ORIE 4999 Final Report Fall 2024
Clinical Trials with Active Surveillance

Members: Jack Jansons, **Mentors:** Arielle Anderer

0. Introduction

In certain scenarios, cancer management involves what is known as “active surveillance.” This treatment approach involves testing for various signs of cancer progression at regular intervals, and beginning additional treatment if these tests show that it is starting to grow more quickly. One disease in which active surveillance is common practice is prostate cancer. The focus of this work is twofold – first, to understand the patterns of progression and develop a model to accurately predict the time to progression of prostate cancer. Second, to use these predictions in a clinical trial setting to better understand the efficacy of different early interventions. The goal here would be to utilize screening records as surrogate markers for disease progression. Ideally the model would be able to use both surrogate information from the active surveillance tests as well as information on patients’ disease progression to accurately determine the efficacy of a new intervention as compared to some control group “current standard of care” before disease progression actually occurs in all patients, effectively reducing the amount of time required to run a clinical trial.

1. Clinical Trials and Prostate Cancer

The first portion of the project involved building a full understanding of the setting in which the model will operate. This included understanding the primary components of a clinical trial and being familiar with the relevant measurements taken for a patient with prostate cancer in active surveillance. The final aspect of this portion of the project was to solidify the structure of the data that would be collected in a clinical trial and could be expected to be an input to our model.

1.1 Relevant Measures for Prostate Cancer

The first part of understanding the setting in which our model will work was being familiar with the relevant measurements taken for a patient with prostate cancer in active surveillance. The most common measure used is Prostate Specific Antigen (PSA) levels [1]. These are taken through a simple blood test and are measured in ng/mL. While this is a very easy measurement to take, the primary downfall of PSA levels is that it is easily affected by various physical activities (some as arbitrary as riding a bike) resulting in these measurements being skewed or noisy making it difficult to accurately predict disease progression solely based on this factor. Another common measure is gleason scores. This score measures the abnormality of the cells in cancer tissue from samples of the prostate taken in a biopsy [2]. This is a more invasive measurement to take and while there are guidelines for the scoring and new AI models to “normalize” grading, high discrepancies in grading persist even between experienced pathologists. Additional measurements include hardness and irregularity measures taken during digital rectal exams (DREs) and imaging used to detect disease spread.

1.2 Formulating Data Structure

A critical aspect in developing a predictive model is understanding the format of the input data. Now that we have developed a strong understanding of the types of measurements taken in for prostate cancer patients in active surveillance, we can solidify the structure of the data we are working with. Specifically, we are working with multiple measurements taken at regular intervals. This type of data is called panel data. We are trying to predict the time until an event occurs: either cancer progression or death. This type of prediction is called time-to-event prediction or survival analysis. With this structure solidified, we can search for a model that is effective at making time-to-event predictions from panel data.

2. Predictive Models for Survival Analysis with Panel Data

The model we are attempting to develop is a survival analysis model. Specifically, this model must be able to make time-to-event predictions by predicting the corresponding survival function. A survival function gives the probability of surviving past a certain time. A survival analysis model must also be able to work with censored data. Censored data points are the result of patients who have not experienced the event that we are attempting to predict and thus are "incomplete." Instead of throwing these data points out, our model must be able to utilize them as they provide valuable information.

2.1 Time to Event Predictions

We first attempt to find models that are capable of making time-to-event predictions. I reviewed a paper that performs a complete review of the applications of machine learning in the survival analysis field [3]. This paper looks at 28 different publications that utilize machine learning methods to make time-to-event predictions. Of those 28, 57% (N=16) of these publications utilize random survival forests. Random forests utilize a combination of decision trees to make predictions in nonlinear settings [4]. Random survival forests are capable of working with censored data by using a log rank statistic for node splitting. The primary downside of using random survival forests in our setting is that they have not been adapted to work with panel data. Other than random forests, 39% (N=11) of the publications use neural networks to make survival function predictions. The study does not specify what kinds of models specifically these publications utilize; however, it is reasonable to assume that they use various deep learning methods. The next goal is to find a deep learning model that can make time-to-event predictions with panel data.

2.2 Predictive Models with Panel Data

Panel data is a type of time-series data. Typically, recurrent neural networks (RNNs) are effective at working with time series data. This includes the various forms of RNNs including long-short term memory (LSTMs) models and gated recurrent units (GRUs). These types of models work well with time series data; however, they are not specifically designed for survival analysis predictions. Another type of model that is specifically designed for survival analysis is deep survival machines (DSMs) [5]. While these models are not initially designed to work with panel data, they have an extension called recurrent deep survival machines (RDSMs) which combine DSMs and RNNs to create a deep learning model specifically designed for time-to-event predictions with panel data [6].

2.3 Deep Survival Machines (and Extensions)

Deep survival machines, published by Auton Lab from Carnegie Mellon University, are specifically designed for survival function prediction. They estimate the survival function as a weighted combination of parametric functions represented by either Weibull or Log-Normal distributions. This allows this model to account for competing risks as each parametric distribution essentially can represent a different risk. The model works by passing the input covariates through an initial neural network to get an embedding version of the data. These embeddings are combined with some tuned hyperparameters to estimate an eta and beta which are used to parameterize the various distributions. The embeddings are also combined with a weight hyperparameter to compute a weighted average of the parametric distributions creating a final survival function prediction. The model is trained with a custom combined loss function which allows DSMs to effectively learn from both censored and uncensored data. Ultimately, DSMs are able to learn nonlinear dependencies between input covariates through the neural network embedding layer, make predictions in competing risk situations with the weighted combination of parametric distributions, and effectively handle learning from censored data with the combined loss function. This makes DSMs the most dynamic model at the time of their publication for survival analysis, and they provide the best representation of the real world scenarios as they do not make as many assumptions as previous methods. However, the initial DSM method does not work with time varying covariates.

Shortly after the publication of the original DSM paper, the authors published an extension: recurrent deep survival machines. These are exactly the same as DSMs; however, they replace the neural network used to generate the embedding with a recurrent neural network or one of its variants (including GRUs and LSTMs). RDSMs are thus able to fit perfectly in our situation as they are able to effectively model real world panel data for survival analysis predictions.

With the publication of the DSM and RDSM paper, the authors provided a Python package with implementations of both models [7]. I was able to use the demo provided in their package to use an RDSM model on the Primary Biliary Cirrhosis (PBC) dataset, a survival analysis panel data dataset matching our problem setup [8, 9]. Now, with a model architecture that matches our problem and data structure, we are able to move forward with finding relevant data and designing a clinical trial.

3. Generating Synthetic Time-Series Survival Analysis Data

Initially, the project was meant to look at patients with prostate cancer in active surveillance; however, after presenting the project to a group of practitioners from Weill Cornell, we learned that the event of progressing from active surveillance to treatment can be somewhat noisy and arbitrary depending on the doctor prosiding over the patient. It is not uncommon for patients who do not necessarily need treatment to receive treatment for a variety of reasons, so predicting this outcome may be difficult as progression is an almost arbitrary measurement.

Also, after scanning through a series of prostate cancer clinical trial papers and observing the measurements and events they observed, it was clear that obtaining clinical trial data in the form we solidified during the first stage of this project was not a trivial task and would likely be a task that spans beyond this semester [13-17]. Thus, we decided to move forward with testing the model and problem setup we have come up with on general datasets that match our structure. Specifically, we decided to move forward with the PBC dataset that I originally trained the RDSM on. The goal now is to generate a larger synthetic dataset that matches the PBC dataset in that the generated samples could be from the original dataset. We would then adjust some of these data points to be “treatment” data points and perform our experiment to see if we can effectively predict the efficacy of a treatment using our DSM model and ultimately expedite clinical trials.

3.1 Synthetic Data Generation Methods

The first data generation method we considered was creating parametric decay functions for each covariate. This would allow us to model time varying covariates that depend on the time invariant covariates which are trivial to generate. The issue with this method is that it requires us to make many assumptions about the relationships in the data. Another data generation method commonly used is generative adversarial networks (GANs). This is a deep learning model that has two parts: a generative model and an adversarial classifier. GANs are trained in a two part fashion. First, the adversarial model is trained to distinguish between the data generated by the generative model and the real data from the dataset. Then, the generative model is trained to fool the adversarial model into believing that the generated data is from the true dataset. This training cycle is then repeated until the generative model is able to generate data that actually appears to mimic the data from the dataset. We are now presented with the same issue that we faced when choosing a model to predict time-to-event data. We need a generative method that is able to generate panel data as well as time-to-event data.

3.2 SurvivalGANS and TimeGANS

There is already work published about using GANs for generating survival analysis data: SurvivalGANs [10]. SurvivalGANs follow a four step process. The first step is to train the actual GAN to generate the synthetic data. This is done in the same fashion as described in the previous section. However, in the case of SurvivalGANs, these models are able to be conditioned by a conditional vector which is able to specify some specific categorical data. The rest of the data is generated conditioned on the categorical data so that it follows the distribution presented in the true dataset. The second step of the process is to train a survival function model (in our case our DSM model) to generate a predicted survival function for the generated dataset. This will give us the time-to-event prediction data. The third step involves training a time regressor. This model is used for generating censored data. It is able to take in the predicted survival function and output the time-to-event prediction for when censoring occurs. This way, our generated data truly matches the censored and uncensored data in our dataset. Finally, the final step is combining all of the three models to generate the data. Namely, generate a conditional vector which is used to generate synthetic data with the GAN. The survival function is then predicted based on this data and is converted into a censored time-to-event prediction by the time regressor. The main issue with this setup is that it has not yet been adapted to generate panel data. A Python package is also available with an implementation of this method [12].

4. Future Work

We are leaving off the semester looking at methods for generating panel data for time-to-event predictions. I have found a promising way forward with TimeGANs which supposedly are able to generate time series data based on a given dataset [11]. In future work, it would be interesting to dive deeper into this model and see if it is possible to substitute TimeGANs into the first part of the SurvivalGAN method. Namely, instead of using a regular GAN to generate the data, see if it is possible to generate time-series data with the TimeGAN and then progress from there. We have already trained an RDSM to work with panel data, so step 2 is complete, so the only remaining step would be to create a time regressor model that is able to work with panel data as well. A simple RNN should work for this instance as the authors of the SurvivalGAN paper specify that many different architectures can work for this model. The last question would be to figure out a way to incorporate conditional generation with these TimeGANs so that the full functionality of the SurvivalGAN method is present for generating synthetic panel data.

On top of further work in generating synthetic panel data for survival analysis, we also need a better understanding of the PBC dataset. Specifically, we do not have a great understanding of what the individual data points are in the dataset itself as they are all normalized and it does not appear to contain any categorical data. This does not align with the documentation provided by the Auton Survival Python package that contains this dataset, so reaching out to the authors to get a better understanding of the data might be necessary [8, 9].

5. References

[1] Bernal, A., et al. (2024). “The Current Therapeutic Landscape for Metastatic Prostate Cancer.” Pharmaceuticals, 17(3), 351. https://doi.org/10.3390/ph17030351

[2] Munjal, A., Leslie, SW. (2023). “Gleason Score.” StatPearls Publishing. https://www.ncbi.nlm.nih.gov/books/NBK553178/

[3] Huang, Y., et al. (2023). “Application of machine learning in predicting survival outcomes involving real-world data: a scoping review.” BMC Med Res Methodol 23, 268 https://doi.org/10.1186/s12874-023-02078-1

[4] Ishwaran, H., et al. (2008). “Random survival forests.” Ann. Appl. Stat. 2 (3) 841 - 860. https://doi.org/10.1214/08-AOAS169

[5] Nagpal, C., et al. (2021). “Deep Survival Machines: Fully Parametric Survival Regression and Representation Learning for Censored Data.”

[6] Nagpal, C., et al. (2021). “Deep Parametric Time-to-Event Regression with Time-Varying Covariates.”

[7] Nagpal, C. et al. (2022). “Auton-survival: an open-source package for regression, counterfactual estimation, evaluation and phenotyping with censored time-to-event data.”

[8] “Primary Biliary Cirrhosis, Sequential Data.” Education and Research at Mayo Clinic - Education and Research at Mayo Clinic. www.mayo.edu/research/documents/pbcseqhtml/doc-10027141.

[9] “R: Mayo Clinic Primary Biliary Cirrhosis, Sequential Data.” stat.ethz.ch/R-manual/R-devel/library/survival/html/pbcseq.html.

[10] Norcliffe, A., et al. (2023). “Survivalgan: Generating time-to-event data for survival analysis.” International Conference on Artificial Intelligence and Statistics.

[11] Yoon, J., et al. (2019). “Time-series generative adversarial networks.” Advances in neural information processing systems.

[12] Qian, Z., et al. (2023). “Synthcity: facilitating innovative use cases of synthetic data in different data modalities.”

[13] Saad, Fred, et al. “Darolutamide in Combination With Androgen-Deprivation Therapy in Patients With Metastatic Hormone-Sensitive Prostate Cancer From the Phase III ARANOTE”

[14] Newcomb, L. F., et al. (2024). “Long-term outcomes in patients using protocol-directed active surveillance for prostate cancer.” JAMA.

[15] Chi, K. N., et al. (2019). “Apalutamide for metastatic, castration-sensitive prostate cancer.” New England Journal of Medicine, 381(1), 13-24.

[16] Freedland, S. J., et al. (2023). “Improved outcomes with enzalutamide in biochemically recurrent prostate cancer.” New England Journal of Medicine, 389(16), 1453-1465.

[17] Gharzai, L. A., et al. (2021). “Intermediate clinical endpoints for surrogacy in localised prostate cancer: an aggregate meta-analysis.” The Lancet Oncology, 22(3), 402-410.