CSE 6250 - Final Paper - Group 69

Mortality Prediction in Intensive Care Unit

Nakul Patel
Georgia Institute of Technology
npatel72@gatech.edu

Ishan Kaul
Georgia Institute of Technology
ikaul7@gatech.edu

Hyeon Woo Shim
Georgia Institute of Technology
hshim30@gatech.edu

Mentor: Ming Liu, Georgia Institute of Technology; mliu302@gatech.edu

Presentation Link: https://youtu.be/adEBlGV7kG4

Abstract

Modern Electronic Health Record (EHR) systems contain large amount of data which can be used to predict the best course of action for patients admitted to a hospital ICU. In this project we use combination of admission time information such as age, sex and SAPS-II score and latent variable topic models (Latent Dirichlet Allocation) to turn clinical notes into meaningful features, and actionable knowledge to predict patient mortality among ICU patients. Physicians can use this to identify patients that need immediate attention and care. We used Postgres DB to generate the SAPS-II and EH comorbidities features and Apache Spark for data processing and building models. Our current result in this draft uses baseline model tested with in ICU, 30 day post discharge and 1 year post discharge mortality using MIMIC III demo database containing 100 patients.

Introduction

The intensive care unit is a challenging environment due to rapidly changing state of patients, and effective mortality prediction in such conditions can help us make informed, quicker decisions and save more lives in such critical states. In this project, we plan to implement the work done in the paper “Unfolding Physiological State: Mortality Modelling in Intensive Care Units” [1] using big data tools (such as Spark) to not only get a deeper understanding of how to predict mortality using MIMIC healthcare data but also gain expertise in data processing using big data technologies..

As stated in [1], the aim is to generate a better understanding of the severity of a patient's condition by using the clinical notes of doctors in order to create better context. The information in clinical notes helps provide the most important aspects of patients’ physiology, and combining such features (along with healthcare information) in a predictive model will help in generating better accuracy, which in turn can provide insight on improving the quality of care. And by employing big-data tools, we have been able create and run experiments more efficiently.

Related Works

Mortality prediction for acute setting such as ICU is a broad area of research and there is range of work done in this area. [2] and [3] uses a list clinical data and machine learning to predict the severity of patients in acute cases. There is also work done in using discharge summaries as features [4], and topic modelling on clinical notes on a sub group of patients [5], which has shown improved predictive models. Work in [7] provides a solid benchmark to evaluate our models’ predictive capabilities.

Data and Methodology

Data Source

We used the ICU data from the MIMIC III (Medical Information Mart for Intensive Care) Clinical Database [6], which contains de-identified health data for diverse set of patients from the Intensive Care Unit (ICU). The dataset consists of 46,520 patients from which we find 26,121 male and 20,399 female patients and 2,078,705 clinical notes. Patient mortality outcomes were queried and served as the ground truth for the machine learning (SVM) models.

Feature Engineering and Related Pre-Processing

We use a combination of different features to test our prediction model, and they are as follows:

1. Admission Baseline Model: This is the benchmark model which will have just 3 features which are computed for each patient at the time of admission to ICU i.e. Age, Sex, SAPS-II score.

    a. From the ICU data, for each patient, we pick the most recent ICU stay based on the date on which they left the ICU. Although it would be possible to treat different ICU stays for the same patient as unique instances, we believed that we may fail to account for the relationships between those stays.

    b. Unlike patients' SAPS-II score and gender, which are directly provided by the dataset, age requires more processing to calculate. To calculate the age at the beginning of their ICU stay, we find the difference between the in-date of the stay and their date of birth. There is a caveat, however, that the age anyone older than 89 within their first ICU stay is masked, rendering their date of birth meaningless. To account for this, we set anyone who appears to have an age of 300, which is the offset of the mask, to be 89 years old. We then filter out any patients who are less than 18 years old.

2. Retrospective Topics Model: In this model, we attempt to derive contextual features with the help of clinical notes. We run LDA topic modelling on all the notes to infer the top 50 topics for each patients thus creating 50 dimensional feature vector as training features. Note that as a retrospective model, it is not for practical use, but the measures obtained are used to evaluate the temporal models.

    a. We filter out discharge notes as they tend to give away the diagnosis of a patient and that won't lead to a generalized model.

    b. We filter all notes that are within the ICU stay of their corresponding patients. For some notes without exact chart times, we utilize their exact store time or their chart date appended with 00:00:00 AM.

    c. We tokenize each note's text and remove stopwords using a list provided by an algorithms course at Princeton. For each patient, using the remaining words, we determine the 500 most informative by applying the term frequency–inverse document frequency (TF-IDF) metric, and the union of all such 500-word lists make up our vocabulary for the LDA. We also filter out any patient who has less than 100 non-stop words in their vocabulary.

3. Retrospective Derived Features Model: We attempt to derive additional features from the admission data. We extract from the patient's stay their 30 EH comorbidities and combine them with their baseline characteristics, resulting in a total of 33 features. This is missing the min, max, and final SAPS-II score that Ghassemi et al. utilize, as we could not obtain the data that indicates patients' mid-stay SAPS-II scores.

4. Time-varying topic model: In this model, we derive contextual information from notes as a function of time from their first note. This is essentially the retrospective topic model that is applied to a time

window that grows by 12 hours from the time of their first note. This helps apply dynamic feature engineering as we use notes available till a given point and helps us to predict mortality at any given instance using only features available. (Unfortunately, we could not test our implementation due to time constraints)

a. To create a time-varying model, we compute the time of the first note submitted within the patient's stay, and let it be t=0. From there, at every 12-hour interval, we filter the training and testing set to keep only the patients who remain alive in the ICU. Patients who have died or discharged before that time mark are no longer considered from that point forward. Note, as stated above, that these time-based datasets are applied to the baseline model as well.

Predictions

In this project, we have identified three types of prediction for our pipeline: mortality in ICU, mortality within 30 days post-discharge, and mortality within 1 year post-discharge. For each type, we generate a label for each patient, giving a 1 if they die in that period and 0 if not, thus formulating a binary classification problem. These labels are paired with the features from a model to produce the training and testing data.

To test our models, we split the data into 70% training data and 30% testing data. Then, as the ratio of positive instances to all instances is very low (hovering around 10%), we subsample the training data to create a 30:70 ratio between the positive and the negative instances. The test data is left unchanged to emulate the reality of the class imbalance. After normalizing the features generated depending on the model, we take the features derived to train a support vector machine by employing `SVMWithSGD` from Spark's MLlib and make predictions on the features derived from the test data using the resulting model. Our goal was to maximize the area under the receiver operating characteristic curve (AUROC or AUC), so we ran 5-fold cross validation on a smaller subset of the training data to determine the AUC-maximizing hyper parameters, which we then used to train the SVM on the full training set.

For the admission baseline model, we use the entire training set of patients to train an SVM, cross-validating to find the best hyper parameters as mentioned. However, to test the model, we filter the test patient set the same way mentioned under the time varying topic model, in intervals of 12 hours since the chart time of their first note. The model trained on the entire set was used to make prediction on each test patient set at different intervals, and the resulting AUCs were recorded.

Results & Analysis

The table below shows the AUC of test-set for the retrospective derived features model and the retrospective topic model for each of the prediction types (i.e In ICU, post 30 Day discharge and post 1 year discharge mortality)

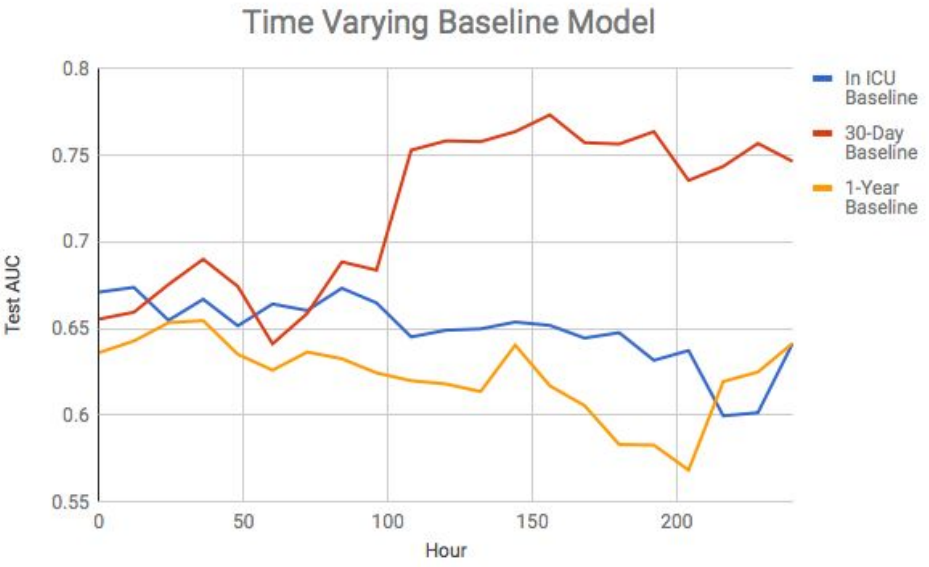
Model	**In ICU**	**30 Day**	**1 Year**
Retrospective Derived Features Model	0.6193971804	0.6233460113	0.6369264503
Retrospective Topic Model	0.6754885294	0.5726433431	0.5951190826

The retrospective derived features model shows similar AUC value for the three prediction types as these were static features computed at the time of admission and seems to have a similar impact for each type of prediction scenarios. Although it is difficult to compare these singular values to derive meaning, we believe that the derived features model may have more predictive power in this case than the retrospective topic model because it already encodes the serious diseases present in the patients, while the topics only indicate words that may imply such diseases. In other words, comorbidities may already contain the information that the topics provide and in a more

generalized way. Topics may simply add more degrees of freedom, having about 20 more features than the derived features, allowing the model to overfit more easily. Comorbidities also record *chronic* diseases or problems present in patients, and thus the information can certainly be more useful for the longer term prediction, as the AUC values evidence. However, more experiments with cross validation and hyper parameter tuning would be able to provide a more solid conclusion. This problem is especially more prominent if we consider that Ghassemi et al. present AUC metrics of 0.9 to 1.0 for retrospective in-hospital mortality and 0.7 to 0.8 for others, which are much higher than our results.

The in-ICU prediction for the retrospective topic models shows the highest AUC. This is expected because as we get the real-time contextual information derived from the notes would have the most impact on in ICU mortality prediction as compared to the other prediction types. The notes are likely to contain immediate concerns or very recent occurrences that can be directly related to the patient's short-term survival. The EH comorbidities, however, point towards the underlying threats to the patient's well-being.

The figure below shows the result, specifically the AUC metrics, of employing the baseline feature vector of age at admission, gender, and SAPS-II score for each patient, as described above.



One aspect that immediately stands out is the 30-day baseline metrics, which suddenly increases after the 100-hour mark. This may be due to the distribution of data points across the control and test sets, as experiments with smaller subsets of the full dataset did not indicate this pattern. The ratio between positive cases and negative cases can become abnormal especially when the testing set becomes smaller. This intuition is confirmed by the table below, which shows that there are significantly fewer positive cases in the 30-day prediction than others. Although, the numbers are for test patients, it is likely that the training set shows similar behavior. The in-ICU baseline predictions shows deterioration in predictive power as time goes on, which is expected and shown in [1]. Overall, however, the resulting AUCs are shown to be much improved from the initial prototyping stage, averaging about 0.07 below what Ghassemi et al. record. Again, with a more thorough investigation of the hyperparameter space, we believe that similar results can be achieved.

Test Patient Numbers				
Hours	Total	Positive In-ICU	Positive 30-Day	Positive 1-Year
0	10365	1159	708	1727
12	9883	982	693	1708
24	8118	887	599	1473
36	6745	808	531	1277
48	5299	736	443	1013
60	4518	673	401	897
72	3661	627	335	751
84	3197	593	302	655
96	2724	560	273	559
108	2457	521	256	506
120	2147	487	232	446
132	1943	430	212	411
144	1736	408	191	367
156	1602	379	181	331
168	1455	357	156	303
180	1357	329	147	286
192	1241	308	132	266
204	1146	277	124	254
216	1076	263	114	236
228	1004	250	106	223
240	940	241	101	209

Conclusion

In the ICU, knowing a patient's state and trajectory can help clinicians make decisions and provide the right care. We implemented admission baseline feature engineering for 3 types of prediction criteria. We then combine these features from structured data with the unstructured features from clinical notes by implementing LDA to automatically discover latent structures present in the free text hospital notes. With our work we try to confirm the idea put forward by Ghassemi et al that there is rich information in hospital notes that provide information on given patient's severity which can be used to augment the features from structured data to improve predictions made about patient's condition.

We found it very challenging to replicate the implementations of Ghassemi et al.. We faced issues primarily in our development environment, specifically its Spark version. Version 1.3.1 does not provide the classes, which are on the other hand available in higher versions, to allow us to perform topic modelling with LDA more effectively and conveniently. Without the functionality provided through Spark, we have ended up investing a significant amount of time testing alternatives or pre-processing data outside of the Spark session. Also, owing to the large size of the date, we were not able to tune hyper-parameters for all the models and there is still scope of improvement which is evident from the difference in results in our experiments and the baseline paper.

References

[1] Ghassemi M, Naumann T, Doshi-Velez F, Brimmer N, Joshi R, Rumshisky A, Szolovits P. Unfolding physiological state. Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD 14. 2014;

[2] A. E. Johnson, A. A. Kramer, and G. D. Clifford. A new severity of illness scale using a subset of acute physiology and chronic health evaluation data elements shows comparable predictive accuracy*. Critical care medicine, 41(7):1711–1718, 2013.

[3] W. Hug and P. Szolovits. Icu acuity: real-time models versus daily models. In AMIA Annual Symposium Proceedings, volume 2009, page 260. American Medical Informatics Association, 2009.

[4] S. Saria, G. McElvain, A. K. Rajani, A. A. Penn, and D. L. Koller. Combining structured and free-text data for automatic coding of patient outcomes. In AMIA Annual Symposium Proceedings, volume 2010, page 712. American Medical Informatics Association, 2010.

[5] M. Ghassemi, T. Naumann, R. Joshi, and A. Rumshisky. Topic models for mortality modeling in intensive care units. In Proceedings of ICML 2012 (Machine Learning for Clinical Data Analysis Workshop), Poster Presentation, Edinburgh, UK, June 2012.

[6] The Laboratory for Computational Physiology, MIT. (2015). MIMIC Critical Care Database. Retrieved from https://mimic.physionet.org/

[7] Harutyunyan H, Khachatrian H, Kale DC, Galstyan A. Multitask Learning and Benchmarking with Clinical Time Series Data. arXiv preprint arXiv:1703.07771. 2017 Mar 22.