ICU Mortality Prediction

Seyoung Song

Contents

- Problem Description
- Data
- Proposed Approach
- Implementation
- Result
- Conclusion
- Discussion

Problem Description

- Predicting in-hospital mortality using electronic health record (EHR).
- Over 83% of 30 million patients visit hospitals use EHR in the United States alone.
- Deep learning models are being increasingly used in clinical healthcare applications.
- However, few works exist that benchmark the performance of the deep learning models using EHR

Data

- (MIMIC-III) Medical Information Mart for Intensive Care III (v1.4) is publicly available dataset, which includes all patients admitted to an ICU at the Beth Israel Deaconess Medical Center from 2001 to 2012.
- supports a diverse range of analytic studies spanning epidemiology, clinical decision-rule improvement, and electronic tool development
- encompasses a diverse and very large population of ICU patients
- contains highly granular data including lab results, vital signs, medications, and more

Approaches

- Use conventional machine learning model (logistic regression) with hand crafted features as benchmark model
- Compare the performance of RNN models (both LSTM and GRU) that takes raw temporal data (discretized by 1 hour interval) against the benchmark model.
- Experiment with both standard RNN and variations (different depth, bidirectional, deep supervision, channel-wise models with or without target replication, applying recurrent dropout, etc)

Approaches (continued)

- AUC-ROC as main metrics. Also report AUC-PR.
- bootstrapping to estimate confidence intervals of the test score.
- For all LSTM and GRU based models, bidirectional models with depth of 2 with 16 hidden units as standard structure
- used recurrent dropout for all RNN models (in the approach similar to Moon et al 2015)
- grid search to tune all hyperparameters based on validation set performance.
- best model for each model is chosen according to the performance on the validation set. The final scores are reported on the test set.

Implementation

- Data: Parquet versions of MIMIC-III tables are already available in the OpenData S3 bucket and we have access to the PhysioNet MIMIC-III bucket.
- ETL (data extraction, cohort selection, cleansing, preprocessing, variables and features selection) with AWS Glue (job type of Python shell running on spark clusters) and S3.
- Features and model results outputted to and accessed in S3 bucket.
- ML models deployed with Amazon SageMaker on ml.m5.xlarge instance.

Results

Models	AUC-ROC	AUC-PR
Logistic regression with handcrafted features	0.849 (0.828-0.867)	0.474 (0.419-0.531)
Standard LSTM	0.843 (0.823-0.863)	0.464 (0.409-0.518)
Standard GRU	0.848 (0.828-0.867)	0.482 (0.429-0.533)
Standard LSTM with target replication	0.849 (0.829-0.868)	0.475 (0.420-0.529)
Standard GRU with target replication	0.849 (0.829-0.868)	0.482 (0.428-0.536)
Channel-wise LSTM	0.853 (0.833-0.871)	0.493 (0.440-0.546)
Channel-wise GRU	0.854 (0.831-0.870)	0.492 (0.438-0.544)
Channel-wise LSTM with target replication	0.853 (0.834-0.872)	0.493 (0.441-0.548)
Channel-wise GRU with target replication	0.854 (0.833-0.872)	0.491 (0.440-0.546)

Table 3: Results for in-hospital mortality prediction.

Results (continued)

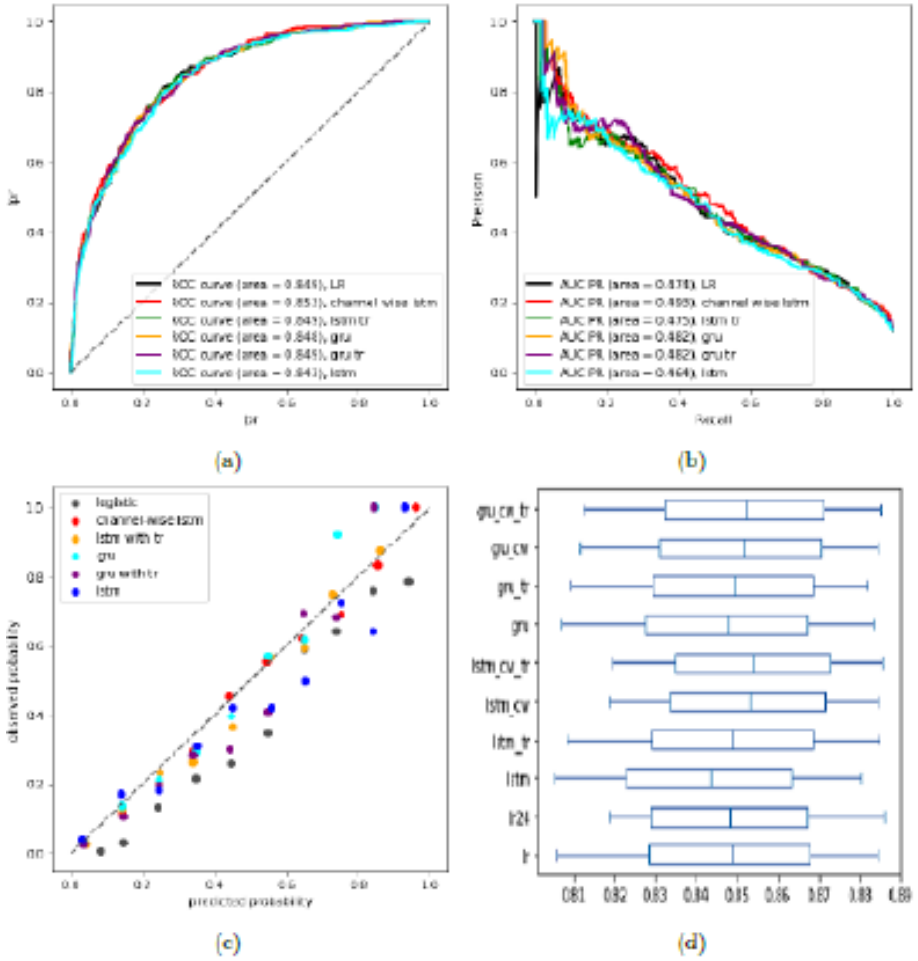

Figure 1: Results for in-hospital mortality prediction.

Conclusion

- Most RNN models do not outperform linear model, although we can expect that as complicated features are used for the linear model.
- Exception is the channel wise variations, which are better than linear models with substantial margins
- When calibration is considered, logistic regression model tend to overestimate the actual probability of mortality however
- Standard bi-directional GRU performs similar to logistic models across all metrics and better calibrated

Discussion (challenges)

- Implementation side: AWS EC2 default instances (ex. ml.t3.medium) barely usable for both ETL and deploying ML models to work on MIMIC III database
- Switched to ml.m5.xlarge, but still has performance problems (ex. cannot train channel wise models that can be doable with local machine) Non-standard EC2 instances (ex. ml.p3) that supports accelerated computing with GPU need to be used
- Not many features were experimented (although features we focused on in vital signs and lab categories make it comparable to related benchmark works), not enough epochs for complicated structures, not through hyperparameter tuning (adjusting learning schedule, weight decays, etc can make difference) as a result
- The number of patients we used are considered to be large compared to the related benchmark works, but the test data remaining is deemed not large and best regularization is needed to generalize