MACHINE LEARNING FOR PRECISION MEDICINE AND HEALTH ECONOMICS: A NOVEL MODEL FOR SURVIVAL ANALYSIS AND SUBGROUP IDENTIFICATION

By

Meimei Chen

Supervisors: Prof Martin Hoyle, Dr Yuanyuan Gu

A thesis submitted to Macquarie University
for the degree of Masters of Research
Macquarie Business School
Centre for the Health Economy
June 2022

MACQUARIE
University
SYDNEY · AUSTRALIA

Examiner's Copy

Acknowledgements

This thesis was supported by Macquarie University and the MQ / Pharma HDR scholarship program.

I would like to thank my supervisors, Prof. Martin Hoyle and Dr. Yuanyuan Gu, for their encouragement and support. Prof. Martin Hoyle and Dr. Yuanyuan Gu provided me with health economics knowledge, guided me through the entire research, and reviewed my thesis constructively. I would also like to thank Dr. Noemi Kreif for the helpful comments.

Finally, thanks to my parents and friends for their encouragement.

Abstract

Background: The emergence of precision medicine, which is described as an approach that tailors health interventions to a group of patients based on their characteristics, brings challenges to Health Technology Assessment (HTA). HTA seeks to provide policymakers with the necessary information to better understand the benefits of health technologies and make better funding decisions, and ultimately improve resource allocation. Precision medicine differs from traditional medicine in that it is based on patient subsets, whereas traditional medicine is based on the entire patient population. Therefore, identifying appropriate patient subgroups is an important part of the evaluation of precision medicines. **Objective:** This research aims to develop Machine Learning tools to address the subgroup identification problem in precision medicine. **Methodology:** A novel Machine Learning model is proposed in this research using Multi-Task Learning and Support Vector Machine. The survival analysis problem is decomposed into a series of classification problems, and Support Vector Machine is applied to improve the classification accuracy. Moreover, a $\ell_{2,1}$ norm is used for feature selection.This model has two functions: (1) predict event times based on censored data; (2) select important covariates. **Results:** The prediction accuracy of the proposed model is compared with two benchmark statistical methods, Cox-LASSO and Cox-EN, and three state-of-the-art Machine Learning methods. The proposed model outperforms the other methods. The proposed model significantly outperforms the Cox models in terms of feature selection. In the simulation study, the proposed model outperforms the other methods in selecting patient subgroups with enhanced treatment effects **Conclusion:**

The model proposed in this research outperforms several other state-of-the-art methods in time-to-event prediction and feature selection. Therefore, the proposed model can be used for precision medicine to select high-risk patients or patient subgroups with enhanced treatment effects and has potential applications in other fields involving survival analysis.

Contents

List of Figures

List of Tables

Statement of Originality

This work has not previously been submitted for a degree or diploma in any university. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the thesis itself.

1
Introduction

Precision medicine, which is described as an approach that tailors health interventions to a group of patients based on their individual characteristics, has grown rapidly over the past decade. The emergence of precision medicine brings challenges for Health Technology Assessment (HTA), which is a subfield of health economics that aims to assess the quality of health technologies by using scientific evidence. The purpose of this thesis is to develop machine learning (ML) models to improve the assessment of health technologies in precision medicine.

This chapter outlines the background, purpose, and objectives of the research. Section 1.1 introduces the background of HTA and precision medicine. Section 1.2 describes the relationship between subgroup identification and survival analysis. Section 1.3 and 1.4 introduce survival analysis and the ML model used in the research. Finally, Section 1.5 lists the research questions and intended contributions of the research.

1.1 Background: Health Technology Assessment and Precision Medicine

The goal of HTA is to provide policymakers with the necessary information to better understand the benefits of health technologies and make better funding decisions. When making decisions between different actions, economic evaluation is the tool for comparing the corresponding costs and consequences (Drummond et al., 2015). Economic evaluation has four categories: cost-effectiveness analysis, cost-minimisation analysis, cost-benefit analysis, and cost-utility analysis. In cost-effectiveness analysis, the incremental cost-effectiveness ratio (ICER) is used to compare the cost-effectiveness of a new health technology versus an old technology or no technology. Suppose A is a new health technology and B is the alternative intervention to be compared, the ICER is calculated as follow:

$$\mathrm{ICER}_{\mathrm{A,B}} = \frac{C_{\mathrm{A}} - C_{\mathrm{B}}}{E_{\mathrm{A}} - E_{\mathrm{B}}} \tag{1.1.1}$$

where C_A and C_B are the costs of health technologies A and B, and E_A and E_B represents their effectiveness. Funding decisions for new health technologies are based on comparing the ICER to a willingness-to-pay threshold. If the ICER of the new health technology A is smaller than the threshold, then we say that the health technology A is cost-effective.

Precision medicine is described as the tailoring of health interventions based on patients' individual characteristics (Joore et al., 2020; Fahr et al., 2019). With advances in genomes, data collection, and data analysis, precision medicine is no longer out of reach. Compared with traditional healthcare, precision medicine improves both disease prevention and treatment (Williamson et al., 2018). In terms of disease prevention, precision medicine uses genetic screening to identify high-risk patients before symptoms appear. Early diagnosis of disease combined with prevention can reduce the risk of severe conditions, thereby increasing health outcomes such as life years gained and reducing the financial burden to healthcare providers. In terms of disease treatment, precision medicine provides personalised treatment to maximise efficacy and minimise side effects. Precision medicine achieves the goal of stratifying patients by collecting multiple data including genomic, environmental and population health data for analysis (Williamson et al., 2018).

Both patients and healthcare systems can benefit from precision medicine, where their treatments are designed based on their baseline risk, estimated treatment outcome, and other important biomarkers (Love-Koh et al., 2018). Precision medicine is increasingly important worldwide over the last five years as many countries such as the UK, US and China recognise it as a funding priority (Williamson et al., 2018). Australia also attaches great importance to precision medicine. In 2018, “using genomics and precision medicine to help Australia become the healthiest country on earth” was identified by Innovation and Science Australia as an ideal National Mission (Williamson et al., 2018). The development of precision medicine brings challenges to health economics. Therefore, the Australian health policy-makers such as HTA agencies should consider how their methods can be improved to better evaluate new precision medicines.

1.2 Background: Subgroup Identification

As described above, precision medicine not only requires optimised treatments based on individuals’ characteristics, but also predictive methods to identify patients at risk for specific health conditions. The process to identify patients is called **subgroup identification**. In precision medicine, the purpose of subgroup identification is to evaluate patient subgroups in which a particular health intervention can be clinically effective and/or cost-effective (Chen et al., 2020).

In precision medicine, health technologies such as drugs are designed for specific patient subgroups with enhanced treatment outcomes, rather than targeting the entire patient population. Therefore, HTA assesses the ICER of new health technologies in precision medicine among specific patient subgroups rather than the entire population. As a result, subgroup identification can directly influence the calculation of the ICER, as it affects E_A in Equation (1.1.1). Here is an example of how funding decisions are made among a subgroup of patients. Gefitinib is a drug for non-small cell lung cancer (NSCLC) tumor patients with activating mutations of EGFR-TK. In 2010, the National Institute for Health and Care Excellence (NICE) in England assessed the cost-effectiveness of gefitinib among NSCLC patients with activating mutations of EGFR-TK. The results showed that gefitinib is cost-effective for this subgroup of patients, so NICE recommended gefitinib as the first-line treatment for them (Gavan et al., 2018). This example illustrates the difference in economic evaluations between traditional and precision medicine – economic evaluations for precision medicine take patient heterogeneity into consideration.

Here are some examples of identifying high-risk patient subgroups. Hill et al. (2019b) built a tree-based predictive model to identity patients who have high-risk of developing atrial fibrillation, and the patients who are identified as high-risk will be tested. They also conducted a cost-effectiveness analysis of this method and the result showed that the screening strategy based on their risk predictive algorithm is more cost-effective than the conventional screening strategy (Hill et al., 2019a). Ling et al. (2018) developed a predictive model using observational data to identify sepsis patients at high risk for prolonged intensive care unit (ICU) stay, and the result can inform the plan to meet the demand for ICU care. Zack et al. (2019) applied a Random Forest model to identify patients who have high-risk of death or congestive heart failure rehospitalisation after percutaneous coronary intervention, and their result can improve the quality of healthcare, reduce healthcare costs, and assist clinical decision-making. These examples show that the identification of specific patient subgroups at risk for certain health conditions is important and can enhance health outcomes and resource allocations.

Dmitrienko et al. (2016) pointed out several contexts in precision medicine where identifying suitable patient subgroups for particular health technologies is needed:

- *A sponsor is interested in 'salvaging' an experimental treatment following a failed Phase III trial by identifying a subgroup with a substantial treatment benefit*;

- *A sponsor is interested in identifying a subset of 'super-responders' in a successful Phase III trial.*

Successfully identifying a proper patient subgroup in these contexts can influence the economic evaluations for these health interventions. For example, bevacizumab is a drug for epithelial ovarian cancer, and evidence of improved median progression free survival time can be found in two Phase III trials when bevacizumab was added to the treatment for ovarian cancer. However, only a subgroup of ovarian cancer patients benefited remarkably from bevacizumab, while the rest benefited marginally or not at all. Winterhoff et al. (2018) developed a predictive model which identified 20% of patients who benefit significantly from bevacizumab, and the ICER of bevacizumab dropped from $360,000 per Quality-Adjusted Life-Year (QALY) for universal treatment to $120,000 per QALY for treating only the predictive patient subgroup. This example shows that predictive modelling has the potential to improve healthcare resource allocations in precision medicine by identifying the suitable patient subgroup for a certain treatment.

From the examples given above, it can be concluded that patient subgroups can be defined according to the following rules:

1. Patients who have a particularly good or poor prognosis, such as the identification of high-risk patients examples (Hill et al., 2019b,a; Ling et al., 2018; Zack et al., 2019).

2. Patients for whom the treatment effect is particularly strong, such as the bevacizumab example (Winterhoff et al., 2018).

3. Patients for whom the treatment is particularly cost-effective, such as the Gefitinib example (Gavan et al., 2018).

As the capability of collecting and managing data grew rapidly over the past decades, ML as the state-of-the-art predictive method is suggested to be applied to precision medicine for cost-effectiveness analysis (Chen et al., 2020). Compared with statistical methods, ML models perform well in discovering non-linear and non-monotone relationships by processing a large number of covariates simultaneously, and outperformed traditional statistical methods in various real-world applications. For example, a study compared ML methods with conventional statistical methods for survival analysis, and the result showed that when the dimension of covariates is high, ML methods outperformed conventional statistical methods (Spooner et al., 2020). Consequently, ML methods will be applied to build predictive models to identify patient subgroups in this research.

1.3 Background: Survival Analysis

Survival analysis, also known as duration analysis, is the branch of econometrics or statistics to analyse the time until the event of interest occurs based on censored datasets. In general, building predictive ML models to identify high-risk patients and suitable patient subgroups for health interventions involves estimating time-to-event outcomes using data from clinical trials or observational studies. There are several associated difficulties. For example, often some of the data are censored – individuals lost to follow-up or have not experienced the event of interest at the end of the observation period (in this research only right-censoring is considered, as this is the norm). Suppose a study lasts 10 years and the event of interest is death. A large proportion of patients are alive at the end of the study, and then they are censored. The duration between the starting point of observation and the event of interest is called the survival time. Standard ML models cannot be applied to censored data directly. One intuitive method is to delete the censored data and use the rest to build the model. However, it is inappropriate since: (1) the censored data provide partial information – the lower bound of the survival time; and (2) in some observations or at the early stage of observations, a large proportion of patients are censored.

Due to the existence of censoring and the demand of identifying patient subgroups based on censored datasets, survival analysis should be adapted to subgroup identification.

1.4 Background: Multi-task Learning

Multi-task learning (MTL) is a branch of ML, which has been intensively studied lately. In an early study, Caruana (1997) defined MTL as ‘an inductive transfer mechanism whose principal goal is to improve generalization performance by leveraging the domain-specific information contained in the training signals of related tasks.' In plain words, MTL enables algorithms to learn from the experiences of other similar tasks. Caruana (1997) also stated that most real-world problems are multi-task problems. Moreover, Caruana (1997) listed three fields MTL can be applied to, and one of them is healthcare.

There are two reasons MTL is chosen as the ML method to build the predictive model in this research. First, several works have applied MTL to survival analysis, and the results showed that it outperformed several other popular ML methods and conventional statistical methods (Yu et al., 2012; Li et al., 2016; Wang et al., 2017; Liu et al., 2018; Wang et al., 2021). Second, MTL has developed fast over the past decade and has outstanding performance in many real-world applications, and the technique can identify patient subgroups using censored data. MTL has been successfully applied in the fields of computer vision, natural language processing, web, bioinformatics, and so on (Zhang and Yang, 2021).

Zhang and Yang (2018) gave a formal definition of MTL:

Definition 1.4.1 (Multi-task learning). *Given m learning tasks $\{\mathcal{T}_i\}_{i=1}^m$ where all the tasks or a subset of them are related but not identical, multi-task learning aims to learn the m tasks together to improve the learning of a model for each task $\mathcal{T}_i$ by using the knowledge contained in all or some of other tasks.*

This learning strategy is naturally used by humans in daily life because people always transfer knowledge between related things, which is the motivate of MTL. For example, learning Japanese and Mandarin can help each other. When learning Japanese and Mandarin at the same time, one can find that the two languages have many characters in common with the same meaning. So, 'Learning Mandarin' and 'Learning Japanese' are two related but not identical tasks. In this way, common information 'characters' are shared in the related tasks 'Learning Mandarin' and 'Learning Japanese', which improves the results of both tasks. Another example is spam filters. Suppose there are two tasks, a spam filter for Russian users and a spam filter for US users. Russian emails are likely to be spam for US users, but not for Russian users, and vice versa.

1.5 Research Objectives

Current patient subgroups identification methods using ML models based on censored datasets are limited (Huber et al., 2019; Alemayehu et al., 2018; Loh et al., 2019).

The models reviewed in the comparative study suffer from at least one limitation, such as inability to handle censored data and/or defective performance on at least one of the seven rules for subgroup identification (Loh et al., 2019), which are: *(a) bias in selection of subgroup variables; (b) probability of false discovery; (c) probability of identifying correct predictive variables; (d) bias in estimates of subgroup treatment effects; (e) expected subgroup size; (f) expected true treatment effect of subgroups; and (g) subgroup stability.*

The purpose of this research is to build a novel predictive ML model based on MTL to identify patient subgroups using censored datasets, and the main goal is to improve the prediction accuracy of survival time. The prediction of survival time is when given a new patient, the proposed model can predict the survival time of the new patient based on the censored dataset. And the prediction accuracy is assessed by concordance index, which is introduced in Chapter 3. This research is designed as follows:

- First, I propose a new ML model based on MTL to improve the prediction accuracy of time-to-event outcomes when data are censored.
- Second, I apply this new model to real-world and simulated datasets.
- Third, I compare the accuracy of my model in these datasets with other state-of-the-art models, including both conventional and ML models, and calculate the change of ICER in simulated datasets based on the method of subgroup identification.

The predictive model proposed in this research will not satisfy all the seven criteria for subgroup identification. There are two reasons. First, the development of ML models for subgroup identification in precision medicine is still in its infancy. Second, satisfying all the seven criteria is unrealistic at this stage as the comparative studies showed no existing model close to this standard (Huber et al., 2019; Alemayehu et al., 2018; Loh et al., 2019). This research makes unique contribution to the literature as: (1) a novel model for survival analysis based on MTL is developed; and (2) to the best of my knowledge, it is the first to use MTL to identify patient subgroups.

In precision medicine, identification of patient subgroups at risk for certain diseases or health events is required. Successful identification of such patient subgroups can help to enhance healthcare quality, reduce healthcare costs, and assist health decision-making. Moreover, predictive models which can identify suitable patient subgroups for a certain treatment have the potential to impact the ICER of the treatment and thereby improve healthcare resource allocations. So successful identification of patient subgroups based on their individual characteristics in the contexts demonstrated above is an important step towards precision medicine. Moreover, as Innovation and Science Australia identified “using genomics and precision medicine to help Australia become the healthiest country on Earth” as an ideal National Mission (Williamson et al., 2018), this research can ultimately help healthcare providers enhance the uptake of appropriate treatments and improve the well-being and quality of life of Australians.

The primary questions of this research is:

Compared with existing models, can the proposed model improve the accuracy of time-to-event prediction based on the censored dataset, as assessed by the concordance index?

Secondary questions that this research will additionally investigate are:

Compared with existing models, can the proposed model select the correct predictive or prognostic biomarkers? and

How will the ICER change when calculated among selected patient subgroups and whole patient population?

2

Literature Review

In this chapter, the current HTA agencies' guidelines on subgroup identification will be discussed, and important works on subgroup identification, survival analysis, and MTL for survival analysis will be reviewed. Finally, the research gaps will be illustrated.

The definitions of prognostic and predictive biomarkers need to be introduced, as they are mentioned in the guidelines.

Typically, subgroups are defined by biomarkers, which refer to covariates, including clinical, genomic, demographic, and other covariates. There are two types of biomarkers: prognostic and predictive biomarkers. Prognostic biomarkers provide information on disease progression and are independent of treatment. Predictive biomarkers provide information on the effectiveness of treatment.

2.1 Guideline Discussion

Many fear that uncontrolled data ‘dredging’ could lead to inflated treatment effects in selected patient subgroups, so HTA agencies from the UK and Australia have guidelines on the identification of subgroups. Data ‘dredging’ is an abuse of data analysis by reporting only statistically significant results. To avoid data ‘dredging’, the common method is to perform randomised Cross-Validation while training the model. The details of Cross-Validation is in section 4.1.

2.1.1 NICE

The National Institute for Health and Care Excellence (NICE) in England is one of the leading HTA reimbursement organisations. NICE evaluates the clinical effectiveness and cost-effectiveness of medicine, medical devices and diagnostic tests in England. According to the **NICE Health Technology Evaluations: the Manual**[1] (NICE, 2022) published on 31st January 2022:

- *“The characteristics of patients in the subgroup should be clearly defined and should preferably be identified based on an expectation of differential clinical or cost effectiveness because of known, biologically plausible mechanisms, social characteristics or other clearly justified factors.”*
- *“Avoid post hoc data ‘dredging’ in search of subgroup effects, this will be viewed sceptically.”*
- Subgroups can be based on *“characteristics of patients”* or *“differences in baseline risk of specific health outcomes”*.
- *“..., potentially relevant subgroups will be identified at the scoping stage.”*
- *“Individual patient data is preferred.”*

To conclude, NICE prefers:

[1] https://www.nice.org.uk/process/pmg36

1. Subgroups being pre-specified is preferred and post hoc subgroup analysis should be avoided.
2. Subgroups should be biologically plausible.
3. Subgroups can be based on predictive biomarkers (treatment-by-characteristics interaction) or prognostic biomarkers (baseline risk).
4. Analysis of individual patient data is preferred.

2.1.2 PBAC

The Pharmaceutical Benefits Advisory Committee (PBAC) is an independent institute in Australia to make recommendations on which health technology should be publicly funded. The recommendations are made based on the safety, clinical effectiveness and cost-effectiveness of new health technologies. According to the **Guidelines for Preparing a Submission to the Pharmaceutical Benefits Advisory Committee**[2] (PBAC, 2016) published on September 2016:

- *"Provide. .. the plausibility of a variation in treatment effect for the subgroup, as it relates to the pharmacological, biological or clinical rationale for using the medicine."*
- *"The PBAC prefers submissions based on the whole population of a randomised trial."*
- *"For each outcome relevant to the submission, present the relative and absolute treatment effect measures for the whole trial population, the subgroup and the complement."*
- *"Test for interaction between the subgroup and its complement to support and quantify the association between the treatment effect and the covariate defining the subgroup."*

[2] `https://pbac.pbs.gov.au`

To conclude, PBAC prefers:

1. Subgroups being pre-specified is preferred.

2. Subgroups should be biologically plausible.

3. Submissions based on the whole patient population of a randomised trial are preferred.

2.1.3 Discussion

As described above, both guidelines recognise subgroup analysis and both prefer patient subgroups to be biologically and statistically plausible and to be pre-specified. Now considering each of the rules in turn. For the requirement that subgroups should be pre-specified, it will never be known which biomarkers are associated with treatment effect or baseline risk until the data analysis is conducted. Therefore, the model presented in this study can be used as a tool in the analysis of data from early clinical trials by pharmaceutical companies to find potential predictive/prognostic biomarkers. The result of the analysis could inform the design of further trials regarding patient subgroups. For the requirement that subgroups be biologically plausible, this should be left to medical researchers to decide.

For the concerns about data 'dredging' and overfitting, these can be avoided by model validation techniques such as Cross-Validation.

2.2 Subgroup Identification

There are generally three types of subgroup identification methods: basic methods, global outcome modelling methods, and global treatment effect modelling methods (Dmitrienko et al., 2016).

2.2.1 Basic Methods

Basic methods include univariate regression and tree-based regression. The univariate regression method assumes that a set of candidate predictive biomarkers has been specified in advance, and then performs the following three steps:

Step 1:

Use the single biomarker and treatment biomarker interaction terms as covariates and treatment effect as the response variable to fit a group of regressions. For example:

$$y = \sum_{i=1}^{n} c_i x_i + \sum_{i=1}^{n} b_i t \cdot x_i + \varepsilon$$

where x_i are candidate biomarkers, y is treatment effect, t is treatment option, c_i and b_i are coefficients, and n is the number of candidate biomarkers.

Step 2:

If the interaction term b_i is significant, the corresponding biomarker is identified as a predictive biomarker.

Step 3:

Use the identified biomarkers to define a patient subgroup S.

The subgroups defined on binary predictive biomarkers (e.g. positive or negative) are straightforward. If the biomarkers are continuous (e.g., age), a pre-specified reasonable cutoff is needed (e.g., $\geqslant 60$ or < 60 years old). A severe disadvantage of the univariate regression method is that it only considers single biomarkers and ignores the interaction between biomarkers.

Tree-based regression models are one of the most popular tools for identifying potential predictive biomarkers in clinical trials. The ‘tree’ in this model is a ML method – decision tree. Decision trees divide the covariate space into separate areas called leaves of the tree, and assign each patient to a region based on each patient’s covariate.

In univariate regression models, cutoff values for continuous biomarkers need to be specified in advance. In tree-based regression models, continuous biomarkers are segmented during tree fitting. Therefore, an advantage of tree-based regression methods compared to univariate regression methods is that cutoff points for continuous biomarkers do not need to be pre-specified, and the optimal cutoff point can be found by this method.

However, the tree-based regression method still has some deficiencies. For example, tree-based regression methods perform well at identifying prognostic biomarkers but poor at identifying predictive biomarkers (Lipkovich et al., 2017).

2.2.2 Global Outcome Modelling

The basic concept of global outcome modelling methods is to build a model with a relatively large number of candidate biomarkers to predict the outcome. Depending on how the outcome model is fitted, there are two methods in this class: parametric and non-parametric (Lipkovich et al., 2017).

Royston and Sauerbrei (2004) applied factional polynomials (details can be found in Royston and Altman (1994)) to model the relationship between treatment outcome and candidate biomarkers. The results of this type of methods are easy to explain, but only suitable for a relatively small group of candidate biomarkers. In some real-world scenarios, there are hundreds or thousands of candidate biomarkers, in which case we should look for other approaches such as penalised regression methods.

The core concept of penalised regression is to put constraints on the regression coefficients to select predictive biomarkers. One famous penalised regression method called FindIt is developed by Imai and Ratkovic (2013). FindIt uses support vector machine (SVM) along with Least Absolute Shrinkage and Selection Operator (LASSO) penalty to put constrains on predictive and prognostic biomarkers.

In the non-parametric approach, a famous model called Virtue Twins (VT) is designed by Foster et al. (2011). VT has two stages. In the first stage, Foster et al. (2011) applied Random Forest to estimate the outcome function. Random Forest is a black box ML method (details of Random Forest can be found in Breiman (2001)).

Then the treatment contrast for each patient is calculated using the fitted model. The treatment contrast is defined as the difference between treatment effect under different arms of Randomised Controlled Trials. In the second stage, a regression tree was generated using treatment contrast as the response variable. Any patient with a treatment contrast greater than a pre-specified threshold was then selected to the identified patient subgroup S.

2.2.3 Global Treatment Effect Modelling

Global treatment effect modelling methods aim to estimate the treatment contrast directly, bypass the problem of estimating the global outcome. The most famous method in this category is the Interaction Tree (IT) proposed by Su et al. (2008). IT changes the splitting criterion in the tree-based regression method, so that the patient subgroups selected by IT are likely to benefit from the treatment. IT has better property on identifying predictive biomarkers compared with the tree-based regression method (Lipkovich et al., 2017).

2.3 Survival Analysis

Several conventional statistical methods describe the survival function under censoring and the commonly used ones can be grouped into three categories: non-parametric, semi-parametric, and parametric models. Non-parametric models, including Kaplan-Meier estimator, Nelson-Aalen estimator, and life-table method, are efficient when there are no underlying distributions for the survival time. These methods are proposed for generating unbiased descriptive statistics, but generally they cannot be used to assess the impact of covariates on response variables. Therefore, non-parametric models will not be compared with the proposed model in this research. Parametric models assume that survival time follows a distributions, e.g.: exponential distribution, Weibull distribution, (log-)normal, (log-)logistic, or gamma distribution. The parametric models are easy to use and computationally efficient, but the disadvantage is obvious: most high dimensional real-world data do not follow such distributions. In the semi-parametric class, the Cox regression is more commonly used. It is based on the proportional hazard assumption, which means the hazards are proportional over time and the ratio of hazards for any two individuals is constant over time. For parameter estimation, Cox models use partial likelihood. In Cox regression, the underlying distribution is usually unknown, hence it was called semi-parametric method. The Cox regression makes no assumption on the underlying distribution of survival time. However, the assumption that covariates have an exponential influence on the hazard function is not realistic in many real-world applications (Klein and Klein, 2013).

The Cox regression and its two variants – Cox-LASSO and Cox-EN will be compared with the new model proposed in this research.

2.4 Multi-task Learning for Survival Analysis

To date, there are no studies applying MTL to subgroup identification, but studies applying MTL to improve the prediction accuracy of survival analysis have been conducted. There are generally two categories, one is to build multi-task frameworks based on conventional statistical methods (Wang et al., 2017; Yu et al., 2012; Liu et al., 2018), and another is to avoid prior statistical assumptions(e.g., underlying distributions, proportional hazard assumptions) and turn the problem into a series of classification problem (Li et al., 2016; Wang et al., 2021).

Wang et al. (2017) formulated two multi-task models based on Cox regression. They applied the conventional semi-parametric statistical method Cox Regression to build the survival functions for different training sets and treat them as single tasks, then applied MTL to simultaneously process these tasks. The differences between their two models are the methods they applied to share the information between tasks. After that, they compared their model with two variants of regularised Cox regression model: Cox-LASSO and Cox-EN. The evaluation metric they used is the concordance index (C-index), which is commonly used in survival analysis. C-index compares the relative risks between two patients instead of the absolute survival time. Two patients are comparable if their survival times satisfying: (1) both are not censored; (2) one is censored the other is not censored, and the survival time of the not censored patients is smaller than then censored time of the censored patient. C-index is widely used in survival analysis because it considers right-censoring. The C-index is defined as follow:

$$c = \mathbb{P}(\hat{T}_1 > \hat{T}_2 | T_1 > T_2)$$

where $\hat{T}_i$ is the estimated survival time of patient i, and T_i is the real survival time. The results showed that their methods outperformed the traditional methods in prediction accuracy, suggesting that, when the dataset is large, MTL performs well when combined with traditional survival analysis. Their study proposed a unified framework for multi-task survival analysis, and it uses the same way to handle censored data as the Cox models.

A local multi-task logistic regression model (MTLR) was first established by Yu et al. (2012). MTLR directly models the survival function by combining multiple local logistic regression models and assume they are related. The prediction of individual survival status at each time interval was regarded as separate tasks, and the information between related tasks are shared by different regularisation terms. Their method is similar with the one proposed by Wang et al. (2017), the difference is that they use logistic regression while Wang et al. (2017) use Cox regression.

The two studies above belong to the first category – building multi-task framework based on conventional statistical methods (Cox regression and logistic regression). The Cox regression assumes that the covariates have an exponential influence on the outcome, which is often violated in real-world applications. To overcome this weakness, Li et al. (2016) reformulated the survival analysis problem as a MTL problem without prior assumptions. They transferred the data to a target matrix Y and a indicator matrix W. Y is the target matrix, which shows the survival status for each patient in different days (1: alive; 0: dead; ?: censored). W is the indicator matrix, which indicate whether a patient is censored at a given day (1: uncensored; 0: censored). Then they reformulated the objective function as following:

$$\min \frac{1}{2} \|\Pi_W (Y - XB)\|_F^2 + R(B)$$

where X is the covariates matrix and B is the coefficient matrix. $R(B)$ is the regularisation term. $\| \cdot \|_F$ is the Frobenius Norm. The only assumption of this model is the linear assumption. The advantage is that there are few restrictions, while the disadvantage is that the method is a black box. Moreover,

$$(\Pi_W(U))_{ij} = \begin{cases} U_{ij} & \text{if } W_{ij} = 1 \\ 0 & \text{if } W_{ij} = 0 \end{cases}$$

This equation means that the loss function become 0 when data are censored. Li et al. (2016) compared their model to a series of semi-parametric and parametric models, and the results showed that, based on C-index, their model outperformed all the models above in predicting survival time using high-dimensional gene expression data.

2.5 Research Gaps

In some real-world scenarios, especially when evaluating genomic data, there are often far more candidate biomarkers than patients. And as mentioned before, in subgroup identification, when the number of candidate biomarkers is large, the basic and parametric methods in global outcome modelling do not work, so the only option is the non-parametric approach.

As shown in the previous section, non-parametric global outcome modelling methods include two steps. The first step is to fit the treatment outcome function and calculate the treatment contrast for all patients. The second step is to find the appropriate patient subgroup S based on the treatment contrasts. The first step is central because almost all the errors come from the modelling of the treatment outcome function. Therefore, if the prediction accuracy of the treatment outcome function can be improved, the accuracy of subgroup identification will also be improved.

Moreover, the treatment outcome modelling is a survival analysis problem when the data are censored. Although MTL models are proved to have an outstanding prediction ability in survival analysis, to date, there is no research that applies MTL in subgroup identification. This research will propose a novel MTL model for survival analysis, which improves prediction accuracy based on the current MTL framework. This research only addresses the outcome function modelling step in the global outcome modelling method, not the second step.

To conclude, when datasets are censored, the central step of subgroup identification is a survival analysis modelling problem. Improving the prediction accuracy of the survival analysis model can also improve the accuracy of subgroup identification. This study aims to propose a novel MTL model to improve the prediction accuracy of survival analysis and ultimately the accuracy of subgroup identification.

3

Methodology

This chapter will describe all the methods applied in this research, as well as the process of constructing the new model. Section 3.1 introduces the basic definitions of ML to help readers from other backgrounds better understand this thesis. In Section 3.2, the multi-task learning and feature selection methods in multi-task learning are described as they will be used in this research. In Section 3.3, SVMs are introduced as they will be used in the new model to achieve higher classification accuracy. In Section 3.4, all the procedures for building a new model are given. Section 3.5 introduces real and simulated datasets. Section 3.6 presents the methods to be compared in this study. Finally, Section 3.7 introduces the evaluation metric – the concordance index.

3.1 Machine Learning Basics

ML is a part of artificial intelligence which studies computer algorithms that can learn from experience. It means that algorithms can make decisions without explicit instructions. An popular definition of ML was given by Tom Mitchell (Mitchell, 1997):

Definition 3.1.1 (Machine Learning)**.** *A computer program is said to learn from experience E with respect to some class of tasks T, and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.*

There are many types of ML based on different properties of task T, performance measure P, and experience E. Generally speaking, ML can be divided into three categories: supervised learning, unsupervised learning and semi-supervised learning. Some common notations and the three categories of ML will be introduced below.

The input $\boldsymbol{x} = (x_1, \ldots, x_d) \in \mathbb{R}^d$ is called **covariates, features or predictors**, and d is the dimension of covariates (i.e., number of input features). The output $\boldsymbol{y}$ is called **target, label or response**. The experience E used to train the algorithm is called **training set**, which can be written as the form of input-output pairs $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^N$. There are N pairs data points in the training set, and N is defined as the **sample size** of the training set (Murphy, 2012).

3.1.1 Empirical Risk Minimisation

Suppose an ML algorithm has a training set $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^N$ and the goal is to find a map $f : X \to Y$ that minimise the prediction error when given an unknown test set. **Loss function** $L : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is defined to quantify how well the predicted outcome $\hat{y}$ approximates the real y (Murphy, 2012). Here are some common loss functions.

Example 3.1.2 (quadratic loss)**.** *The quadratic/square loss is defined as:*

$$L(\hat{y}, y) = (\hat{y} - y)^2$$

Example 3.1.3 (absolute loss)**.** *The absolute loss is defined as:*

$$L(\hat{y}, y) = |\hat{y} - y|$$

Example 3.1.4 (0-1 loss)**.** *The 0-1 loss is defined as:*

$$L(\hat{y}, y) = \begin{cases} 0, & \textit{if } \hat{y} = y \\ 1, & \textit{if } \hat{y} \neq y \end{cases}$$

Example 3.1.5 (hinge loss)**.** *The hinge loss is commonly used in SVMs:*

$$L(\hat{y}, y) = \max\{0, 1 - \hat{y}y\}$$

Empirical risk is defined to quantify how well the algorithm fit the training set (Murphy, 2012):

Definition 3.1.6 (Empirical Risk)**.** *The empirical risk is defined as the average loss within the training set with respect to some loss function:*

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} L(\hat{y}_i, y_i)$$

If $\hat{y} = f(X, \theta)$, then the empirical risk can be written as:

$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N} L(f(x_i, \theta), y_i)$$

where θ represents parameters.

For example, if quadratic loss is chosen as the loss function in an algorithm, then the corresponding empirical risk is:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2$$

When training an algorithm, the aim is to find a θ^* to minimise the empirical risk, such that:

$$\theta^* = \text{argmin}_{\theta \in \Theta}\mathcal{L}(\theta)$$

where Θ is the parameter space. This process is called **empirical risk minimisation.**

3.1.2 Labelled and Unlabelled Data

In ML, all data can be grouped into two categories: labelled and unlabelled. Label is the output $\boldsymbol{y}$ that the algorithms aim to learn or predict. For example, suppose we want to build an algorithm to identify whether an animal in a picture is a dog or a cat, then the label $\boldsymbol{y} \in \{\text{dog, cat}\}$. Labelled data has labels, while unlabelled data has no labels (see Figure 3.1).

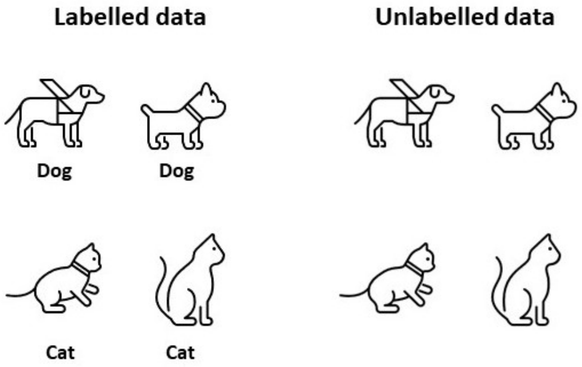


FIGURE 3.1: The label of the pictures on the left is cat or dog, and pictures on the right are unlabelled.

3.1.3 Supervised Learning

Based on the types of the training sets, ML models can be grouped into different categories. Supervised learning is the most common type of ML when the dataset contains data with labels. Supervised models use labelled data in the training set to predict the label of a new data point as shown in Figure 3.2. When given a new image without a label, the model will predict the label of the image. Regression and classification are two common types of supervised learning (Murphy, 2012).

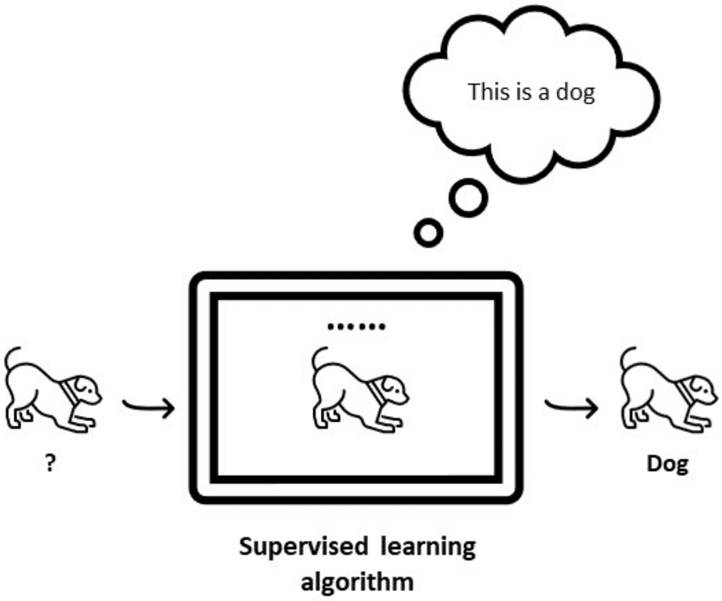


Figure 3.2: Supervised learning predict the label of a new data point.

3.1.4 Unsupervised Learning

The difference between unsupervised learning and supervised learning is that the training set given to unsupervised learning has no label. When the dataset has no labels, there is no target to predict. For example, if we have a dataset with different animals but unlabelled, then we cannot use it to predict the label of a new picture. However, we can still obtain useful information from unlabelled datasets. If we are given a dataset with images of dogs and cats but without labels, unsupervised learning models can divide images to different groups based on their similarities without knowing what each group represents (Murphy, 2012). Figure 3.3 shows an example of unsupervised learning.

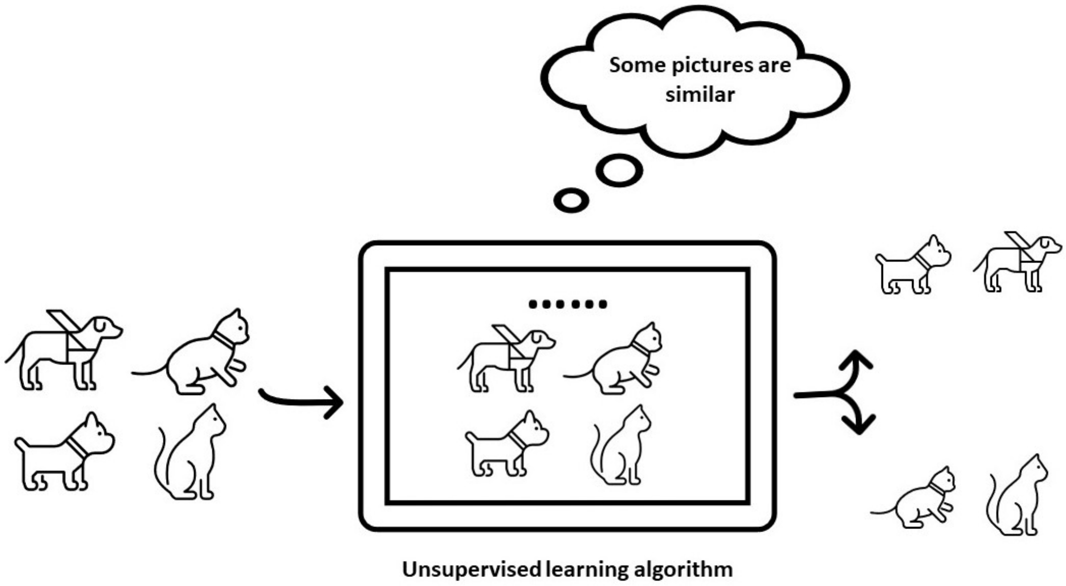


FIGURE 3.3: Unsupervised learning group similar data together.

3.1.5 Semi-supervised Learning

Semi-supervised learning is a method in ML to build models based on datasets which contain a small proportion of labelled and a large proportion of unlabelled data. It falls between supervised learning and unsupervised learning, and aims to combine the advantages of these two methods.

3.2 Multi-task Learning

3.2.1 Introduction

Recall the definition of Multi-task learning (MTL):

Definition 3.2.1 (Multi-task Learning). *Given m learning tasks $\{\mathcal{T}_i\}_{i=1}^m$ where all the tasks or a subset of them are related but not identical, multi-task learning aims to learn the m tasks together to improve the learning of a model for each task $\mathcal{T}_i$ by using the knowledge contained in all or some of other tasks.*

As the definition shows, the basic concept of MTL is to share information among related tasks, rather than training each task individually. When building MTL models, two questions need to be answered (Zhang and Yang, 2021). First, what kind of information is shared? Second, how to share the selected information?

For the first question, according to the literature, three types of information that need to be shared: features, instances (i.e., samples), and parameters (Zhang and Yang, 2021). The feature-based MTL models assume that related tasks share a common set of features. The instance-based MTL models select useful data samples in each task and share them with other tasks. The parameter-based MTL models assume that related tasks share a common set of parameters.

In survival analysis and subgroup identification problems, the dataset is usually a clinical dataset, the features are candidate biomarkers, and the response variable is the patient's survival time. The purpose of these questions is to find prognostic or predictive biomarkers and predict survival time among all candidate biomarkers, so it is reasonable to assume that all tasks share a common set of biomarkers. So, for the second question, the features-based MTL is chosen in this research.

3.2.2 Feature Selection Approach

In this research, the feature-based MTL approach is used to select prognostic or predictive biomarkers. Typically, $\ell_{p,q}$ norm is used to select features in MTL (Zhang and Yang, 2021). $\ell_{p,q}$ norm is a matrix norm, and matrix norm is derived from vector norm. Vector norm is defined as (Royden and Fitzpatrick, 1988):

Definition 3.2.2 (Vector Norm)**.** *Given a vector space* $\mathbf{X}$*, a norm* $\|\cdot\|$ *is a function* $f : X \to \mathbb{R}$ *satisfies:*

1. $\|\boldsymbol{x} + \boldsymbol{y}\| \leqslant \|\boldsymbol{x}\| + \|\boldsymbol{y}\|$ *for any* $\boldsymbol{x}, \boldsymbol{y} \in \mathbf{X}$
2. $\|a\boldsymbol{x}\| = |a| \cdot \|\boldsymbol{x}\|$ *for any* $a \in \mathbb{R}$*,* $\boldsymbol{x} \in \mathbf{X}$
3. $\|\boldsymbol{x}\| = 0$ *if and only if* $\boldsymbol{x} = 0$
4. $\|\boldsymbol{x}\| \geqslant 0$ *for any* $\boldsymbol{x} \in \mathbf{X}$

ℓ_p norm is a type of vector norm that is widely used in feature selection methods in statistics. ℓ_p norm is defined as (Horn and Johnson, 2012):

Definition 3.2.3 (ℓ_p norm)**.** *For a vector $\boldsymbol{x} = (x_1, \ldots, x_n)^\top$, the ℓ_p norm of vector $\boldsymbol{x}$ is:*

$$\|\boldsymbol{x}\|_p = (\sum_{i=1}^{n} |x_i|^p)^{\frac{1}{p}}, \quad p \geqslant 1$$

Note that when $p = 2$,

$$\|\boldsymbol{x}\|_2 = \sqrt{x_1^2 + \cdots + x_n^2}$$

is the Euclidean norm.

The entry-wise matrix norm $\ell_{p,q}$ is defined as (Horn and Johnson, 2012):

Definition 3.2.4 ($\ell_{p,q}$ norm)**.** *Suppose matrix $\mathbf{A} = (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_m) \in \mathbb{R}^{n\times m}$, then the $\ell_{p,q}$ norm of matrix $\mathbf{A}$ is:*

$$\begin{aligned}\|\mathbf{A}\|_{p,q} &= \|(\|\boldsymbol{a}_1\|_p, \ldots, \|\boldsymbol{a}_m\|_p)\|_q \\ &= \left[\sum_{j=1}^{m}\left(\sum_{i=1}^{n}|a_{ij}|^p\right)^{\frac{q}{p}}\right]^{\frac{1}{q}}\end{aligned}$$

The objective function of a feature-based MTL model usually has the following form (Zhang and Yang, 2021):

$$\mathcal{L}(\mathbf{W}, \boldsymbol{b}) + \lambda\|\mathbf{W}\|_{p,q}$$

where $\mathbf{W}$ is the coefficient/weight matrix, and the ith column of $\mathbf{W}$ represents the coefficients of task i. $\boldsymbol{b}$ is the bias vector, representing the bias for all tasks. The $\ell_{p,q}$ norm placed on the coefficient matrix $\mathbf{W}$ forces $\mathbf{W}$ to be row-sparse, which means that most rows of $\mathbf{W}$ are 0. Then non-zero rows represent important features that need to be selected.

3.3 Support Vector Machine

SVMs are one of the best supervised learning algorithms in existence, and many believe it is indeed the best (Ng, 2018). The approach is highly suitable for classification tasks and has shown decent performance. At the finest, SVMs map training samples into multi-dimensional space and attempt to partition the space into different classes. The boundaries are drawn in a way so that the distances between sample points and the boundaries are maximised. This section is based on Chapter 17 of Murphy (2012).

3.3.1 Introduction

SVMs classify sample data by drawing hyperplanes in d-dimensional space which act as boundaries to define regions corresponding to different classes. Here, d is equivalent to the number of features in the data. When d equals to 2, the task is to draw linear lines to separate different classes on a Cartesian plane.

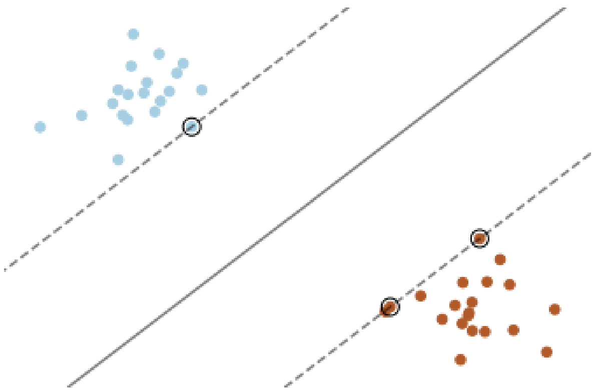


Figure 3.4: An illustration of SVM mechanism in 2-dimensional space with 2 classes of samples.

When data samples are scattered in a way that one cannot draw a linear line to separate classes, the kernel trick is often applied where the original data points $\boldsymbol{x}_i^\top$ are transformed through a kernel function $\emptyset$ reaching new data points $\emptyset(\boldsymbol{x}_i^\top)$. The new data points may be separated into classes with linear lines and this is equivalent to non-linear boundaries on the original data set.

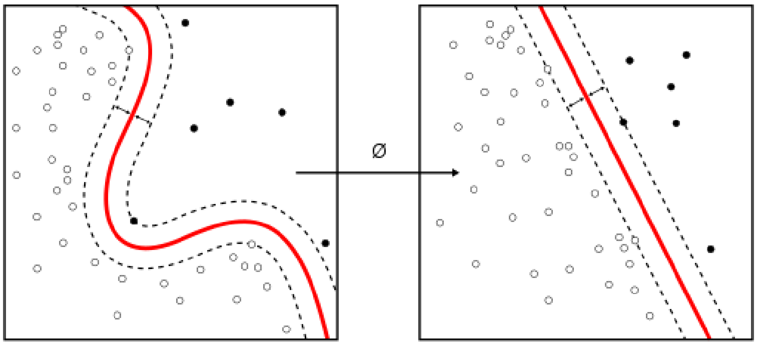


FIGURE 3.5: An illustration of kernel trick with kernel Ø (Adapted from *Kernel Machine* [Image], by Alisneaky, 2011, licensed under CC BY-SA 4.0).

Without loss of generality, denote $\boldsymbol{x}_i^\top$ as the data points where linear boundaries are to be applied over with, whether this is the raw data or the kernel transformed data. Then, SVMs seek to determine the optimal hyperplane of the form

$$\boldsymbol{w}^\top \boldsymbol{x} - \boldsymbol{b} = 0$$

where $\boldsymbol{x}$ here represents a generic variable while $\boldsymbol{w}$ and $\boldsymbol{b}$ are to be determined via optimisation. For simplicity, we only introduce SVMs over binary classification as multi-class classification can be viewed as repeated pairwise binary classifications. We denote the responses as y_i taking values ± 1.

Hard-Margin

The simple scenario is when the data points can be perfectly separated by a hyperplane into two classes. This often means that there exists infinitely many hyperplanes satisfying the classes separation criteria. Hence the definition of optimal hyperplane is needed for a unique solution. Intuitively, the best boundary should be such that its distance to the data points are maximised. Thus two additional parallel hyperplanes can be defined as

$$\boldsymbol{w}^\top \boldsymbol{x} - \boldsymbol{b} = 1 \quad \text{and} \quad \boldsymbol{w}^\top \boldsymbol{x} - \boldsymbol{b} = -1$$

where the distance between the two hyperplanes are given by $\frac{2}{||\boldsymbol{w}||_2}$. Thus the space may be partitioned into 3 regions

$$\begin{cases} \left\{\boldsymbol{x} : \boldsymbol{w}^\top \boldsymbol{x} - \boldsymbol{b} \geqslant 1\right\}; \\ \left\{\boldsymbol{x} : \boldsymbol{w}^\top \boldsymbol{x} - \boldsymbol{b} \leqslant -1\right\}; \\ \left\{\boldsymbol{x} : \boldsymbol{w}^\top \boldsymbol{x} - \boldsymbol{b} \in (-1, 1)\right\}. \end{cases}$$

If the regions are assigned as

$$\begin{cases} \boldsymbol{w}^\top \boldsymbol{x}_i - \boldsymbol{b} \geqslant 1 & \text{if } y_i = 1; \\ \boldsymbol{w}^\top \boldsymbol{x}_i - \boldsymbol{b} \leqslant -1 & \text{if } y_i = -1, \end{cases}$$

they can be reformulated into

$$y_i\left(\boldsymbol{w}^\top \boldsymbol{x}_i - \boldsymbol{b}\right) \geqslant 1$$

as a condition that must be met. Now one may maximise the distance $\frac{2}{||\boldsymbol{w}||}$ by minimising $||\boldsymbol{w}||$ subject to the above constraint. This is a formalised optimisation problem solvable by many algorithms.

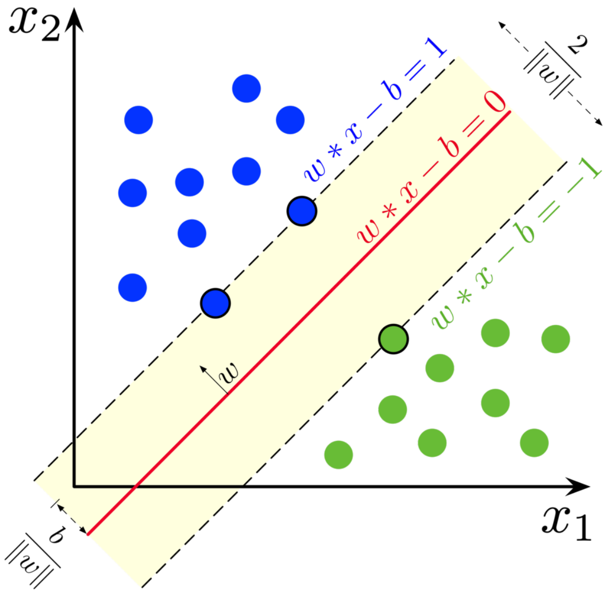


FIGURE 3.6: A visualisation of the optimisation problem in 2-dimensional feature space (Adapted from *SVM margin* [Image], by Larhmam, 2018, liscensed under CC BY-SA 4.0).

Soft-Margin

It is often not possible to perfectly separate two classes of data without extensive search for kernels. Often, it is not ideal to separate the classes perfectly over the training set as this is a recipe for overfitting. Hence, most applications of SVMs adapt soft-margins where some data points are allowed to be separated to the wrong side by the hyperplane boundary. In this case, for some data points x_i, the original constraint

$$y_i\big(\boldsymbol{w}^\top \boldsymbol{x}_i - \boldsymbol{b}\big) \geqslant 1$$

or equivalently

$$1 - y_i\big(\boldsymbol{w}^\top \boldsymbol{x}_i - \boldsymbol{b}\big) \leqslant 0$$

does not necessarily have to be acknowledged. Nevertheless, one still aims to ensure that most points in the training sample are being allocated to the correct region and those being allocated to the wrong region should not be too far from the boundary. The latter means that whenever a point is misallocated,

$$1 - y_i\big(\boldsymbol{w}^\top \boldsymbol{x}_i - \boldsymbol{b}\big)$$

which is always greater than 0, should be minimised. Combining the two goals, SVMs with soft-margins hold objective function

$$\frac{1}{2}\|\boldsymbol{w}\|_2^2 + \Big[C\sum_{i=1}^{n} \max\big(0, 1 - y_i\big(\boldsymbol{w}^\top \boldsymbol{x}_i - \boldsymbol{b}\big)\big)\Big]$$

where $C > 0$ defines a trade-off between keeping the two classes away from each other and correct allocation of training samples. In practice, the value of C is determined via hyperparameter search in the effort of minimising generalisation loss.

Computing SVMs

Let

$$\zeta_i = \max\big(0, 1 - y_i\big(\boldsymbol{w}^\top \boldsymbol{x}_i - \boldsymbol{b}\big)\big) .$$

The primal problem of SVMs is:

$$\begin{aligned} \min_{\boldsymbol{w}} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_{i=1}^{n}\zeta_i \\ \text{s.t.} \quad & y_i\big(\boldsymbol{w}^\top \boldsymbol{x}_i - \boldsymbol{b}\big) \geqslant 1 - \zeta_i \\ & \zeta_i \geqslant 0, \qquad \forall i \end{aligned}$$

Optimisation theories allow further simplification via the duality principle. The dual form can be efficiently solved by any quadratic optimisation problem and it seeks to maximise with c_i:

$$\begin{aligned} \max_{c_i} \quad & \sum_{i=1}^{n} c_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i c_i (\boldsymbol{x}_i^\top \boldsymbol{x}_j) y_j c_j \\ \text{s.t.} \quad & \sum_{i=1}^{n} c_i y_i = 0 \\ & 0 \leqslant c_i \leqslant C, \qquad \forall i \end{aligned}$$

$\boldsymbol{w}$ is subsequently computed as

$$\boldsymbol{w} = \sum_{i=1}^{n} c_i y_i \boldsymbol{x}_i$$

while b can be found with any point $\boldsymbol{x}_i$ on the margin via

$$b = \boldsymbol{w}^\top \boldsymbol{x}_i - y_i \,.$$

3.4 SVM-based MTL for Survival Analysis

In this section, the process of constructing a new algorithm will be described. Section 3.4.1 describes the notation used in the new algorithm, and Section 3.4.2 explains the details of how the objective function is chosen. After the objective function is determined, the problem becomes an optimisation problem. Then Section 3.4.3 proves that the objective function is a convex function defined on a convex set such that every local minimum is a global minimum. Finally, the optimisation problem is solved and the pseudo-code of the algorithm is given.

3.4.1 Notations

Suppose the sample size of the training set is n, and there are d candidate biomarkers. Let $\mathbf{X} \in \mathbb{R}^{n\times d}$ denote the input matrix:

$$\mathbf{X} = \begin{pmatrix} \boldsymbol{x}_1^\top \\ \boldsymbol{x}_2^\top \\ \vdots \\ \boldsymbol{x}_n^\top \end{pmatrix} = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1d} \\ x_{21} & x_{22} & \dots & x_{2d} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nd} \end{pmatrix} \in \mathbb{R}^{n\times d}$$

each row of $\mathbf{X}$ is a sample with d features:

$$\boldsymbol{x}_i^\top = (x_{i1}, \dots, x_{id}), \qquad i = 1, 2, \dots, n$$

Suppose there are m time intervals, then the target matrix $\mathbf{Y}$ is:

$$\mathbf{Y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \dots, \boldsymbol{y}_m) = \begin{pmatrix} y_{11} & y_{12} & \cdots & y_{1m} \\ y_{21} & y_{22} & \cdots & y_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ y_{n1} & y_{n2} & \cdots & y_{nm} \end{pmatrix} \in \mathbb{R}^{n\times m}$$

for any $i \in \{1, \dots, n\}$ and $j \in \{1, \dots m\}$

$$y_{ij} = \begin{cases} 1 & \text{sample } i \text{ is alive at time interval } j \\ -1 & \text{sample } i \text{ is dead at time interval } j \\ ? & \text{sample } i \text{ is censored at time interval } j \end{cases}$$

each column of $\mathbf{Y}$ represent the survival status of all the sample points at the corresponding time interval:

$$y_j = \begin{pmatrix} y_{1j} \\ y_{2j} \\ \vdots \\ y_{nj} \end{pmatrix}, \qquad j = 1, 2, \dots, m$$

An extra matrix $\mathbf{B}$ is used to indicate the censoring information. $\mathbf{B}$ has the same dimensions as $\mathbf{Y}$:

$$\mathbf{B} = \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1m} \\ b_{21} & b_{22} & \dots & b_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ b_{n1} & b_{n2} & \dots & b_{nm} \end{pmatrix} \in \mathbb{R}^{n \times m}$$

where each element of $\mathbf{B}$ represents the censoring status of sample point i at time interval j:

$$b_{ij} = \begin{cases} 1 & \text{sample } i \text{ is uncensored at time interval } j \\ 0 & \text{sample } i \text{ is censored at time interval } j \end{cases}$$

Without loss of generality, assume the map $f : \mathbf{X} \to \mathbf{Y}$ has the form:

$$\mathbf{Y} = \mathbf{X}\mathbf{W} + \boldsymbol{b} = (\mathbf{1}, \mathbf{X}) \begin{pmatrix} \boldsymbol{b}^\top \\ \mathbf{W} \end{pmatrix}$$

where $\boldsymbol{b} \in \mathbb{R}^{m \times 1}$ is a vector represents the bias terms, and $\mathbf{1}^\top = (1, \dots, 1) \in \mathbb{R}^{n \times 1}$ is a vector of 1.

And $\mathbf{W}$ is the coefficient/weight matrix:

$$\mathbf{W} = (\boldsymbol{w}_1, \dots, \boldsymbol{w}_m) = \begin{pmatrix} w_{11} & w_{12} & \dots & w_{1m} \\ w_{21} & w_{22} & \dots & w_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ w_{d1} & w_{d2} & \vdots & w_{dm} \end{pmatrix} \in \mathbb{R}^{d \times m}$$

To make the notation simple, let

$$\mathbf{X} = (\mathbf{1}, \mathbf{X}), \qquad \mathbf{W} = \begin{pmatrix} \boldsymbol{b}^\top \\ \mathbf{W} \end{pmatrix}$$

then the new $\mathbf{X}$ and $\mathbf{W}$ are:

$$\mathbf{X} = \begin{pmatrix} \boldsymbol{x}_1^\top \\ \boldsymbol{x}_2^\top \\ \vdots \\ \boldsymbol{x}_n^\top \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & x_{12} & \dots & x_{1d} \\ 1 & x_{21} & x_{22} & \dots & x_{2d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{nd} \end{pmatrix} \in \mathbb{R}^{n \times (d+1)}$$

where

$$\boldsymbol{x}_i^\top = (1, x_{i1}, x_{i2}, \dots, x_{im}), \qquad i = 1, 2, \dots n$$

and

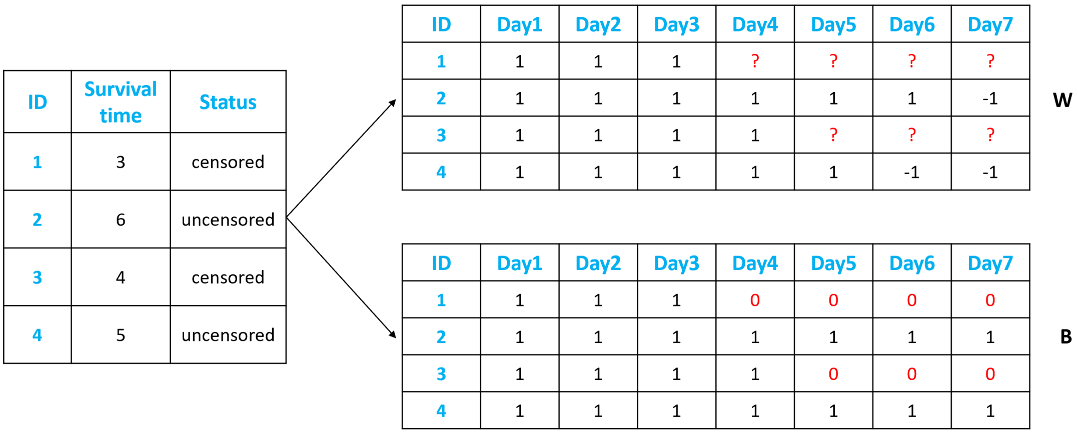
$$\mathbf{W} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \dots, \boldsymbol{w}_m) = \begin{pmatrix} b_1 & b_2 & \dots & b_m \\ w_{11} & w_{12} & \dots & w_{1m} \\ w_{21} & w_{22} & \dots & w_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ w_{d1} & w_{d2} & \vdots & w_{dm} \end{pmatrix} \in \mathbb{R}^{(d+1)\times m}$$

where

$$\boldsymbol{w}_j^\top = (b_j, w_{1j}, \dots, w_{dj}), \qquad j = 1, 2, \dots, m$$

3.4.2 Objective Function

First, the original dataset is decomposed to a series of classification problems follow the work of Li et al. (2016). The transformation process is shown in Figure 3.7.



ID	Survival time	Status
1	3	censored
2	6	uncensored
3	4	censored
4	5	uncensored

W

ID	Day1	Day2	Day3	Day4	Day5	Day6	Day7
1	1	1	1	?	?	?	?
2	1	1	1	1	1	1	-1
3	1	1	1	1	?	?	?
4	1	1	1	1	1	-1	-1

B

ID	Day1	Day2	Day3	Day4	Day5	Day6	Day7
1	1	1	1	0	0	0	0
2	1	1	1	1	1	1	1
3	1	1	1	1	0	0	0
4	1	1	1	1	1	1	1

FIGURE 3.7: The original dataset (left) is transferred into a weight matrix **W** and an indicator matrix **B** (right).

After the data transformation, the goal of the MTL model is to train m SVMs simultaneously on $\mathbf{X}$, and each SVM determine the survival status of all patients at the corresponding time interval. Therefore, for the jth SVM:

$$\hat{\boldsymbol{y}}_j = \text{sign}(\mathbf{X}\boldsymbol{w}_j)$$

where $\boldsymbol{w}_j \in \mathbb{R}^{(d+1)\times 1}$, and $\hat{\boldsymbol{y}}_j$ is the **predicted** survival status of all patients at time interval j.

Before deciding the objective function, it is necessary to carefully observe the characteristics of the problem to be solved and select the most suitable objective function. Observe that:

- *feature selection* The datasets used in this research are survival datasets, so they are assumed to be collected from clinical trials or observational studies with a relatively large number of covariates and contain censored data. In the problem of identifying high-risk patients for a disease or identifying appropriate patients for a treatment, it is natural and reasonable to assume that disease progression or treatment response y is determined by a set of prognostic or predictive biomarkers $\mathbf{X} = (\boldsymbol{x}_1 \cdots, \boldsymbol{x}_v)$, and $y = f(\boldsymbol{X})$ where $f : \mathbf{X} \to Y$ is a map. Individual patients have different prognostic or predictive biomarker values, and therefore different disease progression or response to treatment. The new algorithm has two goals: (1) to select prognostic or predictive biomarkers from a set of candidate biomarkers, which is called feature selection; (2) to predict y. Features selection approaches in MTL have been discussed in section 3.3.2. $\ell_{2,1}$ norm is chosen in the new algorithm to force the coefficient matrix $\mathbf{W}$ to be row-sparse, and select important features.

- *non-recurring* Some events are non-recurring, such as death. For those recurring events such as rehospitalisation, they are treated as one-time events in this research. This means that if a sample i in time interval j has status $y_{ij} = -1$, then for any time interval $k > j$, its status remains $y_{ik} = -1$.

In short, the problem has these characteristics:

1. All tasks share a common set of features;
2. For any sample i, $\hat{y}_{ij} > \hat{y}_{ik}, \forall j < k$.

Based on these characteristics, the objective function has the form:

$$\mathcal{L}(\mathbf{W}) + \lambda_1 R(\mathbf{W}) + \lambda_2 \mathcal{L}_1(\mathbf{W}) \qquad \lambda_1, \lambda_2 > 0 \tag{3.4.1}$$

The first part $\mathcal{L}(\mathbf{W})$ is a common loss function for evaluating prediction errors. The second part $R(\mathbf{W})$ is the regularisation term for selecting features. The third part $\mathcal{L}_1(\mathbf{W})$ is used to ensure non-recurring. Also, λ_1 and λ_2 are two hyperparameters that need to be selected manually. These terms are explained below.

$\mathcal{L}(\mathbf{W})$: A Modified Hinge Loss

Hinge loss is selected in this research as it is the most common loss function in SVMs. Hinge loss is defined as:

$$hinge(y, \hat{y}) = \max\{0, 1 - y\hat{y}\}$$

However, it cannot be directly applied to censored datasets because the exact survival times of censored patients are unknown. Therefore, in this research, the hinge loss is modified to include the censoring information, the same as Li et al. (2016) modifying their loss function.

Define the modified loss function as:

$$L(y_{ij}, \hat{y}_{ij}) = \begin{cases} hinge(y_{ij}, \hat{y}_{ij}) & \text{if } b_{ij} = 1 \\ 0 & \text{if } b_{ij} = 0 \end{cases}$$

It can be written as

$$L(y_{ij}, \hat{y}_{ij}) = b_{ij} \max\{0, 1 - y_{ij}\hat{y}_{ij}\}$$

In this way, the censoring information b_{ij} for each patient is embedded into the loss function.

$R(\mathbf{W})$: $\ell_{2,1}$ Norm for Feature Selection

Follow the general approach of feature-based MTL models (Zhang and Yang, 2021), defined $\|\mathbf{W}\|_{2,1}$ as:

$$\|\mathbf{W}\|_{2,1} = \sum_{i=1}^{d+1} \|\boldsymbol{w}^i\|_2 = \sum_{i=1}^{d+1} \sqrt{\sum_{j=1}^{m} w_{ij}^2} \tag{3.4.2}$$

where $\boldsymbol{w}^i$ is the ith row of weight matrix $\mathbf{W}$. Note the different between $\boldsymbol{w}_i$ and $\boldsymbol{w}^i$, where $\boldsymbol{w}_i$ is the ith column of $\mathbf{W}$. During the minimisation of the objective function, most $\|\boldsymbol{w}^i\|_2$ will be forced to zero. Therefore, the non-zero features are the most important ones. In this way, prognostic and predictive biomarkers are selected.

$\mathcal{L}_1(\mathbf{W})$: Term to Ensure Non-recurring

To ensure non-recurring, $\mathcal{L}_1(\mathbf{W})$ is set in a way so that every time the predicted survival status $\hat{y}_{ij}$ is smaller than $\hat{y}_{ik}$ for any $k > j$, it will incurs a loss. $\mathcal{L}_1(\mathbf{W})$ can be written as:

$$\mathcal{L}_1(\mathbf{W}) = \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=j+1}^{m} I(\hat{y}_{ij} < \hat{y}_{ik})(\boldsymbol{x}_i^\top \boldsymbol{w}_k - \boldsymbol{x}_i^\top \boldsymbol{w}_j)$$

Combining all the terms, the objective function becomes:

$$\sum_{i=1}^{n}\sum_{j=1}^{m} b_{ij} \max\{0, 1 - y_{ij}\boldsymbol{x}_i^\top \boldsymbol{w}_j\} + \lambda_1\|\mathbf{W}\|_{2,1} + \lambda_2 \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=j+1}^{m} I(\hat{y}_{ij} < \hat{y}_{ik})(\boldsymbol{x}_i^\top \boldsymbol{w}_k - \boldsymbol{x}_i^\top \boldsymbol{w}_j) \tag{3.4.3}$$

Observe that $\|\mathbf{W}\|_{2,1}$ is represented by rows $\boldsymbol{w}^i$, while the other two terms are represented by columns $\boldsymbol{w}_j$. Therefore, the form of $\|\mathbf{W}\|_{2,1}$ needs to be changed. The following theorem and corollary allow $\|\mathbf{W}\|_{2,1}$ to be represented by columns (Yang et al., 2012; Wang et al., 2016).

Theorem 3.4.1.

$$\|\mathbf{W}\|_{2,1} = tr(\mathbf{W}^\top \mathbf{D}\mathbf{W}) \tag{3.4.4}$$

where $\mathbf{W} \in \mathbb{R}^{n\times m}, (n, m > 0)$ *and* $\mathbf{D}$ *is a diagonal matrix:*

$$\mathbf{D} = \begin{pmatrix} \frac{1}{\|\boldsymbol{w}^1\|_2} & & \\ & \ddots & \\ & & \frac{1}{\|\boldsymbol{w}^n\|_2} \end{pmatrix}$$

where $\boldsymbol{w}^i$ *denote the ith row of* $\mathbf{W}$*, and* $tr(\cdot)$ *is the trace of a matrix.*

Proof.

$$\begin{aligned}
\mathbf{W}^\top\mathbf{D}\mathbf{W} &= \begin{pmatrix}\boldsymbol{w}^1 & \dots & \boldsymbol{w}^n\end{pmatrix}\begin{pmatrix}\frac{1}{\|\boldsymbol{w}^1\|_2} & & \\ & \ddots & \\ & & \frac{1}{\|\boldsymbol{w}^n\|_2}\end{pmatrix}\begin{pmatrix}\boldsymbol{w}^{1\top}\\ \vdots \\ \boldsymbol{w}^{n\top}\end{pmatrix}\\
&= \begin{pmatrix}\frac{\boldsymbol{w}^1}{\|\boldsymbol{w}^1\|_2} & \dots & \frac{\boldsymbol{w}^n}{\|\boldsymbol{w}^n\|_2}\end{pmatrix}\begin{pmatrix}\boldsymbol{w}^{1\top}\\ \vdots \\ \boldsymbol{w}^{n\top}\end{pmatrix}\\
&= \sum_{i=1}^{n}\frac{\boldsymbol{w}^i\boldsymbol{w}^{i\top}}{\|\boldsymbol{w}^i\|_2}
\end{aligned}$$

So,

$$\begin{aligned}
tr(\mathbf{W}^\top\mathbf{D}\mathbf{W}) &= tr(\sum_{i=1}^{n}\frac{\boldsymbol{w}^i\boldsymbol{w}^{i\top}}{\|\boldsymbol{w}^i\|_2}) = \sum_{i=1}^{n} tr(\frac{\boldsymbol{w}^i\boldsymbol{w}^{i\top}}{\|\boldsymbol{w}^i\|_2}) = \sum_{i=1}^{n}\frac{tr(\boldsymbol{w}^i\boldsymbol{w}^{i\top})}{\|\boldsymbol{w}^i\|_2}\\
&= \sum_{i=1}^{n}\frac{tr(\boldsymbol{w}^{i\top}\boldsymbol{w}^i)}{\|\boldsymbol{w}^i\|_2} \qquad \text{(cyclic property)}\\
&= \sum_{i=1}^{n}\sum_{j=1}^{m}\frac{w_{ij}^2}{\|\boldsymbol{w}^i\|_2} = \sum_{i=1}^{n}\frac{\|\boldsymbol{w}^i\|_2^2}{\|\boldsymbol{w}^i\|_2}\\
&= \sum_{i=1}^{n}\|\boldsymbol{w}^i\|_2 = \|\mathbf{W}\|_{2,1}
\end{aligned}$$

□

Corollary 3.4.2.

$$\|\mathbf{W}\|_{2,1} = \sum_{j=1}^{m}\boldsymbol{w}_j^\top\mathbf{D}\boldsymbol{w}_j \tag{3.4.5}$$

where $\mathbf{W} \in \mathbb{R}^{n\times m}, (n, m > 0)$ *and* $\mathbf{D}$ *is a diagonal matrix:*

$$\mathbf{D} = \begin{pmatrix}\frac{1}{\|\boldsymbol{w}^1\|_2} & & \\ & \ddots & \\ & & \frac{1}{\|\boldsymbol{w}^n\|_2}\end{pmatrix}$$

where $\boldsymbol{w}^i$ *denote the* i*th row of* $\mathbf{W}$, *and* $\boldsymbol{w}_j$ *denote the* j*th column of* $\mathbf{W}$.

Proof. According to Theorem 3.4.1, we know that

$$\begin{aligned}
\|\mathbf{W}\|_{2,1} &= tr(\mathbf{W}^\top \mathbf{D}\mathbf{W}) = tr(\mathbf{D}\mathbf{W}\mathbf{W}^\top) \quad \text{(cyclic property)} \\
&= tr(\mathbf{D}\begin{pmatrix}\boldsymbol{w}_1 & \dots & \boldsymbol{w}_m\end{pmatrix}\begin{pmatrix}\boldsymbol{w}_1^\top \\ \vdots \\ \boldsymbol{w}_m^\top\end{pmatrix}) \\
&= tr(\mathbf{D}\sum_{j=1}^{m}\boldsymbol{w}_j\boldsymbol{w}_j^\top) \\
&= tr(\sum_{j=1}^{m}\mathbf{D}\boldsymbol{w}_j\boldsymbol{w}_j^\top) \\
&= \sum_{j=1}^{m} tr(\mathbf{D}\boldsymbol{w}_j\boldsymbol{w}_j^\top) \\
&= \sum_{j=1}^{m} tr(\boldsymbol{w}_j^\top\mathbf{D}\boldsymbol{w}_j) \\
&= \sum_{j=1}^{m} \boldsymbol{w}_j^\top\mathbf{D}\boldsymbol{w}_j
\end{aligned}$$

□

According to Theorem 3.4.1 and corollary 3.4.1, formula (3.4.3) can be written as

$$\sum_{i=1}^{n}\sum_{j=1}^{m} b_{ij}\max\{0, 1-y_{ij}\boldsymbol{x}_i^\top\boldsymbol{w}_j\}+\lambda_1\sum_{j=1}^{m}\boldsymbol{w}_j^\top\mathbf{D}\boldsymbol{w}_j+\lambda_2\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=j+1}^{m} I(\hat{y}_{ij} < \hat{y}_{ik})(\boldsymbol{x}_i^\top\boldsymbol{w}_k-\boldsymbol{x}_i^\top\boldsymbol{w}_j) \tag{3.4.6}$$

When $\|\boldsymbol{w}^i\| \leqslant \varepsilon, i = 1, \dots, n$, then set $\|\boldsymbol{w}^i\| = \varepsilon$. ε is a small number which can be chosen manually.

According to empirical risk minimisation, the training goal is to find a $\mathbf{W}$ such that:

$$\begin{aligned}
\min_{\mathbf{W}} \sum_{i=1}^{n}\sum_{j=1}^{m} b_{ij}\max\{0, 1-y_{ij}\boldsymbol{x}_i^\top\boldsymbol{w}_j\} + \lambda_1\sum_{j=1}^{m}\boldsymbol{w}_j^\top\mathbf{D}\boldsymbol{w}_j \\
+ \lambda_2\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=j+1}^{m} I(\hat{y}_{ij} < \hat{y}_{ik})(\boldsymbol{x}_i^\top\boldsymbol{w}_k - \boldsymbol{x}_i^\top\boldsymbol{w}_j)
\end{aligned} \tag{3.4.7}$$

where $\lambda_1 > 0$ and $\lambda_2 > 0$.

3.4.3 Optimisation

Formula (3.4.7) can be seen as a function $f : \mathbf{W} \to \mathbb{R}$, where $\mathbf{W} \in \mathbb{R}^{(d+1)\times m}$. The goal is to find a $\mathbf{W}^* \in \mathbb{R}^{(d+1)\times m}$ such that

$$f(\mathbf{W}^*) \leqslant f(\mathbf{W}), \qquad \forall \mathbf{W} \in \mathbb{R}^{(d+1)\times m} \tag{3.4.8}$$

The matrix $\mathbf{W}^*$ which satisfies (3.4.8) is called a **global optimum** or **global minimum** of function f, and the process of finding $\mathbf{W}^*$ is called **optimisation** (Murphy, 2012).

However, sometimes it is difficult to find the global optimum. This process can be simplified by requiring f to be a convex function defined on a convex set, so that every local optimum is a global optimum. Now the definition of convex set and convex function will be explained.

Convexity

This part is based on Chapter 8 of Murphy (2012).

Definition 3.4.3 (Convex Set)**.** *$\mathbf{S}$ is a convex set if and only if for any $\boldsymbol{x} \in \mathbf{S}$ and $\boldsymbol{y} \in \mathbf{S}$,*

$$\lambda \boldsymbol{x} + (1 - \lambda)\boldsymbol{y} \in \mathbf{S}, \quad \forall \lambda \in [0, 1] \tag{3.4.9}$$

Figure 3.8 shows some examples of convex and not convex sets.

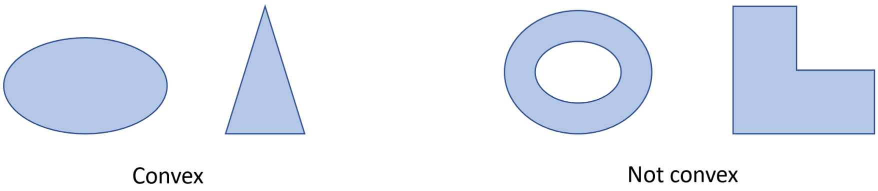


FIGURE 3.8: Examples of convex and not convex sets

Lemma 3.4.4. *According to the definition, the real space $\mathbb{R}^{n\times m}(n, m \geqslant 1)$ is a convex set.*

Definition 3.4.5 (Convex Function)**.** *Suppose* $\mathbf{S}$ *is a convex set. The function* $f : \mathbf{S} \to \mathbb{R}$ *is a convex function if and only if for any* $\boldsymbol{x} \in \mathbf{S}$ *and* $\boldsymbol{y} \in \mathbf{S}$*,*

$$f(\lambda\boldsymbol{x} + (1-\lambda)\boldsymbol{y}) \leqslant \lambda f(\boldsymbol{x}) + (1-\lambda)f(\boldsymbol{y}) \qquad \lambda \in [0,1] \tag{3.4.10}$$

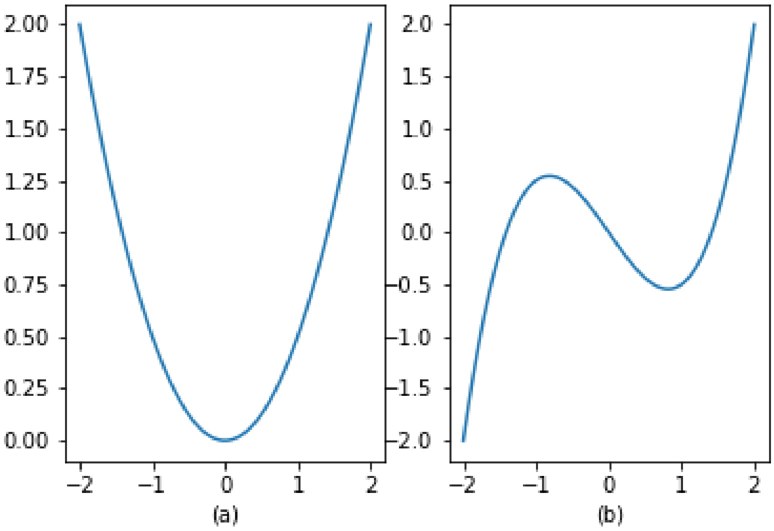


Figure 3.9: Examples of convex (a) and not convex (b) functions.

Example 3.4.6. *The quadratic function*

$$f(x) = x^2$$

is a convex function.

Proof. For any $x, y \in \mathbb{R}$, we have

$$f(\lambda x + (1-\lambda)y) = [\lambda x + (1-\lambda)y]^2$$

Then

$$\begin{aligned} &\lambda f(x) + (1-\lambda)f(y) - f(\lambda x + (1-\lambda)y) \\ =&\lambda x^2 + (1-\lambda)y^2 - [\lambda x + (1-\lambda)y]^2 \\ =&\lambda(1-\lambda)x^2 + \lambda(1-\lambda)y^2 - 2\lambda(1-\lambda)xy \\ =&\lambda(1-\lambda)(x-y)^2 \geqslant 0 \end{aligned}$$

So we have

$$f(\lambda x + (1-\lambda)y) \leqslant \lambda f(x) + (1-\lambda)f(y)$$

for any $\lambda \in [0,1]$. Thus f is a convex function. □

Lemma 3.4.7. *Any affine function*

$$f(x) = ax + b, \quad a, b \in \mathbb{R}$$

is a convex function.

Lemma 3.4.8. *The hinge loss*

$$f(u,y) = max\{0, 1 - uy\}, \qquad y \in \{-1, 1\}$$

is a convex function.

Lemma 3.4.9. *Suppose $f_1, f_2, \ldots, f_n$ are a series of convex functions. Then the positive linear combination*

$$f(x) = \sum_{i=1}^{n} \lambda_i f_i(x) \quad \lambda_i \geqslant 0$$

is a convex function.

Convexity of the Objective Function

Based on all the lemmas and theorems in the above section, I now prove the objective function (3.4.7) in this research is convex.

Theorem 3.4.10. *The objective function*

$$\begin{aligned} f(\mathbf{W}) = & \sum_{i=1}^{n}\sum_{j=1}^{m} b_{ij} \max\{0, 1 - y_{ij}\boldsymbol{x}_i^\top \boldsymbol{w}_j\} + \lambda_1 \sum_{j=1}^{m} \boldsymbol{w}_j^\top \mathbf{D} \boldsymbol{w}_j \\ & + \lambda_2 \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=j+1}^{m} I(\hat{y}_{ij} < \hat{y}_{ik})(\boldsymbol{x}_i^\top \boldsymbol{w}_k - \boldsymbol{x}_i^\top \boldsymbol{w}_j) \end{aligned}$$

is convex.

Proof. According to Lemma 3.4.4, the real space $\mathbb{R}^{(d+1)\times m}$ is convex, hence $f(\mathbf{W})$ is defined on a convex set. Now let

$$f(\mathbf{W}) = f_1(\mathbf{W}) + \lambda_1 f_2(\mathbf{W}) + \lambda_2 f_3(\mathbf{W}) \qquad \lambda_1, \lambda_2 > 0$$

where

$$\begin{aligned} f_1(\mathbf{W}) &= \sum_{i=1}^{n}\sum_{j=1}^{m} b_{ij} \max\{0, 1 - y_{ij}\boldsymbol{x}_i^\top \boldsymbol{w}_j\} \\ f_2(\mathbf{W}) &= \sum_{j=1}^{m} \boldsymbol{w}_j^\top \mathbf{D} \boldsymbol{w}_j \\ f_3(\mathbf{W}) &= \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=j+1}^{m} I(\hat{y}_{ij} < \hat{y}_{ik})(\boldsymbol{x}_i^\top \boldsymbol{w}_k - \boldsymbol{x}_i^\top \boldsymbol{w}_j) \end{aligned}$$

If $f_1(\mathbf{W}), f_2(\mathbf{W})$ and $f_3(\mathbf{W})$ are all convex function, then according to Lemma 3.4.9, $f(\mathbf{W})$ is a convex function.

Observe that $f_1(\mathbf{W})$ is a positive linear combination of hinge losses, since $b_{ij} =$ 0 or 1. According to Lemma 3.4.8, hinge loss is convex, thus $f_1(\mathbf{W})$ is a convex function.

Observe that $f_2(\mathbf{W})$ is a linear combination of quadratic functions with coefficient $\mathbf{D}$. Recall that

$$\mathbf{D} = \begin{pmatrix} \frac{1}{\|\boldsymbol{w}^1\|_2} & & \\ & \ddots & \\ & & \frac{1}{\|\boldsymbol{w}^n\|_2} \end{pmatrix}$$

and according to definition 3.8, it has:

$$\|\boldsymbol{w}^j\|_2 \geqslant 0 \qquad j = 1, \ldots m$$

thus every elements of $\mathbf{D}$ is greater than 0. Therefore, $f_2(\mathbf{W})$ is also a positive linear combination of convex functions, hence it is convex.

Rewrite $f_3(\mathbf{W})$ as

$$f_3(\mathbf{W}) = \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=j+1}^{m} I_{\mathbf{S}}(\mathbf{W})$$

where

$$I_{\mathbf{S}}(\mathbf{W}) = \begin{cases} 0 & \text{if } \mathbf{XW} \in \mathbf{S} \\ \boldsymbol{x}_i^\top \boldsymbol{w}_k - \boldsymbol{x}_i^\top \boldsymbol{w}_j & \text{if } \mathbf{XW} \notin \mathbf{S} \end{cases}$$

where $\mathbf{S}$ is a set of matrices such that every matrix satisfies non-recurring condition. Now we prove $I_{\mathbf{S}}(\mathbf{W})$ is convex.

Suppose $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{(d+1)\times m}$ are two arbitrary matrices. Now we discuss all the situations:

1. If $\mathbf{XA} \in \mathbf{S}$ and $\mathbf{XB} \in \mathbf{S}$, then

$$\mathbf{X}(\lambda\mathbf{A} + (1-\lambda)\mathbf{B}) = \lambda\mathbf{XA} + (1-\lambda)\mathbf{XB} \in \mathbf{S}$$

   since $\mathbf{S}$ is a convex set. Therefore,

$$\begin{aligned} \lambda I_{\mathbf{S}}(\mathbf{A}) &= 0 \\ (1-\lambda)I_{\mathbf{S}}(\mathbf{B}) &= 0 \\ I_{\mathbf{S}}(\lambda\mathbf{A} + (1-\lambda)\mathbf{B}) &= 0 \end{aligned}$$

   This satisfies:

$$I_{\mathbf{S}}(\lambda\mathbf{A} + (1-\lambda)\mathbf{B}) = \lambda I_{\mathbf{S}}(\mathbf{A}) + (1-\lambda)I_{\mathbf{S}}(\mathbf{B})$$

2. If $\mathbf{XA} \notin \mathbf{S}$, $\mathbf{XB} \notin \mathbf{S}$ and $\mathbf{X}(\lambda\mathbf{A} + (1-\lambda)\mathbf{B}) \in \mathbf{S}$, then:

$$\begin{aligned} \lambda I_{\mathbf{S}}(\mathbf{A}) &= \lambda(\boldsymbol{x}_i^\top \boldsymbol{a}_k - \boldsymbol{x}_i^\top \boldsymbol{a}_j) \\ (1-\lambda)I_{\mathbf{S}}(\mathbf{B}) &= (1-\lambda)(\boldsymbol{x}_i^\top \boldsymbol{b}_k - \boldsymbol{x}_i^\top \boldsymbol{b}_j) \\ I_{\mathbf{S}}(\lambda\mathbf{A} + (1-\lambda)\mathbf{B}) &= 0 \end{aligned}$$

   This satisfies the inequality:

$$I_{\mathbf{S}}(\lambda\mathbf{A} + (1-\lambda)\mathbf{B}) \leqslant \lambda I_{\mathbf{S}}(\mathbf{A}) + (1-\lambda)I_{\mathbf{S}}(\mathbf{B})$$

3. If $\mathbf{XA} \notin \mathbf{S}$, $\mathbf{XB} \notin \mathbf{S}$ and $\mathbf{X}(\lambda\mathbf{A} + (1-\lambda)\mathbf{B}) \notin \mathbf{S}$, then:

$$\begin{aligned} \lambda I_{\mathbf{S}}(\mathbf{A}) &= \lambda(\boldsymbol{x}_i^\top \boldsymbol{a}_k - \boldsymbol{x}_i^\top \boldsymbol{a}_j) \\ (1-\lambda)I_{\mathbf{S}}(\mathbf{B}) &= (1-\lambda)(\boldsymbol{x}_i^\top \boldsymbol{b}_k - \boldsymbol{x}_i^\top \boldsymbol{b}_j) \\ I_{\mathbf{S}}(\lambda\mathbf{A} + (1-\lambda)\mathbf{B}) &= \lambda\boldsymbol{x}_i^\top(\boldsymbol{a}_k - \boldsymbol{a}_j) + (1-\lambda)\boldsymbol{x}_i^\top(\boldsymbol{b}_k - \boldsymbol{b}_j) \end{aligned}$$

   This satisfies:

$$I_{\mathbf{S}}(\lambda\mathbf{A} + (1-\lambda)\mathbf{B}) = \lambda I_{\mathbf{S}}(\mathbf{A}) + (1-\lambda)I_{\mathbf{S}}(\mathbf{B})$$

4. If $\mathbf{XA} \in \mathbf{S}$, $\mathbf{XB} \notin \mathbf{S}$ and $\mathbf{X}(\lambda\mathbf{A} + (1-\lambda)\mathbf{B}) \notin \mathbf{S}$, then:

$$\begin{aligned} \lambda I_{\mathbf{S}}(\mathbf{A}) &= 0 \\ (1-\lambda)I_{\mathbf{S}}(\mathbf{B}) &= (1-\lambda)(\boldsymbol{x}_i^\top \boldsymbol{b}_k - \boldsymbol{x}_i^\top \boldsymbol{b}_j) \\ I_{\mathbf{S}}(\lambda\mathbf{A} + (1-\lambda)\mathbf{B}) &= \lambda\boldsymbol{x}_i^\top(\boldsymbol{a}_k - \boldsymbol{a}_j) + (1-\lambda)\boldsymbol{x}_i^\top(\boldsymbol{b}_k - \boldsymbol{b}_j) \end{aligned}$$

Then we have

$$I_{\mathbf{S}}(\lambda\mathbf{A} + (1-\lambda)\mathbf{B}) = (1-\lambda)I_{\mathbf{S}}(\mathbf{B}) + \lambda\boldsymbol{x}_i^\top(\boldsymbol{a}_k - \boldsymbol{a}_j)$$

Since $\mathbf{XA} \in \mathbf{S}$, then we have

$$\lambda\boldsymbol{x}_i^\top(\boldsymbol{a}_k - \boldsymbol{a}_j) < 0$$

Then the inequality

$$I_{\mathbf{S}}(\lambda\mathbf{A} + (1-\lambda)\mathbf{B}) \leqslant \lambda I_{\mathbf{S}}(\mathbf{A}) + (1-\lambda)I_{\mathbf{S}}(\mathbf{B})$$

is satisfied.

5. If $\mathbf{XA} \in \mathbf{S}$, $\mathbf{XB} \notin \mathbf{S}$ and $\mathbf{X}(\lambda\mathbf{A} + (1-\lambda)\mathbf{B}) \in \mathbf{S}$, then:

$$\begin{aligned} \lambda I_{\mathbf{S}}(\mathbf{A}) &= 0 \\ (1-\lambda)I_{\mathbf{S}}(\mathbf{B}) &= (1-\lambda)(\boldsymbol{x}_i^\top \boldsymbol{b}_k - \boldsymbol{x}_i^\top \boldsymbol{b}_j) \\ I_{\mathbf{S}}(\lambda\mathbf{A} + (1-\lambda)\mathbf{B}) &= 0 \end{aligned}$$

Then the inequality

$$I_{\mathbf{S}}(\lambda\mathbf{A} + (1-\lambda)\mathbf{B}) \leqslant \lambda I_{\mathbf{S}}(\mathbf{A}) + (1-\lambda)I_{\mathbf{S}}(\mathbf{B})$$

is satisfied.

Combining all the situations above, we can conclude that $I_{\mathbf{S}}(\mathbf{W})$ is convex, thus $f_3(\mathbf{W})$ is a convex function. As a positive linear combination of convex functions, $f(\mathbf{W})$ is also convex. □

Sub-gradient Descent Method

After proving the objective function is a convex function defined on a convex set, the sub-gradient descent method can be applied to solve the optimisation problem (Murphy, 2012). Let

$$\mathbf{W}_{t+1} = \mathbf{W}_t - \eta_t \frac{\partial f(\mathbf{W}_t)}{\partial \mathbf{W}_t}$$

where $\mathbf{W}_t$ is the value of $\mathbf{W}$ at iteration t, and η_t is the step size at iteration t.

The gradient of $f(\mathbf{W})$ is:

$$\frac{\partial f(\mathbf{W})}{\partial \mathbf{W}} = \frac{\partial f_1(\mathbf{W})}{\partial \mathbf{W}} + \lambda_1 \frac{\partial f_2(\mathbf{W})}{\partial \mathbf{W}} + \lambda_2 \frac{\partial f_3(\mathbf{W})}{\partial \mathbf{W}}$$

$f_1(\mathbf{W})$ is not differentiable since hinge loss is not differentiable, so its sub-gradient will be used.

$$\frac{\partial f_1(\mathbf{W})}{\partial \mathbf{W}} = \begin{cases} -y_{ij}\boldsymbol{x}_i & \text{if } y_{ij}\hat{y}_{ij} < 1 \text{ and } b_{ij} \neq 0 \\ 0 & \text{if } y_{ij}\hat{y}_{ij} \geqslant 1 \text{ or } b_{ij} = 0 \end{cases} \tag{3.4.11}$$

for every $\boldsymbol{w}_j, \quad j = 1, \ldots, m$.

$f_2(\mathbf{W})$ is differentiable, so

$$\frac{\partial f_2(\mathbf{W})}{\partial \mathbf{W}} = 2\mathbf{D}\mathbf{W} \tag{3.4.12}$$

Similarly,

$$\frac{\partial f_3(\mathbf{W})}{\partial \mathbf{W}} = \begin{cases} -\sum_{k=j+1}^{m} \boldsymbol{x}_i & \text{if } \hat{y}_{ik} > \hat{y}_{ij} \\ 0 & \text{if } \hat{y}_{ik} \leqslant \hat{y}_{ij} \end{cases} \tag{3.4.13}$$

for every $\boldsymbol{w}_j, \quad j = 1, \ldots, m$.

After calculating the gradients and sub-gradient, the algorithm of this research is as follows:

Algorithm 1: SVM-based MTL for Survival Analysis

```
input  : $\mathbf{X} \in \mathbb{R}^{n\times(d+1)}, \mathbf{Y} \in \mathbb{R}^{n\times m}, \mathbf{W} \in \mathbb{R}^{(d+1)\times m}, \mathbf{B} \in \mathbb{R}^{n\times m}, \lambda_1, \lambda_2$
output: $\hat{\mathbf{W}}$
Set $\mathbf{W}_0 = \mathbf{0}$;
$t = 1$;
while not convergence do
    $\eta_t = \frac{1}{\lambda_1 t}$;
    Calculating $\mathbf{D}_t$ with respect to $\mathbf{W}_t$;
    Calculating $\frac{\partial f(\mathbf{W}_t)}{\partial \mathbf{W}_t}$ according to (3.4.11), (3.4.12), (3.4.13);
    $\mathbf{W}_{t+1} = \mathbf{W}_t - \eta_t \frac{\partial f(\mathbf{W}_t)}{\partial \mathbf{W}_t}$;
    $t = t + 1$;
end
```

3.5 Data Description

This section demonstrates the real and simulated data used. The real data are used to test the prediction accuracy of the proposed model in this research, while the simulated data are used to study the prediction accuracy, feature selection ability of the model, as well as the change of ICER.

3.5.1 Real Data

Four high-dimensional open-source gene expression datasets are used in this study, all the datasets can be found here `http://user.it.uu.se/~liuya610/download.html`.

NSBCD: The Norway/Stanford breast cancer data from Sorlie et al. (2003) contains gene expression measurements of 115 women with breast cancer, and 77 patients are censored. The response variable is survival time, and the maximum survival time is 188 months. The dataset contains 549 features.

VDV: Van de Vijver's Microarray Breast Cancer data (van't Veer et al., 2002) contains gene expression measurements of 78 women with breast cancer, and 44 patients are censored. The response variable is survival time, and the maximum survival time is 13 years. The dataset contains 4707 features.

DBCD: The Dutch Breast Cancer Data from van Houwelingen et al. (2006) contains gene expression measurements of 295 women with breast cancer, and 216 patients are censored. The response variable is survival time, and the maximum survival time is 18 years. The dataset contains 4919 features.

LUNG: Gene-expression profiles of lung adenocarcinoma from Beer et al. (2002) is a dataset containing observations of 86 early-stage lung adenocarcinoma patients. The response variable is survival time, and the maximum survival time is 110 months. The dataset contains 7129 features and 62 patients are censored.

Datasets	Patients	Censored	Features
NSBCD	115	77	549
VDV	78	44	4707
DBCD	295	216	4919
LUNG	86	62	7129

Table 3.1: Summaries of all real datasets.

3.5.2 Simulated Data

The simulated datasets are designed to demonstrate the feature selection as well as the prediction ability of the new model in this research. The simulated datasets are generated by the following distribution:

$$\mathbf{T} \sim \exp(\alpha(\boldsymbol{v}^\top \mathbf{X})^2 + \beta \boldsymbol{u}^\top \mathbf{X})$$

where $\mathbf{T}$ is the survival time, $\mathbf{X}$ is the feature space, and $exp()$ is the exponential distribution. $\boldsymbol{v}$ and $\boldsymbol{u}$ are two vectors, and α and β are two arbitrary positive real numbers.

Suppose there are 100 features, and these features follow a multivariate normal distribution:

$$\mathbf{X} \sim N(\boldsymbol{\mu}, \mathbf{I})$$

where each elements of $\boldsymbol{\mu}$ follows the uniform distribution $\mathrm{U}[0, 1]$.

Vectors $\boldsymbol{v}$ and $\boldsymbol{u}$ control how many features determine the survival time. In this research, the first 20 features are supposed to decide the survival time, so

$$\begin{aligned} \boldsymbol{u}^\top &= (u_1, u_2, \ldots, u_{20}, 0, \ldots, 0) \\ \boldsymbol{v}^\top &= (v_1, v_2, \ldots, v_{20}, 0, \ldots, 0) \end{aligned}$$

To imitate censoring, around 20% samples are randomly assigned a survival time which is shorter than the real survival time.

3.6 Comparison Methods

This section introduces the related popular state-of-the-art methods for survival analysis, including two Cox models as the benchmark statistical methods, random survival forest and gradient boosted model as the benchmark ML methods, and Multi-Task Learning model for Survival Analysis (MTLSA) proposed in Li et al. (2016).

3.6.1 Cox Regression

In survival analysis, when the survival distribution is continuous and differentiable, the **hazard function** $\lambda(t)$ is defined as (Klein and Klein, 2013):

$$\lambda(t)dt = \mathbb{P}(T < t + dt | T \geqslant t)$$

where t is the variable representing time and T is the event time. Intuitively, if the patient does not experience the event at time t, then the hazard function is the probability that the patient will experience the event between time t and $t + dt$. Cox regression (Cox, 1972) assumes that the hazard function of all patients satisfies the proportional hazards assumption:

$$\lambda(t|\mathbf{X}) = \lambda_0(t) \exp(\mathbf{X}^\top \boldsymbol{\beta})$$

where $\lambda_0(t)$ is the baseline hazard function, $\mathbf{X}$ is the covariates matrix, and $\boldsymbol{\beta}$ is the coefficients vector. $\hat{\boldsymbol{\beta}}$ is estimated by maximising the partial likelihood function $\ell(\boldsymbol{\beta})$.

3.6.2 Cox-LASSO

When the dimension of the covariates $\mathbf{X}$ is high, some feature selection methods can be applied to Cox regression to reduce the dimension of the covariates space (Klein and Klein, 2013). The Cox-LASSO puts the ℓ_1 constraints on $\boldsymbol{\beta}$, and $\boldsymbol{\beta}$ is estimated by minimising

$$-\ell(\boldsymbol{\beta}) + \lambda\|\boldsymbol{\beta}\|_1$$

where $\lambda > 0$ is the parameter of the ℓ_1 penalty.

3.6.3 Cox-EN

The Elastic Net penalty is to solve the problem that Cox-LASSO can only identify one feature when two highly correlated features are both critical to the survival function (Klein and Klein, 2013). EN penalty is a mixture of ℓ_1 and ℓ_2 norm, and the $\hat{\boldsymbol{\beta}}$ in the Cox-EN is estimating by minimising

$$-\ell(\boldsymbol{\beta}) + \lambda_1\|\boldsymbol{\beta}\|_1 + \lambda_2\|\boldsymbol{\beta}\|_2^2$$

3.6.4 Gradient Boosted Model

Gradient boosted model is a branch of ML model which combines a series of base learners to improve the overall model (Pölsterl, 2020). Base learners are usually simple models which are not very powerful when used alone. The gradient boosted model has the form:

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \beta_i g(\boldsymbol{x}; \theta_i)$$

where $g(\boldsymbol{x}; \theta_i)$ is a base learner and β_i is the weight of the base learner. The base learner chosen in this research is the accelerated failure time model, and thus will be referred as accelerated failure time model in this research.

3.6.5 Random Survival Forest

Random Survival Forest (Ishwaran et al., 2008) is a benchmark ML method for survival analysis. It is an alternative of the standard Random Forest and is specially designed for survival analysis.

3.6.6 Multi-task Learning

The detailed description of MTL and its variants for survival analysis can be found in Section 2.4. The model MTLSA from Li et al. (2016) will be compared with the new model proposed in this thesis.

3.7 Evaluation Metric

The concordance index (C-index) is used in this research to compare the prediction accuracy of survival analysis, rather than the absolute error (Harrell et al., 1982). The absolute error is defined as $|\hat{T} - T|$, where $\hat{T}$ is the predicted survival time and T is the real survival time. C-Index is similar to the area under the receiver operating characteristic (ROC) curve but takes into account censored data.

Definition 3.7.1 (Concordance Index)**.** *The C-index is calculated by:*

$$\textit{C-index} = \mathbb{P}(\hat{T_1} > \hat{T_2} | T_1 > T_2)$$

where T_1, T_2 are the survival time or censored time of an arbitrary pair of patients, and $\hat{T_1}, \hat{T_2}$ are the predicted survival time of them. Two kinds of patient pairs satisfy the rules. First, both patients are not censored. Second, one of the patient is censored, and the censored time is larger than then survival time of another patient.

Compared with the absolute error, C-index has two advantages. First, C-index assumes that patients with smaller survival time have higher risks, thus combines two aspects of survival problems: survival time and relative risk. Second, C-index takes into account censored data.

4

Results and Analysis

This chapter is designed as follows. Section 4.1 introduces the model validation technique cross-validation. Section 4.2 lists the results of the real datasets. Section 4.3 gives the results of the simulated datasets.

4.1 Cross-Validation

When training the model, the hyperparameters λ_1 and λ_2 in the proposed new model should be chosen so that the model has the highest prediction accuracy. After choosing suitable hyperparameters, model performance should be evaluated in a generalised manner and biases should be avoided. Cross-validation (CV) is a model validation technique in statistics designed to evaluate the performance of a model when tested on independent datasets. Independent here means that the test set is independent of the training set. CV can also be used to select hyperparameters. In this research, a branch of CV called Nested CV is applied to evaluate model performance and select appropriate hyperparameters to avoid data 'dredging' and overfitting. The first objective is called model evaluation and the second objective is called model selection. CV and Nested CV will be introduced below.

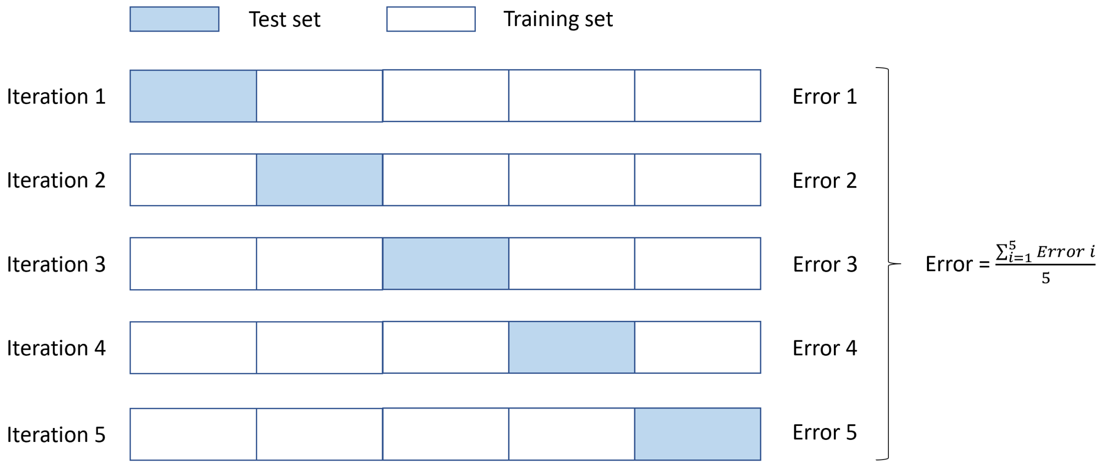


Figure 4.1: 5-Fold Cross-Validation

The basic idea of CV is to split the dataset into k folds, take k-1 folds as the training set and the other fold as the test set, and repeat the process k times until all k folds are used as the test set. The overall error of the model is the arithmetic mean of the errors for k iterations. Figure 4.1 illustrates the structure of the 5-fold CV.

In K-fold CV, the training and test sets are separated, thus avoiding overfitting. In addition, all folds are used as the test sets in different iterations. This reduces bias in model evaluation, as it reduced the risk of accidentally finding a test set with good performance. CV also reduces the risk of data 'dredging' by repeating the model evaluation process and taking the average error. In short, applying CV in model evaluation can reduce the risk of overfitting and data 'dredging'.

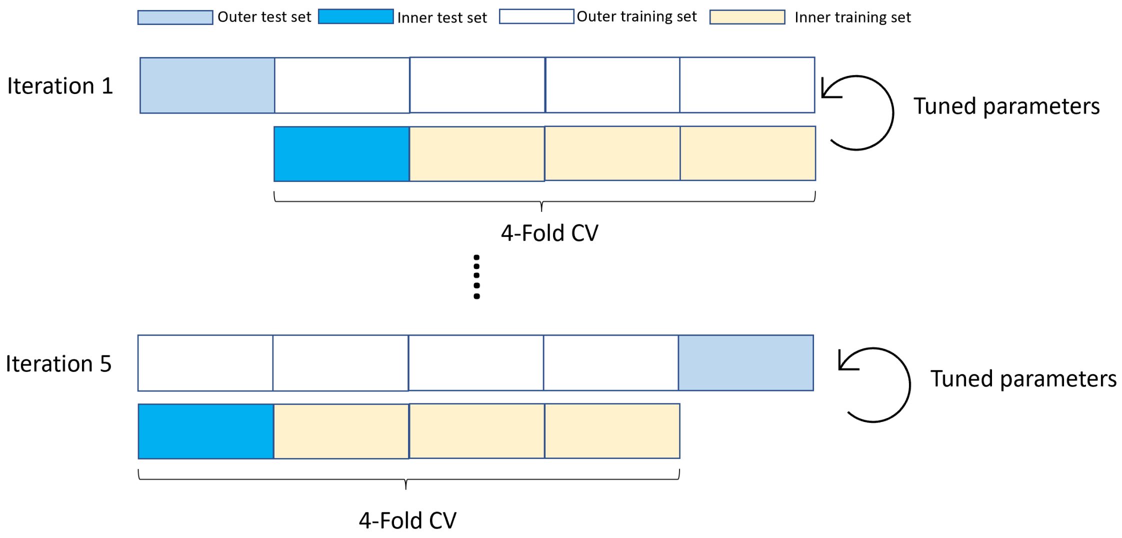


Figure 4.2: The Nested Cross-Validation

However, model evaluation and model selection cannot be processed simultaneously in one K-fold CV, as this may lead to 'information leakage'. 'Information leakage' occurs because of manual selection of hyperparameters when error is at its lowest or prediction accuracy is at its highest. And in the step of choosing hyperparameters, the test set is no longer independent of the training set. Thus, the test set leaks information if it is used again for model evaluation. Nested CVs (Krstajic et al., 2014) are designed to avoid 'information leakage' and reduce bias. The structure of a Nested 3 × 4 Folds CV is shown in Figure 4.2.

As shown in Figure 4.2, each iteration of the Nested CV is a K-fold CV. For each iteration, the inner training and test sets are used for choosing hyperparameters. The hyperparameters are chosen such that the error reaches lowest or the prediction accuracy reaches the highest at the inner test set. After choosing the hyperparameters, the model is run on the the outer test set for model evaluation. Therefore, the model selection and model evaluation are separated and independent. The overall error is the average of the errors for all the iterations. In a word, the inner training and test sets are used to choose hyperparameter, and the outer test sets are used to estimate performance.

The following algorithm summaries the Nested $K \times I$ Folds CV (Krstajic et al., 2014).

Algorithm 2: Nested $K \times I$ Folds Cross-Validation

```
Randomly divide the dataset into $K$ folds;
for $i = 1, \ldots, K$ do
    Let fold $i$ be the outer test set;
    Randomly divide the other $K - 1$ folds into $I$ folds;
    for $j = 1, \ldots, I$ do
        Let fold $j$ be the inner test set and the remaining be the training sets;
        Train all hyperparameters on the training sets and records the error on
          test set;
        Choose the best hyperparameters when the error reaches the lowest;
    end
    Train a model with the best hyperparameters on the $K - 1$ folds and
      record the error on the test fold $i$;
end
Return the average error;
```

4.2 Real Data

This section introduces the results of real datasets.

4.2.1 NSBCD

The dataset **NSBCD** contains gene expression measurements of 115 women with breast cancer, and 77 patients are censored. The censorship percentage is 70.0%. The response variable is survival time, and the maximum survival time is 188 months. The dataset contains 549 features. According to the sample size, a Nested 4×3 Folds CV is reasonable to be applied to the **NSBCD** dataset. The detailed procedure is as follows.

Step 1: As shown in Table 2, split the dataset into 4 folds randomly, each fold should contain approximately the same number of censored and uncensored samples.

This process is achieved by using `train_test_split` function from `scikit-learn` (Pedregosa et al., 2011). Similar procedures below use the same technique.

	Fold 1	Fold 2	Fold 3	Fold 4
censored	20	19	19	20
uncensored	9	10	10	9

Table 4.1: Split the dataset **NSBCD** into 4 folds.

Step 2: Take fold 1 as the outer test set, and conduct a 3-Fold CV on the rest folds. λ_1 and λ_2 are chosen from

$$\lambda_1 \in \{0.0001, 0.0002, \ldots, 0.001\}$$
$$\lambda_2 \in \{0.1, 0.2, \ldots, 1\}$$

Train the model with all pairs of λ_1 and λ_2, and select the best λ_1 and λ_2 such that the C-index reaches the highest.

Step 3: Train a model with the selected λ_1 and λ_2 in step 2 on folds 2, 3, and 4, then use fold 1 as test set to calculated the C-index. Then we get the C-index for iteration 1, denoted as C_1.

Step 4: Repeat steps 2 and 3 till all folds are used as outer test sets and calculated the C-indices C_2, C_3 and C_4.

Step 5: The final C-index is calculated as

$$\text{C-index} = \frac{C_1 + C_2 + C_3 + C_4}{4}$$

And the standard deviation of the overall C-index is calculated using C_1, C_2, C_3 and C_4.

The result is as follows:

	Iteration 1	**Iteration 2**	**Iteration 3**	**Iteration 4**
Best λ_1	0.0001	0.0001	0.0001	0.0005
Best λ_2	0.1	0.1	0.1	0.2
C_i	0.721	0.683	0.756	0.688
C-index	**0.712**			
SD	0.029			

Table 4.2: The Results of the Nested 4×3 Folds CV for dataset **NSBCD**.

According to Table 3, the overall C-index of the new model on dataset **NSBCD** is 0.712, and the standard deviation is 0.029.

4.2.2 VDV

The **VDV** dataset contains gene expression measurements of 78 women with breast cancer, and 44 of them are censored. The censorship percentage is 56.4%. The dataset contains 4707 features and the maximum survival time is 161 months. According to the sample size, a Nested 4×3 Folds CV is applied on the **VDV** dataset.

Step 1: Split the dataset into 4 folds as shown in Table 4, each fold should contain approximately the same number of censored and uncensored data.

	Fold 1	Fold 2	Fold 3	Fold 4
censored	11	11	11	11
uncensored	9	9	8	8

Table 4.3: Split the dataset **VDV** into 4 folds.

Step 2: Take fold 1 as the outer test set, and conduct a 3-Fold CV on the rest folds. λ_1 and λ_2 are chosen from

$$\lambda_1 \in \{0.0001, 0.0002, \ldots, 0.001\}$$

$$\lambda_2 \in \{0.1, 0.2, \ldots, 2\}$$

Train the model with all pairs of λ_1 and λ_2, and select the best λ_1 and λ_2 such that the C-index reaches its highest point.

Step 3: Train a model with the selected λ_1 and λ_2 on folds 2, 3, and 4, then use fold 1 as the outer test set to calculated the C-index for iteration 1, denoted as C_1.

Step 4: Repeat step 2 and 3 for folds 2, 3, and 4, till all folds are used as the outer test sets, and calculated the C-indices C_2, C_3 and C_4.

Step 5: The final C-index is calculated as

$$\text{C-index} = \frac{C_1 + C_2 + C_3 + C_4}{4}$$

And the standard deviation of the overall C-index is calculated using C_1, C_2, C_3 and C_4.

The result is as follows:

	Iteration 1	**Iteration 2**	**Iteration 3**	**Iteration 4**
Best λ_1	0.0001	0.0001	0.0009	0.0005
Best λ_2	1	1	1	1
C_i	0.752	0.660	0.752	0.703
C-index	**0.717**			
SD	0.039			

Table 4.4: The Results of the Nested 4 × 3 Folds CV for dataset **VDV**.

According to Table 5, the overall C-index of the new model on dataset **VDV** is 0.717, and the standard deviation is 0.039.

4.2.3 DBCD

The **DBCD** dataset contains gene expression measurements of 295 women with breast cancer, and 216 of them are censored. The censoring percentage is 73.2%. The dataset contains 4919 features and the maximum survival time is 18 years. According to the sample size, a Nested 5 × 4 Folds CV is applied on the **DBCD** dataset.

Step 1: Split the dataset into 5 folds as shown in Table 6, each fold should contain approximately the same number of censored and uncensored samples.

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
censored	44	43	43	43	43
uncensored	15	16	16	16	16

Table 4.5: Split the dataset **DBCD** into 5 folds.

Step 2: Take fold 1 as the outer test set, and conduct a 4-Fold CV on the rest folds. λ_1 and λ_2 are chosen from

$$\lambda_1 \in \{0.01, 0.02, \ldots, 0.1\}$$
$$\lambda_2 \in \{0.1, 0.2, \ldots, 1, 2\}$$

Train the model with all pairs of λ_1 and λ_2, and select the best λ_1 and λ_2 such that the C-index reaches its highest point.

Step 3: Train a model with the selected λ_1 and λ_2 on folds 2, 3, 4, and 5, then use fold 1 as test set to calculated the C-index for iteration 1, denoted as C_1.

Step 4: Repeat steps 2 and 3 till all folds are used as outer test sets, and calculated the C-indices C_2, C_3, C_4 and C_5.

Step 5: The final C-index is calculated as

$$\text{C-index} = \frac{C_1 + C_2 + C_3 + C_4 + C_5}{5}$$

And the standard deviation of the overall C-index is calculated using C_1, C_2, C_3, C_4 and C_5.

The result is as follows:

	Iteration 1	**Iteration 2**	**Iteration 3**	**Iteration 4**	**Iteration 5**
Best λ_1	0.1	0.1	0.1	0.01	0.01
Best λ_2	1	1	0.5	2	0.1
C_i	0.757	0.739	0.820	0.725	0.765
C-index	**0.761**				
SD	0.033				

Table 4.6: The Results of the Nested 5 × 4 Folds CV for dataset **DBCD**.

According to Table 7, the overall C-index of the new model on dataset **DBCD** is 0.761, and the standard deviation is 0.033.

4.2.4 LUNG

The **LUNG** dataset contains gene expression measurements of 86 patients with early-stage lung adenocarcinoma, and 62 patients are censored. The censorship percentage is 72.1%. The dataset contains 7129 features and the maximum survival time is 100 months. According to the sample size, a Nested 4 × 3 Folds CV is applied on the **LUNG** dataset.

Step 1: Split the dataset into 4 folds as shown in Table 8, each fold should contain approximately the same number of censored and uncensored samples.

	Fold 1	Fold 2	Fold 3	Fold 4
censored	16	16	15	15
uncensored	6	6	6	6

Table 4.7: Split the dataset **LUNG** into 4 folds.

Step 2: Take fold 1 as the outer test set, and conduct a 3-Folds CV on the rest folds.

λ_1 and λ_2 are chosen from

$$\lambda_1 \in \{10^{-5}, 0.0001, \ldots, 0.1\}$$

$$\lambda_2 \in \{10^{-5}, 0.0001, \ldots, 0.1\}$$

Train the model with all pairs of λ_1 and λ_2, and select the best λ_1 and λ_2 such that the C-index reaches its highest point.

Step 3: Train a model with the selected λ_1 and λ_2 on folds 2, 3, and 4, then use fold 1 as test set to calculated the C-index for iteration 1, denoted as C_1.

Step 4: Repeat steps 2 and 3 till all folds are used as outer test sets, and calculated the C-indices C_2, C_3 and C_4.

Step 5: The final C-index is calculated as

$$\text{C-index} = \frac{C_1 + C_2 + C_3 + C_4}{4}$$

And the standard deviation of the overall C-index is calculated using C_1, C_2, C_3 and C_4.

The result is as follows:

	Iteration 1	**Iteration 2**	**Iteration 3**	**Iteration 4**
Best λ_1	10^{-5}	10^{-5}	10^{-5}	10^{-5}
Best λ_2	0.01	0.01	0.01	0.01
C_i	0.635	0.584	0.797	0.640
C-index	**0.664**			
SD	0.080			

TABLE 4.8: The Results of the Nested 4 × 3 Folds CV for dataset **LUNG**.

According to Table 9, the overall C-index of the new model on dataset **LUNG** is 0.664, and the standard deviation is 0.080.

4.2.5 Comparison with Other Methods

Two semi-parametric models and two state-of-the-art ML models for survival analysis are compared with the new model in this research. The Cox-LASSO, Cox-EN, Random

Survival Forest, and Gradient Boosted Model with Accelerated Failure Time model as the base learner are from Python module `scikit-survival` (Pölsterl, 2020), and the C-index function is from the Python module `lifelines` (Davidson-Pilon, 2019). The model MTLSA (Li et al., 2016) is trained on MATLAB.

All the compared models are evaluated using K-Fold CV and the results are as follows. Bold number in a column indicates the highest C-index, and numbers in parentheses are standard deviations.

	NSBCD	**VDV**	**DBCD**	**LUNG**
Cox-LASSO	0.598 (0.058)	0.582 (0.063)	0.670 (0.037)	0.558 (0.098)
Cox-EN	0.603 (0.076)	0.578 (0.064)	0.684 (0.034)	0.558 (0.098)
Accelerated Failure Time	0.624 (0.071)	0.634 (0.066)	0.644 (0.056)	0.573 (0.108)
Random Survival Forests	**0.732** (0.078)	0.705 (0.079)	0.760 (0.040)	0.628 (0.098)
MTLSA	0.682 (0.045)	0.701 (0.033)	0.758 (0.030)	0.633 (0.075)
The new model	0.712 (0.029)	**0.717** (0.039)	**0.761** (0.033)	**0.664** (0.080)

Table 4.9: The C-index of all methods for real datasets. Bold number in a column indicates the highest C-index for the dataset, and numbers in parentheses are standard deviations.

Here is the summary of findings. First, the new model proposed in this research outperformed the other models in 3 out of 4 real datasets. Overall, 3 ML methods performed better than the Cox models. Second, Random Survival Forest has the best performance on dataset **NSBCD**. However, the standard deviation of the Random Survival Forest is relatively high (SD = 0.078). Third, the new model outperformed other methods on the dataset **VDV**, but the improvement was marginal, 1.67% compared with the second best method, Random Survival Forest. However, the standard deviation of my model is significantly smaller than the Random Survival Forest. This shows that my model is more stable. Fourth, the new model, Random Survival Forest and MTLSA have almost the same C-index on the dataset **DBCD**, my model is slightly higher than the other two methods. Also, the standard deviations are at the same level. Finally, the new model outperformed the other methods in the dataset **LUNG** by 4.90% compared with the second highest method. In addition, the standard deviations are relatively high for all methods because the dataset **LUNG** has a small sample size (N = 86) and a large number of covariates (N = 7129).

4.3 Simulated Data

Datasets with different sample sizes (N = 200, 500, and 1000) are generated as described in section 3.5.2. The coefficients are set as:

$$\begin{aligned}
\boldsymbol{v}^\top &= (1, 2, \ldots, 20, 0, \ldots, 0) \\
\boldsymbol{u}^\top &= (1, 1, \ldots, 1, 0, \ldots 0) \\
\alpha &= 0.1 \\
\beta &= 1 \\
\mu_i &\sim \mathrm{U}[0, 1] \text{ for } i = 1, \ldots, 20 \\
\mu_i &= 0 \text{ for } i = 21, \ldots, 100
\end{aligned}$$

and the covariates follow the distribution:

$$\mathbf{X} = (x_1, x_2, \ldots, x_{100}) \sim N(\boldsymbol{\mu}, \mathbf{I})$$

and the survival time follows the distribution:

$$\mathbf{T} \sim \exp(\alpha(\boldsymbol{v}^\top \mathbf{X})^2 + \beta \boldsymbol{u}^\top \mathbf{X})$$

4.3.1 C-index Comparison

For each dataset, the models are trained in the same way as section 4.2, and the results are as follows.

	N = 200	N = 500	N = 1000
Cox-LASSO	0.555 (0.037)	0.572 (0.046)	0.651 (0.048)
Cox-EN	0.555 (0.036)	0.572 (0.046)	0.651 (0.048)
Accelerated Failure Time	0.599 (0.062)	0.598 (0.047)	0.700 (0.023)
Random Survival Forests	0.581 (0.018)	0.624 (0.039)	0.682 (0.023)
MTLSA	0.613 (0.034)	0.664 (0.087)	0.699 (0.013)
The new model	**0.665** (0.065)	**0.695**(0.023)	**0.723** (0.020)

Table 4.10: The C-index of all methods for simulated datasets. Bold number in a column indicates the highest C-index for the dataset, and numbers in parentheses are standard deviations.

Here is the summary of findings. First, the new model outperformed others on all simulated datasets. Overall, the three ML models outperform the Cox model. Second, the improvements of the new model proposed in this research compared with the second best model are mediocre on simulated datasets, 7.8%, 4.7%, and 4.7%, respectively. Moreover, all models performed better when the sample size increased.

4.3.2 Feature Selection Comparison

In subgroup identification, it is important to select the correct predictive or prognostic biomarkers from a relatively large set of candidate biomarkers. In the simulated datasets, 20 out of 100 biomarkers are related to the survival time, and the other 80 are not.

As described in the Methodology, the new model proposed in this research forces the weight matrix to be row-sparse, and important features have higher coefficients. The process of feature selection is embedded in the training process, so no additional training and analysis is required. The same is true for Cox models, features with higher coefficients are more important.

The feature selection accuracy is defined as:

$$F = \frac{\text{The number of correct biomarkers being selected}}{\text{The number of correct biomarkers}}$$

For example, if 18 of the 20 biomarkers associated with the response variable are selected, the feature selection accuracy is 90%.

Now compare the feature selection ability of my model with the Cox model. The feature selection process is as follows:

1. During the K-Fold CV, at the end of each iteration, calculate the row norms $\|\boldsymbol{w}^1\|_2, \|\boldsymbol{w}^2\|_2, \ldots, \|\boldsymbol{w}^{100}\|_2$ of the weight matrix $\mathbf{W}$.

2. Select biomarkers $\boldsymbol{w}^i$ with higher norms.

3. calculate the feature selection accuracy F_i for each iteration.

4. The final feature selection accuracy F is the average of F_i. The standard deviations are calculated using F_i.

	N = 200	N = 500	N = 1000
Cox-LASSO	0.490 (0.066)	0.615 (0.032)	0.66 (0.037)
Cox-EN	0.490 (0.066)	0.615 (0.032)	0.66 (0.037)
The new model	**0.75** (0.032)	**0.920** (0.020)	**0.870** (0.024)

Table 4.11: The feature selection accuracy of the simulated datasets.

Below is the summary of findings. First, the new model outperformed the Cox model in feature selection across all simulated datasets. Second, the improvements of the feature selection accuracy of the new model compared with the Cox models are significant, 53%, 50%, and 32% respectively.

4.4 Subgroup Identification of Simulated Data

In this section, the subgroup identification ability of the new model will be compared with the Cox-LASSO, Gradient Boosted Model with Accelerated Failure Time model as base learner, and Random Survival Forest using simulated dataset.

Here are the assumptions of the simulated study. First, suppose there is a treatment A, and the cost of A/patient is \$11000, independent of survival time. Second, the survival time of patients under treatment A is the simulated dataset with the same distribution and coefficients as before, and the sample size is $N = 1000$. Third, 40% of patients are identified as benefiting from the treatment. Fourth, the baseline comparison is that all patients receive treatment A.

Now randomly divide the dataset into training set with sample size $N = 900$, and test set with sample size $N = 100$.

The process of identifying patients with enhanced treatment effect is as follows:

1. Train each model on the training set, and predict the survival time of the test set.

2. Select patients with predicted survival time in the top 40 centile among all patients in the test set as the patient subgroup 'Predict to Benefit'.

3. Select 40% of patients in the test set with shorter predictive survival time as the patient subgroup 'Predict to Not Benefit'.

The Kaplan-Meier curve of the real survival/censor time of the selected subgroups and the whole test set is as follows:

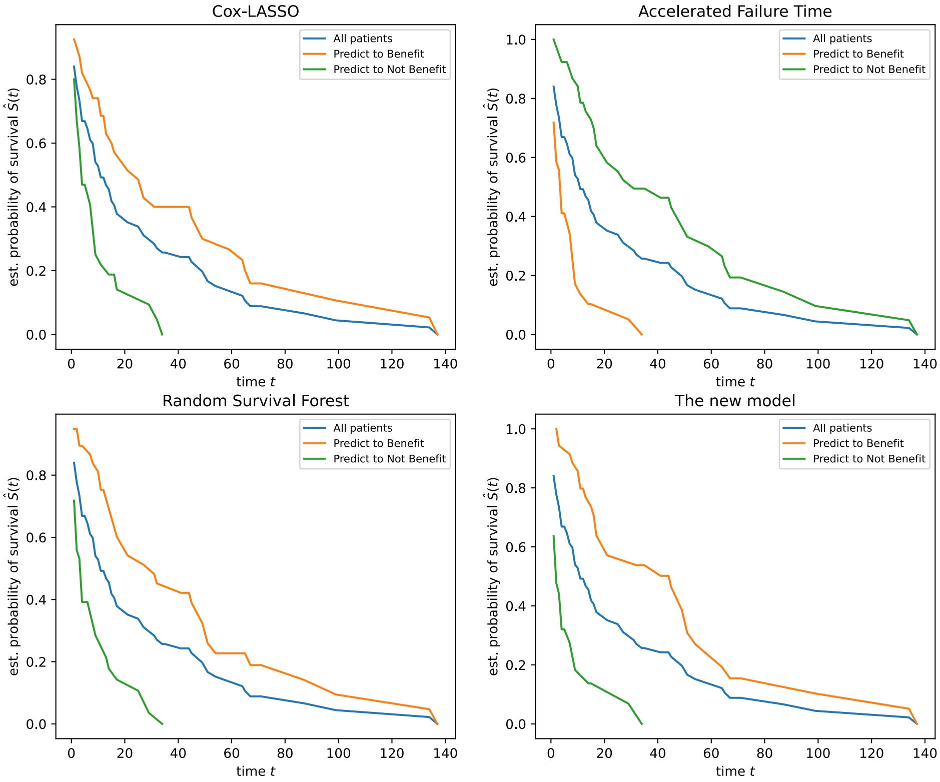


FIGURE 4.3: The Kaplan-Meier Curves of the Selected Patient Subgroups

The estimated median survival time based on Kaplan-Meier estimates for the entire test set and selected patient subgroups is as follows:

	Overall	Predict to Benefit	Predict to Not Benefit
Cox-LASSO	11	25	4
Cox-EN	11	25	4
Accelerated Failure Time	11	31	4
Random Survival Forest	11	31	4
The new model	11	45	2

TABLE 4.12: The Median Survival Time (Months) of the Selected Patient Subgroups

Based on the estimated median survival time of each method, the corresponding ICERs are calculated using the formula:

$$\text{ICER} = \frac{C_A - C_B}{E_A - E_B}$$

The effectiveness of A is evaluated by the Quality Adjusted Life Year (QALY). Suppose the cost of A is \$11000 per patient, independent of the survival time. Moreover, suppose the comparison treatment option B cost \$10000 per patient, and the median survival time under treatment B is 11 months.The results are as follows:

	Cox-LASSO	Cox-EN	AFT	RSF	The new model
Predict to Benefit	\$857	\$857	\$600	\$600	\$353

Table 4.13: The ICERs (per QALY) of ALL Patient Subgroups

Below is the summary of findings. In this simulated study, the new model proposed in this research can select the patient subgroup with the highest median survival time compared with other methods. Moreover, the ICER of the patient subgroup selected by the new model is the lowest. According to the definition of the ICER, it can be conclude that the treatment A is more cost-effective on the patient subgroup selected by the new model.

5 Discussion

5.1 Summary of Findings

In this research, a new SVM-based MTL model for survival analysis was established. The new model was compared with 5 models, including two benchmark statistical models: Cox-LASSO and Cox-EN, two benchmark ML models: Random Survival Forest and Gradient Boosted Model with Accelerated Failure Time model as the base learner, and one MTL model for survival analysis: MTLSA. Three important features were compared, including prediction accuracy of survival time, feature selection accuracy, and changes in ICER in the selected patient subgroup. The prediction accuracy of survival time/risk is evaluated by the C-index, and the feature selection accuracy is assessed by the average probability of selecting the correct biomarker. Two types of datasets were used in this research – the high dimension gene expression datasets and the simulated datasets with different sample size. To avoid data 'dredging' and overfitting, the models were trained using Nested Cross-Validation.

The prediction accuracy comparison using the real datasets shows that my model outperformed the other 5 models in 3 of the 4 datasets when evaluated by the C-index. Also, the comparison of standard deviations of the C-index shows that my model is stable, since the standard deviations are relatively small. However, the overall improvement is marginal at less than 3% in two datasets and around 5% in one dataset.

Three datasets with sample sizes $N = 200, 500$, and 1000 were simulated, each with a censorship percentage of 20%. The prediction accuracy comparison shows that my model outperforms the other 5 models when evaluated by the C-index across all simulated datasets. The C-index improvements for the three datasets are 7.8%, 4.7%, and 4.7%, respectively.

The feature selection accuracy of my model was compared with the Cox models on the simulated datasets. The feature selection accuracy is defined as the average probability of selecting correct biomarkers in the simulated datasets. The result shows that my model outperforms the Cox model in feature selection, with impressive improvements of 53%, 50% and 32% respectively.

In the simulated dataset, each method selected a patient subgroup consisting of 40% samples with longer predicted survival times. Then the ICER was calculated for each selected subgroup. The effectiveness was the median survival time, which was estimated using the Kaplan-Meier method. The results show that my model can select a patient subgroup with longest median survival time compared with other methods. Therefore, the patient subgroup selected by my model has the lowest ICER.

5.2 Strengths

This is the first research on applying the state-of-the-art ML method MTL in subgroup identification, as well as health economics. In the era of big data and artificial intelligence, the use of ML to improve data analytics in different fields is of global interest (Williamson et al., 2018). It is foreseeable that developing ML models for health economics will become increasingly important in the near future (Chen et al., 2020).

One of the strengths of the new model proposed in this research is that it predicts survival time. When given data from a new patient, the new model can directly predict survival time without estimating the baseline function, whereas other comparison methods only predict risk scores.

Another strength is that the new model has a relatively high C-index compared with other methods, and an impressively high feature selection accuracy compared with the Cox models.

As a result of these strengths, the patient subgroup selected by the model has a longer median survival time and a lower ICER compared with other methods. Therefore, the model presented in this study can be used as a tool in the analysis of data from early clinical trials by pharmaceutical companies to find potential predictive/prognostic biomarkers. The result of the analysis could inform the design of further trials regarding patient subgroups.

5.3 Limitations

There are five limitations of this research.

First, the convergence rate of the algorithm can be slow, and the computation can be intensive when the dataset is large. This is because the optimisation technique for the new model is sub-gradient descent, a first-order method. Sub-gradient descent uses all samples at each iteration to update the weight matrix, which is computationally intensive and can be slow when the sample size is large. Therefore, the new model has the same weakness as the first-order optimisation.

Second, the new model is not specially designed for recognising predictive biomarkers. It may have difficulty in selecting predictive biomarkers if the predictive biomarkers are much weaker than the prognostic biomarkers. Therefore, after important biomarkers have been selected, additional analysis is required to identify predictive biomarkers.

Third, this model can only predict the survival time up to the longest observation point. For example, if the longest survival time for the observed patients is 10 years, then the maximum survival time predicted by my model is 10 years.

Fourth, all the real datasets are small size with high amount of censoring. The sample sizes of all real datasets are 78, 86, 115, and 295, respectively. Meanwhile, all the real datasets are highly censored, with censorship percentages ranging from 56% to 73%. The simulated datasets are all based on exponential distribution.

Fifth, experiments showed that C-index tend to overestimate the performance when the censorship percentage exceeds approximately 66% for small sample datasets (Pölsterl, 2020).

5.4 Further Research

As mentioned in the limitations, the real datasets in this research are all high dimensional gene expression datasets with small sample sizes. So, the ability and stability of the new model can be further studied with additional real world datasets.

The optimisation technique applied in this research is a first-order method – sub-gradient decent. Sub-gradient decent is a good choice for non-smooth optimisation (the modified hinge loss in my algorithm is non-smooth), but it is slow and computationally intensive. There are several new optimisation methods designed specifically for non-smooth optimisation. For example, smoothing approximation of non-smoothing functions (Nesterov, 2005), projected gradient method, and proximal gradient method. These methods can be applied in this research to speed up the training of the algorithm.

The new model in this research can be applied in the framework of other subgroup identification methods to replace the time-to-event prediction model. According to the literature review of subgroup identification methods, the central step of the global outcome modelling methods is to build models to predict the treatment outcome function (e.g., survival time). Since the new model has a high prediction accuracy compared with other state-of-the-art methods, it can be applied in the global outcome modelling framework to identify predictive biomarkers in randomised controlled trials, and to improve the general accuracy of subgroup identification.

5.5 Conclusion

For the primary question ‘compared with existing models, can the proposed model improve the accuracy of time-to-event prediction based on the censored dataset, as assessed by the concordance index?’, the answer appears to be ‘yes’ according to the results in Chapter 4. For the secondary question ‘compared with existing models, can the proposed model select the correct predictive or prognostic biomarkers?’, according to the results in Chapter 4, the answer appears to be ‘yes’ when compared with the Cox models. For the secondary question ‘how will the ICER change when calculated among selected patient subgroups and whole patient population?’, the answer appears to be that the ICER reaches its lowest when calculated among the patient subgroup selected by the new model.

We are still on the road to precision medicine. ‘Using genomics and precision medicine to help Australia become the healthiest country on earth’ was identified as an ideal national mission (Williamson et al., 2018). As part of the healthcare system, HTA plays an important role in the allocation of medical resources by providing policymakers with the necessary information to make better funding decisions.

This research shows the new model has enough predictivity to improve the time-to-event estimation and feature selection for subgroup identification. As such, the new model has the power to identify patients subgroups for which treatments are particularly good value for money in precision medicine, and ultimately improving the allocation of resources and quality of life for Australians.

With the dramatic increase in data collection capabilities and the variety of covariates, machine learning and data science also play a crucial role in precision medicine. It is foreseeable that in the near future, the development of ML models for HTA will become increasingly popular and gain more attention (Williamson et al., 2018; Chen et al., 2020). In addition, the model provides theoretical improvements to the application of ML to survival analysis. Thus, this research can be applied to other areas beyond health economics, as survival analysis is used widely in a range of other disciplines, such as ecology and engineering.

A

Appendix

A.1 Code of the New Model

```
def mymodel(X, Y, B, max_iter = 100, lambda1 = 1, lambda2 = 1):
    N, M = X.shape
    Ny, T = Y.shape
    if N != Ny:
        return(’wrong dimension’)
    #initialise W, D
    W = np.zeros((M, T))
    D = np.diag(np.ones(M)) * 1000
    for i in range(1, max_iter):
        eta = 1 / i * lambda1
        Y_prediction = X @ W
        L1 = gradient1(X, Y, B, Y_prediction, N, M, T)
        L2 = gradient2(X, Y, Y_prediction, N, M, T)
        dif = eta * (2 * lambda1 * D @ W + L1 + lambda2 * L2)
        error = LA.norm(dif, 'fro')
        if error < 0.0001:
            print(f'The optimisation end at iteration {i}')
            break
        W = W - dif
        D = calculate_D(W)
    return W, D, error

def gradient1(X, Y, B, Y_prediction, N, M, T):
    result = np.zeros((M, T))
    for j in range(T):
        for i in range(N):
            if Y_prediction[i, j] * Y[i, j] < 1 and B[i, j] == 1:
                result[:, j]  -= Y[i, j] * X[i, :]
    return result

def gradient2(X, Y, Y_prediction, N, M, T):
    result = np.zeros((M, T))
    for i in range(N):
        for j in range(T-1):
            if Y_prediction[i, j] < 0:
                for k in range(j + 1, T):
                    if Y_prediction[i, k] > 0:
                        result[:, j] -= X[i, :]
                        result[:, k] += X[i, :]
    return result
```

```
def calculate_D(W):
    M, T = W.shape
    Diag = []
    for i in range(M):
        a = max(0.001, LA.norm(W[i, :]))
        Diag.append(1/a)
    D = np.diag(np.array(Diag))
    return D
```

References

D. Alemayehu, Y. Chen, and M. Markatou. A comparative study of subgroup identification methods for differential treatment effect: Performance metrics and recommendations. *Statistical methods in medical research*, 27(12):3658–3678, 2018. 8, 9

D. Beer, S. Kardia, C.-C. Huang, T. Giordano, A. Levin, D. Misek, L. Lin, G. Chen, G. Tarek, D. Thomas, M. Lizyness, R. Kuick, S. Hayasaka, J. Taylor, M. Iannettoni, M. Orringer, and S. Hanash. Gene-expression profiles predict survival of patients with lung adenocarcinoma. *Nature medicine*, 8:816–24, 09 2002. doi: 10.1038/nm733. 52

L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001. 16

T. Cai, L. Tian, P. H. Wong, and L. Wei. Analysis of randomized comparative clinical trial data for personalized treatment selections. *Biostatistics*, 12(2):270–282, 2011.

R. Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997. doi: 10.1023/A:1007379606734. URL `https://doi.org/10.1023/A:1007379606734`. 6, 7

Y. Chen, V. V. Chirikov, X. L. Marston, J. Yang, H. Qiu, J. Xie, N. Sun, C. Gu, P. Dong, and X. Gao. Machine learning for precision health economics and outcomes research (p-heor): Conceptual review of applications and next steps. *J Health Econ Outcomes Res*, 7(1):35–42, 2020. ISSN 2327-2236 (Electronic); 2326-697X (Print); 2326-697X (Linking). doi: 10.36469/jheor.2020.12698. 3, 5, 77, 79

D. Cox. Regression models and life table. *Journal of the Royal Statistical Society. Series B*, 34, 01 1972. doi: 10.1007/978-1-4612-4380-9_37. 53

C. Davidson-Pilon. lifelines: survival analysis in python. *Journal of Open Source Software*, 4(40):1317, 2019. doi: 10.21105/joss.01317. URL `https://doi.org/10.21105/joss.01317`. 67

A. Dmitrienko, C. Muysers, A. Fritsch, and I. Lipkovich. General guidance on exploratory and confirmatory subgroup analysis in late-stage clinical trials. *Journal of biopharmaceutical statistics*, 26(1):71–98, 2016. 4, 14

M. F. Drummond, M. J. Sculpher, K. Claxton, G. L. Stoddart, and G. W. Torrance. *Methods for the economic evaluation of health care programmes*. Oxford university press, 2015. 2

P. Fahr, J. Buchanan, and S. Wordsworth. A review of the challenges of using biomedical big data for economic evaluations of precision medicine. *Applied health economics and health policy*, 17(4):443–452, 2019. ISSN 1175-5652. 2

J. C. Foster, J. M. Taylor, and S. J. Ruberg. Subgroup identification from randomized clinical trial data. *Statistics in medicine*, 30(24):2867–2880, 2011. 16

S. P. Gavan, A. J. Thompson, and K. Payne. The economic case for precision medicine. *Expert review of precision medicine and drug development*, 3(1):1–9, 2018. 4, 5

F. E. Harrell, R. M. Califf, D. B. Pryor, K. L. Lee, and R. A. Rosati. Evaluating the yield of medical tests. *Jama*, 247(18):2543–2546, 1982. 55

N. Hill, B. Sandler, R. Mokgokong, S. Lister, T. Ward, R. Boyce, U. Farooqui, and J. Gordon. Cost-effectiveness of targeted screening for the identification of patients with atrial fibrillation: Evaluation of a machine learning risk prediction algorithm. *Journal of Medical Economics*, 23:1–1, 12 2019a. doi: 10.1080/13696998.2019.1706543. 4, 5

N. R. Hill, D. Ayoubkhani, P. McEwan, D. M. Sugrue, U. Farooqui, S. Lister, M. Lumley, A. Bakhai, A. T. Cohen, M. O'Neill, et al. Predicting atrial fibrillation in primary care using machine learning. *PloS one*, 14(11):e0224582, 2019b. 4, 5

R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge university press, 2012. 30

Y. Huang, P. B. Gilbert, and H. Janes. Assessing treatment-selection markers using a potential outcomes framework. *Biometrics*, 68(3):687–696, 2012.

C. Huber, N. Benda, and T. Friede. A comparison of subgroup identification methods in clinical drug development: Simulation study and regulatory considerations. *Pharmaceutical statistics*, 18(5):600–626, 2019. 8, 9

K. Imai and M. Ratkovic. Estimating treatment effect heterogeneity in randomized program evaluation. *The Annals of Applied Statistics*, 7(1):443–470, 2013. 16

H. Ishwaran, U. B. Kogalur, E. H. Blackstone, and M. S. Lauer. Random survival forests. *The annals of applied statistics*, 2(3):841–860, 2008. 54

M. Joore, S. Grimm, A. Boonen, M. de Wit, F. Guillemin, and B. Fautrel. Health technology assessment: a framework. *RMD Open*, 6:e001289, 11 2020. doi: 10.1136/rmdopen-2020-001289. URL `https://rmdopen.bmj.com/content/rmdopen/6/3/e001289.full.pdf`. 2

J. P. Klein and Klein. *Handbook of Survival Analysis*. Crc Press, 2013. 18, 53, 54

D. Krstajic, L. J. Buturovic, D. E. Leahy, and S. Thomas. Cross-validation pitfalls when selecting and assessing regression and classification models. *Journal of cheminformatics*, 6(1):1–15, 2014. 59

V. Kumar, J. T. Cohen, D. van Klaveren, D. I. Soeteman, J. B. Wong, P. J. Neumann, and D. M. Kent. Risk-targeted lung cancer screening: a cost-effectiveness analysis. *Annals of internal medicine*, 168(3):161–169, 2018.

Y. Li, J. Wang, J. Ye, and C. K. Reddy. A multi-task learning formulation for survival analysis. In *Proceedings of the 22nd ACM SIGKDD International Conference on*

Knowledge Discovery and Data Mining, KDD '16, page 1715–1724, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2939857. URL `https://doi.org/10.1145/2939672.2939857`. 7, 18, 19, 20, 38, 40, 53, 55, 67

Y. Ling, Y. Chen, V. Chirikov, J.-F. Xie, H. Qiu, Z. Otgonsuren, P. Dong, and X. Gao. A prediction model for length of stay in the icu among septic patients: A machine learning approach. *Value in Health*, 21:S5, 05 2018. doi: 10.1016/j.jval.2018.04.016. 4, 5

I. Lipkovich, A. Dmitrienko, and R. B D'Agostino Sr. Tutorial in biostatistics: data-driven subgroup identification and analysis in clinical trials. *Statistics in medicine*, 36(1):136–196, 2017. 16, 17

B. Liu, Y. Li, Z. Sun, S. Ghosh, and K. Ng. Early prediction of diabetes complications from electronic health records: A multi-task survival analysis approach. In *AAAI*, 2018. 7, 18

W.-Y. Loh, L. Cao, and P. Zhou. Subgroup identification for precision medicine: A comparative review of 13 methods. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 9(5):e1326, 2019. 8, 9

J. Love-Koh, A. Peel, J. C. Rejon-Parrilla, K. Ennis, R. Lovett, A. Manca, A. Chalkidou, H. Wood, and M. Taylor. The future of precision medicine: potential impacts for health technology assessment. *Pharmacoeconomics*, 36(12):1439–1451, 2018. 3

T. Mitchell. *Machine Learning.* McGraw-Hill, 1997. 24

K. P. Murphy. *Machine learning: a probabilistic perspective.* MIT press, 2012. 24, 25, 26, 27, 31, 44, 50

National Institute for Health and Care Excellence. *NICE health technology evaluations: the manual*, 2022. URL `https://www.nice.org.uk/process/pmg36`.

Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical programming*, 103(1):127–152, 2005. 78

A. Ng. *Machine Learning [lecture notes]*, 2018. 30

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 61

Pharmaceutical Benefits Advisory Committee. *Guidelines for preparing a submission to the Pharmaceutical Benefits Advisory Committee*, 2016. URL `https://pbac.pbs.gov.au`.

S. Pölsterl. scikit-survival: A library for time-to-event analysis built on top of scikit-learn. *Journal of Machine Learning Research*, 21(212):1–6, 2020. URL `http://jmlr.org/papers/v21/20-729.html`. 54, 67, 78

H. L. Royden and P. Fitzpatrick. *Real analysis*, volume 32. Macmillan New York, 1988. 29

P. Royston and D. Altman. Regression using fractional polynomials of continuous covariates: Parsimonious parametric modelling (with discussion). *Appl Stat*, 43: 425–467, 01 1994. 16

P. Royston and W. Sauerbrei. A new approach to modelling interaction between treatment and continuous covariates in clinical trials by using fractional polynomials. *Statistics in medicine*, 23:2509–25, 08 2004. doi: 10.1002/sim.1815. 16

X. Song and M. S. Pepe. Evaluating markers for selecting a patient's treatment. *Biometrics*, 60(4):874–883, 2004.

T. Sorlie, R. Tibshirani, J. Parker, T. Hastie, J. Marron, A. Nobel, S. Deng, H. Johnsen, R. Pesich, S. Geisler, J. Demeter, C. Perou, P. Lønning, P. Brown, A. Borresen-Dale, and D. Botstein. Repeated observation of breast tumor subtypes in independent gene expression data sets. *PROCEEDINGS OF THE NATIONAL ACADEMY OF SCIENCES OF THE UNITED STATES OF AMERICA*, 100:8418–8423, 01 2003. 51

A. Spooner, E. Chen, A. Sowmya, P. Sachdev, N. A. Kochan, J. Trollor, and H. Brodaty. A comparison of machine learning methods for survival analysis of high-dimensional clinical data for dementia prediction. *Scientific reports*, 10(1):1–10, 2020. 6

X. Su, T. Zhou, X. Yan, J. Fan, and S. Yang. Interaction trees with censored survival data. *The international journal of biostatistics*, 4(1), 2008. 17

H. C. van Houwelingen, T. Bruinsma, A. A. M. Hart, L. J. van't Veer, and L. F. A. Wessels. Cross-validated cox regression on microarray gene expression data. *Statistics in Medicine*, 25(18):3201–3216, 2006. doi: https://doi.org/10.1002/sim.2353. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.2353`. 52

L. van't Veer, H. Dai, M. Vijver, Y. He, A. Hart, M. Mao, H. Peterse, K. Kooy, M. Marton, A. Witteveen, G. Schreiber, R. Kerkhoven, C. Roberts, P. Linsley, R. Bernards, and S. Friend. Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415:530–6, 02 2002. doi: 10.1038/415530a. 51

L. Wang, Y. li, J. Zhou, D. Zhu, and J. Ye. Multi-task survival analysis. pages 485–494, 11 2017. doi: 10.1109/ICDM.2017.58. 7, 18, 19

P. Wang, T. Shi, and C. K. Reddy. A novel tensor-based temporal multi-task survival analysis model. *IEEE Transactions on Knowledge & Data Engineering*, 33(09):3311–3322, sep 2021. ISSN 1558-2191. doi: 10.1109/TKDE.2020.2967700. 7, 18

S. Wang, X. Chang, X. Li, Q. Z. Sheng, and W. Chen. Multi-task support vector machines for feature selection with shared knowledge discovery. *Signal Processing*, 120:746–753, 2016. 41

R. Williamson, W. Anderson, S. Duckett, I. Frazer, C. Hillyard, J. M. Emma Kowal, C. McLean, K. North, and A. Turner. The future of precision medicine in australia. *Report for the Australian Council of Learned Academies*, 2018. 2, 3, 9, 76, 79

B. Winterhoff, S. Kommoss, F. Heitz, G. Konecny, S. Dowdy, S. Mullany, T. Park-Simon, K. Baumann, F. Hilpert, S. Brucker, A. du Bois, W. Schröder, A. Burges, S. Shen, J. Wang, R. Tourani, S. Ma, J. Pfisterer, and C. Aliferis. Developing a clinico-molecular test for individualized treatment of ovarian cancer: The interplay of precision medicine informatics with clinical and health economics dimensions. *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*, 2018:1093–1102, 2018. ISSN 1559-4076. 5

Y. Yang, Z. Ma, A. G. Hauptmann, and N. Sebe. Feature selection for multimedia analysis by sharing information among multiple tasks. *IEEE Transactions on Multimedia*, 15(3):661–669, 2012. 41

C.-N. Yu, R. Greiner, H.-C. Lin, and V. Baracos. Learning patient-specific cancer survival distributions as a sequence of dependent regressors. 05 2012. 7, 18, 19

C. J. Zack, C. Senecal, Y. Kinar, Y. Metzger, Y. Bar-Sinai, R. J. Widmer, R. Lennon, M. Singh, M. R. Bell, A. Lerman, et al. Leveraging machine learning techniques to forecast patient prognosis after percutaneous coronary intervention. *Cardiovascular Interventions*, 12(14):1304–1311, 2019. 4, 5

Y. Zhang and Q. Yang. An overview of multi-task learning. *National Science Review*, 5:30–43, 01 2018. doi: 10.1093/nsr/nwx105. 7

Y. Zhang and Q. Yang. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, 2021. doi: 10.1109/TKDE.2021.3070203. 7, 29, 30, 40

L. Zhao, L. Tian, T. Cai, B. Claggett, and L.-J. Wei. Effectively selecting a target population for a future comparative study. *Journal of the American Statistical Association*, 108(502):527–539, 2013.