DPM360: New Additions to Advanced Disease Progression Modeling

Akira Koseki, Italo Buleje, Prithwish Chakraborty, Elif Eyigoz, Mohamed Ghalwash, Takashi Itoh, Toshiya Iwamori, Michharu Kudo, Pablo Meyer, Kenney Ng, Parthasarathy Suryanarayanan, Hiroki Yanagisawa, Jianying Hu

Background

Disease Progression Modeling (DPM)[1] aims to characterize the progression of a disease and its comorbidities overtime using a wide range of analytics models. Typical approaches include predictive modeling[2], time-to-event estimation[3,4,5,6,7], and state-based modeling[8,9,10] for key disease-related events. DPM has applications throughout the healthcare ecosystem, from providers, to payers, and pharmaceutical companies. But the complexity of building effective DPM models can be a roadblock for their rapid experimentation and adoption when adopting cutting-edge algorithms. Some of this is addressed by standardization of data model and tooling for data analysis and cohort selection. However, there are still unmet needs to facilitate the development of advanced machine learning techniques such as recent deep learning and probabilistic modeling.

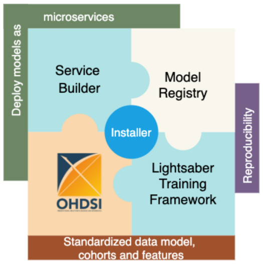

Figure 1. DPM360 component view

To address this, we have been developing Disease Progression Modeling Workbench 360(DPM360) as an opensource project (https://ibm.github.io/DPM360/). DPM360 is an easy-to-install system to help research and development of DPM models (Figure 1). It manages the entire modeling life cycle, from data analysis (e.g, cohort identification) to machine learning algorithm development and prototyping. While we showed general features of DPM360 and predictive analysis in the past OHDSI[11] event[12], we now demonstrate advanced modeling capability including OHDISI data tooling, and extensible training framework which exploits recent achievements of time-to-event estimation, and state-based modeling.

Methods

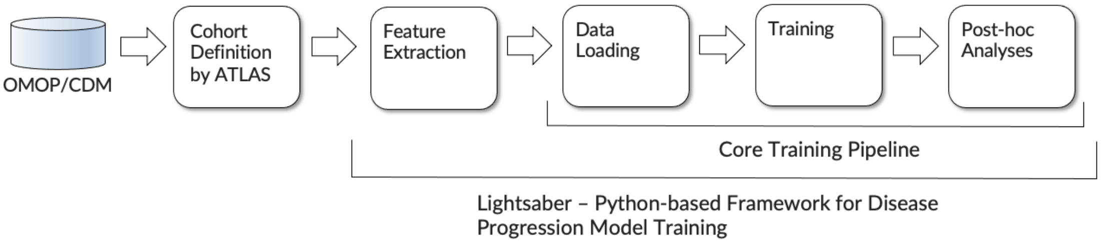
OMOP/CDM
Cohort Definition by ATLAS
Feature Extraction
Data Loading
Training
Post-hoc Analyses
Core Training Pipeline
Lightsaber – Python-based Framework for Disease Progression Model Training

Figure 2. Model training pipeline of Lightsaber in DPM360

Our general training pipeline is illustrated in Figure 2. At first, being connected to an OMOP/CDM database, a user defines cohorts of interest using the GUI of ATLAS. Next, using feature extraction modules in Lightsaber, features as explanatory variables, and outcome as the dependent variable, if necessary, are extracted. In most cases, such features are formed as time series of standard clinical data including diagnosis, drug prescription, procedures, lab test results, and so on. Subsequently, the user trains models using the core training pipeline consisting of the components of Data Loader, Trainer, and Post-hoc Analyzer. Note that modules running on the framework are extensible, and the user can easily add new functionalities using ordinary Python programing conventions for machine learning.

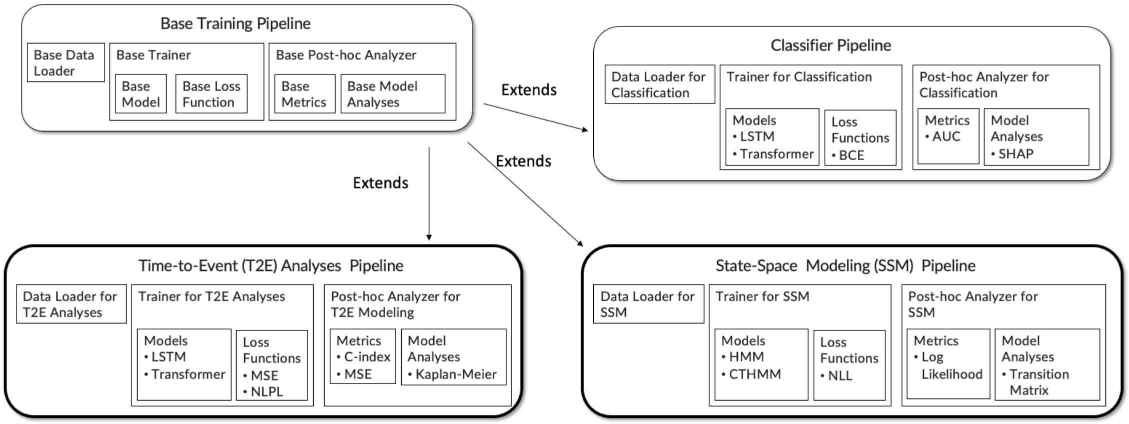

Figure 3. Extending training pipeline of Lightsaber for advanced disease progression modeling

As described in Figure 2, the core training pipeline is constructed by Data Loader, Trainer, and Post-hoc Analyzer. The Data Loader performs necessary data pre-processing such as data transformation, conversion, scaling, imputation, and so on. The subsequent Trainer then conducts mini-batch loop to update parameters of the models to optimize the specified loss for the input data. Finally, the user evaluates the learned models using metrics and other standard analyses in the problem domain.

For extending the pipeline, the user can modify and override the base components of the core training pipeline whenever it is needed. Figure 3 illustrates the base pipeline extended for three typical disease progression problems, predictive modeling using classification[2], time-to-event estimation using survival analyses[3,4,5,6,7], and state-based modeling using Hidden Markov models[8,9,10]. Since the first extension has already been discussed[11], we focus on the latter two. First, an extension for time-to-event analyses solves survival problems when censored patients are also included. The extended Data Loader provides event flags as extra information. The typical Trainer contains representation learning and the prediction layer with specific loss functions to obtain correct rankings between censored and uncensored patients such as Negative Partial Log Likelihood studied in Cox Regression[3]. In the Post-hoc Analyzer, the users conduct Kaplan-Meier estimation[13] to measure the performance of the learned models. Second, an extension for Space-State Modeling finds hidden states and the time-series progression of patients. This is unsupervised modeling thus the extended Data Loader does not provide outcome information. In the Trainer, events are typically modeled as state-transitions using Hidden Markov models and its variant algorithms, whose parameters are trained to maximize the likelihood of the observations. In the Post-hoc Analyzer, the users can examine the learned transition matrices and other estimated parameters to describe the disease progression.

Results

To demonstrate the training system, we constructed modeling pipelines using MIMIC III[14]. Specifically, we defined and extracted cohorts related to mortality in critical care settings from an OMOP CDM version of the MIMICIII dataset[15]. Using ATLAS, we defined our target cohort consisting of adult patients who have been hospitalized for the first time for at least two days, and who have at least one measurement recorded within the first 48 hours, forming our time-series features, and our outcome cohort of adult patients who died. In the past OHDSI event, we demonstrated classification pipelines (https://ibm.github.io/DPM360/) where we predicted whether the patient died in hospital within 30 days of the first admission. Following up, we want to demonstrate our capabilities to do time-to-event and space-state modeling using the same cohort setting. For time-to-event analyses, we estimate the survival rate in the presence of censored patient using the time-series features. We provide traditional algorithms including Cox Regression[3] and Random Survival Forests[4], as well as more recently published advanced methods such as RankSVX[5], DeepHit[6], VAECox[7]. For space-state modeling, transitions among hidden states during 48 hours of observations are modeled. Hidden states with their typical parameter distributions, a transition matrix among those states, and state-transition pattern for a patient are estimated. We provide a traditional discrete-time Hidden Markov model[8] as well as a continuous-time Hidden Markov model[9] where modeling using continuous-time transitions is possible.

Conclusion

We explained and showed the capability of Python-based training framework for disease progression modeling. We plan to incorporate those components to our opensource project of DPM360. It is believed that this can facilitate disease progression modeling using OHDSI data accumulation, especially in the Python community, and encourages the developments and commitments of advanced models leveraging opensource activities.

References/Citations

1. Wang X, Qi J, Yang Y, Yang P. A Survey of Disease Progression Modeling Techniques for Alzheimer's Diseases. In: 2019 IEEE 17th International Conference on Industrial Informatics (INDIN). vol. 1. IEEE; 2019. p. 1237–1242.
2. Chakraborty P, Codella J, Madan P, Li Y, Huang H, Park Y, et al.. Blending Knowledge in Deep Recurrent Networks for Adverse Event Prediction at Hospital Discharge. AMIA 2021 Virtual Informatics Summit. 2021.
3. Cox DR. Regression Models and Life-Tables. Journal of the Royal Statistical Society, B34(2):187–220, 1972.
4. Hothorn T, and Jung HH. RandomForest4Life: a Random Forest for predicting ALS disease progression. Amyotrophic LateralSclerosis & Frontotemporal Degeneration, 15(5-6):444–452, September 2014.
5. Liu B, Li Y, Sun Z, Ghosh S, Ng K. Early prediction of diabetes complications from electronic health records: A multi-task survival analysis approach. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 32;2018.
6. Lee C, Zame WR, Yoon J, Schaar M. DeepHit: A Deep Learning Approach to Survival Analysis With Competing Risks. In AAAI, July 2018.
7. Kim S, Kim K, Choe J, Lee I, Kang J. Improved survival analysis by learning shared genomic information from pan-cancer data. *Bioinformatics*. 2020;36(Suppl_1):i389-i398.
8. Kwon BC, et al.. Progression of Type 1 Diabetes from Latency to Symptomatic Disease is Predicted by Distinct Autoimmune Trajectories. Nature Communications, 2022.

9. Mohan A, Sun Z, Ghosh S, Li Y, Sathe S, Hu J, Sampaio C. A Machine-Learning Derived Huntington's Disease Progression Model: Insights for Clinical Trial Design. Movement Disorders, 2022.
10. Severson K, Chahine LM, Smolensky LA, Frasier M, Ng K, Ghosh S, Hu J. Discovery of Parkinson's Disease States and Disease Progression Modeling: a Longitudinal Data Study Using Machine Learning. The Lancet Digital Medicine, 2021.
11. Hripcsak G, Duke JD, Shah NH, Reich CG, Huser V, Schuemie MJ, et al. Observational Health Data Sciences andInformatics (OHDSI): opportunities for observational researchers. Studies in health technology and informatics.2015;216:574.
12. Parthasarathy S, Chakraborty P, et al.. Disease Progression Modeling Workbench 360, OHDSI Collaborator Showcase. 2021.
13. Kaplan EL, Meier P. Nonparametric estimation from incomplete observations. Journal of the American statistical association, 53(282):457–481, 1958.
14. Harutyunyan H, Khachatrian H, Kale DC, Ver Steeg G, Galstyan A. Multitask learning and benchmarking with clinical time series data. Scientific Data. 2019 Jun;6(1).
15. Johnson AE, Pollard TJ, Shen L, Li-Wei HL, Feng M, Ghassemi M, et al. MIMIC-III, a freely accessible criticalcare database. Scientific data. 2016;3(1):1–9.