A Reproduction and Analysis of the INPREM Model for Healthcare

Brayden Turner and Joshua M. Smith
{brturne2, jms28}@illinois.edu

Group ID: 85, Paper ID: 159
Presentation link: https://youtu.be/8teJ8AcxOyM
Code link: https://github.com/braydenturner/DL4H_CourseProject

1 Introduction

The paper "INPREM: An Interpretable and Trustworthy Predictive Model for Healthcare"(Zhang et al., 2020) addresses the general problem of building predictive multi-label classification models based on Electronic Health Record (EHR) data for personalization of healthcare. Current approaches to applying deep learning in clinical prediction, i.e. predicting the medical code for the next clinical visit, lack transparency in both their interpretability and their trustworthiness. Interpretability requires the ability for medical experts to be able to identify the contributions of specific medical events to the overall prediction, and models such as RETAIN have been developed to help address this. Trustworthiness requires the models to represent their level of uncertainty about their predictions,an issue that has been unaddressed. INPREM addresses both interpretability and trustworthiness in its design, and the paper puts forth a case study to demonstrate its effectiveness.

2 Scope of reproducibility

2.1 Addressed claims from the original paper

- INPREM will be a robust machine learning model for Encounter Sequence Modeling (ESM) that can additonally output probabilities of health condition (ICD-9 code) predictions.
- INPREM will outperform existing approaches such as RETAIN for predicting medical codes by providing predictions with higher accuracy.
- INPREM will predict a binary classification for disease risk prediction, identifying if the patient is at risk of heart failure, diabetes, or chronic kidney disease (CKD). We do not have the data to replicate this claim.

3 Methodology

3.1 Model descriptions

INPREM is designed as a linear model for interpretability with non-linear relationships, that is later extended to a Bayesian Neural Network to address trustworthiness. First, it encodes its non-linear relationships into learning weights which represent dependencies between and within visits. These are then implemented with two attention modules: a sparse visit attention module for modeling dependencies between visits and a variable model for modeling dependencies within visits. While this addresses interpretability, it does not yet address trustworthiness. For this, INPREM sets a random gate with a Bernoulli distribution, blocking each weight of the network to aid in uncertainty estimation. The model also leverages an additional branch at the end of the pipeline to eliminate the noise of each data point and address aleatoric uncertainty.

For a model performance baseline, we compare the INPREM model against a RETAIN model, as was done in the paper.

3.1.1 Interpretability Framework

INPREM first constructs two embedding layers to (1) learn the representation of medical codes within each visit, and (2) encode the order information of each visit. These are then joined with a linear mapping along with an α (non-linear dependencies between visits) and β (non-linear dependencies within visits) attention modules, creating a patient-level representation of each visit's medical code and order in time. The logit for prediction is calculated by multiplication of the learning parameters and the patient representation with a bias. Finally, a $Softmax$ function is applied to estimate probability for prediction.

From this result, for interpretability findings,

we produce a contribution matrix CM, where $CM[i, j][k]$ is the contribution of the j-th medical event in the i-th visit, when the predicted class is k.

The attention components α and β are formed via stacked multi-head attention. Each self-attention layer, representing each visit, is fed with a set of key-query pairs and their values. These are then concatenated in parallel to form the multi-head attention, fully connected layers. The α layer is produced by averaging a $Sparsemax$ and $Softmax$ function, while the β module is produced with the $\tanh()$ function.

3.1.2 Trustworthiness Framework

Trustworthiness in the INPREM model is implemented in two methods, (1) to address epistemic uncertainty by extending the linear model to a Bayesian Neural Network (BNN) and (2) to address aleatoric uncertainty by building in robustness to input data noise. INPREM uses a Bernoulli distribution to place random gates (0 or 1, on or off) over weights in the model, in order to approximate variational distributions and capture epistemic uncertainty. To address aleatoric uncertainty, INPREM uses a Gaussian distribution over the output logit, and the variance of the distribution is shared among the binary classifications of each data point.

3.2 Data descriptions

The MIMIC-III dataset was used for experimentation with the INPREM model. The dataset consists of 7499 Intensive Care Unit (ICU) patients and their ICD-9 diagnosis codes, with the objective to predict the ICD-9 codes for the following clinical visit. The second hierarchy of the ICD-9 codes is used as a category code to more easily reference in the data.

There was a second real-world dataset used for disease prediction. For this, the researchers identified three cohorts: Heart Failure, Diabetes, and Chronic Kidney Disease. with 5, 3, and 3 control patients respectively, with the goal of performing binary classification on the dataset for disease prediction. However, we do not have access to this dataset and cannot reproduce these results.

3.3 Hyperparameters

Many hyperparameters are given in the original paper and we used these hyperparameters wherever possible. The researchers used Adam with a batch size of 32 and a learning rate of 0.0005. They set weight decay to $\lambda = 0.0001$ with a dropout rate of 0.5 for each model. Dimensions for embedding and hidden states are also set in line with the paper, at 256. This number includes the baseline RETAIN as well as the INPREM model.

For stacked multi-head attention layer in the model, we use the paper's specified number of heads m of 2 and the number of times to stack S as 2.

3.4 Implementation

The code used in the original paper is unavailable. We used our own code, provided in the GitHub repository linked at the top of page 1. We process the dataset, construct a collate function for both the baseline and the INPREM model, and construct both RETAIN (as a baseline) and INPREM models. After that, we run an evaluation to check each model's performance. A Jupyter notebook is provided in the repository containing all of the code we used to reproduce the results and descriptions of each step.

3.5 Computational requirements

The INPREM researchers used a single Nvidia Titan Xp, which is a Pascal architecture GPU launched in 2017 containing 12GB of VRAM. Both students have access to a newer, Ampere-based Nvidia RTX 3080 GPU, each with 10GB of VRAM, running on an Nvidia CUDA PyTorch Docker container. The RTX 3080 is roughly 40-50% faster than the Nvidia Titan Xp. The training took 60s per epoch on the Nvidia Titan Xp, and we estimated that it would take roughly 35-45s per epoch on each of our single RTX 3080 GPUs.

Testing and training the model did not present issues for our Nvidia GPUs, using on average about 60% of the available VRAM on the RTX 3080 graphics cards we tested on. Training took about 20-25 seconds per epoch to run, lower than initial estimates. We also took measurements of CPU utilization and RAM utilization while training, but we are training on powerful enough workstations (Intel i7-8700K, 32 GB of DDR4 RAM) that neither of these dimensions were significantly impacted.

4 Results

Accuracy metrics for diagnosis prediction are defined in another paper referenced by the original authors: "KAME: Knowledge-based attention model for diagnosis prediction in healthcare" (Ma et al.,

2018). They use two measurements for predicting medical diagnosis performance: the accuracy across all predicted codes and the accuracy at k codes. The researchers in the 2018 paper vary the value of k between 5 and 30.

The method for "ranking" was not outlined clearly in either paper, so we rank results by taking the difference between our y_{score} probability and y_{true} probability. For example, if we arrive at a y_{score} of 0.919 for a given medical code, and that code exists as a 1 in our y_{true} vector, we take the difference of 0.081 and sort the predicted codes by this difference. We then use this ranking to calculate the top k to use in accuracy calculations.

4.1 Result 1

Accuracy of diagnosis: prediction accuracy is the ratio of correct ICD-9 codes in the top ranked k over y_t, which is the number of medical codes in the prediction visit $(t+1)$, and $k = y_t$. We take the average of the accuracy across these visits.

We were unable to get the model evaluation to produce meaningful results after this ranking function. Accuracy and precision for the data repeatedly showed as NaN values. We investigated this thoroughly and cannot tell if the failure to produce a meaningful evaluation is in the way that we are setting up the data, the model reconstruction, the evaluation reconstruction, or the way that we are calculating the accuracy and ranking k results.

4.2 Result 2

Accuracy of diagnosis @ k: prediction accuracy at k is the ratio of the correct ICD-9 codes in k over the minimum of (k, y_t). We take the average of the accuracy across these visits.

Similar to above, accuracy and precision for the data repeatedly showed as NaN values using our ranking. We cannot tell if the failure to produce a meaningful evaluation is in the way that we are setting up the data, the model reconstruction, the evaluation reconstruction, or the way that we are calculating the accuracy and ranking k results.

4.3 Additional results not present in the original paper

We decided on a couple of ablations to test theories from the original paper.

First: variation of the learning rate between the original hyperparameter of 0.0005 to 0.0001 and 0.001. We didn't get meaningful training loss changes from altering the learning rate. This could be because we did not vary the learning rate enough, or it could be due to issues with the data setup, similar to the issues we saw with our ability to compute accuracy.

Secondly, variation of the number of epochs. The number of epochs was not specified in the original paper but we varied the number of epochs between 10 and 100. When we tested with a larger number of epochs, the difference in the training loss tapered off. We expect this result, but are not sure if there are issues in our data causing the training loss to taper off sooner than expected.

5 Discussion

Based on our experience, the paper was not easily reproducible, if reproducible at all. The paper alters between being broad about different aspects to incredibly specific about other aspects. This seems to be a focus on the novel, INPREM-specific aspects of their model, without enough context around the other papers they used when setting up the model and writing their code. If the code had been provided we might have gotten more context and been able to reproduce the paper much more easily.

In our opinion, this paper is actually quite a difficult paper to reproduce (see discussion on "What was difficult" below). The only area which we found easy was the computational

5.1 What was easy

The ability to use our knowledge and available code we had worked through on homework assignments for the course, such as RNN, RETAIN, and others, made it easier to replicate the baseline RETAIN model used in the paper.

The code for the model for diagnosis prediction also uses a standardized dataset, MIMIC-III. A standardized dataset makes it easier to evaluate given our experience in the course. The authors also gave detailed information on the hyperparameters for the model, which made some parts of the model easy (however, this did not make replication of the model easy overall, see below).

5.2 What was difficult

The authors did not provide code for the INPREM model or the discussed performance evaluations. Without code illustrating the implementation details, many aspects of the model and the evaluation were unclear, leading to a greatly increased difficulty in making sure we were on the right track as

we replicated the paper.

The paper was also unclear at times about the details surrounding the model, which made implementation much more difficult than we initially expected. There was a lack of clarity when the paper was discussing two different approaches for the model, specifically diagnosis prediction and risk prediction. Often, the authors switched back and forth between the two discussing different evaluation methods and different formulas for the model. As a result, we spent time implementing details for one model instead of the other. We think some clarity and separation would have benefitted the paper and made replication easier.

Implementing evaluations was incredibly difficult due to the paper's description of accuracy and precision calculations. Without specification of how probabilities are ranked, the term "top k" was not meaningful. We ended up formulating our own ranking and using that.

5.3 Recommendations for reproducibility

There were a few key aspects we found lacking for reproducability.

The first issue is the lack of provided code. Details such as calculation of the stacked multi-head attention were implemented by us with only the mathematical representation and the description of functions being used (e.g. $Sparsemax$).

The second is description of the evaluation, specifically the ranking of results to use to document accuracy and accuracy@k. The original paper referenced also did not specify this so it's unclear how the researchers determined their ranking metrics.

The third is access to the dataset for risk prediction. We had access to the MIMIC-III dataset for diagnosis prediction, but the results of the risk prediction are sourced from local hospitals and are not easily accessible. We understand that this is not an easy problem to solve, but perhaps the paper could include guidance on who to contact for access to the dataset.

The fourth is more clarity between the two different applications (diagnosis prediction and risk prediction). Since we only had data for one of the predictions, we made several mistakes in implementation which we had to go back and repair, which may have been avoided had there been more clarity.

6 Communication with original authors

We reached out to the original authors of the INPREM paper but they did not respond. We told them we were attempting to reproduce their paper, and requested the code, and a couple of clarifications. We reached out on two separate occasions, to two different authors, and got no response.

We will send a follow-up to them now that our project is complete with our recommendations for reproducability.

References

Fenglong Ma, Quanzeng You, Houping Xiao, Radha Chitta, Jing Zhou, and Jing Gao. 2018. Kame: Knowledge-based attention model for diagnosis prediction in healthcare. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, page 743–752, New York, NY, USA. Association for Computing Machinery.

Xianli Zhang, Buyue Qian, Shilei Cao, Yang Li, Hang Chen, Yefeng Zheng, and Ian Davidson. 2020. Inprem: An interpretable and trustworthy predictive model for healthcare. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*, KDD '20, page 450–460, New York, NY, USA. Association for Computing Machinery.