
2. EHR to physician diagnoses
3. Cohort
4. Methods
5. Results

Problem: Massive amounts of Electronic Health Records (EHR) are available today, physicians have little time/tools to analyze them. Predict outcomes based on longitudinal time stamped EHR

Novelty: Use RNN to predict the diagnosis and medication categories for a subsequent visit. Multi-label prediction

Results: The trained RNN is able to achieve 79% Recall@30 for diagnosis prediction. Prove potential of RNNs in transfer learning

Overview

1. Quick Summary of paper
2. EHR to physician diagnoses
3. Cohort
4. Methods
5. Results

Electronic Health Records

….. contain patient's medical history, medications, diagnoses, treatment plans, laboratory and test results, etc

EHR represent the longitudinal experience of both patients and doctors. And are being used with increasing frequency to predict future events

Doctor AI

Is interested in whether historical EHR data can be used to predict future physician diagnoses and medication orders

As a secondary goal, it predicts the time to the patients' next visit

Leverages the power of RNNs for sequential modelling

Overview

1. Quick Summary of paper
2. EHR to physician diagnoses
3. Cohort
4. Methods
5. Results

Population and source of data

The source population for the study were primary care patients from Sutter Health Palo Alto Medical Foundation

Dataset was extracted from a density sampled case-control study for heart failure

Dataset consists of Enounter orders, medication orders, problem list records and procedure orders

Data processing

ICD-9 codes were extracted from the records

Generic Product Identifier (GPI) medication codes and CPT procedure codes were extracted

Excluded patients that made less than two visits

Grouping Medical Codes

More than 11,000 Unique ICD-9 codes and 18,000 GPI medication codes in the dataset

Pumonary tuberculosis (ICD-9 code 011) has 70 subcategories (ICD-9 code 011.01, … 011.96)

For diagnoses codes, 3-digit ICD-9 codes are used – 1183 unique codes

For medication codes, GPI drug Class is used – 595 unique groups

Table 1: Basic statistics of the the clinical records dataset.

# of patients	263,706	Total # of codes	38,594
Avg. # of visits	54.61	Total # of 3-digit Dx codes	1,183
Avg. # of codes per visit	3.22	# of top level Rx codes	595
Max # of codes per visit	62	Avg. duration between visits	76.12 days

Label Y_i at each time step is a 1,778- dimensional vector (i.e. 1183+595) for the grouped diagnoses codes and medication codes

Overview

1. Quick Summary of paper
2. EHR to physician diagnoses
3. Cohort
4. Methods
5. Results

Problem setting

For each patient, the observations are (t_i, x_i) for $i = 1,....n$

Each pair represents an event, during which multiple medical codes such as ICD-9 diagnosis codes, procedure codes or medication codes are documented in the patient record

x_i is a multi-hot label vector $\in \{0,1\}^p$

Gated Recurrent Units Prelims.

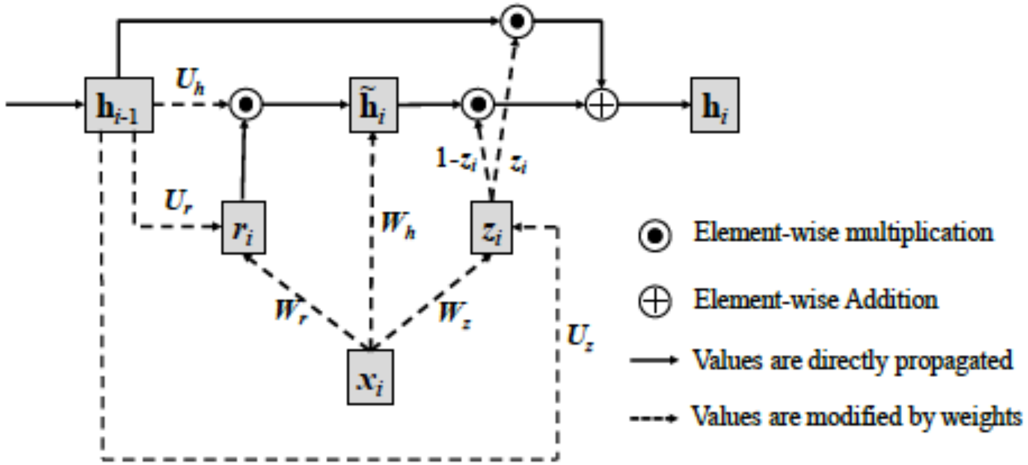


Figure 4: Architecture of GRU

We first reiterate the mathematical formulation of GRU so that the reader can see Figure 4 and the formulations together.

$$
\begin{aligned}
\boldsymbol{z}_i &= \sigma(\boldsymbol{W}_z\boldsymbol{x}_i + \boldsymbol{U}_z\boldsymbol{h}_{i-1} + \boldsymbol{b}_z) \\
\boldsymbol{r}_i &= \sigma(\boldsymbol{W}_r\boldsymbol{x}_i + \boldsymbol{U}_r\boldsymbol{h}_{i-1} + \boldsymbol{b}_r) \\
\tilde{\boldsymbol{h}}_i &= \tanh(\boldsymbol{W}_h\boldsymbol{x}_i + \boldsymbol{r}_i \circ \boldsymbol{U}_h\boldsymbol{h}_{i-1} + \boldsymbol{b}_h) \\
\boldsymbol{h}_i &= \boldsymbol{z}_i \circ \boldsymbol{h}_{i-1} + (1 - \boldsymbol{z}_i) \circ \tilde{\boldsymbol{h}}_i
\end{aligned}
$$

Neural Network Architecture

Goal is to learn effective vector representation for the patient status at each t_i

Predict the diagnosis and medication categories in the next visit $\boldsymbol{Y_{i+1}}$ and the time duration until the next visit $d_{i+1} = t_{i+1} - t_i$

Softmax layer is used to predict the diagnosis and the medication codes, and a rectified linear unit to predict the time duration until next week.

Neural Network Architecture

Softmax layer stacked on top of the GRU

$Y'_{i+1} = \mathrm{softmax}(W_{code}{}^{T} h_i + b_{code})$

Predicting the time duration until next visit

$d_{i+1} = \max(W_{time}{}^{T} h_i + b_{time}, 0)$

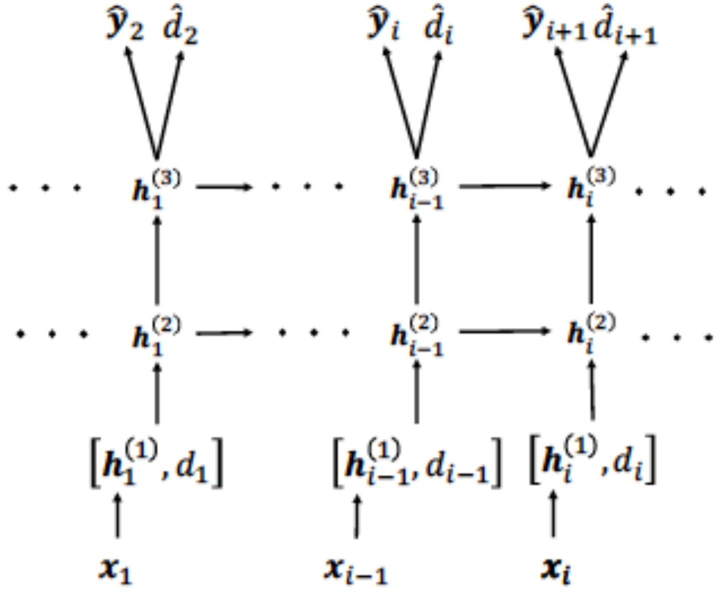
Values of all W's and U's are initialized to orthonormal matrices using singular value decomposition of matrices

Loss function

$$\mathcal{L}(\boldsymbol{W}, \boldsymbol{U}, \boldsymbol{b}, \boldsymbol{w}_{time}, b_{time}) = \sum_{i=1}^{n-1} \left\{ \Big(\boldsymbol{y}_{i+1} \log(\widehat{\boldsymbol{y}}_{i+1}) + (1 - \boldsymbol{y}_{i+1}) \log(1 - \widehat{\boldsymbol{y}}_{i+1}) \Big) + \frac{1}{2} \| d_{i+1} - \widehat{d}_{i+1} \|_2^2 \right\}$$

Neural Network Architecture

Figure 1: This diagram shows how we have applied RNNs to solve the problem of forecasting of next visits' time and the codes assigned during each visit. The first layer simply embeds the high-dimensional input vectors in a lower dimensional space. The next layers are the recurrent units (here two layers), which learn the status of the patient at each timestamp as a real-valued vector. Given the status vector, we use two dense layers to generate the codes observed in the next timestamp and the duration until next visit.



Overview

1. Quick Summary of paper
2. EHR to physician diagnoses
3. Cohort
4. Methods
5. Results

Experiment setup

85% of the patients as the training set and 15% as the test set

RNN models are trained for 20 epochs

Regularization: Drop-out and L2

Size of the hidden layer h_i set to 2000

Experiment setup

Four different variations of Doctor AI:

- **RNN-1**: RNN with a single hidden layer initialized with a random matrix for W_{emb}
- **RNN-2:** RNN with two hidden layers initialized with a random matrix for W_{emb}
- **RNN-1-IR:** RNN with a single hidden layer initialized with a pre-trained W_{emb}
- **RNN-2-IR:** RNN with two hidden layers initialized with a pre-trained W_{emb}

Prediction Performance

Results are reported in three settings

- Predicting only diagnosis codes (Dx)
- Predicting only medication codes (Rx)
- Predicting Dx codes, Rx codes, and time duration to next visit

Top-k recall:

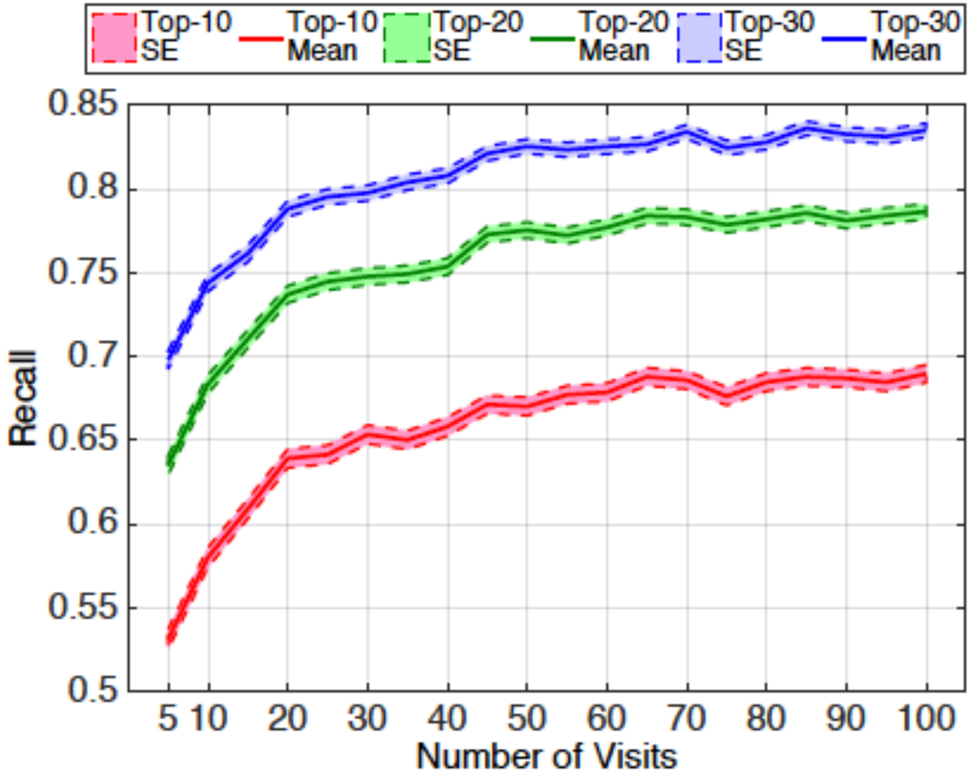
$$\text{top-}k \text{ recall} = \frac{\#\text{ of true positives in the top } k \text{ predictions}}{\#\text{ of true positives}}$$

Prediction Performance

Algorithms	Dx Only Recall @k			Rx Only Recall @k			Dx,Rx,Time Recall @k			
	$k=10$	$k=20$	$k=30$	$k=10$	$k=20$	$k=30$	$k=10$	$k=20$	$k=30$	R^2
Last visit	29.17			13.81			26.25			—
Most freq.	56.63	67.39	71.68	62.99	69.02	70.07	48.11	60.23	66.00	—
Logistic	43.24	54.04	60.76	45.80	60.02	68.93	36.04	46.32	52.53	0.0726
MLP	46.66	57.38	64.03	47.62	61.72	70.92	38.82	49.09	55.74	0.1221
RNN-1	63.12	73.11	78.49	67.99	79.55	**85.53**	53.86	65.10	71.24	0.2519
RNN-2	63.32	73.32	78.71	67.87	79.47	85.43	53.61	64.93	71.14	0.2528
RNN-1-IR	63.24	73.33	78.73	**68.31**	**79.77**	85.52	54.37	65.68	71.85	0.2492
RNN-2-IR	**64.30**	**74.31**	**79.58**	68.16	79.74	85.48	**54.96**	**66.31**	**72.48**	**0.2534**

Understanding the behavior of the network

Used the best performing model to predict the diagnosis codes at visits at different times



Transfer Learning

A different dataset is used – MIMIC II

2,695 patients available. And 767 unique diagnosis codes

Two experiments are performed:

- Trained model only on the MIMIC II dataset
- Initialized the coefficients of model from model trained on Sutter data

Transfer Learning

Figure 3: The impact of pre-training on improving the performance on smaller datasets. In the first experiment, we first train the model on a small dataset (red curve). In the second experiment, we pre-train the model on our large dataset and use it for initializing the training of the smaller dataset. This procedure results in more than 10% improvement in the performance.