Breast Cancer Survival Prediction with Neural Networks and Cox Regression

Dariya Mamyrbek, Margo Thompson, Hong Phuc Nguyen

Abstract

This project explores 3 common variations of the Cox regression method to predict breast cancer survival time using the METABRIC (The Molecular Taxonomy of Breast Cancer International Consortium) dataset.

1 Introduction

Survival analysis, also called time-to-event analysis, is used to estimate the lifespan of a specific study population. Cox regression (or proportional hazards regression) is a revolutionary survival analysis model in statistics. It is a method for investigating the effect of several variables on the time required for a particular event to happen. The model assumes that the effects of the predictor variables upon survival are constant over time and are additive in one scale [3]. This research project will explore an extension of the Cox proportional hazards model with neural networks proposed by Kvamme, Borgan, and Scheel (2019) [2]. By parameterizing the relative risk function of a Cox model with neural networks, it is possible to model rich relationships between the covariates and event times. The resulting methods have the flexibility of neural networks while modeling event times continuously [2].

Applications of time-to-event prediction can be found in a variety of settings such as cancer patient survival prediction, customer churn, credit scoring, and system failure times. This project will focus on predicting breast cancer survival time using the following continuous-time survival analysis methodologies: Cox-Time, Cox-CC, and Cox-PH.

The outline of the project is as follows: we will discuss methods of Cox regression, non-proportional Cox-Time, Cox proportional hazards model parameterized with a neural network including Cox-PH (DeepSurv) and Cox-CC, as well as the dataset which will be used. Finally, we summarize and discuss our findings.

2 Methods

In this project, we use methods proposed by Kvamme, Borgan, and Scheel [2] who proposed an alternative loss function that scales well for both the proportional and the non-proportional cases.

The survival function is defined as

$$S(t) = P(T^* > t) = 1 - F(t) \tag{1}$$

where T^* is the time to the event and $F(t)$ is the CDF of the probability distribution that the event happens at time t. In the case of the METABRIC dataset, T^* is the time until a certain patient will fail to survive while enduring breast cancer.

The survival function can be retrieved through the cumulative hazard, $H(t) = \int_0^t h(s)\, ds$, by

$$S(t) = exp[-H(t)] \tag{2}$$

Preprint. Under review.

We calculate the hazard function $h(s)$ and the survival function $S(t)$ using 3 regression methods: Cox-PH, Cox-Time and Cox-CC to predict whether the patient can survive in a certain number of months past the initial examination.

2.1 Cox-PH (DeepSurv)

The hazard function of the Cox proportional hazard, or Cox-PH, regression model is given by:

$$h(t \mid X) = h_0(t) \exp \left(X_1\beta_1 + \cdots + X_p\beta_p \right) \tag{3}$$

In this model, the hazard is assumed "proportional" to a baseline hazard $h_0(t)$. That is, we note that $\exp \left(X_1\beta_1 + \cdots + X_p\beta_p \right)$ is a constant value with respect to time, hence the hazard at any given time is only a multiple of the baseline hazard. With this in mind, we can extend the Cox-PH model where instead of

$$\left(X_1\beta_1 + \cdots + X_p\beta_p \right), \tag{4}$$

we use a neural network to approximate $h_0(t)$. The neural network that was originally used in [1] is called "DeepSurv", hence this method is also referred to as Cox-MLP (DeepSurv) or Cox-PH (DeepSurv).

2.2 Cox-CC

Cox-MLP (CC) or Cox-CC is a proportional version of the Cox-Time model. We will fit the Cox model with mini-batch stochastic gradient descent (SGD) to better scale to large data sets. An approximation of the loss that is easily batched was proposed in the paper. We can approximate the risk set R_i with a sufficiently large subset $\tilde{R}_i$, and weight the likelihood accordingly with weights w_i.

$$loss = \prod_{n=1} \left(\frac{exp[g(x_i)]}{w_i \sum_{j \in \tilde{R}_i} exp[g(x_j)]} \right. \tag{5}$$

As the weights w_i do not contribute to the gradients of the logarithm of (5), we can simply drop them from the loss function, finally obtaining the following loss function for Cox-CC by averaging the loss to make it independent of the data set size,

$$loss = \frac{1}{n} \sum_{i:Di=1} log(\sum_{j \in \tilde{R}_i} exp[g(x_j) - g(x_i)]) \tag{6}$$

2.3 Cox-Time

Cox-Time is a risk model proposed by Kvamme, Borgan and Scheel in [2] that attempts to remove the proportionality assumption from the Cox-CC model. That is, we no longer need to assume that the relative hazards in the Cox-CC model are constant over time with different patients. In exchange, we need to include time as a parameter for the relative hazard function g. The hazard function then becomes:

$$h(t|\mathbf{x}) = h_0(t) exp[g(t, \mathbf{x})] \tag{7}$$

And the loss function becomes:

$$loss = \frac{1}{n} \sum_{i:Di=1} log(\sum_{j \in \tilde{R}_i} exp[g(T_i, \mathbf{x}_j) - g(T_i, \mathbf{x}_i)]) \tag{8}$$

In practice, we only need to include the time value as a regular covariate to calculate the hazard at a certain time. The Cox-Time model is more flexible than the Cox-CC model, as Cox-Time can be used even when the relative hazard varies. However, since g is now time-dependent and we can no longer reuse the same $g(\mathbf{x})$ for every time point, we need to recalculate $g(t, \mathbf{x})$ for every time t when Cox-Time is used, making the Cox-Time method much more computationally expensive compared to Cox-CC.

2.4 Data Sources and Software

We gathered our data from the Molecular Taxonomy of Breast Cancer International Consortium (METABRIC) dataset which contains gene expression data and clinical features for 1,980 breast cancer patients. The analysis is made using the Python programming language and its packages PyTorch, pycox, and NumPy.

2.5 Our Analysis

For the project we used a python package called "pycox" designed for survival analysis and time-to-event prediction with PyTorch. After loading the METABRIC data set, splitting it in train, test and validation (where the test set and validation set each constitute 20% of the original dataset), and feature transforming, we created a simple MLP with two hidden layers: ReLU activations, batch norm and dropout. For this, we used MLPVanilla function (for both Cox-CC and Cox-PH) and MLPVanillaCox-Time (for Cox-Time) from the "torchtuples" python package. We trained all three models and obtained survival estimates for the test set for prediction. The training of the neural network was limited to a maximum of 512 epochs, with early stopping enabled. However, in practice, it only took about 15 to 30 epochs for the neural network to finish training in each experiment. For speed and visibility, we have also limited the prediction to predict the survival rate only on 5 individuals.

For the evaluation criteria, we used three metrics: time-dependent concordance index, the integrated IPCW Brier score (inverse probability of censoring weighted Brier score), and the integrated IPCW (negative) binomial log-likelihood. Of these metrics:

- The Concordance Index is explained in further details in Section 4.1 of [2]. It is a measure of whether for a random pair of individuals, the predicted survival times of the two individuals have the same ordering as their true survival times. [2] The higher the Concordance Index, the better.
- The Integrated Brier Score (IBS) is the integral of the IPCW Brier score across the entire simulated duration. The Brier score can be thought of as the mean-squared error of the survival probability estimates at a certain time [2]. Hence, a lower IBS is more preferable. The IBS is explained in more details in Section 4.2 of [2].
- The Integrated Negative Binomial Log-Likelihood (INBLL) is similar to the Integrated Brier Score in motivation, but with a different mathematical formulation based on log-likelihood instead of mean-squared error. The INBLL is explained in more details in Section 4.3 of [2].

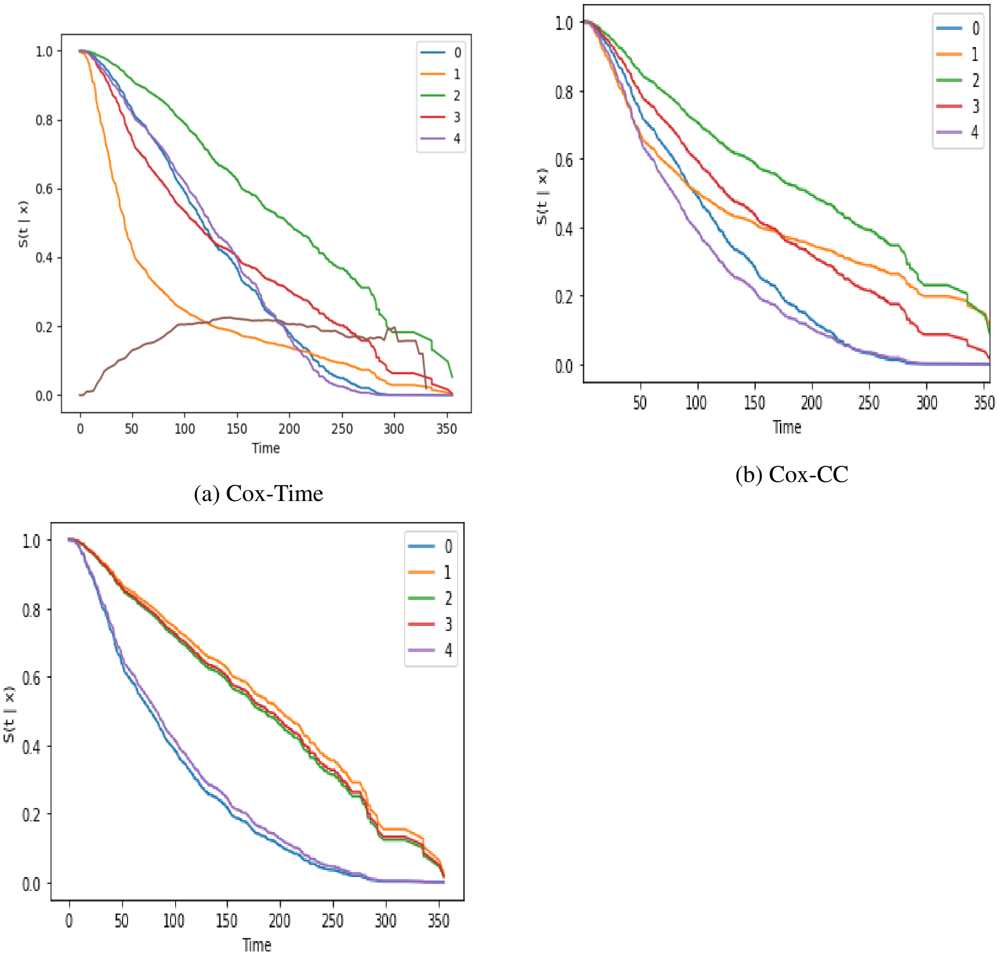
3 Results and Discussion

As was stated above, we used the concordance index which evaluates a method's discriminative performance, and the Brier score and binomial log-likelihood which also evaluate the calibration of the survival estimates. For the METABRIC dataset, the following results presented in Table 1 were obtained for the three models we used - Cox-CC, Cox-Time, Cox-PH. In this experiment, the program was initialized with the seed "1234". The PyTorch package in particular was also initialized with the seed "123".

Method	Concordance	IBS	INBLL
Cox-Time	0.659	0.169	0.503
Cox-CC	0.656	0.167	0.497
Cox-PH	0.654	0.167	0.495

Table 1: Evaluation scores of an experiment using Cox-Time, Cox-CC and Cox-PH to predict breast cancer survival time using METABRIC.

The survival rate prediction with each of these three methods in this first experiment is represented in Figure 1 below, where 0, 1, 2, 3, 4 in the legend denotes the 5 different individuals whose survival rate over time is being predicted and the time is measured in months:



(a) Cox-Time

(b) Cox-CC

(c) Cox-PH

Figure 1: Predicted survival rate over time of 5 individuals in the METABRIC data

We are not entirely sure what the brown line in the graph for Cox-Time (Figure 1a) represents, but we guess that it is the baseline hazard function $h_0(t)$.

With all 3 methods, the 5 individuals have their survival rates flatten out (typically at 0.2 or below) past the 300 months mark, and have very low survival rates past the 350 months mark. Hence, there are no stark differences between the 3 methods in terms of how long a breast cancer patient is expected to survive.

The experiment was then repeated 4 more times, each with a different (but consistent) seed for the program and the PyTorch package. The means of the scores received from all 5 simulations are compiled in Table 2 below:

Method	Concordance	IBS	INBLL
Cox-Time	0.673	0.166	0.499
Cox-CC	0.650	0.168	0.507
Cox-PH	0.648	0.169	0.511

Table 2: Means of evaluation scores of 5 simulations using Cox-Time, Cox-CC and Cox-PH.

The scores did not have large variations across the 5 simulations. For example, the concordance index for Cox-Time varied from 0.649 to 0.704, a difference of less than 10%.

The three models (Cox-Time, Cox-CC and Cox-PH) are very comparable in performance. As noted in Section 2.3, Cox-Time is a more flexible version of Cox-CC, so naturally, Cox-Time returned slightly better results than Cox-CC with a higher Concordance Index, lower IBS and lower INBLL. However, as mentioned in Section 2.3, Cox-Time is much more computationally expensive than Cox-CC and in this case, the difference in performance is so small that Cox-CC might still be worth considering for predicting the survival rate of breast cancer patients using the METABRIC data. We can deduce, then, that the hazard function is proportional for these 5 individuals in the METABRIC data. The Cox-CC model, in turn, performs very slightly better than the Cox-PH model.

4 Conclusion

In this project, we explored extensions of the Cox proportional hazards model to predict breast cancer survival time using the METABRIC dataset. We used models proposed by Kvamme, Borgan and Scheel [2], namely Cox-Time which is a relative risk model that extends Cox regression beyond the proportional hazards, Cox-CC - a proportional version of the Cox-Time model, and Cox-PH - a Cox proportional hazards model also referred to as DeepSurv.

Parametrizing the Cox relative risk function with neural networks allows the team to model rich interactions between covariates and event time, which, in turn, results in models that are no longer constrained by the proportionality assumption of the Cox model. After analyzing these models' performance using 3 different metrics, we conclude that the Cox-Time model has the best overall performance in terms of concordance index, integrated Brier score (IBS) and integrated binomial log-likelihood (IBLL), in exchange for using more computational resources. We can also note that all three models perform almost similarly.

References

[1] Katzman, J.L., Shaham, U., Cloninger, A., Bates, J., Jiang, T. & Kluger, Y. (2018) *Deepsurv: personalized treatment recommender system using a Cox proportional hazards deep neural network.* BMC Medical Research Methodology, 18(1), 2018. Retrieved March 1, 2022, from `https://bmcmedresmethodol.biomedcentral.com/articles/10.1186/s12874-018-0482-1`

[2] Kvamme, H., Borgan, Ø. & Scheel, I. (2019) *Time-to-Event Prediction with Neural Networks and Cox Regression.* Journal of Machine Learning Research 20 (2019) 1-30

[3] *Cox (proportional hazards) regression*. StatsDirect. Retrieved March 1, 2022, from `https://www.statsdirect.com/help/survival_analysis/cox_regression.htm`