UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Modeling of long-term survival data with unobserved dispersion via neural network

Teh Led Red

Dissertação de Mestrado do Programa Interinstitucional de Pós-Graduação em Estatística (PIPGEs)

ICMC USP SÃO CARLOS

Teh Led Red

Modeling of long-term survival data with unobserved dispersion via neural network

Master dissertation submitted to the Institute of Mathematics and Computer Sciences – ICMC-USP and to the Department of Statistics – DEs-UFSCar, in partial fulfillment of the requirements for the degree of the Master Interagency Program Graduate in Statistics. *FINAL VERSION*

Concentration Area: Statistics

Advisor: Prof. Dr. Vicente Garibay Cancho

USP – São Carlos
December 2023

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

Teh, Led Red
T253m Modelagem via redes neurais de dados de sobrevivência de longa duração com dispersão não observada / Led Red Teh; orientador Vicente Garibay Cancho. -- São Carlos, 2023.
73 p.

Dissertação (Mestrado - Programa Interinstitucional de Pós-graduação em Estatística) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2023.

1. ANÁLISE DE SOBREVIVÊNCIA. I. Cancho, Vicente Garibay, orient. II. Título.

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

Teh Led Red

Modelagem via redes neurais de dados de sobrevivência de longa duração com dispersão não observada

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP e ao Departamento de Estatística – DEs-UFSCar, como parte dos requisitos para obtenção do título de Mestra em Estatística – Programa Interinstitucional de Pós-Graduação em Estatística. *VERSÃO REVISADA*

Área de Concentração: Estatística

Orientador: Prof. Dr. Vicente Garibay Cancho

USP – São Carlos
Dezembro de 2023

UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia
Programa Interinstitucional de Pós-Graduação em Estatística

Folha de Aprovação

Defesa de Dissertação de Mestrado da candidata Teh Led Red, realizada em 08/12/2023.

Comissão Julgadora:

Prof. Dr. Vicente Garibay Cancho (USP)

Prof. Dr. Edwin Moises Marcos Ortega (ESALQ/USP)

Prof. Dr. Fabio Nogueira Demarqui (UFMG)

O Relatório de Defesa assinado pelos membros da Comissão Julgadora encontra-se arquivado junto ao Programa Interinstitucional de Pós-Graduação em Estatística.

This work is dedicated to my family whom I hold dear. To my beloved husband, Chee, who has always believed and supported me. To my daughter, Clara, and son, Caio Vinicius, may this motivate them to persevere in their lifelong learning journey and inspire them to achieve their dreams.

ACKNOWLEDGEMENTS

As Steve Jobs once said, “Great things in business are never done by one person, they are done by a team of people.” With that, I am deeply grateful to my advisor, Prof. Dr. Vicente Garibay Cancho, for his unwavering patience and invaluable guidance in the course of my work. I would also like to to extend my sincere appreciation to the members of my qualification examination committee, Prof. Dr. Josemar Rodrigues and Prof. Dr. Antonio Carlos Pedroso de Lima, for their diligent and meticulous review, which included identifying errors and providing extremely helpful suggestions for improvement in my work. A special thanks goes to Professor Josemar for his thoughtful consideration and generous insights/guides into potential avenues for future research expansion.

The supplementary resources, including the Python implementation code and the medical images dataset, provided by the authors in the referenced article (XIE; YU, 2021b), greatly eased the challenges in executing the simulation studies and application within this work. I would also like to acknowledge the support and motivation from my colleagues, Juan, who has always pushed me to the improvement and completion of this work, and Amanda, for her assistance in editing the Portuguese text. Both of them are Ph.D. graduates from PIPGEs.

The pursuit of my master’s degree would not have been possible without the steadfast support of my husband. I am profoundly thankful for his understanding, emotional support, and assistance in revising the English text. I would also like to thank my two lovely children who have been a pillar of strength.

Last but not least, I would to end by saying that I owe a mountain of debt to this beautiful country, Brazil, which I have adopted as my home and thank her for the opportunity to fulfill a dream that I thought was not possible. Obrigada, Brasil.

“I have not failed.
I’ve just found 10,000 ways that won’t work.”
(Thomas A. Edison)

RESUMO

TEH, L. R. **Modelagem via redes neurais de dados de sobrevivência de longa duração com dispersão não observada**. 2023. 73 p. Dissertação (Mestrado em Estatística – Programa Interinstitucional de Pós-Graduação em Estatística) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2023.

Os modelos tradicionais na análise de sobrevivência pressupõem que todos os sujeitos eventualmente experimentarão o evento de interesse do estudo, como a morte ou a recidiva de uma doença, de modo que a função de sobrevivência é própria. O modelo de cura, proposto há setenta anos, é usado para incorporar uma fração de cura. Isso significa que existe uma fração de indivíduos que nunca experimentarão o evento de interesse, que pode ser tratado imune ou curado no contexto de tratamento oncológico. Na literatura, vários modelos de cura foram amplamente estudados e geralmente foram aplicados aos dados estruturados com pouca quantidade de covariáveis. O uso de rede neural convolucional, uma técnica poderosa de aprendizado profundo para o processamento de imagens, tem crescido rapidamente na área médica nos últimos anos. Imagens médicas, como imagens histológicas e ressonâncias magnéticas (RMIs), estão diretamente relacionadas aos fatores prognósticos de um paciente, tornando razoável introduzi-las como preditoras no modelo de cura. Com base no artigo de Xie and Yu (2021b), no qual uma rede neural foi usada para modelar os efeitos das preditoras não estruturadas no modelo de tempo de promoção, faremos uma expansão para casos em que os dados apresentam sobredispersão. Chamaremos nossa extensão de modelo de cura de binomial negativa integrado, e a estimação dos parâmetros será realizada por meio do algoritmo de Expectativa-Maximização.

Palavras-chave: Análise de sobrevivência, Modelo de cura, Rede Neural Convolucional, Conjunto de dados da doença de Alzheimer OASIS-3.

ABSTRACT

TEH, L. R. **Modeling of long-term survival data with unobserved dispersion via neural network**. 2023. 73 p. Dissertação (Mestrado em Estatística – Programa Interinstitucional de Pós-Graduação em Estatística) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2023.

Traditional models in survival analysis assume that every subject will eventually experience the event of interest in the study, such as death or disease recurrence, so the survival function is said to be proper. Cure rate model, which was first proposed seven decades ago, has since been used to account for the presence of cure fraction, this means that a certain fraction of the individuals will never experience the occurrence of an event of interest for which they can be treated as immune or cured subjects in the context of cancer treatment. In the literature, various cure rate models have been widely studied and commonly applied to structured data with small quantities of covariates. The use of convolutional neural network, a powerful deep learning technique for image processing problem, has become increasingly more common in the medical field in recent years. Medical images such as histological slides and magnetic resonance images (MRIs) are directly related to a patient's prognostic factors, therefore, it is reasonable to introduce these images as predictors in cure model. In this work, we extend upon the article of Xie and Yu (2021b) in which a neural network was used to model the unstructured predictor's effect in the promotion time cure model's setting to the cases of overdispersed data. We will call our extension as integrated negative binomial cure rate model, and its parameters will be estimated through the Expectation-Maximization algorithm.

Keywords: Survival analysis, Cure rate model, Convolutional Neural Network, OASIS-3 Alzheimer's disease dataset.

LIST OF FIGURES

LIST OF ALGORITHMS

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

AI	Artificial Intelligence
ANN	Artificial Neural Networks
AUC	Area Under the ROC Curve
cdf	cumulative distribution function
CDR	Clinical Dementia Rating
CNN	Convolutional Neural Networks
DL	deep learning
DNN	Deep neural networks
EM	Expectation-Maximization
FPR	False Positive Rate
GPU	Graphics Processing Units
iid	independent and identically distributed
INBCM	integrated negative binomial cure rate model
IPCM	integrated promotion time cure rate model
MCM	mixture cure model
ML	machine learning
MLE	Maximum Likelihood Estimator
MLP	Multilayer Perceptron
MRI	Magnetic Resonance Imaging
MSE	mean squared error
PCM	promotion time cure model
pdf	probability density function
pgf	probability generating function
pmf	probability mass function
ReLU	Rectified Linear Unit
RNN	recurrent neural networks
ROC	Receiver Operating Characteristic Curve
SGQ	Stochastic Gradient Descent
SVM	support vector machine
SVR	support vector regression
TPR	True Positive Rate
XGBoost	Extreme Gradient Boosting

LIST OF SYMBOLS

S — Survival Function

P — Probability

F — Cumulative Distribution Function (CDF)

f — Probability Density Function

λ — Hazard function

Λ — Cumulative hazard function

G — Probability generating function

$\mathbb{E}$ — Expectation

$\mathbb{R}$ — Real number set

CONTENTS

CHAPTER

1

INTRODUCTION

Survival analysis is a branch of statistics that deals with data modeling in which the response variable is the period of time until the occurrence of a specific event of interest. This can encompass various scenarios, such as the duration of time from the diagnosis of a patient's illness until its passing, the duration until cancer recurrence after initial treatment, the lifespan of an equipment before failure, the number of days until a user discontinues its service subscription, and more. In traditional survival analysis models, which account for censored data, the assumption typically holds that all individuals in the study are equally susceptible to the specified event. However, this assumption does not always hold true, particularly in healthcare. Thanks to medical advancements, there are cases where individuals are not only able to prolong their lives or have their illnesses brought under control, but they have also achieved complete recovery. The presence of a proportion of individuals who are immune or not susceptible to the event's occurrence makes the use of the traditional model impractical. This proportion is generally termed as the cure fraction or cure rate.

Models that incorporate the cure fraction in survival analysis are commonly referred to as cure models. Amico and Keilegom (2018) extensively reviewed some principal models proposed in the literature. This review delved into various aspects of these models, including their construction, identifiability, inference, predictive capabilities, and the assessment of goodness of fit. In summary, the reviewed models, based on the different techniques or approaches, can be grouped into mixture models, promotion time models, and unified models. For a thorough understanding of the decision-making process regarding the selection of an appropriate model, as well as their interpretation and comparison of the advantages inherent in each approach, readers are encouraged to refer to Legrand and Bertrand (2019, Chapter 4).

The mixture model proposed by Berkson and Gage (1952) contains a binary component that represents the proportion of cured and uncured individuals in the survival function of the entire population. Drawing inspiration from the biological processes that underlie tumor recurrence, Yakovlev, Tsodikov and Asselain (1996) introduced the promotion time cure model

as an alternative to the mixture model. For a comprehensive understanding on how to apply this modeling technique through parametric, non-parametric, and semi-parametric approaches, one can refer to the thorough review article authored by Tsodikov, Ibrahim and Yakovlev (2003). Furthermore, this model offers an advantage of being applicable in the Bayesian approach. The unified model, initially presented by Yin and Ibrahim (2005), represents a more generalized version as it encompasses both the mixture model and the promotion time model as specific cases. Its construction is based on the Box-Cox transformation.

The models mentioned in the previous paragraphs have been commonly applied in the field of healthcare and demonstrated good inferential and predictive capabilities. However, their utility has been constrained to structured data and low-dimensional datasets featuring only a limited number of covariates. With the development of powerful GPUs designed for processing complex data, Convolutional Neural Networks (CNN), one of the deep learning techniques, have been increasingly used in computer vision. Meyer *et al.* (2018) conducted research and proposed procedures for integrating deep learning techniques into radiotherapy study, while Lundervold and Lundervold (2019) compiled various CNN architectures specifically applied for Magnetic Resonance Imaging (MRI) tasks.

Driven by the understanding that the quantity of cancerous cells appeared in histological images and the condition of organs demonstrated in MRIs may hold relevance to the pathogenic factors affecting oncology patients, these medical images can be introduced via CNN as predictors into the promotion time model (XIE; YU, 2021b). Expanding upon this approach, where the number of cancer cells is assumed to follow a Poisson distribution, we aim to extend it to accommodate scenario of overdispersed data by considering the negative binomial distribution, which is well-known in the literature for its effectiveness in modeling count data with greater variability. The Poisson distribution is actually a limiting case when the dispersion parameter in negative binomial approaches zero.

The organization of this work is described as follows: firstly, in Chapter 2, we will provide a brief overview of the fundamental elements for survival analysis. Then, Chapter 3 covers a simple understanding of neural networks, encompassing their structures, the back-propagation algorithm employed for parameter estimation, some commonly used activation functions, a concise introduction to the CNN and concluded with a brief exploration of the adaption of machine learning and deep learning techniques for survival analysis. Chapter 4 delves into the proposed cure model, namely the integrated two-stage cure rate model. The term “integrated” represents the integration of CNN in the cure rate model in order to introduce the images as predictors, while “two-stage” gives a more explicit idea on the model’s construction. We will study the model’s construction, estimation procedures using the Expectation-Maximization (EM) algorithm and the two resulting models based on different discrete distributions in this chapter. Chapter 5 presents the results of simulation study. We will compare the performance of the proposed model on simulated Poisson and negative binomial scenarios. We should expect

the proposed negative binomial model to give a more reasonable estimation result for the overdispersed data. Finally, in Chapter 6, we will apply the integrated negative binomial cure model to the OASIS-3 dataset, which is the same dataset employed by Xie and Yu (2021b); we will then compare the outcomes achieved in our application to those presented in the referenced article.

CHAPTER

2

BASIC CONCEPTS

In this chapter, we will review some fundamental concepts used in survival analysis as outlined in the book authored by Colosimo and Giolo (2006).

2.1 Censoring

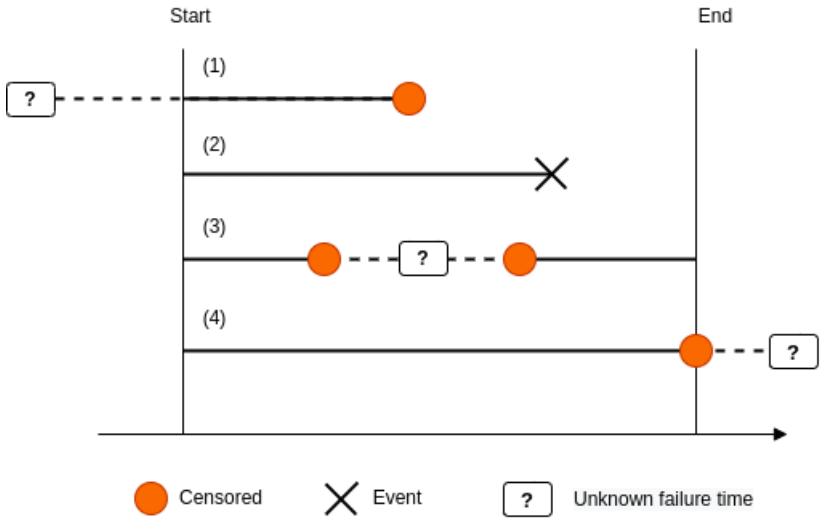
What distinguishes survival analysis from other statistical models is the presence of censored data. Censoring occurs when it becomes impossible to observe the exact failure times for some individuals in a study. There can be various reasons causing the absence of an observed event, such as the dropping out of a participant before the study ends or instances where the event of interest did not occur because of the effectiveness of a treatment. Eventually, the observed time for censored data is a partial or incomplete record because the exact failure time exceeds the time range adopted in the study.

In general, there are three types of censoring, classified based on the sequencing of failure time occurrence and the observed time as explained below:

1. Left censoring: The exact failure time occurs prior to the commencement of the study. In this context, the true failure time is shorter than the observed time.
2. Interval censoring: The failure time falls within a specified time range.
3. Right censoring: The exact failure time surpasses the observed time.

The presence of censored data poses a challenge for statistical modeling. The lack of knowledge about the exact failure times of these data means that simply removing them from the dataset could lead to valuable information being lost, which can potentially introduce bias into the estimated survival times. Consequently, survival analysis seeks methods that take into account the censored data to accurately estimate the distribution of failure time through statistical modeling.

Figure 1 – Types of censoring: (1) Left censoring; (2) No censoring; (3) Interval censoring and (4) Right censoring.



According to Cox D.R. (1984), the determination of failure time necessitates three crucial elements: the precise definition of the initial time, the time measurement scale must be consistent with the experimental context, and a clear definition of the event of interest (failure). To address the challenge posed by censored data, the solution involves the introduction of a censoring indicator variable (δ). In a study with sample size of n, the i-th individual's data is represented by a triplet (X_i, y_i, δ_i), where

- $X_i \in \mathbb{R}^{1\times P}$ is a vector of covariates,
- δ_i is an indicator variable, $\delta_i = 0$ if it is a censored data and $\delta_i = 1$ otherwise, and
- y_i represents the observed time,

$$y_i = \begin{cases} T_i & \text{if } \delta_i = 1 \\ C_i & \text{if } \delta_i = 0 \end{cases} \tag{2.1}$$

in which T_i is the observed failure time and C_i the censored time.

Note that for each individual i, either the failure time T_i or the censoring time C_i can be observed. Therefore, if the data is of the right-censoring type, which is more common in practice, it is easy to see that $y_i = \min(T_i, C_i)$.

2.2 The description of survival data

In survival analysis, the response variable is denoted as T and represents the time until the event of interest or failure occurs, T is a non-negative, continuous variable. It is typically characterized by two fundamental functions: the survival function and the hazard function (or failure rate). The survival function, denoted as S, encapsulates the probability of an observation

not experiencing the event until the time t. This function exhibits a monotonically decreasing behavior over time.

$$S(t) = P(T \geq t), \quad 0 < S(t) < 1 \text{ e } t > 0. \tag{2.2}$$

At the initial time ($t = 0$), all individuals are alive, so $S(t) = 1$ and the function will decrease as t becomes larger, $\lim_{t\to\infty} S(t) = 0$. Conversely, the cumulative distribution function (cdf), denoted as $F(t)$, represents the probability that an observation does not survive until time t. Clearly, F is the complementary function of S, $F(t) = 1 - S(t)$.

One can obtain the probability density function (pdf) f by differentiating F, such that $f(t) = \frac{d}{dt}F(t)$, which satisfies the normalization condition.

$$\int_0^\infty f(u)du = 1. \tag{2.3}$$

Then, Equation 2.2 can be expressed by

$$S(t) = \int_t^\infty f(u)du. \tag{2.4}$$

The failure rate within an interval $[t, t+\Delta t)$ is defined as the probability of an event occurring within this interval, given that it did not occur before t, divided by the width of the interval. As $\Delta t \to 0$, the hazard function or failure rate becomes an instantaneous failure rate at time t, conditioned on no failure occurring up to time t. Its mathematical expression is as follows:

$$\lambda(t) = \lim_{\Delta t\to 0} \frac{\Pr(t \leq T < t+\Delta t \mid T \geq t)}{\Delta t} = \lim_{\Delta t\to 0} \frac{F(t+\Delta t) - F(t)}{\Delta t \times S(t)} = \frac{f(t)}{S(t)}. \tag{2.5}$$

The λ is a non-negative function and can exhibit various curve shapes. Typical shapes include monotonically increasing (e.g., aging case) or decreasing (e.g., rejuvenation case), constant, and U-shaped. The behavior of the λ function can offer guidance in selecting an appropriate parametric model. Models with an increasing λ function are most frequently used in cancer epidemiological studies. The cumulative hazard function is expressed as follows:

$$\Lambda(t) = \int_0^t \lambda(u)du.$$

In short, the relationships between the characterization functions of $t \in [0, \infty)$ are:

- $f(t) = -\frac{d}{dt}S(t)$;
- $\lambda(t) = \frac{f(t)}{S(t)} = -\frac{d}{dt}S(t) \times \frac{1}{S(t)} = -\frac{d}{dt}\big[\ln S(t)\big]$;
- $S(t) = \exp\big\{-\Lambda(t)\big\}$.

Returning to the notation for observed time in Equation 2.1, let $(Y_i, \delta_i), i = 1, \ldots, n$, be independent and identically distributed (iid) from (Y, δ), the likelihood function for survival data

is given by:

$$L = \prod_{i=1}^{n} f(Y_i)^{\delta_i} S(Y_i)^{1-\delta_i}$$
$$= \prod_{i=1}^{n} \{\lambda(Y_i)\}^{\delta_i} S(Y_i),$$

where $f(Y_i) = \lambda(Y_i) S(Y_i)$. It is intuitive that uncensored observations will contribute to the likelihood of incidence rates, while censored observations will contribute to the likelihood of survival time distribution. In the case of the cure model, the likelihood retains the same form, with the exception of the interpretation of the survival function, which becomes an improper distribution.

2.3 Improper distribution

One of the assumptions in survival analysis is that all individuals in the study will experience the event of interest given a sufficiently long follow-up time. However, there are situations in which a proportion of individuals is immune or not susceptible to failure. These individuals will never experience the failure event, leading to $\lim_{t\to\infty} S(t) > 0$. As per Yakovlev, Tsodikov and Asselain (1996, Section 1.2), the cure model can be formulated from an improper survival distribution and we present its mathematical meaning as follows.

Consider a relaxed normalization condition in Equation 2.3 with

$$\int_0^\infty f(u)du = A \leq 1. \tag{2.6}$$

Therefore the condition in terms of survival function is

$$\lim_{t\to\infty} S(t) = 1 - A. \tag{2.7}$$

And a more generalized version of Equation 2.4 is given by

$$S(t) = 1 - A + \int_t^\infty f(u)du, \quad t \geq 0.$$

Mathematically, we say that the survival time distribution is improper whenever $A < 1$, $1 - A$ is a constant value and is called the survival fraction. This fraction can be interpreted as a probability of cure when examining tumor recurrence. It can also be interpreted as the expected proportion of subjects that would remain tumor-free after being exposed to a chemical carcinogen.

The construction of the cure rate model will be studied with more details in Chapter 4.

2.4 Probability generating function

The probability generating function (pgf) is a mathematical function that can be used as a technique for specifying the distribution of a discrete random variable and estimating its

parameters. Below, we present its mathematical definition, key properties, and examples for the Poisson and negative binomial distributions.

Definition 1. Let X be a random discrete variable of some non-negative integers $\{0,1,2,\ldots\}$. A pgf of X is defined as

$$\begin{aligned} G_X(s) &= \mathbb{E}(s^X) \\ &= \sum_{x=0}^{\infty} s^x \, P(X=x), \quad \forall s \in \mathbb{R} \text{ for which the sum will converge.} \end{aligned} \tag{2.8}$$

Some of its properties:

1. $G_X(1) = 1$.

Proof. $G_X(1) = \sum_{x=0}^{\infty} 1^x P(X=x) = \sum_{x=0}^{\infty} P(X=x) = 1$ □

2. $\mathbb{E}(X) = G'_X(1)$.

Proof.

$$\begin{aligned} G_X(s) &= \sum_{x=0}^{\infty} s^x \, P(X=x) = \sum_{x=0}^{\infty} x \, s^{x-1} \, P(X=x) \\ \Rightarrow \quad G'_X(1) &= \sum_{x=0}^{\infty} x \, P(X=x) = \mathbb{E}(X) \end{aligned}$$

□

If $X \sim \text{Poisson}(\theta)$ with probability mass function (pmf) $\mathbb{P}(X=x) = \frac{\theta^x e^{-\theta}}{x!}$, $x = 0,1,2,\cdots$, then, its pgf is given by

$$\begin{aligned} G_X(s) = \sum_{x=0}^{\infty} s^x \, P(X=x) &= \sum_{x=0}^{\infty} s^x \, \frac{\theta^x e^{-\theta}}{x!} \\ &= e^{-\theta} \sum_{x=0}^{\infty} \frac{(\theta s)^x}{x!} \\ &= e^{-\theta} e^{\theta s} \\ &= e^{-\theta(1-s)}, \quad \forall s \in \mathbb{R}. \end{aligned}$$

If $X \sim$ negative binomial(ϕ, θ) with pmf is in the form of $P(X=x) = \frac{\Gamma\left(\frac{1}{\phi}+x\right)}{\Gamma\left(\frac{1}{\phi}\right) x!} \left(\frac{\phi\theta}{1+\phi\theta}\right)^x \left(\frac{1}{1+\phi\theta}\right)^{\frac{1}{\phi}}$,

$x = 0, 1, 2, \cdots, \theta > 0$ e $\phi > -\frac{1}{\theta}$, then, its pgf is given by

$$
\begin{aligned}
G_X(s) &= \sum_{x=0}^{\infty} s^x \, P(X = x) \\
&= \sum_{x=0}^{\infty} s^x \, \frac{\Gamma\left(\frac{1}{\phi} + x\right)}{\Gamma\left(\frac{1}{\phi}\right) x!} \left(\frac{\phi\theta}{1+\phi\theta}\right)^x \left(\frac{1}{1+\phi\theta}\right)^{\frac{1}{\phi}} \\
&= \sum_{x=0}^{\infty} \frac{\Gamma\left(\frac{1}{\phi} + x\right)}{\Gamma\left(\frac{1}{\phi}\right) x!} \left(\frac{\phi\theta s}{1+\phi\theta}\right)^x \left(\frac{1}{1+\phi\theta}\right)^{\frac{1}{\phi}} \\
&= \left(\frac{1}{\cancel{1+\phi\theta}}\right)^{\frac{1}{\phi}} \left(\frac{\cancel{1+\phi\theta}}{1+\phi\theta(1-s)}\right)^{\frac{1}{\phi}} \sum_{x=0}^{\infty} \frac{\Gamma\left(\frac{1}{\phi} + x\right)}{\Gamma\left(\frac{1}{\phi}\right) x!} \left(\frac{\phi\theta s}{1+\phi\theta}\right)^x \left(\frac{1+\phi\theta(1-s)}{1+\phi\theta}\right)^{\frac{1}{\phi}} \\
&= \left(\frac{1}{1+\phi\theta(1-s)}\right)^{\frac{1}{\phi}}, \quad \forall s \in \mathbb{R}.
\end{aligned}
$$

2.5 Piecewise exponential distribution

In this work, we will consider the survival times of risk factors following the piecewise exponential distribution as in Xie and Yu (2021b). The piecewise exponential distribution is an extension of the exponential distribution. While the choice of the latter assumes a constant failure rate for survival data, the piecewise exponential distribution allows for the varying of the hazard curve over time. In this distribution, the observed times are partitioned into multiple intervals, and the failure rate is assumed to remain constant within each interval but may vary between intervals.

The construction of the distribution is as follows: we do a finite partition J on the time axis, $0 < s_1 < \cdots < s_J$ such that $s_J > \max\limits_{1 \le i \le n} (y_i)$, then, there will be J time intervals $(0, s_1], (s_1, s_2], \ldots, (s_{J-1}, s_J]$ and a hazard rate for the j-th interval denoted as $\lambda_j, j = 1, \cdots, J$, $\lambda_j > 0$. The cdf for the survival times is given by

$$
F(t \mid \boldsymbol{\lambda}) = 1 - \exp\left\{-\lambda_j \left(t - s_{j-1}\right) - \sum_{g=1}^{j-1} \lambda_g \left(s_g - s_{g-1}\right)\right\}. \tag{2.9}
$$

When $J = 1$, the $F(t \mid \lambda)$ will reduce to a normal exponential distribution. Note that the degree of the nonparametricity is controlled by J, the model tends to be more nonparametric if the values of J are getting big. It is recommended to choose a small to moderate value for J, between 5 and 7, in cure rate modeling. Xie and Yu (2021b) adopted $J = 5$ in their work as instructed by Chen and Ibrahim (2001).

CHAPTER

3

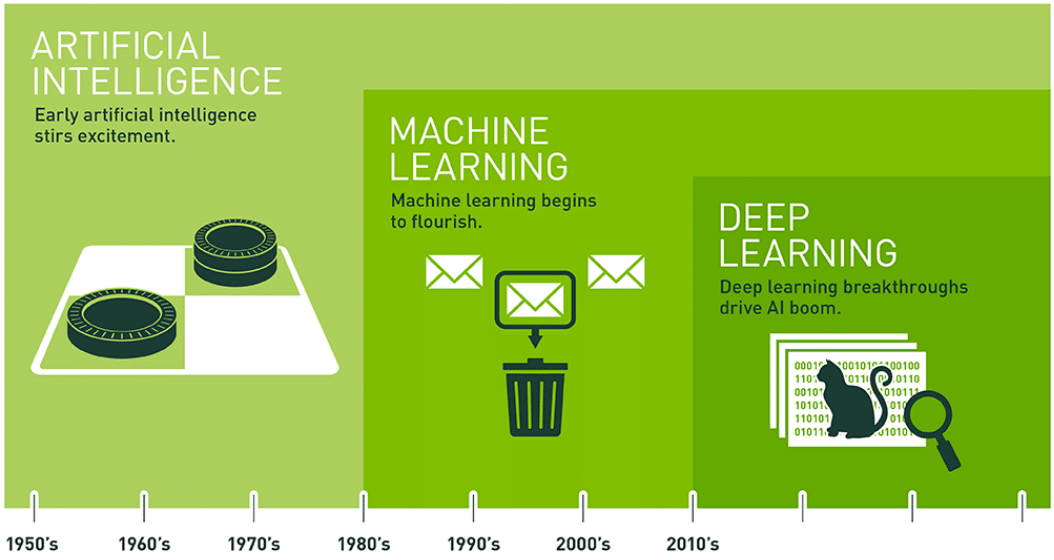
NEURAL NETWORKS AND DEEP LEARNING

The creation of Artificial Neural Networks (ANN), initially inspired by the learning mechanism in biological organisms, can be traced back to the 1940s. The introduction of the perceptron algorithm by Rosenblatt in the late 1950s sparked considerable excitement within the field of Artificial Intelligence (AI). However, the single layer perceptron has its own constraints due to the limited ability to model only linear separable problems and its inability to tackle the XOR[1], or "exclusive or", problem. Moreover, the lack of computational infrastructure had led to a decade-long period of stagnation in the exploration of this technique, often referred to as the "winter era" in the annals of artificial intelligence. The resurgence of ANNs began in the mid-1980s with the advent of the backpropagation algorithm and the development of the Multilayer Perceptron (MLP). This breakthrough not only resolved the aforementioned XOR problem but also made parameter estimation a feasible task. Subsequently, with the easy access to data and the emergence of Graphics Processing Units (GPU) that facilitates parallel computing, there has been a significant growth of network architectures with increased numbers of hidden layers since 2012. Such networks are now commonly referred to as deep learning in the literature.

Figure 2 illustrates succinctly the evolution of AI, starting from the early attempts to train a machine to play checkers in the 1950s up until the early 2010s when machine learning (ML) with its series of algorithmic developments for classification tasks, could distinguish spams from regular emails for example. The deep learning (DL) techniques have demonstrated their ability to successfully solving numerous challenging tasks and through varying their architecture structures, they can achieve levels of performance almost comparable to tasks executed by humans, like image classification, text analysis, speech recognition, and so on. The techniques are also particularly useful when applied to high-dimensional data where traditional statistical methods fall short in producing the desired results.

[1] <https://www.educative.io/answers/xor-problem-in-neural-network>

Figure 2 – The evolution of AI for the last five decades. Source: blog nvidia.



Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.

In the following sections, we will briefly examine the structure of a neural netwrok, the application of the backpropagation algorithm in parameter estimation, various network tuning parameters, and provide a simple description of CNN's layers. The contents is drawn from Efron and Hastie (2016, Chapter 18) and Roberts, Yaida and Hanin (2022, Chapter 2.2). We will conclude this chapter by exploring the recent adaption of deep learning methods for survival analysis.

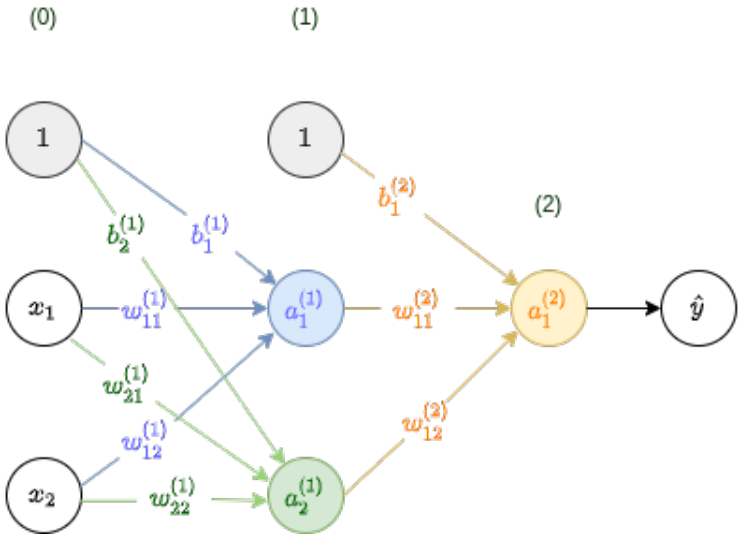
3.1 Structure of a neural network

A neural network is a profoundly parameterized model and is used as a function approximator. The process of fitting the model's parameters is commonly denoted as a learning algorithm. Figure 3 gives an example of a fully connected feed-forward neural network. This network consists of an input layer containing two predictors, x_1 and x_2, and an intermediate hidden layer with two neurons. In this hidden layer, each neuron $a_l^{(1)}$ can be expressed mathematically by

$$a_l^{(1)} = \sigma^{(1)}\left(\sum_{j=1}^{2} w_{lj}^{(1)} x_j + b_l^{(1)}\right), \; l = 1,2, \tag{3.1}$$

where the super-scripted (1) means that the component belongs to layer 1. Each neuron is connected to the neurons of the input layer through some weight parameters ($w^{(1)}$) and bias ($b^{(1)}$). The index notation used in Equation 3.1 follows the opposite direction to the layer order. In this context, w_{lj} represents the weight connecting the j-th neuron of the preceding layer to the l-th neuron of the current layer. While this order may initially seem confusing, it becomes more

Figure 3 – Diagram of a neural network containing only one hidden layer.



intuitive when represented in matrix form. We can simply say that index l denotes neurons in the outgoing layer and the index j refers to neurons in the incoming layer. After weighing the values received from the preceding layer, it is applied to a non-linear function denoted as $\sigma^{(1)}$, this operation is also known as activation. A very commonly used activation function is the sigmoid function, defined as $\sigma(x) = \dfrac{1}{1-e^{-x}}$. The output layer contains only one neuron, denoted as $a_1^{(2)}$. It is obtained through a similar operation as in $a_1^{(1)}$, i.e., $a_1^{(2)} = \sigma^{(2)}\left(\sum_{j=1}^{2} w_{1j}a_j^{(1)} + b_1^{(2)}\right)$. Note that the activation function $\sigma^{(2)}$ can be a different non-linear function.

In their work, Hornik, Stinchcombe and White (1989) introduced the Universal Approximation Theorem in which they claim that even with just one hidden layer and the use of arbitrary squashing functions, standard multilayer feedforward networks can approximate any function, as long as a sufficient number of hidden units are available. In this context, multilayer feedforward networks can be considered as universal approximators. When dealing with high-dimensional data, multiple layers are often employed, and there are numerous ways to combine the number of layers and the quantity of neurons in each layer. These two hyper-parameters - depth and width - determine a network's architecture. However, increased flexibility in network architecture can lead to issues like over-fitting, prompting the adoption of various regularization techniques to enhance generalization capacity.

A neural network can be succinctly described as a complex function, denoted as $f(\boldsymbol{x}, \mathbf{W})$, where $\mathbf{W}$ encompasses the complete set of weights. With a training dataset at hand and a chosen loss function, the network proceeds to learn the mapping from inputs to outputs. This process represents an optimization problem, as it searches for an appropriate set of weights $\mathbf{W}$ that can effectively make predictions using the function f on the test dataset. For regression problems, mean squared error can be used as the loss function accompanied by a linear activation function

at the output layer. Conversely, for classification problems, sigmoid or softmax functions can be chosen as activation functions and cross-entropy utilized as the loss function.

3.2 Parameter estimation (backpropagation)

As previously mentioned, a neural network can be described as a complex function $f(\boldsymbol{x}, \mathbf{W})$, where $\boldsymbol{x}$ represents the covariate vector and $\mathbf{W}$ is the collection of weights. Typically, a differentiable activation function is chosen. In the context of supervised learning, with a training dataset and a loss function L, the goal is to find the solution that minimizes the loss:

$$\underset{\mathbf{W}}{\operatorname{argmin}} \left\{ \frac{1}{n} \sum_{i=1}^{n} L\left[y_i, f\left(x_i; \mathbf{W}\right)\right] + \lambda J(\mathbf{W}) \right\} \tag{3.2}$$

where $J(\mathbf{W})$ stands for the regularization or penalty term, with λ serving as a non-negative tuning parameter. The network is trained via the backpropagation algorithm, utilizing gradient descent to seek the optimal values for $\mathbf{W}$ in order to minimize L. The computation of the gradient of L with respect to the elements of $\mathbf{W}$ employs the chain rule, since the neurons in each layer encapsulate functions of the previous layer.

Algorithm 1 outlines the steps for computing gradients of the first term in Equation 3.2 for the i-th observation. The idea is to compute $\delta_\ell^{(k)}$ which measures the error of each neuron contributes to the prediction of y. The $\delta_\ell^{(K)}$ values in the last layer can be easily obtained because the chain rule can be applied directly to $a_\ell^{(K)}$. In the intermediate layers, $\delta_\ell^{(k)}$ becomes a weighted sum of errors from neurons that take $a_\ell^{(k)}$ as inputs in layer $k+1$.

Using matrix notation, the Equation 3.3 becomes

$$\delta^{(K)} = \nabla_a L \odot \sigma'\left(z^K\right), \tag{3.6}$$

the Equation 3.4 becomes

$$\delta^{(k)} = \left(\left(w^{k+1}\right)^T \delta^{k+1}\right) \odot \sigma'\left(z^l\right), \tag{3.7}$$

and the Equation 3.5 becomes

$$\frac{\partial L[y, f(x; \mathbf{W})]}{\partial \mathbf{W}^{(k)}} = \delta^{(k+1)} a^{(k)'}, \tag{3.8}$$

in which $\odot$ denotes the Hadamard product. During the iterative process, the gradient descent updates

$$W^{(k)} \leftarrow W^{(k)} - \eta \left(\Delta W^{(k)} + \lambda W^{(k)}\right), k = 1, \ldots, K-1, \tag{3.9}$$

with $\Delta \mathbf{W}^{(k)} = \frac{1}{n} \sum_{i=1}^{n} \frac{\partial L\left[y_i, f\left(x_i; \mathbf{W}\right]\right.}{\partial \mathbf{W}^{(k)}}$ and η denotes the learning rate, $\eta \in (0, 1]$. When n becomes large, it is computationally more economical to use Stochastic Gradient Descent (SGQ). Instead of processing every observation in the dataset to update the estimated $\mathbf{W}$, the SGQ approach calculates $\Delta \mathbf{W}^{(k)}$ using batches, which are subsets of the complete dataset.

Algorithm 1 – Backpropagation algorithm

1: Given a pair of (x, y), calculates $a_l^{(k)}$ of each layer, $k = 2, \cdots, K$.
2: For every neuron in the output layer K, calculates

$$\begin{aligned}\delta_\ell^{(K)} &= \frac{\partial L[y, f(x, \mathbf{W})]}{\partial z_\ell^{(K)}} \\ &= \frac{\partial L[y, f(x; \mathbf{W})]}{\partial a_\ell^{(K)}} \sigma'^{(K)}\left(z_\ell^{(K)}\right).\end{aligned} \tag{3.3}$$

3: For every neuron ℓ in the hidden layers $k = K-1, K-2, \cdots, 2$, do

$$\delta_\ell^{(k)} = \left(\sum_{j=1}^{p_{k+1}} w_{j\ell}^{(k)} \delta_j^{(k+1)}\right) \sigma'^{(k)}\left(z_\ell^{(k)}\right). \tag{3.4}$$

4: Calculates the gradients

$$\frac{\partial L[y, f(x; \mathbf{W})]}{\partial w_{\ell j}^{(k)}} = a_j^{(k)} \delta_\ell^{(k+1)}. \tag{3.5}$$

3.3 Other tuning parameters

There are other important aspects to consider beyond the parameters $\mathbf{W}$. Below, we will briefly discuss some parameters that contribute to improve the quality of an artificial neural network.

Number of hidden layers and their widths

The number of neurons used in a layer determines its width and can be regarded as a tuning parameter. The greater the quantity, the higher the complexity. Research indicates that using a greater number of neurons while controlling over-fitting through regularization is a preferable approach. The number of layers defines a network's depth and is related to the objectives of the task. For image classification problems, the number of layers is related to the extent of feature extraction and pattern recognition.

The functions of activation

Table 1 presents several primary activation functions commonly employed in neural networks. These functions are represented as $\sigma(z)$, which applies a nonlinear transformation to the pre-activation z. Pre-activation is computed as a dot product of the weights and the neuron vector from the preceding layer.

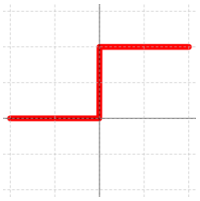
Regularization, early stopping and drop-out

Neural networks often comprise a vast number of parameters, which can lead to the risk of over-fitting. To mitigate this, several techniques have been proposed to enhance the model's generalization ability. Similar to its application in regression, Lasso regularization, commonly denoted as L_1, has the effect of ignoring irrelevant features, while Ridge regularization, denoted as L_2, aims to reduce collinearity among parameters. The "early stopping" technique, as its name suggests, prematurely terminates the training process when it exhibits no improvement in quality as the number of iterations increases. Typically, the stopping criterion involves the continuous monitoring of the loss function across both the training and testing datasets with a predefined tolerance of t steps or epochs. If the loss starts to rise beyond t steps in the testing dataset, the training process stops earlier at t steps.

Table 1 – Some primary activation functions, graphs were created using R.

Perceptron

$$\sigma(z) = \begin{cases} 1, & z \geq 0, \\ 0, & z < 0. \end{cases}$$



It is a step, binary function of 0 or 1, and only gives signals whenever z is greater and equal to zero. It is very restrictive and not used in deep neural network.

Sigmoid

$$\sigma(z) = \frac{1}{1+e^{-z}} = \frac{1}{2} + \frac{1}{2}\tanh\left(\frac{z}{2}\right),$$

$\sigma : \mathbb{R} \to (0,1)$ is a logistic function which is smooth and differentiable. $\sigma(z) = 1$ as $z \to \infty$ and $\sigma(z) = 0$ as $z \to -\infty$. In statistical modeling, this function is frequently used in logistic regression to predict the probability of a binary class because the coefficients can be interpreted as odd ratios in a logarithmic scale. Its derivation is simple, if $p = (1+e^{-z})^{-1}$, then, $\sigma' = p(1-p)$.
However, the function is not practical to use for deep networks because of the vanishing gradient problem. This occurs when the output of sigmoid function saturates, i.e., the curve tends to reach plateau and becomes parallel to the the z-axis for extreme values resulting in decreased gradient in these regions and contributes nothing to learning algorithm.

Tanh

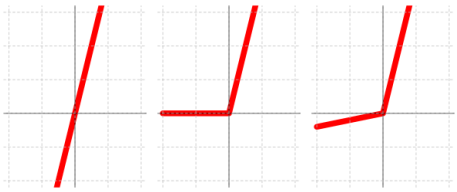
$$\sigma(z) = \tanh(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}} = \frac{e^{2z} - 1}{e^{2z} + 1}$$

The hyperbolic tangent is a shifted and scaled version of the sigmoid function, $tanh(z) = 2 \cdot \text{sigmoid}(2z) - 1$. It is mapped from real domain to the range of (-1, 1) and is preferred over the sigmoid function as $\sigma(0) = 0$. If $p = \frac{e^{2z}-1}{e^{2z}+1}$, then $\sigma' = 1 - p^2$.

Scale-invariant

$\sigma(\lambda z) = \lambda\sigma(z)$, $\forall\lambda > 0$. Any scaling done $\to \lambda z$ can be undone by applying the inverse function $\sigma(z) \to \lambda^{-1}\sigma(z)$ where $\sigma(z)$ is of the following form

$$\sigma(z) = \begin{cases} a_+ z, & z \geq 0, \\ a_- z, & z < 0. \end{cases}$$



linear: $a_+ = a_- = a$;
ReLU: $a_+ = 1, a_- = 0$;
leaky ReLU: $a_+ = 1, a_- = a$.

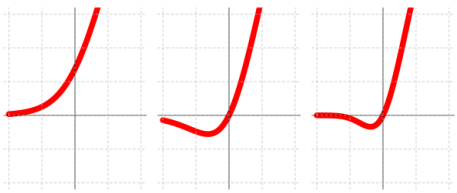
ReLU-like

These functions were proposed in order to introduce smoothness to the Rectified Linear Unit (ReLU):

Softplus: $\sigma(z) = \log(1+e^{z})$

SWISH: $\sigma(z) = \dfrac{z}{1+e^{-z}}$;

GELU: $\sigma(z) = \left[\dfrac{1}{2} + \dfrac{1}{2}\,\text{erf}\left(\dfrac{z}{\sqrt{2}}\right)\right] \times z$, where $\text{erf}(z) \equiv \dfrac{2}{\sqrt{\pi}} \int_0^z dt e^{-t^2}$



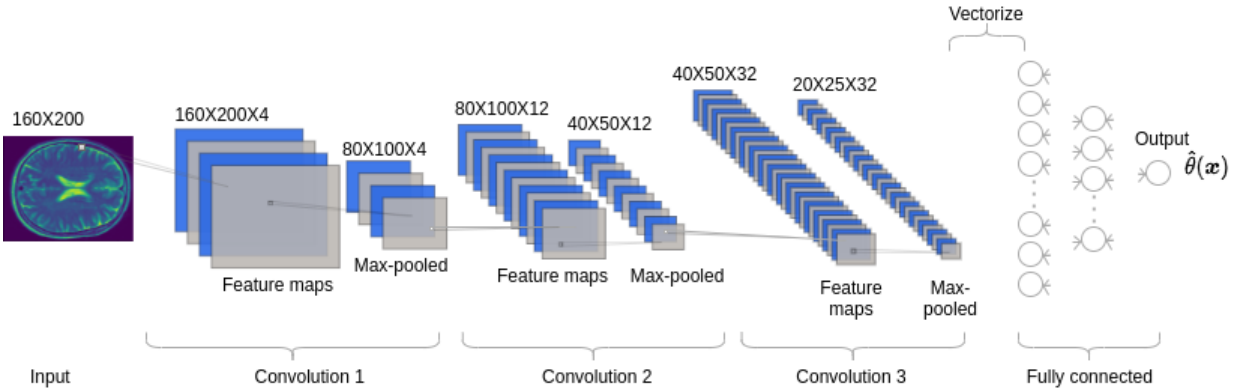
3.4 Deep learning

CNN are the primary architecture for deep learning in image analysis. In this process, each image is converted into a matrix with elements representing pixel intensity values. The CNN architecture is designed to efficiently process data with spatial structures, a characteristic often found in medical images. The inputs initially pass through several successive layers of convolution and pooling with the goal of extracting representative features, and are then processed in a fully connected layer to generate an output.

Let elements of an input matrix of size $k \times k$ be denoted as $\{x_{i,j}\}$. In the convolutional layer, a set of filters represented as f, is applied by sliding across the entire input matrix to capture distinctive features. These filters are typically square matrices of size $q \times q$, often with $q = 3$. They act as detectors for specific features in the image, such as horizontal, vertical, diagonal lines, and more. The output will be a convoluted image, also called a feature map with elements $\tilde{x}_{i,j} = \sum_{\ell=1}^{q} \sum_{\ell'=1}^{q} x_{i+\ell,j+\ell'} f_{\ell,\ell'}$.

A pooling layer reduces the dimension of the convolutional layer it follows. The max-pooling operation is the most common technique in which the maximum value is selected for each non-overlapping block. If the input is a matrix containing 16×16 pixels, by using a block size of 2×2 and a stride of 2, the input matrix would be reduced to 2×2. The reason for applying the maximum function in pooling is that after convolution, the convoluted image, denoted as $\{\tilde{x}_{i,j}\}$, contains elements with high values in regions where the applied filter captures specific patterns in the input image. Max-pooling retains the most prominent feature detected by the filters. This property is sometimes referred to as "local translation invariance" because it ensures that the network can recognize features even if they are shifted slightly within a local region of the input.

Figure 4 – The convolutional neural network architecture used to estimate θ in the survival cure model applied to OASIS-3 data.



We refer to the work of Meyer *et al.* (2018, pages 130-133), in which they showed visualization of the convolutional and pooling computing details. Figure 4 illustrates the CNN architecture applied to the OASIS-3 data discussed in Chapter 6. The inputs are MRI images with dimensions of 160×200 pixels. The images pass through three convolutional and max-pooling layers before being fed into a fully connected layer. The network's output provides an estimate for the mean number of risk factors, denoted as $\hat{\theta}$ in the survival cure model. The ReLU function is chosen as the activation function in the convolutional layers because of its improved computational efficiency.

3.5 Deep learning techniques in survival analysis

In this section, we aim to provide a brief and simple review of recently published articles that focus on applying machine learning or deep learning methods in survival analysis, particularly in cure models.

P and C (2022) conducted a systematic review of 37 articles to assess the utility of machine learning and deep learning techniques in cancer survival prediction. These techniques distinguish themselves from traditional statistical survival models by their ability to handle high-dimensional data and capture non-linear associations between features and outcomes. In contrast, statistical models face challenges in this aspect. While these advantages can result in improved predictive performance, it comes with a drawback - the lack of interpretability for clinicians. Consequently, this challenge has driven the development of explainable AI methods. Examples of several techniques listed in the review are:

1. Machine learning: Random forest, support vector machine (SVM), support vector regression (SVR), Extreme Gradient Boosting (XGBoost), MLP, and more.

2. Deep learning: Deep neural networks (DNN), autoencoders, recurrent neural networks (RNN), CNN, and more.

When there is a potential cure fraction, Ezquerro, Cancela and López-Cheda (2023) investigated the reliable use of machine learning models with a focus on the mixture cure model approach. They categorized the deep learning approaches for classical survival analysis into three groups:

1. Neural networks where the output corresponds to the survival status of a subject.

2. Neural networks based on a Cox Proportional-Hazards model with the output representing the survival time: Faraggi–Simon network, DeepSurv (CoxPH), CoxTime and CoxCC.

3. Neural networks based on the discrete time survival likelihood, with the output representing discrete survival time: DeepHit, Nnet-survival and Deep Survival Machines.

In their simulation study evaluating the reliability of the mentioned techniques for estimating cure probabilities, they made an interesting discovery. While the existing literature indicates that Deep Survival Machines, Random Survival Forests, and DeepHit perform well in classic survival analysis, especially in terms of the concordance index, these methods show poor performance when tested against a known survival distribution that contains a cure fraction. Ezquerro, Cancela and López-Cheda (2023) concluded that nonparametric mixture cure model and deep learning Cox-based approaches provide the best approximation for the cure rate distribution in terms of mean squared error (MSE).

In Table 2, we provide a brief summary of a few recent articles that explore the integration of machine learning and deep learning techniques with mixture cure model (MCM) and promotion time cure model (PCM).

Table 2 – Integration of machine learning and deep learning techniques with cure models

Cure model	Reference	ML/ DL	Latency part	Description
MCM	Li *et al.* (2020)	SVM	Cox PH/ AFT	These studies integrated a ML classifier, referred to as $g(\boldsymbol{x})$, as an alternative to the logistic link function typically employed in the mixture cure model. The primary advantage of employing an ML classifier lies in its ability to model the incidence part $\pi(\boldsymbol{x})$ more accurately by effectively capturing nonlinear boundaries that separate cured and non-cured patients based on covariates. The output of the classifier is a binary response variable, serving as a latent indicator for cured status. The probability of incidence is determined through the Platt scaling method: $\pi(\boldsymbol{x}) = (1+\exp(Ag(\boldsymbol{x})+B))^{-1}$.
	Pal and Aselisewine (2023b)	Decision trees	Cox PH	
	Xie and Yu (2021a)	CNN	Cox PH	This study integrated a CNN in order to model the incident $\pi(\boldsymbol{x})$ from unstructured predictors.
PCM	Pal and Aselisewine (2023a)	SVM	non parametric	This study is similar to Li *et al.* (2020) but has been adapted to the PCM framework. The classifier models the probability of incidence $\pi(\boldsymbol{x})$ and the mean number of competing risk is then obtained by $\theta(\boldsymbol{x}) = -\log(1-\pi(\boldsymbol{x}))$.
	Xie and Yu (2021b)	CNN	Piecewise exponential	This study integrated a CNN in order to model the mean number of competing risk from unstructured predictors.

The studies listed in Table 2 adopt a similar EM algorithm-based estimation procedure. The fundamental step involves defining an unobserved latent variable, which is the latent cured status indicator in MCM and the number of competing risks in PCM, respectively. Subsequently, the complete data log likelihood function can be derived. This complete data log likelihood is decomposed into two components. The first component involves parameters related to the incident part only, and the second component involves parameters related to the latency part only. The E-step of the EM algorithm computes the conditional expectation of the complete data log

likelihood function given the observed data and the current parameter values. It is necessary to deduce the conditional expectation of the latent variable in this step. In the M-step of the EM algorithm, the complete data log-likelihood is maximized in order to update the parameter estimates in the current iteration.

Finally, we came across an arXiv preprint that proposed a novel approach using DNN with PCM (Medina-Olivares; Lessmann; Klein, 2023). The authors named their method as "Deep Promotion Time Cure Model". They reformulated the PCM and placed a strong focus on its application in credit risk management. The model they have developed is an end-to-end DNN that directly integrates all parameters associated with incidence or latency distribution, eliminating the need for considering latent variables as seen in EM algorithm-based estimation methods. This results in a significant reduction in execution time. Another interesting aspect of their work is the introduction of an orthogonalization layer within the framework which addresses the identifiability issue. To demonstrate the model's scalability, the authors applied their framework to a large US mortgage portfolio which contains approximately 200,000 loans. It is worth noting that the authors have made their framework available as an open-source Python package, named `deepcure`. This provides a valuable resource to the research community and practitioners in the field.

CHAPTER

4

INTEGRATED TWO-STAGE CURE RATE MODEL

As Tsodikov, Ibrahim and Yakovlev (2003, Section 3.1) had pointed out, a cure model could be formulated as a two-stage model based on tumor recurrence mechanistic reasoning. At the first stage, one assumes there is an unobservable discrete random variable M which represents a number of risk factors such as surviving carcinogenic cells after an initial treatment, where M is characterized by distribution $P(M|\boldsymbol{\beta},\boldsymbol{x})$ that depends on a vector of covariates $\boldsymbol{x}$ and a vector of regression coefficients $\boldsymbol{\beta}$. The first stage can be interpreted as a latent tumor generating process where each remaining clonogen is associated with a latent progression time during which it will develop into a detectable tumor. At the second stage, the observed failure time T is generated when one of the risk factors is activated and the observed survival function can be obtained by $S_p(t|\boldsymbol{\beta},\boldsymbol{x}) = \mathbb{E}\left[S(t)^M|\boldsymbol{x}\right]$.

As mentioned earlier, our goal is to integrate CNN with cure model in order to introduce medical images as predictors of the cure probability on overdispersed data. The two-stage, long-term survival model mentioned above is actually a unified long-term survival model as described in Rodrigues *et al.* (2009). In this chapter, we propose an integrated two-stage cure rate model combines with CNN to relate clinical images to the latent number of risk factors at the first stage.

4.1 An integrated two-stage cure rate model

4.1.1 First stage

Let M be a discrete random variable denoting the number of risk factors or damaged cells with probability mass function

$$p_m = P_\Theta(M = m),$$

where $\theta = E(M)$. Since the parameter θ is positive[1], we consider the following parametrization

$$\theta(\boldsymbol{x}) = e^{\eta(\boldsymbol{x})}, \tag{4.1}$$

in which $\boldsymbol{x} \in \mathbb{R}^p$ is a vector of covariates with p dimensions. From the probabilistic view (RODRIGUES *et al.*, 2009, Section 2), the first stage can be characterized by the probability generating function of M given by

$$A_M(s) = \mathbb{E}\left[s^M\right] = \sum_{m=0}^{\infty} s^m p_m. \tag{4.2}$$

4.1.2 The convolutional neural network link function

The exponent $\eta(\boldsymbol{x})$ in Equation 4.1 is modelled after CNN and we followed the example provided by Xie and Yu (2021b) for explication purpose. The model of a fully connected neural network with two hidden layers can be shown as

$$\eta(\boldsymbol{x}) = \text{act}\left(\text{act}\left(\boldsymbol{B}_2^T \text{act}\left(\boldsymbol{B}_1^T \boldsymbol{x}_i + \boldsymbol{b}_1\right)^{\boldsymbol{t}} + \boldsymbol{b}_2\right)^{\boldsymbol{t}} \boldsymbol{\beta_3}\right), \tag{4.3}$$

in which

- $\boldsymbol{x}_i \in \mathbb{R}^p$ is an input vector of covariates.
- $\boldsymbol{B}_1 \in \mathbb{R}^{p \times k}$ is the weight matrix of the first hidden layer containing k neurons.
- $\boldsymbol{b}_1 \in \mathbb{R}^k$ is the bias vector of the first hidden layer.
- $\boldsymbol{B}_2 \in \mathbb{R}^{k \times m}$ is the weight matrix of the second hidden layer containing m neurons.
- $\boldsymbol{b}_2 \in \mathbb{R}^m$ is the bias vector of the second hidden layer.
- $\text{act}(\cdot)$ denotes activation function, eg: ReLU and tanh.
- $\boldsymbol{\beta}_3 \in \mathbb{R}^m$ denotes the weight vector of the output layer.

Note that the output layer does not include the intercept term (bias) due to the identifiability issue as mentioned by Li *et al.* (2001). For simpler notation, we use $\boldsymbol{w}$ to denote all the weights as in $\boldsymbol{w} = (\boldsymbol{b}_1, \boldsymbol{B}_1, \boldsymbol{b}_2, \boldsymbol{B}_2, \boldsymbol{\beta}_3)$. The output of integrated neural network gives estimates of the mean number of risk factors in logarithmic scale.

[1] Depends on the assumed distribution for M

4.1.3 Second stage

Let $Z_k, k = 1, \ldots, M$ be the unobservable progression time for kth latent risk factor where all Z_k are iid with a distribution function $F(\cdot)$ and survival function $S(\cdot) = 1 - F(\cdot)$. Also, Z_k are independent of M. Thus, the observable lifetime T is defined by

$$T = \min\{Z_1, \ldots, Z_M\}. \tag{4.4}$$

The susceptible individuals are the observations with $M \geq 1$, and the cured individuals are those with $M = 0, P(Z_0 = \infty) = 1$. As in Rodrigues *et al.* (2009), we can obtain the long-term survival model by

$$S_p(t) = A_M(S(t)). \tag{4.5}$$

Proof.

$$\begin{aligned}
S_p(t) &= P(\text{No cancer by time } t) \\
&= P(T > t) \\
&= \sum_{m=0}^{\infty} P(T > t \mid M = m) P(M = m) \\
&= P(M = 0) + \sum_{m=1}^{\infty} P(T > t \mid M = m) P(M = m) \\
&= P(M = 0) + P(\min\{Z_1, \ldots, Z_M\} > t \mid M = m) P(M = m) \\
&= P(M = 0) + P(Z_1 > t, \ldots, Z_M > t \mid M = m) P(M = m) \\
&= P(M = 0) + \sum_{m=1}^{\infty} [P(Z_1 > t)]^m P(M = m) \qquad \big(Z \perp M \text{ and } Z\text{s are iid } \mid M\big) \\
&= P(M = 0) + \sum_{m=1}^{\infty} S(t)^m P(M = m) \\
&= \sum_{m=0}^{\infty} S(t)^m P(M = m) \\
&= A_M\big((S(t)\big),\ 0 < S(t) < 1.
\end{aligned}$$

□

In Equation 4.5, note that the latency part $S(t)$ is a proper survival function while the populational survival function $S_p(t)$ is improper. The cure probability is defined as the limit as t approaches infinity, $\lim_{t \to \infty} S_p(t) = P(M = 0) = p_0$.

4.2 Estimation procedures

The estimation method for the proposed integrated two-stage cure rate models is based on the frequentist approach. The EM algorithm has been widely implemented in the literature of

cure model for finding the Maximum Likelihood Estimator (MLE) of the parameters. Gallardo *et al.* (2016) proposed a simplified estimation procedure for the power series cure rate model by maximizing the likelihood function of the complete data instead of the observed likelihood function, this approach requires the computation of the expected values of latent variable M which will facilitate the maximization step to be discussed next.

For a sample of size n, the complete data are $\boldsymbol{D}_{\text{comp}} = (\boldsymbol{t}, \boldsymbol{\delta}, \boldsymbol{x}, \boldsymbol{M})$ where $\boldsymbol{t} = (t_1, \cdots, t_n)^T$, $\boldsymbol{\delta} = (\delta_1, \cdots, \delta_n)^T, \boldsymbol{x} = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n)^T$ and $\boldsymbol{M} = (M_1, \cdots, M_n)^T$, for the ith individual,

- t_i is the survival time.
- δ_i is the censoring indicator, $\delta_1 = 1$ if t_i is a failure time, otherwise it is right censored.
- $\boldsymbol{x}_i$ is a $p \times 1$ covariate vector.
- M_i denotes the number of latent risk factors or competing causes which is unobservable.

The observed data are defined as $\boldsymbol{D}_{\text{obs}} = (\boldsymbol{t}, \boldsymbol{\delta}, \boldsymbol{x})$. Let $\boldsymbol{\psi} = (\boldsymbol{w}, \boldsymbol{\alpha})$ denotes the set of parameters in which $\boldsymbol{w}$ is related to the weights of the neural network and $\boldsymbol{\alpha}$ corresponds to the parameters associated with the cdf $F(t)$ of the latent progression times of clonogenic cells. With the assumption of $Z \perp M$ and $Z_k \perp Z_l \mid M$ for $k \neq l$, the complete likelihood function is defined as

$$
\begin{aligned}
L\left(\boldsymbol{\psi}; \boldsymbol{D}_{\text{comp}}\right) &= \prod_{i=1}^{n} P\left(t_i, \delta_i \mid m_i\right) P_{\Theta_i}\left(M_i = m_i\right) \\
&= \prod_{i=1}^{n} \left\{f\left(t_i; \boldsymbol{\alpha}\right)\right\}^{\delta_i} \left\{S\left(t_i; \boldsymbol{\alpha}\right)\right\}^{1-\delta_i} P_{\Theta_i}\left(M_i = m\right) \\
&= \prod_{i=1}^{n} \left\{m_i \left[1 - f\left(t_i; \boldsymbol{\alpha}\right)\right]^{m_i - 1} f\left(t_i; \boldsymbol{\alpha}\right)\right\}^{\delta_i} \left\{1 - f\left(t_i; \boldsymbol{\alpha}\right)\right\}^{m_i(1-\delta_i)} P_{\Theta_i}\left(M_i = m_i\right) \\
&= \underbrace{\left\{\prod_{i=1}^{n} \left[m_i f\left(t_i; \boldsymbol{\alpha}\right)\right]^{\delta_i} \left[S\left(t_i; \boldsymbol{\alpha}\right)\right]^{m_i - \delta_i}\right\}}_{L_2(\boldsymbol{\alpha}; \boldsymbol{D}_{\text{comp}})} \underbrace{\left\{\prod_{i=1}^{n} P_{\Theta_i}\left(M_i = m_i\right)\right\}}_{L_1(\boldsymbol{w}; \boldsymbol{D}_{\text{comp}})} \\
&= L_1(\boldsymbol{w}; \boldsymbol{D}_{\text{comp}}) L_2(\boldsymbol{\alpha}; \boldsymbol{D}_{\text{comp}})
\end{aligned} \tag{4.6}
$$

where $\theta_i \in \Theta$ depends on some covariates in a relationship of $\theta(\boldsymbol{w}, \boldsymbol{x}_i) = e^{\eta(\boldsymbol{w}, \boldsymbol{x}_i)}$. The assumptions of $Z \perp M$ and $Z_k \perp Z_l \mid M$ are crucial to the factorization in 4.6 which allows a direct integration of the CNN, via $L_1(\boldsymbol{w}; \boldsymbol{D}_{\text{comp}})$, to obtain the MLE of $\boldsymbol{w}$. Hence, the complete log likelihood can be written as a decomposition

$$
l(\boldsymbol{\psi}; \boldsymbol{D}_{\text{comp}}) = l_1(\boldsymbol{w}; \boldsymbol{D}_{\text{comp}}) + l_2(\boldsymbol{\alpha}; \boldsymbol{D}_{\text{comp}}),
$$

where $l_1(\boldsymbol{w}; \boldsymbol{D}_{\text{comp}})$ only depends on the parameters related to the mean number of risk factors (1st stage) and $l_2(\boldsymbol{\alpha}; \boldsymbol{D}_{\text{comp}})$ involves only parameters related to the distribution of the risk factor

lifetime (2nd stage) when m_i is given. This complete log likelihood can also be maximized independently and separately. Prompted by the work of Xie and Yu (2021b), we present in the next paragraph a modified EM algorithm to accommodate CNN as an integrated component for unstructured data processing.

Let $\boldsymbol{\psi}^{(k)} = (\boldsymbol{\alpha}^{(k)}, \boldsymbol{w}^{(k)})$ be the estimate of $\boldsymbol{\psi}$ at the kth iteration of the EM algorithm. At the $(k+1)$th iteration, we perform:

- E-step: Compute the conditional expected value of M given $\boldsymbol{D}_{\text{obs}}$ and the current parameter estimates, for $i = 1, \cdots, n$,

$$m_i^{(k+1)} = \mathbb{E}[M_i + \delta_i \mid \boldsymbol{D}_{\text{obs}}, \boldsymbol{\alpha}^{(k)}, \boldsymbol{w}^{(k)}].$$

- M-step:
  - Find $\boldsymbol{w}^{(k+1)}$ that maximizes $l_1(\boldsymbol{w}; \boldsymbol{D}_{\text{comp}})$ via CNN where

$$l_1(\boldsymbol{w}; \boldsymbol{D}_{\text{comp}}) = \sum_{i=1}^{n} \log P_{\theta_i^{(k)}}(M_i = m_i^{(k+1)}),$$
$$\theta_i^{(k)} = e^{\eta(\boldsymbol{w}^{(k)}, \boldsymbol{x}_i)}.$$

  - Find $\boldsymbol{\alpha}^{(k+1)}$ that maximizes

$$l_2(\boldsymbol{\alpha}; \boldsymbol{D}_{\text{comp}}) = \sum_{i=1}^{n} \left\{ \delta_i \log m_i^{(k+1)} + \delta_i \log \frac{f(t_i; \boldsymbol{\alpha}^{(k)})}{S(t_i; \boldsymbol{\alpha}^{(k)})} + m_i^{(k+1)} \log S(t_i; \boldsymbol{\alpha}^{(k)}) \right\}.$$

We repeat the iterations until $\|\boldsymbol{w}^{(k+1)} - \boldsymbol{w}^{(k)}\|_2^2 < \varepsilon$ and $\|\boldsymbol{\alpha}^{(k+1)} - \boldsymbol{\alpha}^{(k)}\|_2^2 < \varepsilon$, where ε is a defined tolerance and $\|\cdot\|$ a L_2 norm.

4.3 Some integrated models

In this section, we consider two distributions for M–Poisson and negative binomial– which will result in an integrated promotion time cure rate model (IPCM) and integrated negative binomial cure rate model (INBCM) respectively.

We consider a piecewise exponential distribution for T with time partitioned into G intervals, $0 < s_1 < \cdots < s_G$, with $s_G > \max_{1 \le i \le n}(t_i)$. We also assume a constant hazard α_g in the gth interval, $(0, s_1], (s_1, s_2], \cdots, (s_{g-1}, s_g]$. Therefore, the cdf for the lifetime of risk factors is given by

$$F(t; \boldsymbol{\alpha}) = 1 - \exp\left\{ -\alpha_g \left(t - s_{g-1}\right) - \sum_{j=1}^{g-1} \alpha_j \left(s_j - s_{j-1}\right) \right\}, \tag{4.7}$$

in which $t \in (s_{g-1}, s_g]$ and $S(t;\boldsymbol{\alpha}) = 1 - F(t;\boldsymbol{\alpha})$. As shown in Equation 4.6, the factorization of the complete likelihood function for $\boldsymbol{\psi} = (\boldsymbol{\alpha}, \boldsymbol{w})$ is given by

$$\begin{aligned} L\left(\boldsymbol{\psi}; D_{\text{comp}}\right) &= L_1(\boldsymbol{w}; \boldsymbol{D}_{\text{comp}}) L_2(\boldsymbol{\alpha}; \boldsymbol{D}_{\text{comp}}) \\ &= \left\{ \prod_{i=1}^{n} P_{\Theta_i}(M_i = m_i) \right\} \left\{ \prod_{i=1}^{n} \left[m_i f(t_i;\boldsymbol{\alpha}) \right]^{\delta_i} \left[S(t_i;\boldsymbol{\alpha}) \right]^{m_i - \delta_i} \right\}. \end{aligned} \tag{4.8}$$

The term of $L_1(\boldsymbol{w}; \boldsymbol{D}_{\text{comp}})$ varies according to its assumed distribution for risk factors and the formulas are organized in Table 3. The $L_2(\boldsymbol{\alpha}; \boldsymbol{D}_{\text{comp}})$ term is the same for both models. Using

Table 3 – Formulas for the first component of likelihood, $L_1(\boldsymbol{w}; \boldsymbol{D}_{\text{comp}})$, and log likelihood, $l_1(\boldsymbol{w}; \boldsymbol{D}_{\text{comp}})$, related to the weights of the convolutional neural network. C denotes a constant independent of $\boldsymbol{\psi}$.

$M \sim$ Poisson	
$L_1(\boldsymbol{w}; \boldsymbol{D}_{\text{comp}})$	$\prod_{i=1}^{n} \frac{\left(e^{\eta(\boldsymbol{w},\boldsymbol{x}_i)}\right)^{m_i} e^{-e^{\eta(\boldsymbol{w},\boldsymbol{x}_i)}}}{m_i!}$
$l_1(\boldsymbol{w}; \boldsymbol{D}_{\text{comp}})$	$\sum_{i=1}^{n} m_i \eta(\boldsymbol{w},\boldsymbol{x}_i) - e^{\eta(\boldsymbol{w},\boldsymbol{x}_i)} + C$
$M \sim$ Negative binomial	
$L_1(\boldsymbol{w}; \boldsymbol{D}_{\text{comp}})$	$\prod_{i=1}^{n} \frac{\Gamma\left(\phi^{-1} + m_i\right)}{\Gamma(\phi^{-1}) m_i!} \left(\frac{\phi e^{\eta(\boldsymbol{w},\boldsymbol{x}_i)}}{1 + \phi e^{\eta(\boldsymbol{w},\boldsymbol{x}_i)}} \right)^{m_i} \left(\frac{1}{1 + \phi e^{\eta(\boldsymbol{w},\boldsymbol{x}_i)}} \right)^{\phi^{-1}}$
$l_1(\boldsymbol{w}; \boldsymbol{D}_{\text{comp}})$	$\sum_{i=1}^{n} m_i \eta(\boldsymbol{w},\boldsymbol{x}_i) - m_i \log\left(1 + \phi e^{\eta(\boldsymbol{w},\boldsymbol{x}_i)}\right) - \phi^{-1} \log\left(1 + \phi e^{\eta(\boldsymbol{w},\boldsymbol{x}_i)}\right) + C$

Equation 4.7, we can write the second component of the complete log likelihood as follow:

$$\begin{aligned} l_2(\boldsymbol{\alpha}; \boldsymbol{D}_{\text{comp}}) &= \sum_{i=1}^{n} \left\{ \delta_i \log m_i + \delta_i \log \frac{f(t_i;\boldsymbol{\alpha})}{S(t_i;\boldsymbol{\alpha})} + m_i \log S(t_i;\boldsymbol{\alpha}) \right\} \\ &= \sum_{i=1}^{n} \left\{ \delta_i \log m_i + \delta_i \log \alpha_g - m_i \left\{ \alpha_g (t - s_{g-1}) - \sum_{j=1}^{g-1} \alpha_j (s_j - s_{j-1}) \right\} \right\}. \end{aligned} \tag{4.9}$$

For each model, we first present the long-term survival representation based on Equation 4.5; a proposition followed by a corollary and using this corollary, we replace the values of the unobservable variable with its conditional expected values at the E-step of the estimation procedures; we then finalize each subsection with the integrated EM algorithm steps.

4.3.1 Integrated promotion cure rate model (IPCM)

If $M \sim \text{Poisson}(\theta)$ with $P(M = m) = \frac{\theta^m}{m!} e^{-\theta}$, $m = 0, 1, 2, \ldots$, then

$$S_p(t) = A_M\left(S(t)\right) = e^{-\theta\left(1 - S(t)\right)} = e^{-\theta F(t)}, \tag{4.10}$$

and $p_0 = e^{-\theta}$.

Proposition 1. If $M_i \sim \text{Poisson}(\theta_i)$, then $M_i \mid \boldsymbol{D}_{\text{obs}}; \psi \sim \text{Poisson}\left(\delta_i + \theta_i S\left(t_i; \boldsymbol{\alpha}\right)\right)$.
See proof in Appendix A.

Corollary 1. $\mathbb{E}\left(M_i \mid \boldsymbol{D}_{\text{obs}}; \psi\right) = \delta_i + \theta_i S\left(t_i; \boldsymbol{\alpha}\right)$, for $i = 1, \cdots, n$.

Algorithm 2 – Integrated promotion cure rate model

while $|\psi^{(k)} - \psi^{(k-1)}| < \varepsilon$, **do**
E step: calculates for $i = 1, \cdots, n$,

$$\widehat{m}_i^{(k+1)} = e^{\eta(\boldsymbol{w}^{(k)}, \boldsymbol{x})} S(t_i; \boldsymbol{\alpha}^{(k)}) + \delta_i,$$

$$\text{where } S(t_i; \boldsymbol{\alpha}^{(k)}) = \exp\left\{-\alpha_g^{(k)}\left(t - s_{g-1}\right) - \sum_{j=1}^{g-1} \alpha_j^{(k)}\left(s_j - s_{j-1}\right)\right\}$$

M step:

$$\boldsymbol{w}^{(k+1)} = \arg\max_{\boldsymbol{w}} \left\{\sum_{i=1}^{n} \widehat{m}_i^{(k+1)} \eta\left(\boldsymbol{w}^{(k)}, \boldsymbol{x}_i\right) - e^{\eta(\boldsymbol{w}^{(k)}, \boldsymbol{x}_i)}\right\}$$

$$\alpha_g^{(k+1)} = \frac{\sum\limits_{s_{g-1} < t_i \leq s_g} \delta_i}{\sum\limits_{s_{g-1} < t_i \leq s_g} \widehat{m}_i^{(k+1)}\left(t_i - s_{g-1}\right) + \sum\limits_{t_i > s_g} \widehat{m}_i^{(k+1)}\left(s_g - s_{g-1}\right)}$$

end while

4.3.2 Integrated negative binomial cure rate model (INBCM)

If $M \sim \text{Negative Binomial}(\frac{1}{\phi}, \frac{1}{1+\phi\theta})$ with

$$P(M = m) = \frac{\Gamma\left(\frac{1}{\phi} + m\right)}{\Gamma\left(\frac{1}{\phi}\right) m!} \left(\frac{\phi\theta}{1+\phi\theta}\right)^m \left(\frac{1}{1+\phi\theta}\right)^{\frac{1}{\phi}}, m = 0, 1, 2, \ldots, \tag{4.11}$$

for $\theta > 0$ and $\phi > -1/\theta$ such that $\mathbb{E}(M) = \theta$ e $\mathbb{V}(M) = \theta + \phi\theta^2$, then

$$S_p(t) = A_M\big((S(t)\big) = \left(\frac{1}{1 + \phi\theta\left(1 - S(t)\right)}\right)^{\frac{1}{\phi}} \tag{4.12}$$

and $p_0 = \left(1 + \phi\theta\right)^{-\frac{1}{\phi}}$. As stated in Rodrigues *et al.* (2009), the Equation 4.11 contains a dispersion parameter ϕ, where $\phi > 0$ indicates over dispersion and under dispersion when $\phi < 0$. More interestingly, Equation 4.12 is a flexible model because:

- it converges to the IPCM when $\phi \to 0$;
- it becomes the mixture cure model proposed by Berkson and Gage (1952) when $\phi = -1$.

Proposition 2. If $M_i \sim \text{Negative binomial}(\frac{1}{\phi}, \frac{1}{1+\phi\theta_i})$, then

$$M_i \mid \boldsymbol{D}_{\text{obs}}; \psi \sim \text{Negative binomial}\left(\frac{1}{\phi} + \delta_i, \frac{1 + \phi\theta_i F\left(t_i; \boldsymbol{\alpha}\right)}{1 + \phi\theta_i}\right)$$

See proof in Appendix B.

Corollary 2. The conditional expectation of M_i given the observed data and parameter values is

$$\mathbb{E}\left(M_i \mid \boldsymbol{D}_{\text{obs}}; \boldsymbol{\psi}\right) = \frac{(1+\phi\,\delta_i)\,\theta_i S\left(t_i;\boldsymbol{\alpha}\right)}{1+\phi\,\theta_i F\left(t_i;\boldsymbol{\alpha}\right)}, \text{ for } i = 1,\cdots,n.$$

Algorithm 3 – Integrated negative binomial cure rate model

while $|\boldsymbol{\psi}^{(k)} - \boldsymbol{\psi}^{(k-1)}| < \varepsilon$, **do**

E step: calculates for $i = 1,\cdots,n$,

$$\widehat{m}_i^{(k+1)} = \frac{(1+\phi\,\delta_i)\,e^{\eta(\boldsymbol{w}^{(k)},\boldsymbol{x}_i)} S(t_i;\boldsymbol{\alpha}^{(k)})}{1+\phi e^{\eta(\boldsymbol{w}^{(k)},\boldsymbol{x}_i)} F\left(t_i;\boldsymbol{\alpha}^{(k)}\right)},$$

$$\text{where } S(t_i;\boldsymbol{\alpha}^{(k)}) = \exp\Big\{-\alpha_g^{(k)}\left(t - s_{g-1}\right) - \sum_{j=1}^{g-1} \alpha_j^{(k)}\left(s_j - s_{j-1}\right)\Big\}$$

M step:

$$\boldsymbol{w}^{(k+1)} = \arg\max_{\boldsymbol{w}} \left\{ \sum_{i=1}^{n} \widehat{m}_i^{(k+1)} \log\left\{ \frac{\phi e^{\eta(\boldsymbol{w}^{(k)},\boldsymbol{x}_i)}}{1+\phi e^{\eta(\boldsymbol{w}^{(k)},\boldsymbol{x}_i)}} \right\} - \frac{1}{\phi} \log\left(1+\phi e^{\eta(\boldsymbol{w}^{(k)},\boldsymbol{x}_i)}\right) \right\}$$

$$\alpha_g^{(k+1)} = \frac{\displaystyle\sum_{s_{g-1}<t_i\leq s_g} \delta_i}{\displaystyle\sum_{s_{g-1}<t_i\leq s_g} \widehat{m}_i^{(k+1)}\left(t_i - s_{g-1}\right) + \sum_{t_i > s_g} \widehat{m}_i^{(k+1)}\left(s_g - s_{g-1}\right)}$$

end while

CHAPTER

5

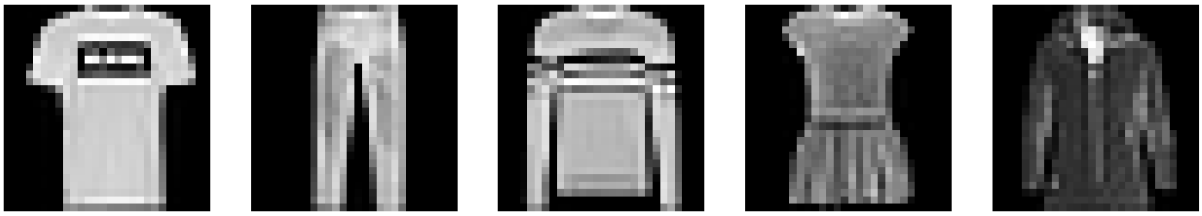
SIMULATION STUDY

In this chapter, we aim to assess the performance of our proposed method through simulations. We explore two scenarios for the number of competing risks: one where M follows a Poisson distribution, and another where it follows a negative binomial distribution. Our implementation procedure closely aligns with the methodology outlined in Xie and Yu (2021b).

We implemented the methodology outlined in Chapter 4 by using Python within the Google Colab Pro environment. The key advantage of utilizing Google Colab Pro is its access to GPU resources which significantly accelerates the execution of deep neural networks. The CNN model was developed using the `TensorFlow` library (v2.9.2). Our approach involved running 30 iterations of the EM algorithm as it reached reasonable tolerance. The image predictors $\boldsymbol{x}$ are samples taken from the Fashion-MNIST dataset, with each entry being a grayscale image with a dimension of 28×28 and associated with one of the 10 clothing classes labeled from 0 to 9. These image predictors, denoted as $\boldsymbol{x}$, will be linked to the mean parameter through a neural network, defined as $\theta(\boldsymbol{x}) = e^{\eta(\boldsymbol{x})}$. In our simulation study, we only consider the first five classes as shown in Figure 5, and each labeled class is then assigned a fixed numeric value for $\theta(\boldsymbol{x})$.

The convolutional network model is the same as the one used in the application of Xie and Yu (2021b). We followed the authors' recommendations regarding the determination of network architecture. Among three different network structures, we selected the one that yielded

Figure 5 – The first five clothing classes corresponding to labels 0 through 4, 0 - T-shirt/top, 1 - Trousers, 2 - Pullover, 3 - Dress and 4 - Coat.



the highest likelihood value of $l_1(\boldsymbol{w}; \boldsymbol{D}_{\text{comp}})$, as shown in Table 3. The details of the model's architecture are presented in Table 4.

Table 4 – Architecture of the convolutional network model applied on the Fashion-MNIST sample images

Layer	Dimension	Activation σ	Number of parameters
Convolution 1	(28, 28, 4)	ReLU	104
Pooling 1	(14, 14, 4)	max	-
Convolution 2	(14, 14, 12)	ReLU	1212
Pooling 2	(7, 7, 12)	max	-
Convolution 3	(7, 7, 32)	ReLU	9632
Pooling 3	(3, 3, 32)	max	-
Flatten	(288)	-	-
Fully connected	(128)	tanh	36992
Output	(1)	-	128
Total	-	-	48068

The lifetimes for the latent risk factors $Z_1, \cdots, Z_M$ are generated from an exponential distribution characterized by a constant hazard rate of 1, leading to the survival function of the form $S_1(t) = e^{-t}$. For the estimation of the latency part, we consider a piecewise exponential distribution and employ $G = 5$ for partitioning the observed times. Intervals are determined to ensure approximately equal numbers of events in each partition. In all of the scenarios, we maintain the average cure rate and overall censoring rate at approximately 40% and 45%, respectively. To explore the impact of varying sample sizes, we utilize three distinct training set sizes (n_{train}): 500, 1000, and 5000. And the corresponding test set sizes (n_{test}) are fixed at 125, 250, and 1250, respectively.

To evaluate the estimation accuracy, we follow the methodology outlined in the referenced paper by performing 100 replications on the estimations and calculate the following mean differences on the test sets:

- The mean squared error for the populational survival function

$$\Delta\overline{S}_p(t) = \frac{1}{R \times n_{test}} \sum_{r=1}^{R} \sum_{i=1}^{n_{test}} \left(\widehat{S}_p(t_i) - S_p^0(t_i) \right)^2$$

- The mean squared error for the competing risk survival lifetime function

$$\Delta\overline{S}_1(t) = \frac{1}{R \times n_{test}} \sum_{r=1}^{R} \sum_{i=1}^{n_{test}} \left(\widehat{S}_1(t_i) - S_1^0(t_i) \right)^2$$

- The mean squared error for the cure rate

$$\Delta\overline{p}_0 = \frac{1}{R \times n_{test}} \sum_{r=1}^{R} \sum_{i=1}^{n_{test}} \left(\widehat{p}_0(t_i) - p_0^0(t_i) \right)^2$$

- The mean squared error for the mean risk factors in logarithm scale

$$\Delta\overline{\eta}(x) = \frac{1}{R \times n_{test}} \sum_{r=1}^{R} \sum_{i=1}^{n_{test}} \left(\widehat{\eta}\left(\boldsymbol{x}_i\right) - \eta^0\left(\boldsymbol{x}_i\right)\right)^2$$

- The mean absolute error for the number of risk factors

$$\Delta\overline{m} = \frac{1}{R \times n_{test}} \sum_{r=1}^{R} \sum_{i=1}^{n_{test}} \mid \widehat{m}_i - m_i^0 \mid$$

where R represents the number of replicates and the superscripted 0 represents the true measures.

To evaluate the predictive capacity of the studied models in terms of the cure fraction, we adopt the Area Under the ROC Curve (AUC) metric as employed in the referenced paper. This AUC, as introduced by Asano and Hirakawa (2017, Section 2.2), is specifically tailored for cure models. The construction of the Receiver Operating Characteristic Curve (ROC) curve involves an imputation method that incorporates the cure fraction estimator $\hat{p}_0$ in the computation of the True Positive Rate (TPR) and False Positive Rate (FPR) for a given threshold $c,\ 0 \leq c \leq 1$.

$$\widehat{TPR}(c) = \frac{\sum_{i=1}^{n} \mathbf{1}\left(\hat{p}_0\left(\boldsymbol{x}_i\right) \leq c\right)\left(1 - \hat{p}_0\left(\boldsymbol{x}_i\right)\right)}{\sum_{i=1}^{n} 1 - \hat{p}_0\left(\boldsymbol{x}_i\right)},$$

$$\widehat{FPR}(c) = \frac{\sum_{i=1}^{n} \mathbf{1}\left(\hat{p}_0 \leq c\right)\hat{p}_0\left(\boldsymbol{x}_i\right)}{\sum_{i=1}^{n} \hat{p}_0\left(\boldsymbol{x}_i\right)}.$$

These rates are commonly referred to as sensitivity and 1 - specificity, respectively. Once these rates are calculated, and the ROC curve is constructed, the AUC can be determined using the trapezoidal method.

5.1 Case where M follows the Poisson distribution

We first study the integrated promotion cure model case in which the competing risk variable M follows the Poisson distribution. In this case, we set $\theta(\boldsymbol{x}) = 0.1, 0.8, 1.5, 2, 2.5$ and then we generate m_i first for each observation $i, \cdots, n$ according to the given value of $\theta(\boldsymbol{x}_i)$. For cases where $m_i = 0$, we assign an infinite lifetime $y_i = \infty$. When $m_i > 0$, we proceed to generate the promotion time of each risk factors $z_1, \cdots, z_{m_i}$, and the corresponding lifetime would be $y_i = \min\{z_1, \cdots, z_{m_i}\}$. Next, we generate the censoring time c_i from a uniform distribution, the observed lifetime is then obtained from $t_i = \min\left(y_i, c_i\right)$. If $t_i < c_i$, we would set the censoring indicator variable $\delta_i = 1$, otherwise, $\delta_i = 0$.

Table 5 – Estimation accuracy of both models presented as the mean errors calculated from the test set.

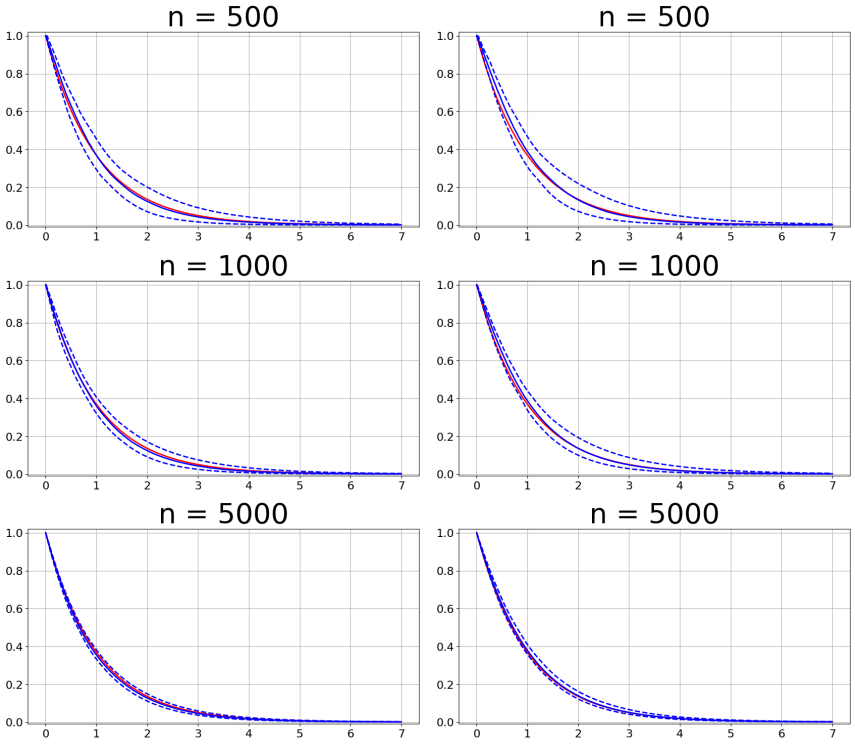
Model	time	n_{train}	$\Delta\overline{S}_p(t)$	$\Delta\overline{S}_1(t)$	$\Delta\overline{\eta}(x)$	$\Delta\overline{p_0}$	$\Delta\overline{m}$
IPCM	23min 34s	500	0.031885	0.001125	0.811638	0.042243	0.713748
	22min 30s	1000	0.023671	0.000319	0.592476	0.031686	0.629459
	58min 47s	5000	0.017309	0.000123	0.436762	0.023141	0.567622
INBCM with fixed ϕ	28min 7s	500	0.034323	0.001741	0.923800	0.043460	0.797420
	31min 34s	1000	0.026620	0.000755	0.676092	0.033795	0.706891
	63min 9s	5000	0.018399	0.000172	0.461016	0.023513	0.600151

The results of estimation accuracy are displayed in Table 5. When the proposed integrated negative binomial model is applied to this dataset, the dispersion parameter is set at a fixed value of 0.1. This choice is made because the model converges to the IPCM for small value of ϕ. It is noteworthy that the outcomes from both models exhibit a high degree of similarity across all sample sizes. The results align consistently with the estimated $S_1(t)$, which can be visualized in Figure 6. The true $S_1(t)$ curves are depicted by solid red lines, while the estimated curves and their corresponding empirical 95% confidence interval are represented by solid and dotted blue lines respectively. In both models, the confidence band narrows as the sample size increases. The estimated AUC values for both models demonstrate exceptional predictive capability as shown in Table 6, with the integrated negative model surpassing 85%. The reason for the slight decrease in AUC with an increased sample size in both models is unclear, it may be linked to the presence of censored data. In the case of INBCM, the inclusion of the dispersion parameter might have introduced minor interference.

Table 6 – AUC of both models for three different sample sizes.

Model	n_{train}	Train set	Test set
IPCM	500	0.862100	0.844204
	1000	0.848263	0.839596
	5000	0.833405	0.830950
INBCM with fixed ϕ	500	0.868450	0.852270
	1000	0.868474	0.859518
	5000	0.855207	0.852680

Figure 6 – Estimated curves of $S_1(t)$, assuming that the competing risk variable follows a Poisson distribution in the generated dataset. (a) Results obtained from the integrated promotion cure model and (b) Results obtained from the integrated negative binomial model.



(a) IPCM (b) INBCM

5.2 Case where M follows the negative binomial distribution

When we assume that the competing risk variable M follows a negative binomial distribution, we explore three distinct values for the dispersion parameter ϕ, 0.1, 1 and 2. The data generation process is analogous to the one described in the Poisson scenario.

From Table 7, it is convincing to show that the integrated promotion cure model does not yield satisfactory results when applied to the overdispersed dataset. For datasets generated with a low dispersion value, $\phi = 0.1$, the estimation accuracy obtained by the integrated negative binomial model is slightly lower, but the difference is relatively small. For datasets generated with ϕ equal to 1 and 2, the model, which does not account for the overdispersion parameter, produces larger mean differences on $\Delta\overline{S}_p(t)$, $\Delta\overline{p_0}$, $\Delta\overline{\eta}(x)$ and $\Delta\overline{S}_1(t)$. This is evident in Figure 7, where the estimated $S_1(t)$ curves and their confidence intervals appear shifted to the left from the true curve, indicating that the model overestimates hazard rates. In contrast, the estimated $S_1(t)$ curves and their confidence intervals stay closer to the true curve when using the integrated negative binomial model, as demonstrated in Figure 8. As ϕ increases, the estimated mean absolute errors $\Delta\overline{m}$ also increase. However, both models exhibit similar accuracy in this measurement. Lastly, the AUC values obtained from the integrated negative binomial model consistently surpass those

Table 7 – Estimation accuracy of both models, presented as the mean errors calculated from the test set.

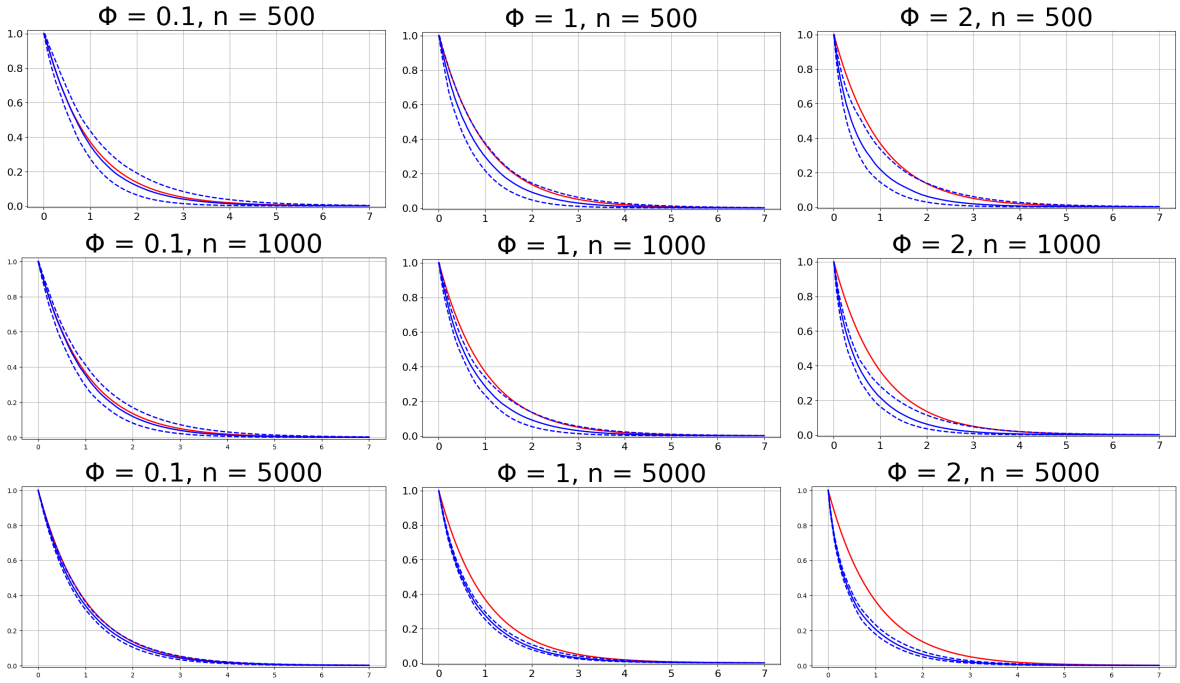
Model	ϕ	time	n_{train}	$\Delta\overline{S}_p(t)$	$\Delta\overline{S}_1(t)$	$\Delta\overline{\eta}(x)$	$\Delta\overline{p_0}$	$\Delta\overline{m}$
IPCM	0.1	25min 29s	500	0.029124	0.000988	0.826139	0.039508	0.711111
		26min 19s	1000	0.022288	0.000480	0.616062	0.031316	0.647315
		59min 57s	5000	0.016053	0.000356	0.437020	0.022225	0.587356
	1	21min 10s	500	0.018581	0.003541	0.625853	0.031741	0.959991
		20min 54s	1000	0.016349	0.004146	0.586346	0.028994	0.977712
		32min 30s	5000	0.008335	0.004666	0.295462	0.016766	0.874762
	2	17min 43s	500	0.020738	0.014814	1.234360	0.041123	1.736038
		18min 39s	1000	0.019442	0.015617	1.175383	0.039116	1.881228
		27min 32s	5000	0.018116	0.016917	1.109764	0.037153	1.839809
INBCM with fixed ϕ	0.1	26min 05s	500	0.032145	0.001381	0.930052	0.041554	0.788707
		36min 10s	1000	0.025263	0.000635	0.707812	0.033504	0.702467
		77min 17s	5000	0.017001	0.000083	0.451592	0.022220	0.603041
	1	21min 11s	500	0.019935	0.001490	0.610912	0.026581	1.076470
		30min 46s	1000	0.016274	0.000740	0.496323	0.020851	1.045644
		29min 32s	5000	0.004367	0.000086	0.102307	0.005509	0.837149
	2	20min 33s	500	0.015165	0.001373	0.668547	0.019151	1.734697
		26min 40s	1000	0.008824	0.000585	0.362249	0.010266	1.750102
		44min 30s	5000	0.002831	0.000092	0.106550	0.003175	1.550582

from the integrated promotion cure model. This suggests that the proposed model is better in predictive capacity particularly as the level of overdispersion intensifies. We observe a trend where the AUC tends to decrease as ϕ increases. This may be associated with the presence of heterogeneity or clustering, given that the dataset was generated under a negative binomial scenario which might have introduced some degree of uncertainty.

Table 8 – AUC of both models applied to overdispersed data for three different sample sizes

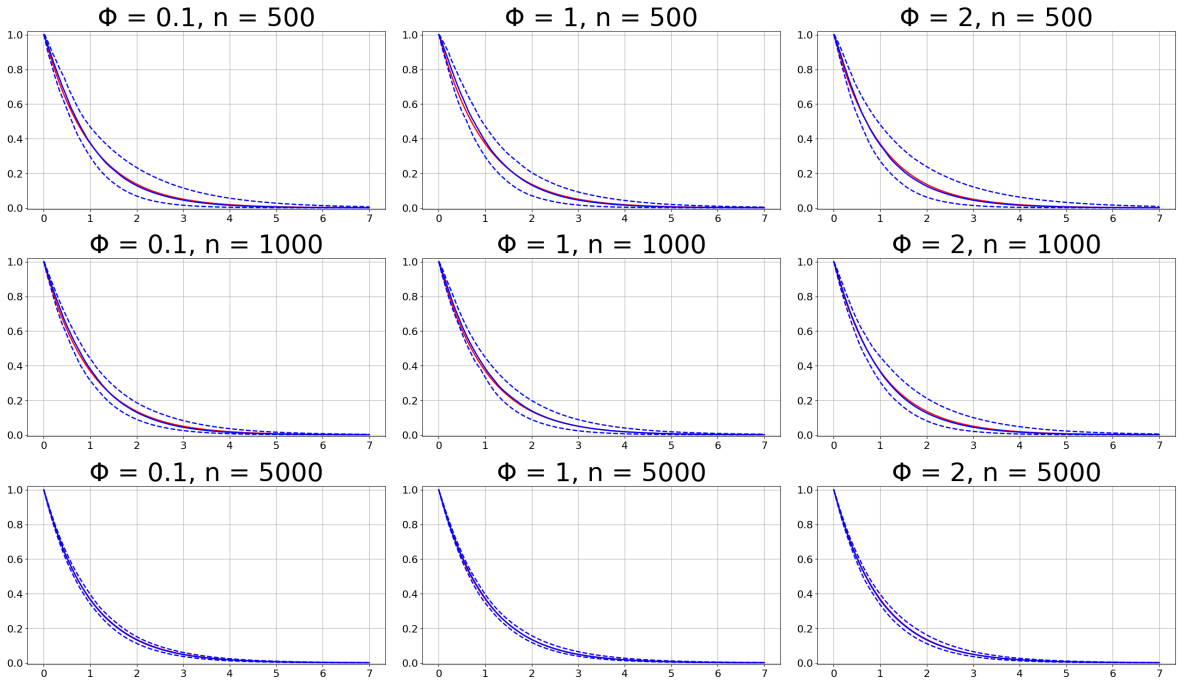
		IPCM		INBCM with fixed ϕ	
ϕ	n_{train}	Train set	Test set	Train set	Test set
0.1	500	0.839108	0.823816	0.848120	0.833626
	1000	0.826987	0.819386	0.847051	0.838269
	5000	0.806683	0.804761	0.829856	0.827632
1	500	0.684517	0.662885	0.740800	0.714441
	1000	0.638764	0.629546	0.702033	0.689066
	5000	0.578341	0.576895	0.628346	0.625909
2	500	0.594320	0.580597	0.671367	0.650770
	1000	0.578684	0.570287	0.633805	0.623909
	5000	0.536130	0.535386	0.579396	0.578014

Figure 7 – Estimated curves of $S_1(t)$, assuming that the competing risk variable follows a negative binomial distribution in the generated dataset. All results are obtained from the IPCM applied to data generated with (a) $\phi = 0.1$ (b)$\phi = 1$ and (c) $\phi = 2$.



(a) $\phi = 0.1$ (b) $\phi = 1$ (c) $\phi = 2$

Figure 8 – Estimated curves of $S_1(t)$, assuming that the competing risk variable follows a negative binomial distribution in the generated dataset. All results are obtained from the INBCM applied to data generated with (a) $\phi = 0.1$ (b)$\phi = 1$ and (c) $\phi = 2$.



(a) $\phi = 0.1$ (b) $\phi = 1$ (c) $\phi = 2$

5.3 Conclusion on simulation results

In this chapter, we have presented simulation results for our proposed model under two scenarios of datasets: one with the presence of overdispersion and the other without it. The outcomes aligned with our expectations. When the number of competing risk factors follows a Poisson distribution, both the integrated promotion cure model and the integrated negative binomial model demonstrate good fits. The dispersion parameter ϕ in the latter model was set as fixed value because we couldn't find a closed-form solution for it.

In conclusion, the integrated negative binomial model takes longer execution times. However, our simulation results consistently indicate that this model outperforms the integrated promotion cure model when handling overdispersed data. When considering the presence of overdispersion, the negative binomial model consistently delivers better estimation accuracy and predictive ability, as evidenced by the higher AUC values. These results suggest that in scenarios involving overdispersion, the integrated negative binomial model is a more robust and accurate choice for modeling over the integrated promotion cure model.

CHAPTER

6

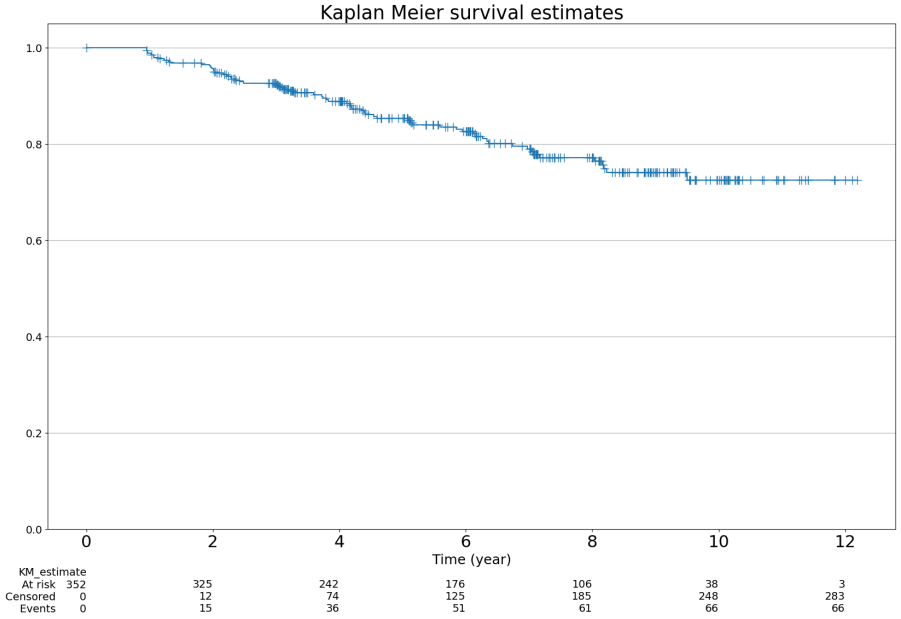
APPLICATION

6.1 OASIS-3 dataset

In this chapter, we will implement the proposed integrated negative binomial model using the OASIS-3 dataset. OASIS-3 is an open source dataset available at <www.oasis-brains.org>. The resource comprises a collection of MRI and clinical data spanning over 15 years, collected from various studies conducted at the University of Washington Alzheimer's Disease Research Center. The data used in this application were sourced from the supplementary material made available by Xie and Yu (2021b). The sample consists of 352 individuals with the maximum observed time of 12.2 years.

The dependent variable is defined as the time until the onset of Alzheimer's disease from

Figure 9 – Estimated Kaplan-Meier survival curve for the OASIS-3 data.



the date each individual is enrolled in the study. Event confirmation is determined by the Clinical Dementia Rating (CDR) scale, where an event is defined as CDR > 0. The event proportion corresponds to approximately one-fifth of the samples. From the fitted Kaplan-Meier survival curve in Figure 9, there is strong evidence of the presence of a non-susceptible group, as the plateau begins at around 9 years. The graph was plotted in Python using the `KaplanMeierFitter` function from the `lifetimes` package. Our interest lies in predicting the cure rate using MRI data, as described in Chapter 4. The MRIs are in grayscale with dimensions of 160×200, and the pixels are normalized to the range $[0, 1]$ before being fed into the network.

The division of the training and testing sets and the percentage of censoring are shown in Table 9. A moderate choice ($G = 5$) was adopted for the interval division of observed times and the number of events is approximately equal in each sub-interval, as presented in Table 10.

Table 9 – Data set partition and censoring percentage

Data	Size	Observed events	Number of censored data	% Censoring
Training	280	52	228	81,43
Test	72	14	58	80,56
Total	352	66	286	81,25

Table 10 – Partition of time

g	(0, 2,062]	(2,062, 3,360]	(3,360, 4,495]	(4,495, 6,320]	(6,320, 12,2]
Number of events	11	10	10	10	11

Table 11 presents the architecture of the applied CNN and the numbers of parameters are over 2 million units in total. The visualization of the architecture is shown in Figure 4. The architecture is the same as being used in the referenced article.

Table 11 – Architecture of the convolutional network applied to the MRIs in the OASIS-3 dataset

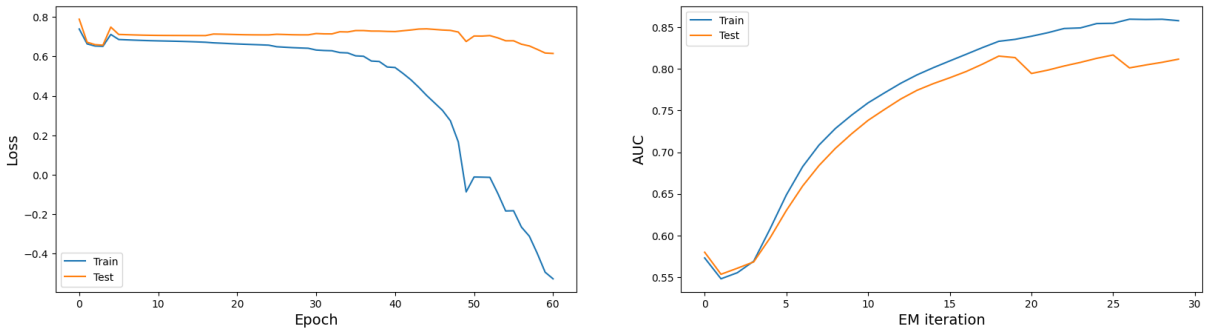
Layer	Dimension	Activation σ	Number of parameters
Convolution 1	(160, 200, 4)	ReLU	104
Pooling 1	(80, 100, 4)	max	-
Convolution 2	(80, 100, 12)	ReLU	1212
Pooling 2	(40, 50, 12)	max	-
Convolution 3	(40, 50, 32)	ReLU	9632
Pooling 3	(20, 25, 32)	max	-
Flatten	(16000)	-	-
Fully connected	(128)	tanh	2048128
Output	(1)	-	128
Total	-	-	2,059,204

6.2 Results

The estimation of the overdispersion parameter is based on the profile log-likelihood technique. By choosing a predefined range of fixed values for ϕ, we search for the ϕ' within this set that maximizes the observed log-likelihood function. Initially, we conducted a search within the set of $\{0.1, 0.2, \cdots, 1.4, 1.5\}$, which resulted in $\phi' = 0.1$. Subsequently, we performed a further second search within the set of $\{0.01, 0.02, \cdots, 0.1\}$, leading to the determination of $\phi' = 0.01$. So it is expected that the model will produce similar results to those obtained from IPCM.

In Figure 10, the graph on the left illustrates the progression of the loss function presented in Table 3 across epochs during the estimation of $\hat{\theta}$ while the graph on the right displays the estimated AUC values at the conclusion of each iteration in EM algorithm. The blue and orange curves represent the training and test sets respectively. At the end of this iterative process, we attained an AUC of 0.858 for the training set and 0.814 for the test set.

Figure 10 – (left) Loss function monitoring across epochs. (right) AUC values increase with the progression of iterations.

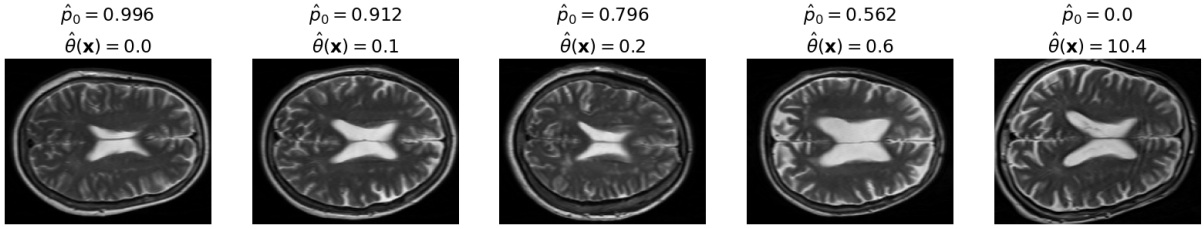


Alzheimer's disease is known to lead to the degeneration of brain cells and the impairment of mental functions, often affecting older people. Clinically, this neurodegeneration manifests as cerebral atrophy. A recent study conducted by Inglese *et al.* (2022) suggests that Alzheimer's disease may potentially be detectable and diagnosed through a single magnetic resonance imaging examination. Also, AlSaeed and Omar (2022) stated that CNN have increasingly been used in the diagnosis and prediction of Alzheimer's disease from MRI images. This technique automatically extracts accurate features to measure the brain's size and the number of its cells, thus detecting atrophy areas in the temporal and parietal lobes.

Figure 11 illustrates varying patterns in MRI scans at different levels of $\hat{p}_0$ and $\hat{\theta}(\mathbf{x}) = e^{\hat{\eta}(\mathbf{x})}$. Since these two quantities are inversely proportional, we can observe noticeable enlargement of the cerebral ventricles, which are the chambers within the brain containing cerebrospinal fluid, when $\hat{\theta}(\mathbf{x})$ increases or $\hat{p}_0$ decreases. Simultaneously, the sulci which are the grooves or furrows in the brain tend to deepen and widen.[1] These observations align with clinical evidence.

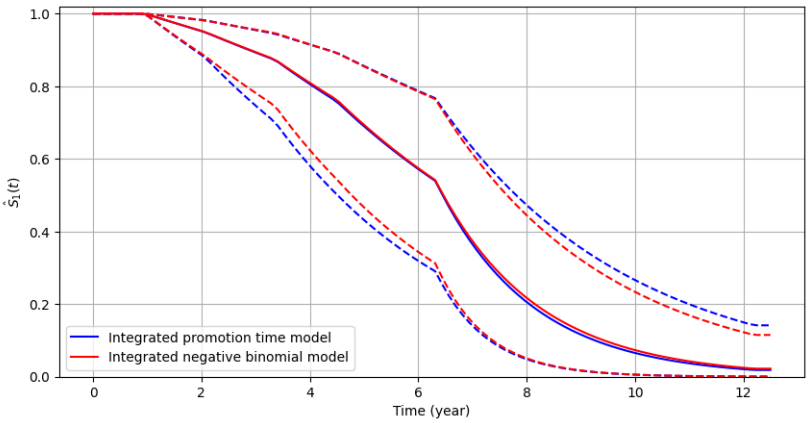
[1] https://www.brightfocus.org/alzheimers/infographic/brain-alzheimers-disease

Figure 11 – MRIs (160×200 pixels) at different quantiles of $\hat{p}_0$ and $\hat{\theta}(\mathbf{x})$ from the test set, from left to right: 0%, 25%, 50%, 75% and 100%.



We anticipate that the proposed integrated negative binomial model will exhibit a good fit similar to the integrated promotion time model when ϕ approaches zero. To assess the model performances, we will conduct estimations based on 100 bootstrapped samples and compare the results of the two models. This comparison will involve an examination of the estimated survival curves $S_1(t)$ along with their corresponding pointwise 95% confidence intervals and the AUC values. Figure 12 illustrates that the two curves are nearly overlapping. Notably, the confidence intervals of the integrated negative binomial model, represented by the red dotted lines, are slightly narrower than those of the integrated promotion time model, depicted by the blue dotted lines. Also, the distributions of AUC of the models are almost identical as demonstrated in Figure 13. The numeric descriptive statistics of the calculated AUCs and the estimated $\boldsymbol{\alpha}$ are presented in Table 12.

Figure 12 – Comparison of estimated $S_1(t)$ curves from the two models. The dotted lines represent estimated pointwise 95% confidence intervals obtained from 100 bootstrapped samples.



Based on the aforementioned results, we can conclude that the proposed integrated negative binomial model produces satisfactory outcomes when applied to the OASIS-3 dataset. These results closely align with those presented in the referenced paper, where the model used is an integrated promotion time model.

Figure 13 – Comparison of AUC values of the two models. AUC values are obtained from the results based on 100 bootstrapped samples.

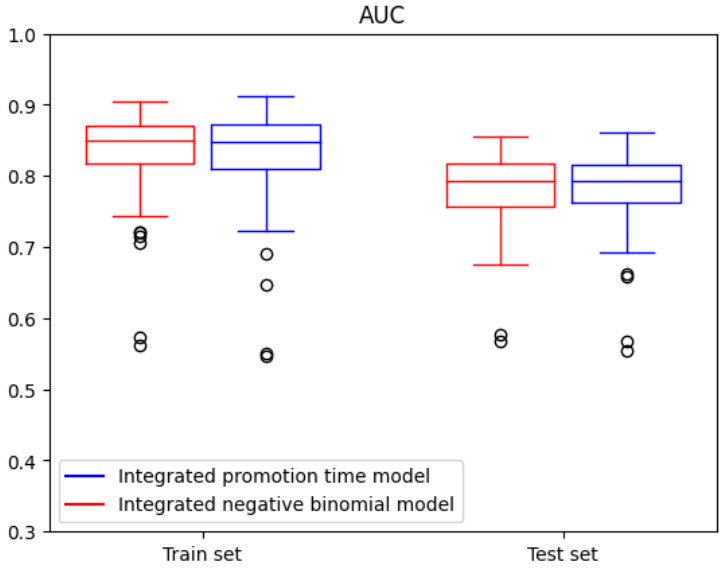


Table 12 – Descriptive summary of AUCs and estimated parameters of survival curves $S_1(t)$.

	Integrated negative binomial model						
	AUC_{train}	AUC_{test}	$\hat{\alpha}_1$	$\hat{\alpha}_2$	$\hat{\alpha}_3$	$\hat{\alpha}_4$	$\hat{\alpha}_5$
mean	0.832066	0.781006	0.048659	0.070641	0.133353	0.181767	0.588516
std deviation	0.059323	0.049832	0.023333	0.029389	0.058779	0.057860	0.201634
1st quartile	0.709700	0.680361	0.017370	0.027592	0.052378	0.084563	0.321970
2nd quartile	0.850377	0.792656	0.046545	0.064515	0.118985	0.188655	0.542635
3rd quartile	0.895008	0.844128	0.111836	0.130714	0.277550	0.304864	1.079108
	Integrated promotion time model						
	AUC_{train}	AUC_{test}	$\hat{\alpha}_1$	$\hat{\alpha}_2$	$\hat{\alpha}_3$	$\hat{\alpha}_4$	$\hat{\alpha}_5$
mean	0.831481	0.783034	0.050816	0.073376	0.135361	0.184000	0.589287
std deviation	0.063529	0.049128	0.025328	0.037387	0.059180	0.059353	0.202318
1st quartile	0.667735	0.660349	0.016570	0.029967	0.049733	0.083310	0.287260
2nd quartile	0.848589	0.792627	0.047271	0.065014	0.123469	0.187220	0.572247
3rd quartile	0.898700	0.845525	0.115514	0.175137	0.293136	0.301436	1.065154

CHAPTER

7

DISCUSSION AND CONCLUSION

In this work, we explore the possibility of modeling long-term survival data with unobserved dispersion using unstructured predictors. Taking inspiration from the work of Xie and Yu (2021b), where medical images were utilized to model the effect in the promotion time cure model, we chose to employ the negative binomial distribution to account for the presence of overdispersion in the data. Since all cure models can be represented as a two-stage model, we proposed the term "integrated two-stage cure rate model" to convey the notion of integrating CNN within the cure rate model. Consequently, our model can be referred to as the "integrated negative binomial cure rate model". The formulation of the proposed integrated two-stage cure rate model was presented in Chapter 4. To investigate the usefulness of our proposed model, we executed a simulation study, detailed in Chapter 5. The results affirm that our model is a better choice as it can accommodate both scenarios with and without overdispersion. Concluding our work, we applied the proposed model to the OASIS-3 dataset, and the obtained results closely align with those presented in the referenced paper.

Two principal aspects need to be considered and improved. First, in application chapter, we adopted the time-consuming profiling likelihood method for estimating the dispersion parameter. As the log likelihood function $L_1(\boldsymbol{w}; \boldsymbol{D}_{\text{comp}})$, which deals with the mean number of risk factors and dispersion in the EM algorithm, involves a significant number of parameters, it is desirable to find a more efficient method for estimating the dispersion parameter. Second, it is not feasible to use traditional statistical inference methods to calculate the estimated parameter's standard error.

On top of these, for the purpose of enhancing the clarity and interpretability of the application context, it is advisable to apply the proposed model in future studies on medical images associated with cancer diagnosis.

APPENDIX

A

PROOF OF PROPOSITION 1

$$
\begin{aligned}
f\left(m_i \mid \boldsymbol{D}_{\text{obs}}; \boldsymbol{\psi}\right) &= \frac{f\left(t_i, \delta_i, m_i; \boldsymbol{\psi}\right)}{f\left(t_i, \delta_i; \boldsymbol{\psi}\right)} \\
&= \frac{\left\{S\left(t_i; \boldsymbol{\alpha}\right)\right\}^{m_i-\delta_i} \left\{m_i f\left(t_i; \boldsymbol{\alpha}\right)\right\}^{\delta_i} \frac{\theta_i^{m_i} e^{-\theta_i}}{m_i!}}{\sum_{m_i=0}^{\infty} \left\{S\left(t_i; \boldsymbol{\alpha}\right)\right\}^{m_i-\delta_i} \left\{m_i f\left(t_i; \boldsymbol{\alpha}\right)\right\}^{\delta_i} \frac{\theta_i^{m_i} e^{-\theta_i}}{m_i!}}
\end{aligned}
$$

- $\delta_i = 0$,

$$
\begin{aligned}
f\left(m_i \mid \boldsymbol{D}_{\text{obs}}; \boldsymbol{\psi}\right) &= \frac{\frac{\left(\theta_i S(t_i;\boldsymbol{\alpha})\right)^{m_i} e^{-\theta_i}}{m_i!}}{e^{-\theta_i} e^{\theta_i S(t_i;\boldsymbol{\alpha})} \sum_{m_i=0}^{\infty} \frac{\left(\theta_i S(t_i;\boldsymbol{\alpha})\right)^{m_i} e^{-\theta_i S(t_i;\boldsymbol{\alpha})}}{m_i!}} \\
&= \frac{\frac{\left(\theta_i S(t_i;\boldsymbol{\alpha})\right)^{m_i} e^{-\theta_i}}{m_i!}}{e^{\theta_i} e^{\theta_i S(t_i;\boldsymbol{\alpha})}} \\
&= \frac{\left(\theta_i S\left(t_i; \boldsymbol{\alpha}\right)\right)^{m_i} e^{-\theta_i S(t_i;\boldsymbol{\alpha})}}{m_i!} \text{ which is a Poisson } \left(\theta_i S\left(t_i; \boldsymbol{\alpha}\right)\right)
\end{aligned}
$$

- $\delta_i = 1$,

$$
\begin{aligned}
f\left(m_i \mid \boldsymbol{D}_{\text{obs}}; \boldsymbol{\psi}\right) &= \frac{\cancel{\frac{f(t_i;\boldsymbol{\alpha})}{S(t_i;\boldsymbol{\alpha})}} e^{-\theta_i} e^{\theta_i S(t_i;\boldsymbol{\alpha})} \frac{\left(\theta_i S(t_i;\boldsymbol{\alpha})\right)^{m_i} e^{-\theta_i}}{(m_i-1)!}}{\cancel{\frac{f(t_i;\boldsymbol{\alpha})}{S(t_i;\boldsymbol{\alpha})}} \sum_{m_i=0}^{\infty} m_i \frac{(\theta_i S(t_i;\boldsymbol{\alpha}))^{m_i} e^{-\theta_i S(t_i;\boldsymbol{\alpha})}}{m_i!}} \\
&= \frac{\frac{\left(\theta_i S(t_i;\boldsymbol{\alpha})\right)^{m_i} e^{-\theta_i}}{(m_i-1)!}}{e^{-\theta_i} e^{\theta_i S(t_i;\boldsymbol{\alpha})} \theta_i S\left(t_i; \boldsymbol{\alpha}\right)} \\
&= \frac{\left(\theta_i S\left(t_i; \boldsymbol{\alpha}\right)\right)^{m_i-1} e^{-\theta_i S(t_i;\boldsymbol{\alpha})}}{(m_i-1)!} \text{ which is a one unit shifted Poisson } \left(\theta_i S\left(t_i; \boldsymbol{\alpha}\right)\right)
\end{aligned}
$$

Thus, $M_i \mid \boldsymbol{D}_{\text{obs}}; \boldsymbol{\psi} \sim \text{Poisson}\left(\delta_i + \theta_i S\left(t_i; \boldsymbol{\alpha}\right)\right)$.

APPENDIX

B

PROOF OF PROPOSITION 2

First, we calculate for $f(t_i, \delta_i; \psi)$,

$$
\begin{aligned}
f(t_i, \delta_i; \psi) &= \sum_{m_i=0}^{\infty} \{S(t_i;\boldsymbol{\alpha})\}^{m_i-\delta_i} \{m_i f(t_i;\boldsymbol{\alpha})\}^{\delta_i} \frac{\Gamma\left(\phi^{-1}+m_i\right)}{\Gamma(\phi^{-1})m_i!} \left(\frac{\phi\theta_i}{1+\phi\theta_i}\right)^{m_i} (1+\phi\theta_i)^{\phi^{-1}} \\
&= \left(\frac{f(t_i;\boldsymbol{\alpha})}{S(t_i;\boldsymbol{\alpha})}\right)^{\delta_i} \left(\frac{1}{1+\phi\theta_i}\right)^{\frac{1}{\phi}} \sum_{m_i=0}^{\infty} m_i^{\delta_i} \frac{\Gamma\left(\phi^{-1}+m_i\right)}{\Gamma(\phi^{-1})m_i!} \left(\frac{\phi\theta_i S(t_i;\boldsymbol{\alpha})}{1+\phi\theta_i}\right)^{m_i} \\
&= \frac{\left(\frac{f(t_i;\boldsymbol{\alpha})}{S(t_i;\boldsymbol{\alpha})}\right)^{\delta_i} \left(\frac{1}{\cancel{1+\phi\theta_i}}\right)^{\frac{1}{\phi}}}{\left(\frac{1+\phi\theta_i F(t_i;\boldsymbol{\alpha})}{\cancel{1+\phi\theta_i}}\right)^{\frac{1}{\phi}}} \sum_{m_i=0}^{\infty} m_i^{\delta_i} \underbrace{\frac{\Gamma\left(\phi^{-1}+m_i\right)}{\Gamma(\phi^{-1})m_i!} \left(\frac{\phi\theta_i S(t_i;\boldsymbol{\alpha})}{1+\phi\theta_i}\right)^{m_i} \left(\frac{1+\phi\theta_i F(t_i;\boldsymbol{\alpha})}{1+\phi\theta_i}\right)^{\frac{1}{\phi}}}_{\sim \text{Negative Binomial}\left(\phi^{-1}, \frac{1+\phi\theta_i F(t_i;\boldsymbol{\alpha})}{1+\phi\theta_i}\right)}.
\end{aligned}
$$

- $\delta_i = 0$,

$$
f(t_i, \delta_i; \psi) = (1+\phi\theta_i F(t_i;\boldsymbol{\alpha}))^{\phi^{-1}}.
$$

- $\delta_i = 1$,

$$
\begin{aligned}
f(t_i, \delta_i; \psi) &= \left(\frac{f(t_i;\boldsymbol{\alpha})}{\cancel{S(t_i;\boldsymbol{\alpha})}}\right) \left(\frac{1}{1+\phi\theta_i F(t_i;\boldsymbol{\alpha})}\right)^{\frac{1}{\phi}} \left(\frac{\theta_i \cancel{S(t_i;\boldsymbol{\alpha})}}{1+\phi\theta_i F(t_i;\boldsymbol{\alpha})}\right) \\
&= \left(\theta_i f(t_i;\boldsymbol{\alpha})\right) \left(1+\phi\theta_i F(t_i;\boldsymbol{\alpha})\right)^{-(\phi^{-1}+1)}.
\end{aligned}
$$

Hence, combining $\delta_i = 0$ and $\delta_i = 1$, we have

$$
f(t_i, \delta_i; \psi) = \left(\theta_i f(t_i;\boldsymbol{\alpha})\right)^{\delta_i} \left(1+\phi\theta_i F(t_i;\boldsymbol{\alpha})\right)^{-(\phi^{-1}+\delta_i)}.
$$

Now, we wish to identify the conditional probability distribution:

$$\begin{aligned} f(m_i \mid \boldsymbol{D}_{\text{obs}};\psi) &= \frac{f(t_i,\delta_i,m_i;\psi)}{f(t_i,\delta_i;\psi)} \\ &= \frac{\{S(t_i;\boldsymbol{\alpha})\}^{m_i-\delta_i}\{m_i f(t_i;\boldsymbol{\alpha})\}^{\delta_i}\frac{\Gamma\left(\phi^{-1}+m_i\right)}{\Gamma(\phi^{-1})m_i!}\left(\frac{\phi\theta_i}{1+\phi\theta_i}\right)^{m_i}(1+\phi\theta_i)^{\phi^{-1}}}{(\theta_i f(t_i;\boldsymbol{\alpha}))^{\delta_i}\left(1+\phi\theta_i F(t_i;\boldsymbol{\alpha})\right)^{-(\phi^{-1}+\delta_i)}} \\ &= \frac{m_i^{\delta_i}\left(\frac{f(t_i;\boldsymbol{\alpha})}{S(t_i;\boldsymbol{\alpha})}\right)^{\delta_i}\frac{\Gamma\left(\phi^{-1}+m_i\right)}{\Gamma(\phi^{-1})m_i!}\left(\frac{\phi\theta_i S(t_i;\boldsymbol{\alpha})}{1+\phi\theta_i}\right)^{m_i}(1+\phi\theta_i)^{\phi^{-1}}}{(\theta_i f(t_i;\boldsymbol{\alpha}))^{\delta_i}\left(1+\phi\theta_i F(t_i;\boldsymbol{\alpha})\right)^{-(\phi^{-1}+\delta_i)}} \end{aligned}$$

- $\delta_i = 0$,

$$f(m_i \mid \boldsymbol{D}_{\text{obs}};\psi) = \frac{\Gamma\left(\phi^{-1}+m_i\right)}{\Gamma(\phi^{-1})m_i!}\left(\frac{\phi\theta_i S(t_i;\boldsymbol{\alpha})}{1+\phi\theta_i}\right)^{m_i}\left(\frac{1+\phi\theta_i F(t_i;\boldsymbol{\alpha})}{1+\phi\theta_i}\right)^{\phi^{-1}}.$$

- $\delta_i = 1$,

$$\begin{aligned} f(m_i \mid \boldsymbol{D}_{\text{obs}};\psi) &= \frac{m_i\left(\frac{f(t_i;\boldsymbol{\alpha})}{S(t_i;\boldsymbol{\alpha})}\right)\frac{\Gamma\left(\phi^{-1}+m_i\right)}{\Gamma(\phi^{-1})m_i!}\left(\frac{\phi\theta_i S(t_i;\boldsymbol{\alpha})}{1+\phi\theta_i}\right)^{m_i}(1+\phi\theta_i)^{\phi^{-1}}}{\theta_i f(t_i;\boldsymbol{\alpha})\left(1+\phi\theta_i F(t_i;\boldsymbol{\alpha})\right)^{-(\phi^{-1}+1)}} \\ &= \frac{m_i\left(\frac{f(t_i;\boldsymbol{\alpha})}{S(t_i;\boldsymbol{\alpha})}\right)\frac{\Gamma\left(\phi^{-1}+m_i\right)}{\Gamma(\phi^{-1})m_i!}\left(\frac{\phi\theta_i S(t_i;\boldsymbol{\alpha})}{1+\phi\theta_i}\right)^{m_i}\left(\frac{1}{1+\phi\theta_i}\right)^{\frac{1}{\phi}}}{\theta_i f(t_i;\boldsymbol{\alpha})\left(\frac{1}{1+\phi\theta_i F(t_i;\boldsymbol{\alpha})}\right)\left(\frac{1}{1+\phi\theta_i F(t_i;\boldsymbol{\alpha})}\right)^{\frac{1}{\phi}}} \\ &= \frac{\Gamma\left(\phi^{-1}+m_i\right)}{\Gamma(\phi^{-1})(m_i-1)!}\left(\frac{\phi\theta_i S(t_i;\boldsymbol{\alpha})}{1+\phi\theta_i}\right)^{m_i}\left(\frac{1+\phi\theta_i F(t_i;\boldsymbol{\alpha})}{1+\phi\theta_i}\right)^{\frac{1}{\phi}}\left(\frac{1+\phi\theta_i F(t_i;\boldsymbol{\alpha})}{\theta_i S(t_i;\boldsymbol{\alpha})}\right) \\ &= \frac{\Gamma\left(\left(\phi^{-1}+1\right)+(m_i-1)\right)}{\Gamma(\phi^{-1}+1)(m_i-1)!}\left(\frac{\phi\theta_i S(t_i;\boldsymbol{\alpha})}{1+\phi\theta_i}\right)^{m_i-1}\left(\frac{1+\phi\theta_i F(t_i;\boldsymbol{\alpha})}{1+\phi\theta_i}\right)^{\frac{1}{\phi}+1}. \end{aligned}$$

Hence, combining $\delta_i = 0$ and $\delta_i = 1$, we have

$$M_i \mid \boldsymbol{D}_{\text{obs}};\psi \sim \text{Negative Binomial}\left(\frac{1}{\phi}+\delta_i, \frac{1+\phi\theta_i F(t_i;\boldsymbol{\alpha})}{1+\phi\theta_i}\right).$$

BIBLIOGRAPHY

ALSAEED, D.; OMAR, S. F. Brain mri analysis for alzheimerrsquo;s disease diagnosis using cnn-based feature extraction and machine learning. **Sensors**, v. 22, n. 8, 2022. ISSN 1424-8220. Available: <https://www.mdpi.com/1424-8220/22/8/2911>. Citation on page 61.

AMICO, M.; KEILEGOM, I. V. Cure models in survival analysis. **Annual Review of Statistics and Its Application**, v. 5, n. 1, p. 311–342, 2018. Available: <https://doi.org/10.1146/annurev-statistics-031017-100101>. Citation on page 21.

ASANO, J.; HIRAKAWA, A. Assessing the prediction accuracy of a cure model for censored survival data with long-term survivors: Application to breast cancer data. **Journal of Biopharmaceutical Statistics**, Informa UK Limited, v. 27, n. 6, p. 918–932, mar 2017. Citation on page 53.

BERKSON, J.; GAGE, R. P. Survival curve for cancer patients following treatment. **Journal of the American Statistical Association**, Taylor Francis, v. 47, n. 259, p. 501–515, 1952. Available: <https://www.tandfonline.com/doi/abs/10.1080/01621459.1952.10501187>. Citations on pages 21 and 49.

CHEN, M.-H.; IBRAHIM, J. G. Maximum likelihood methods for cure rate models with missing covariates. **Biometrics**, Wiley, v. 57, n. 1, p. 43–52, mar 2001. Citation on page 30.

COLOSIMO, E.; GIOLO, S. **Análise de sobrevivência aplicada**. Edgard Blücher, 2006. (ABE - Projeto Fisher). ISBN 9788521203841. Available: <https://books.google.com.br/books?id=g0-uOgAACAAJ>. Citation on page 25.

COX D.R., . O. D. **Analysis of Survival Datas**. 1. ed. [S.l.]: CRC Press, 1984. 1 p. Citation on page 26.

EFRON, B.; HASTIE, T. **Computer Age Statistical Inference: Algorithms, Evidence, and Data Science**. [S.l.]: Cambridge University Press, 2016. (Institute of Mathematical Statistics Monographs). Citation on page 32.

EZQUERRO, A.; CANCELA, B.; LóPEZ-CHEDA, A. On the reliability of machine learning models for survival analysis when cure is a possibility. **Mathematics**, v. 11, n. 19, 2023. ISSN 2227-7390. Available: <https://www.mdpi.com/2227-7390/11/19/4150>. Citations on pages 39 and 40.

GALLARDO, D. I. *et al.* A simplified estimation procedure based on the em algorithm for the power series cure rate model. **Communications in Statistics - Simulation and Computation**, 07 2016. Citation on page 46.

HORNIK, K.; STINCHCOMBE, M.; WHITE, H. Multilayer feedforward networks are universal approximators. **Neural Networks**, v. 2, n. 5, p. 359–366, 1989. ISSN 0893-6080. Available: <https://www.sciencedirect.com/science/article/pii/0893608089900208>. Citation on page 33.

INGLESE, M. *et al.* A predictive model using the mesoscopic architecture of the living brain to detect alzheimer's disease. **Communications Medicine**, Springer Science and Business Media LLC, v. 2, n. 1, jun 2022. Citation on page 61.

LEGRAND, C.; BERTRAND, A. Cure models in cancer clinical trials. In: ____. [S.l.: s.n.], 2019. p. 465–492. ISBN 9781315112084. Citation on page 21.

LI *et al.* Identifiability of cure models. **Statistics Probability Letters**, v. 54, p. 389–395, 10 2001. Available: <https://www.sciencedirect.com/science/article/abs/pii/S0167715201001055?via%3Dihub>. Citation on page 44.

LI, P.; PENG, Y.; JIANG, P.; DONG, Q. A support vector machine based semiparametric mixture cure model. **Computational Statistics**, v. 35, n. 3, p. 931–945, September 2020. Available: <https://ideas.repec.org/a/spr/compst/v35y2020i3d10.1007_s00180-019-00931-w.html>. Citation on page 40.

LUNDERVOLD, A. S.; LUNDERVOLD, A. An overview of deep learning in medical imaging focusing on mri. **Zeitschrift für Medizinische Physik**, v. 29, n. 2, p. 102–127, 2019. ISSN 0939-3889. Special Issue: Deep Learning in Medical Physics. Available: <https://www.sciencedirect.com/science/article/pii/S0939388918301181>. Citation on page 22.

Medina-Olivares, V.; Lessmann, S.; Klein, N. The Deep Promotion Time Cure Model. **arXiv e-prints**, p. arXiv:2305.11575, May 2023. Citation on page 41.

MEYER, P.; NOBLET, V.; MAZZARA, C.; LALLEMENT, A. Survey on deep learning for radiotherapy. **Computers in Biology and Medicine**, v. 98, p. 126–146, 2018. ISSN 0010-4825. Available: <https://www.sciencedirect.com/science/article/pii/S0010482518301318>. Citations on pages 22 and 39.

P, D.; C, G. A systematic review on machine learning and deep learning techniques in cancer survival prediction. **Progress in Biophysics and Molecular Biology**, v. 174, p. 62–71, 2022. ISSN 0079-6107. Available: <https://www.sciencedirect.com/science/article/pii/S0079610722000761>. Citation on page 39.

PAL, S.; ASELISEWINE, W. A semiparametric promotion time cure model with support vector machine. **The Annals of Applied Statistics**, Institute of Mathematical Statistics, v. 17, n. 3, p. 2680 – 2699, 2023. Available: <https://doi.org/10.1214/23-AOAS1741>. Citation on page 40.

PAL, S.; ASELISEWINE, W. On the integration of decision trees with mixture cure model. **Statistics in Medicine**, v. 42, n. 23, p. 4111–4127, 2023. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.9850>. Citation on page 40.

ROBERTS, D. A.; YAIDA, S.; HANIN, B. **The Principles of Deep Learning Theory**. [S.l.]: Cambridge University Press, 2022. <https://deeplearningtheory.com>. Citation on page 32.

RODRIGUES, J.; CANCHO, V. G.; de Castro, M.; LOUZADA-NETO, F. On the unification of long-term survival models. **Statistics Probability Letters**, v. 79, n. 6, p. 753–759, 2009. ISSN 0167-7152. Citations on pages 43, 44, 45, and 49.

TSODIKOV, A. D.; IBRAHIM, J. G.; YAKOVLEV, A. Y. Estimating cure rates from survival data. **Journal of the American Statistical Association**, Taylor Francis, v. 98, n. 464, p. 1063–1078, 2003. PMID: 21151838. Available: <https://doi.org/10.1198/016214503000000001007>. Citations on pages 22 and 43.

XIE, Y.; YU, Z. Mixture cure rate models with neural network estimated nonparametric components. **Comput. Stat.**, Kluwer Academic Publishers, USA, v. 36, n. 4, p. 2467–2489, dec 2021. ISSN 0943-4062. Available: <https://doi.org/10.1007/s00180-021-01086-3>. Citation on page 40.

XIE, Y.; YU, Z. Promotion time cure rate model with a neural network estimated nonparametric component. **Statistics in Medicine**, v. 40, n. 15, p. 3516–3532, 2021. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.8980>. Citations on pages 7, 9, 10, 22, 23, 30, 40, 44, 47, 51, 59, and 65.

YAKOVLEV, A. Y.; TSODIKOV, A. D.; ASSELAIN, B. **Stochastic Models of Tumor Latency and Their Biostatistical Applications**. WORLD SCIENTIFIC, 1996. Available: <https://www.worldscientific.com/doi/abs/10.1142/2420>. Citations on pages 21 and 28.

YIN, G.; IBRAHIM, J. G. Cure rate models: A unified approach. **Canadian Journal of Statistics**, v. 33, n. 4, p. 559–570, 2005. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/cjs.5550330407>. Citation on page 22.


ICMC
USP
SÃO CARLOS