CPH 200B: CPH Cornerstone **Winter 2025**

Project 1: Survival Analysis and Prediction

Instructor: Ahmed Alaa *Total points: 30 pts*

Note: *LaTeX template courtesy of UC Berkeley EECS dept.*

Many clinical trials and observational studies involve following patients for a long time. The primary event of interest in those studies may include death, relapse, or the onset of a new disease. The follow-up time for a trial or a study may range from few weeks to many years. To analyze this data, we typically conduct time-to-event analysis and build predictive models that learn time-to-event distributions. The goal of this project is to test your ability to conduct survival analysis as well as develop deep learning models for survival prediction.

Instructions: This is not a group project and students will be graded individually. Please refer to the course Github link for further instructions on the coding component of the project. This project has two deliverables:

- **A report summarizing your results**. The report should include answers to the questions below.
- **A zip file with your code**. Submit your code files along with your report in one folder.

Your code submission can include a notebook with steps to reproduce your results, or a readme file with commands to run to reproduce your outputs for each task. You can also create a private Github repo and add me as a collaborator (Username: `ahmedmalaa`). Your approach to experiment management and hyperparameter optimization will not be evaluated in this project. However, you are encouraged to follow best practices for experiment management that you learned through project 1 in CPH 200A.

Please submit your report and code via bcourses by Tuesday 2/4 11:59 PST. You will receieve your overall grade, a grading sheet and personalized feedback on your bcourses account.

1.1 Nonparametric Survival Analysis in Heart Failure [7 pts]

Nonparametric models of survival data do not make parametric assumptions on the distribution of time-to-event outcomes. They are widely used in clinical studies to derive descriptive statistics of survival in a population. In this task, we will apply standard nonparametric estimators to analyze survival of heart failure patients in a recent, widely-recognized study [1].

1.1.1 Setup and Dataset

The dataset we will use in this task was extracted from the electronic health records (EHRs) of 299 heart failure patients from the Faisalabad Institute of Cardiology and at the Allied Hospital in Faisalabad (Punjab, Pakistan), during April–December 2015. The cohort included 105 women and 194 men, and their ages range between 40 and 95 years old. All 299 patients had left ventricular systolic dysfunction and had previous heart failures (HF) that put them in classes III or IV of New York Heart Association (NYHA) classification of the stages of heart failure (read more about the NYHA classification system here). The dataset contains 13 features, which report clinical, body, and lifestyle information. The patients were followed up for 130 days on average (maximum follow-up period was 285 days). The event of interest was death during the follow-up period. The dataset is publicly accessible and was shared with the class through UCSF Box.

1.1.2 Tasks and Deliverables

Please conduct the following tasks on the dataset described above. Your report should include the results for every task (i.e., tables or plots) along with your answers to the questions associated with each task.

Task 1.1.1 [3 pt]. Implement the Kaplan-Meier point estimator from scratch in Python. Apply your estimator to the dataset described above to estimate survival in HF patients. Then, compare your results with the built-in functions in lifelines library. Additionally, compare your survival estimates with the prognoses of HF in the US as reported in the literature. If there are differences, explain what might be causing them.

Task 1.1.2 [2 pt]. Instead of the nonparametric Kaplan-Meier estimator, one can estimate a survival curve using a parametric model that makes assumptions about the distribution of survival times. Assume that the survival time in this population follows an exponential distribution. Propose an algorithm for fitting the parameters of this parametric model. Compare the results of the fitted exponential distribution with the Kaplan-Meier estimate and comment on the limitations of the parametric model.

Task 1.1.3 [2 pt]. The Kaplan-Meier estimator generates population-level survival curves. Imagine modifying it to produce patient-specific survival curves instead. Propose a variant of the Kaplan-Meier estimator that incorporates a nearest-neighbor approach to estimate patient-level survival probabilities. Implement the proposed procedure and evaluate its performance using the Concordance Index (C-index).

1.2 Survival Prediction in HF patients using the Cox Model [7 pts]

1.2.1 Setup and Dataset

In Task 1.1, we used nonparametric estimators to derive descriptive statistics of survival in a population. However, these estimators did not learn the relationship between the patient features and their survival. In this Task, we will fit one of the most widely used survival models, the Cox Proportional Hazards (PH) model, to understand how patient features influence their survival. We will use the same dataset in Task 1.

1.2.2 Tasks and Deliverables

Task 1.2.1 [3 pt]. Fit a Cox PH model using the dataset of HF patients described earlier and report all model coefficients. To implement the Cox model, you can model its linear risk function using a 1-layer neural network and fit the model coefficients by applying gradient descent on the Cox partial likelihood loss. Based on your trained model, what is the effect of a one-year increment in age on patient survival?

Task 1.2.2 [2 pt]. Evaluate the predictive accuracy of the Cox PH within the training sample using the Concordance index (C-index). Explain the differences between the C-index and the AUC-ROC metrics.

Task 1.2.3 [2 pt]. After presenting the Cox model fitted in Task 1.2.1 to your clinical collaborator, they mention that they believe that age is a bigger risk factor for HF in males compared to females. Your clinical collaborator asks you to test his hypothesis. Propose a new Cox model to test this hypothesis by modifying the original feature space. Fit this new model using the same dataset, and comment on the validity of the clinician's hypothesis based on your fitted model.

1.3 Deep Survival Prediction for Heart Transplantation [8 pts]

In Task 1.2, we applied a standard linear model for survival prediction. In this task, you will develop your own deep survival prediction model to capture non-linear relationships and interactions between features and outcomes. Here, we will use a dataset of patients who underwent heart transplant surgeries in the US.

1.3.1 Clinical Background

Heart transplantation is a critical surgical intervention designed to save lives by replacing a patient's diseased heart with a healthy one obtained from a deceased donor. This procedure is typically considered when no other viable medical or surgical alternatives exist for the patient. The majority of heart transplantations are performed on individuals facing "end-stage" heart failure. "End-stage" refers to a condition that has progressed to such an advanced state that all available treatments, except for a transplant, are ineffective. Due to the scarcity of donor hearts, individuals in need of a heart transplant undergo a meticulous selection process at heart transplant centers, and those deemed eligible are then placed on a waiting list. Predicting survival before and after a transplant is crucial for determining patients' placement on the waiting list.

1.3.2 Setup and Dataset

For this task, we will use data collected by the United Network for Organ Sharing (UNOS) [2], a non-profit organization that administers the only Organ Procurement and Transplantation Network (OPTN) in the US. UNOS is involved in many aspects of the organ transplant and donation process in the US, including data collection and maintenance, providing assitance to patients and care takers, and informing policy makers on the best use of the limited supply of organs and give all patients a fair chance at receiving the organ they need. UNOS manages the heart transplant waiting list, i.e., the list of terminally-ill patients waiting for donor heart. In order to determine the order of priority for receipt of a donor heart, individuals are classified by degrees of severity for a donor heart, blood type, body weight, and geographic location.

This Task will focus on the cohort of terminally-ill patients who are enrolled in the wait-list for heart transplantation. In this setup, our goal is to predict the patients who are less likely to survive in order to prioritize them for receiving donated organs. The UNOS data covers 30 years of heart transplantation data in the US, spanning the years from 1985 to 2015. We will use data for patients who were on the wait-list for heart transplantation from 1985 to 2010 (27,926 patients) to train your model. A held-out test set of 8,403 patients enrolled in the wait-list between 2010 and 2015 will be used by the instructor to evaluate your model.

1.3.3 Tasks and Deliverables

Task 1.3.1 [3 pt]. Propose a survival model that uses deep learning to estimate a nonlinear survival function $S(t|X)$. You can adopt one of the models we discussed in class, any other model in the literature, or a new model that you propose. Explain the rationale behind your model architecture and loss function. Which of the variables in UNOS will you **exclude** from the feature set X input to your model and why?

Task 1.3.2 [5 pt]. Implement your proposed model as Python class with the following simple specification [3 pt]. Your model class should contain `fit(X, T, C)` and `predict(X)` methods, where `X` is a numpy matrix of patient features, `T` and `C` are numpy arrays with survival times (in days) and censoring indicators ($C = 1$ means the patient is censored). The predict function should return 20 predictions for each patient, corresponding to 10-year survival predictions with 6-month increments (0, 6, 12, 18,.... months from baseline), where time 0 corresponds to the time of enrollment in the wait-list. Train your model using the UNOS

training sample. Compare the average survival curves predicted by your model for patients in the training data with the Kaplan-Meier estimate for the UNOS population.

Please submit your model weights with your codebase. Your model will be tested on the held-out set, and you will receive a full grade only if your model generalizes to the test set (C-index > 0.5).

1.4 Handling Informative Censoring via Domain Adaptation [8 pts]

A key assumption underlying all methods we studied in class is that the censoring event $C = 1$ is random and does not depend on the survival time or patient features. This assumption is also known in the literature as "uninformative censoring". When this assumption is violated, we run the risk of obtaining biased estimates of survival since patients with incomplete observations of survival time have a different risk profile.

In this task, you will reflect on how the informative censoring setup relates to the domain adaptation problem covered in Lecture 4. To proceed, please install the `PyCox` package for survival analysis in Python.

Task 1.4.1 [3 pt]. Propose a method to check if the uninformative censoring assumption holds. Use your proposed method to check if censoring is informative in the `flchain`, `gbsg`, `metabric`, `nwtco` and `support` benchmark datasets for survival prediction. You can load each of these datasets in `PyCox` as follows:

```
from pycox import datasets

df = datasets.<dataset_name>.read_df()
```

Propose a procedure to generate semi-synthetic versions of the above benchmark datasets (i.e., `flchain_synth`, `gbsg_synth`, `metabric_synth`, etc.) that introduce a controllable level of bias in the censoring events.

Task 1.4.2 [5 pt]. Describe how the informative censoring setup can be reframed as a domain adaptation problem. Building on this reframing, modify the model you developed in Task 1.3 to address biases induced by informative censoring. Apply your modified model to the semi-synthetic datasets you generated in Task 1.4.1. Compare the performance of your original model from Task 1.3 and the modified model in terms of the C-index in each dataset. Comment on your results.

References

[1] Chicco, Davide, and Giuseppe Jurman. "Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone." *BMC Medical Informatics and Decision Making*, vol. 20, no. 1 (2020): 1-16.

[2] Weiss, Eric S., Lois U. Nwakanma, Stuart B. Russell, John V. Conte, and Ashish S. Shah. "Outcomes in bicaval versus biatrial techniques in heart transplantation: an analysis of the UNOS database." *The Journal of heart and lung transplantation*, vol. 27, no. 2 (2008): 178-183.