Reproducing SeqMed: Recommending Medication Combination with Sequence Generative Adversarial Nets

Michael Gambino
mgambin2@illinois.edu

Group ID: 212, Paper ID: 48
Presentation link: TODO
Code link: TODO

1 Introduction

The application of Medicine Combination Prediction (MCP) through electronic medical records (EMR) has fast become one of the more interesting applications of deep learning for healthcare. Traditionally, the complexity of effective MCP would scale in consideration with the physicians expertise, as well as the complications of the patients underlying ailments. With the application of deep learning, models are trained from EMR diagnosis, procedure, and medical codes to predict a medicine combination to improve the health and ultimately cure the patient of a certain disease.

Previous models created for MCP can broadly be generalized into two categories: multi-label binary classifiers (Choi et al., 2016; Shang et al., 2019b,a) or sequence predictors (Zhang et al., 2017; Wang et al., 2019). These previous works exhibit inefficiencies in either taking into account the longitude of EMR, failing to demonstrate drug synergism upon recommendation, or ability to converge. Here, we present SeqMed, a proposed alternative model (Wang, 2020) that utilizes Sequential Generative Adversarial Networks (GANs) for MCP. Sequential GANs, demonstrated by (Yu et al., 2017), employs the GAN architecture for reinforcement learning (RL), where the generator acts as the policy agent and uses the discriminator as the action-value function.

2 Scope of reproducibility

The usage of sequential GANs for MCP are reported to take into account drug synergism over data temporality whilst being able to converge. Therefore, the purpose of this study is to implement SeqMed as described in (Wang, 2020) and investigate the validity of those claims from the novel methodology. We can accomplish that through the following quantifiable studies. First, we assess drug synergism and predictive capabilities through evaluating the Jaccard, Recall, Precision, and f-1 score of the implemented model and comparing them against the reported metrics seen in Table II (Wang, 2020). Following that, we will assess the claim that SeqMed achieves faster convergence due to the use of the discriminator probability as RL reward and complete sequence for policy gradient by comparing our observed convergence point to that seen in other sequence based models (Zhang et al., 2017; Wang et al., 2019).

2.1 Addressed claims from the original paper

- Assess drug synergism and predictive capabilities through evaluating the Jaccard, Recall, Precision, and f-1 with 5% tolerance.
- Assess faster convergence due to the use of the discriminator probability as RL reward and complete sequence for policy gradient.

3 Methodology

For this reproducibility study, SeqMed will be implemented from scratch following the description of the architecture seen in (Wang, 2020). Features of SeqMed, in particular the original architecture of SeqGAN (Yu et al., 2017), will be used heavily as reference but will be modified due to differences in the policy gradient and addition of additional network modules. Data and pre-processing conditions used to evaluate SeqMed are identical to that used in (Shang et al., 2019b).

3.1 Model descriptions

The description for the framework composing of SeqMed can be adequately broken up into four main modules: The Clinical Information Extraction Module, Health Status Update Module, Generative model of SeqMed, and Discriminative model of SeqMed. Per each module, we will provide more

adequate details into model architecture and loss objectives.

3.1.1 Clinical Information Extraction Module

The Clinical Information Extraction Module looks to extract key clinical infromation from diagnosis and procedure code EMR data. This module roughly follows the flow of (Shang et al., 2019b) but has some differences in implementation. Individually, diagnosis and procedure codes are projected onto an embedding matrices. Then, each embedding matrix is respectively passed through a three-layer deep convolutional neural network. Output, after using with activation function, is then passed through a fully connect layer. Once done piece-wise for both diagnosis and procedure codes, both outputs are concatenated. The implemented Clinical Information Extraction Module has 221628 parameters.

3.1.2 Health Status Update Module

The Health Status Update Module is used to update the health status of the patient dynamically through some vector of predicted medicines. This is implemented roughly through an attention mechanism, in which every time new medicine is predicted from our generator, both the medicine vector and health status will be updated. The implemented Health Status Update Module has 379200 parameters.

3.1.3 Generative Model

The Generator used for SeqMed is basely built off of a Gate Recurrent Unit (GRU), but is additionally modified with the aforementioned Health Status Update Module. As mentioned above, the use of the Health Status Update Module is to dynamically update both the health status and medicine of the patient. Here, we apply constraints to not allow multiple medications to be predicted or added again. Loss is measured as a cumulative sum of all complete sequential medicine predictions. With this, we can also define our policy gradient for RL to be the generation of medicine sequences. Where SeqMed differs from (Yu et al., 2017) is that we do not utilize Monte Carlo search to go to the end of the sequence but rather evaluate a continually updated "complete" sequence, where additional medicines are added iteratively. The implemented Generator has 401943 parameters.

3.1.4 Discriminative Model

The Discriminator used for SeqMed as well utilizes a GRU. Input into our discriminator is the complete medication recommendation generated from the generator and the ground truth medication record. The loss used measures the cumulative total difference of each sequence predicted medication combination and that of the ground truth. Similarly, we can let our discriminator evaluate the predicted sequence as the action-value function for RL. With this, we let our reward for RL be the actual estimated probability. As mentioned prior for the generative model, we do not allow the generator to use Monte Carlo search, therefore we only allow our reward to be the prediction of the entire sequence with continual update of the medicine vector. The implemented Discriminator has 388673 parameters.

3.2 Data descriptions

The study for SeqMed utilized EMR data from MIMIC-III (Johnson et al., 2016) and follows preprocessing conditions of (Shang et al., 2019b), in which patients selected have more than one visit between 2001-2012 and only the set of medications prescribed by doctors during the first 24-hour. As explained in both (Wang, 2020; Shang et al., 2019b), the medication prescription within the first 24-hours is deemed to be the most critical for patient health. Preprocessed data in it's entirety can be found in the (Shang et al., 2019b) repository `https://github.com/sjy1203/GAMENet/tree/master/data` and is used in it's entirety for our reproduction study.

3.3 Hyperparameters

TODO

3.4 Implementation

We will match the conditions for the reproducibility study as much as possible to what was described in (Wang, 2020). The pre-processed MIMIC-III dataset will be randomly divided into a training set, validation set, and test set with 2/3 : 1/6 : 1/6 and be passed through the model as described in section 3.1. Adam (Kingma and Ba, 2015) optimizer will be used for the Clinical Information Extraction Module and generator, while Adagrad (Duchi et al., 2010) is used for the parameters and discriminator. Our implementation however will be performed using Python 3.8 and PyTorch 1.11.

To our knowledge, there does not exist a public repository to SeqMed. Therefore, for our study, we will be reproducing SeqMed in it's entirety. However in SeqMed, the majority of its

novelty comes from the application of Sequential GANs. Because of that, we will heavily reference the architecture described in (Yu et al., 2017) and the implementation done by Surag Nair in https://github.com/suragnair/seqGAN. Additional modifications to be made will include updating policy gradient, action-value function, and addition of specific model modules.

3.5 Computational requirements

Current machine for reproducibility study has 16-core CPU and Nvidia GeForce RTX 2060. Running similar studies (Shang et al., 2019b) on 32-core CPU as assignement yielded CPU time of 29s and Wall time of 14.6s. However, the major difference lies in the usage of Sequential GANs, which are inherently much more computationally expensive and require extensive GPU usage moreso than the architecture used prior.

TODO: Add further detail about computational resources upon final evaluation

4 Results

As is the case with GANs, model success and ability to interpret results depend on the ability of the model to converge. Given the description of the model from (Wang, 2020) and framework modeled around (Yu et al., 2017), the model has failed to converge and cannot therefore be used currently to investigate the claims. To address this inability to converge in the meantime as the result of our study, we will therefore analyze components within the model. After distinguishing components of failure, we will then outline potential plans and solutions to allow for claims to be investigated.

4.1 Drug Synergism by Metric Comparison

Adequate comparison of Jaccard, Recall, Precision, and f-1 score cannot be done given a model that cannot converge. We will discuss the results further once model convergence has been addressed

4.2 Convergence Point

We will use this section about convergence point to discuss the shortcomings of the currently implemented sequential GAN, as it aptly related to this investigation point. The main source of failure is thought to be directly tied into the formulation of our reinforcement learning objective.

One of the most fundamental differences between SeqMed and its direct inspiration is the implementation of the policy gradient. Sequential GANs (Yu et al., 2017) previously implemented rely on a roll-out policy of Monte Carlo search to expand tokens of intermediate state to some final state. For SeqMed, there is no reliance on a Monte Carlo search to expand intermediate states, but rather uses one reward for the entire medication sequence without addition of any other tokens. This, therefore, puts the implemented model at odds with a core feature of the Sequential GAN. It is hypothesized that without extreme fine-tuning to not only model parameters but also encoding from subsequent modules, reward formulation is a lot more linear and constricted as opposed to one that allows Monte Carlo searches.

One support to this hypothesis is with regards to the implemented Clinical Information Extraction Module. Details mentioned in (Wang, 2020) for implementation adequately failed to provide key information for convolution setup (i.e kernel size, pad, stride, etc). As well, language used in (Wang, 2020) seemed to contradict with itself. Description of the three layer convolution network more so sounded like parameters for LSTM: "... use two 3-layers deep convolutional neural networks as the basic model with the hidden size for each layer st to 128 and output size set to 100"

The proposed plan to combat the issue of convergence is therefore focused on two major areas - addressing shortcomings in the policy gradient and addressing the data being passed into the generator. In addition to re-examining the full details of (Wang, 2020), we will investigate maintaining some Monte Carlo search for evaluation of the action value along the sequence of visits for the policy gradient. Similarly, if the re-examination of the Clinical Information Extraction Module fails to succeed, our implementation may depend on the use of the Patient Query component described in (Shang et al., 2019b).

4.3 Additional results not present in the original paper

TODO

5 Discussion

TODO

5.1 What was easy

TODO

5.2 What was difficult

TODO

5.3 Recommendations for reproducibility

TODO

6 Communication with original authors

Author was contacted on April 10th 2022 via email but has not responeded.

References

E. Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua A. Kulas, Andy Schuetz, and Walter F. Stewart. 2016. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. In *NIPS*.

John C. Duchi, Elad Hazan, and Yoram Singer. 2010. Adaptive subgradient methods for online learning and stochastic optimization. In *J. Mach. Learn. Res.*

Alistair E. W. Johnson, Tom J. Pollard, Lu Shen, Li wei H. Lehman, Mengling Feng, Mohammad Mahdi Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. 2016. Mimic-iii, a freely accessible critical care database. *Scientific Data*, 3.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Junyuan Shang, Tengfei Ma, Cao Xiao, and Jimeng Sun. 2019a. Pre-training of graph augmented transformers for medication recommendation. In *IJCAI*.

Junyuan Shang, Cao Xiao, Tengfei Ma, Hongyan Li, and Jimeng Sun. 2019b. Gamenet: Graph augmented memory networks for recommending medication combination. *ArXiv*, abs/1809.01852.

Shanshan Wang, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Jun Ma, and M. de Rijke. 2019. Order-free medicine combination prediction with graph convolutional reinforcement learning. *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*.

Shuai Wang. 2020. Seqmed: Recommending medication combination with sequence generative adversarial nets. *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 2664–2671.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*.

Yutao Zhang, Robert Chen, Jie Tang, Walter F. Stewart, and Jimeng Sun. 2017. Leap: Learning to prescribe effective and safe treatment combinations for multimorbidity. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.