

Thesis for the Master of Science

Learning to Directly Maximize Evaluation Metrics:
Focus on Matching Efficiency and Concordance Index



Sang-Kyun Ko

Graduate School of Hanyang University

February 2024

Thesis for Master of Science

Learning to Directly Maximize Evaluation Metrics:
Focus on Matching Efficiency and Concordance Index

Thesis Supervisor: Yung-Kyun Noh

A Thesis submitted to the graduate school of
Hanyang University in partial fulfillment of the requirements for
the degree of Master of Science

Sang-Kyun Ko

February 2024

Department of Computer Science
Graduate School of Hanyang University


This thesis, written by Sang-Kyun Ko,
has been approved as a thesis for the Master of Science.

February 2024

Committee Chairman: Dong-Kyu Chae

(Signature) 

Committee member: Yung-Kyun Noh

(Signature) 

Committee member: Cheongjae Jang

(Signature) 

Graduate School of Hanyang University

Table of Contents

Abstract	v
1 Introduction	1
1.1 Identifying additional b-jets in the $t\bar{t}b\bar{b}$ process	1
1.2 Censored data in survival analysis	3
1.3 Organization	5
2 Problem Definition	6
2.1 Identifying additional b-jets based on machine learning	6
2.1.1 Previous approach: binary classification	6
2.1.2 Reconstruction to $t\bar{t}b\bar{b}$ events	7
2.1.3 Matching efficiency	8
2.2 Survival data reconstruction	9
2.2.1 Concordance index	9
2.2.2 Data reconstruction	12
2.2.3 Concordance index for reconstructed data	14
3 Methods	16
3.1 Prediction model	16
3.2 Surrogate loss	17
4 Experiments	18
4.1 Synthetic data	18
4.1.1 Details	18
4.1.2 Results	19
4.2 Simulated $t\bar{t}b\bar{b}$ process data	21
4.2.1 Details	21

4.2.2	Results	23
4.3	Survival data.....	23
4.3.1	Details.....	24
4.3.2	Results	25
5	Conclusion and Discussion	27
	References.....	29
A	Appendix	34
	국문요지	34



List of Tables

4.1	Mean of Matching Efficiency (standard deviation) across all 10 trials in each training data size.....	23
4.2	Table presents sample data count, right-censored data ratio, and ran- dom variable dimension from left to right.	25
4.3	Average of the c-index performance of other methods and our ap- proach, and the standard deviation of the 10-fold validation results is bracketed.	26
A.1	P-values by 10-trial t-test for each performance that compared the proposed methods and others.	34
A.2	C-index results of 10-fold validation with SUPPORT2 data.	35
A.3	C-index results of 10-fold validation with AIDS3 data.....	36
A.4	C-index results of 10-fold validation with COLON DEATH data.....	37

List of Figures

1.1	Feynman diagram for the $t\bar{t}b\bar{b}$ process	1
2.1	Order graph of acceptable pairs in terms of time for both complete and right-censored data. Solid black dots represent data where the event of interest occurred and are marked in the $\delta = 1$ column, while empty circles represent right-censored data where the event did not occur within the observation period and are marked in the $\delta = 0$ column.	10
4.1	(a) A single test matrix of synthetic data and the decision boundaries for each method. (b) Decision boundaries of each linear classifier in the overall scattered test data. (c) and (d) Changes in matching efficiency and loss in Eq.(2) and Eq.(17) over 10,000 epochs, respectively.....	20

Abstract

Learning to Directly Maximize Evaluation Metrics: Focus on Matching Efficiency and Concordance Index

Sang-Kyun Ko

Department of Computer Science
Graduate School of Hanyang University

In particle physics, unraveling the mysteries of the Higgs Boson is a crucial task. In particular, gaining a precise understanding of the $t\bar{t}b\bar{b}$ process, which is related to the $t\bar{t}H(b\bar{b})$ and the associated issues, is closely related to elucidating the properties of the Higgs Boson. Therefore, in the $t\bar{t}b\bar{b}$ event, classifying (or identifying) b-jets originating from top quark decay and the additional b-jets generated from gluon splitting is critical. Since no simple physical rules exist for this identification problem, previous studies have employed machine-learning approaches based on simulated data. Existing methods train without individually structuring each $t\bar{t}b\bar{b}$ event; instead, they perform binary classification on all b-jets and the additional b-jets, minimizing the binary cross-entropy loss. However, the performance metric **Matching Efficiency** of the prediction model evaluates the ability to identify the number of additional b-jets for every structured $t\bar{t}b\bar{b}$ data. Given the metric's evaluation purpose, we propose reconstructing the data into the $t\bar{t}b\bar{b}$ structure. Furthermore, we introduce the surrogate loss function to directly maximize this evaluation metric for achieving improved performance compared to existing approaches. To compare the concepts of the previous and proposed methods, we performed synthetic data experiments to evaluate two loss functions. We generated synthetic data that pre-

served the structural characteristics, making learning challenging via naive binary classifiers but effectively trainable with our proposed method. We quantitatively evaluated the performance using elaborately simulated $t\bar{t}b\bar{b}$ process data to compare the performance of the proposed method with that of binary classification.

Furthermore, the proposed surrogate loss can also be applied to survival analysis. The **Concordance index** (c-index) metric, used as a criterion of model performance in survival analysis, constructs test data into pairs and assesses the ability to identify the data with longer survival times in each pair. We consider this metric similar to the matching efficiency; hence, survival analysis can be cast as an identifying problem similar to the initial issue in the first paragraph. Therefore, we propose a method with surrogate loss to directly maximize the c-index by reconstructing the entire data into pairs, as used in this metric. The proposed learning method with surrogate loss enables us to utilize right-censored data that have incomplete target values and are unusable for training data in the classic machine learning approach. We used three real-world datasets to evaluate the efficiencies of the data reconstruction strategies. The proposed method achieved competitive performance with partial likelihood, rank methods, and Wasserstein metric, proven effective in survival analysis, and showed no statistically significant difference.

1 Introduction

1.1 Identifying additional b-jets in the $t\bar{t}b\bar{b}$ process

The Higgs Boson, discovered in 2012 [2, 12], plays a crucial role in particle physics for explaining the universe based on the Standard Model of elementary particles. The Higgs Boson is responsible for granting mass to elementary particles, strongly interacting with the heaviest top quark, and the bottom quark plays a vital role in studying mass-forming mechanisms through its interactions with the Higgs Boson. Therefore, various preliminary studies are necessary to unravel the mysteries of the Higgs Boson. Among them, the $t\bar{t}(H)b\bar{b}$ process [32, 1] is one of the crucial interactions involving the Higgs Boson and top quark. To accurately understand this interaction, it is essential to interpret the $t\bar{t}b\bar{b}$ process as a function of the **additional b-jets** property [14].

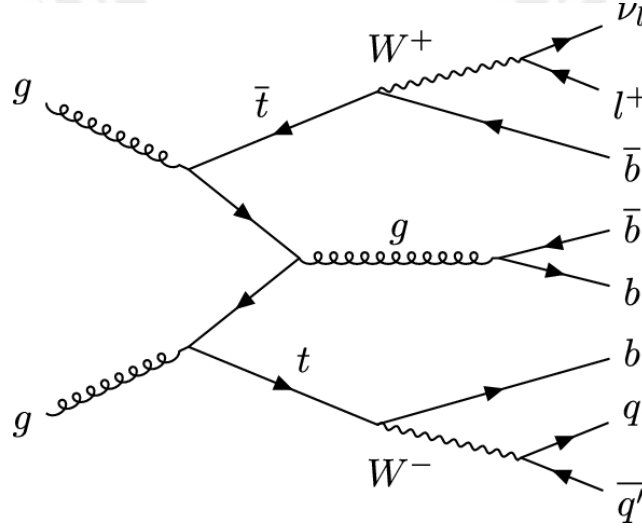


Figure 1.1: Feynman diagram for the $t\bar{t}b\bar{b}$ process

The $t\bar{t}b\bar{b}$ process, as shown in Figure 1.1, which can be described through high-energy particle collision experiments, involves a collision event generating **b-jets**

originating from the decay of the top quark and **additional b-jets** created from gluon splitting. Due to the lack of physical rules to clearly identify the additional b-jets in a $t\bar{t}b\bar{b}$ event consisting of multiple b-jets, research to distinguish them [16] using machine learning methods is actively being conducted [14, 19].

Multiple b-jets constituting an $t\bar{t}b\bar{b}$ event data are made up of collections of all b-jet pairs (including additional b-jets), and these pairs are represented by multidimensional feature vectors composed of observable momenta. We aim to distinguish the unique additional b-jet pair generated from gluon splitting among the multiple b-jet pairs created in the $t\bar{t}b\bar{b}$ event.

However, existing approach, binary classification through neural network training [14], classifies every b-jet pair and all additional b-jet pairs in unconstructed shuffled data, disregarding the structures that can be bundled into each $t\bar{t}b\bar{b}$ event. This method not only fails to utilize the configuration that additional b-jets data is unique for each $t\bar{t}b\bar{b}$ event data, but it also optimizes the learning for binary cross-entropy, which has a different objective from the evaluation metric matching efficiency for assessing the prediction model. In contrast, the **Matching Efficiency** metric evaluates the model's ability to distinguish additional b-jets by counting the number of success signals identified in structured $t\bar{t}b\bar{b}$ events and assigning a score. Therefore, approaching it as a simple binary classification problem, which ignores the structures tied into the $t\bar{t}b\bar{b}$ process, may somewhat improve the target matching efficiency metric but cannot effectively maximize it. Based on this, we consider a learning approach that directly maximizes the matching efficiency metric. However, matching efficiency is a non-smooth function using the Kronecker delta, making neural network training using gradient methods highly limited [6]. Therefore, this paper proposes a data reconstruction method for training neural networks, using a function that appropriately relaxes this step function as a surrogate loss function.

1.2 Censored data in survival analysis

Secondly, this study offers several benefits by applying the proposed method to survival analysis, which uses a function similar to matching efficiency as an evaluation metric. Primarily used in medical statistics for estimating the onset of patient diseases or predicting machinery defects in industries, survival analysis investigates the occurrence (survival) time T of interest events from the time of observing and the covariates \mathbf{x} that influence the event's occurrence. As survival analysis tracks the subjects of interest over a limited research period and measures survival time, the data is meaningful for analysis only if an event occurs within the observation period. However, it is common for events of interest not to occur during the observation period, leading to the termination of observation. This creates a situation where one cannot define a precise survival time T for such data. This case of data is referred to as **right-censored** [31]. Right-censored data cannot be used in traditional statistical methods or as regular machine learning data, as the incomplete survival time T serves as the target value. Moreover, survival analysis data collection requires an observation period, which is relatively expensive. Ultimately, the missing data problem due to right-censoring can critically harm survival analysis, which usually involves small datasets.

However, it is also inappropriate to treat all right-censored data as missing data. Because right-censored data can be used as test data in the most commonly used evaluation metric in survival analysis, the **Concordance index** (c-index). The c-index evaluates how well the ordering of survival times from a pair of data points matches the ordering from the prediction model f , i.e., that is, if the survival times T for the pair $(\mathbf{x}_i, \mathbf{x}_j)$ satisfy $T_i < T_j$, and predicted values from the model f follow the same order $f(\mathbf{x}_i) < f(\mathbf{x}_j)$, then the ordering is considered to be concordant, and scores are awarded. We know that the right-censored data survived to at least the threshold, therefore, survival time comparison is feasible if the survival time value of the intact data is below that of the censored data. This pair is called an

acceptable pair [37], and our study focuses on these c-index characteristics, which use pair of two data for comparison. This task aim can be redefined as identifying the highest survival time among the data points in the pair, similar to the problem described in Chapter 1.1. Therefore, the approach utilized in Chapter 1.1 can be applied similarly to this problem. Ultimately, to facilitate neural network training for right-censored data, we propose reconstructing the acceptable pair set into a matrix and introducing a matrix c-index for its evaluation. We transform the discrete matrix c-index into a c-index optimizable with the gradient method to directly maximize it as the neural network’s surrogate objective function. Subsequently, we present the derivation process for converting it into a form of commonly used error functions. This approach offers the advantage of leveraging right-censored data while directly optimizing the model’s evaluation metric.

There has been a continuous study of utilizing right-censored data for analysis by optimizing the c-index. Cox suggested a proportional hazard model (Cox PH model) [17] in 1972, considering censored data. This method is still frequently used as a survival prediction model due to the advantage of being able to process some right-censored data using partial likelihood. Later, Rakar’s study [36] revealed that Cox’s partial likelihood method approximates maximizing the c-index. In addition, it proved that survival analysis is naturally treated as a Ranking problem [10, 15]. This study used right-censored, optimizing the c-index by applying the previously proposed ranking methods. Newly, Sylvain’s study [25, 37] proposed a new approach to applying the Wasserstein distance. Unlike treating the problem definition of existing survival analysis as a regression problem, this study deals with right-censored as a classification method. In addition, studies [38, 13, 3, 8] to optimize directly or explore the c-index have various approaches.

1.3 Organization

This paper proposes the same approach for two distinct domains and describes the advantages for each domain when the proposed method of data reconstruction and surrogate loss are applied. Chapter 2 introduces the problems of both domains through mathematical expressions. Chapter 2.1 discusses the existing approaches introduced in Chapter 1.1 and how the problem can be redefined through our approach. Chapter 2.2 describes reconstructing the data to apply the problems and goals defined in Chapter 2.1 to survival analysis. Chapter 3 introduces our methods applicable to the integrated defined of problems in Chapter 2 and ultimately presents the derivation of the surrogate loss function. Finally, in the experimental section, we validate the effectiveness of our approach through experiments on data from two domains. Chapter 4.1 implements and compares the data structure with synthetic data to explain the apparent differences between the binary classification method and the problem defined in Chapter 2.1. Chapter 4.2 reports various performance comparison experiments using finely designed $t\bar{t}b\bar{b}$ process simulation data between the binary classification method and our approach. Subsequently, Chapter 4.3 compares our proposed method with other state-of-the-art methods using publicly available survival data, where right-censored data are employed as the training data.

2 Problem Definition

This chapter discusses how problems can integrately redefined by reconstructing data for two domains. First, in Chapter 2.1, we define the problem of identifying additional b-jets in a $t\bar{t}b\bar{b}$ event via a proposed approach using mathematical expressions. Then, in Chapter 2.2, we redefine the problem of survival analysis by restructuring the survival data in a form similar to that in Chapter 2.1.

2.1 Identifying additional b-jets based on machine learning

The objective of the problem introduced in Chapter 1.1 can be summarized as “Distinguishing the unique additional b-jet pair (signal) generated from gluon splitting among multiple b-jet pairs (background) generated in the $t\bar{t}b\bar{b}$ process” (Problem 1). A single event datum of the $t\bar{t}b\bar{b}$ process consists of pairs of b-jet and additional b-jet, each represented by a row of feature vectors in the matrix of event. Therefore, a $t\bar{t}b\bar{b}$ event contains at least two additional b-jets [14], which are pairs, and the goal is to distinguish the additional b-jet pair among the background b-jet pair data.

2.1.1 Previous approach: binary classification

Consider N $t\bar{t}b\bar{b}$ events, each composed of c pairs of b-jet (or vectors).¹ Where, each pair of b-jet is represented as D -dimensional feature vector data \mathbf{x} . For supervised learning, it is possible to label the target values for each type of b-jet pair in binary: label the additional b-jet pair that needs to be distinguished with a target value of $y^{01} = 1$, and label the background b-jet pair with $y^{01} = 0$. Therefore, the

¹The number of b-jet pair included in each event data for the $t\bar{t}b\bar{b}$ process varies, meaning the number of b-jet pairs constituting each $t\bar{t}b\bar{b}$ event also differs. However, for simplicity in explanation, we assume that there are c b-jet pairs in each event. When preprocessing for data training, all events are defined in the same dimension based on the maximum number of b-jet pair c^{max} in any event. For event data with fewer rows than c^{max} , the remaining rows are filled with zeros.

set of data expressed as $\mathbf{x} \in \mathbb{R}^D, y^{01} \in \{0, 1\}$ can be defined as

$$\mathcal{D}^{01} = \{\mathbf{x}_i, y_i^{01}\}_{i=1}^{c \cdot N}, \quad (1)$$

this is suitable for binary classification in supervised learning. The binary classifier employs a fully connected neural network model that takes D -dimensional vector data \mathbf{x} as input and as $f^{01} : \mathbb{R}^D \rightarrow [0, 1]$. It utilizes binary cross-entropy loss as follows

$$\mathcal{L}_{bce} = -\frac{1}{cN} \sum_{i=1}^{cN} (y_i^{01} \log f^{01}(\mathbf{x}_i) + (1 - y_i^{01}) \log (1 - f^{01}(\mathbf{x}_i))), \quad (2)$$

the model is trained to minimize this loss function. While this approach outperforms methods that identify using rules that consider physical properties [14], it does not train specifically for the aforementioned Problem 1. More specifically, training to minimize Eq. (2) aims to find the naive boundary to classify N additional b-jets and $(c - 1) \cdot N$ background b-jets with ignore data structure.

2.1.2 Reconstruction to $t\bar{t}b\bar{b}$ events

To evaluate matching efficiency, which is our ultimate goal, as mentioned in the Introduction, we require the results in the form of a structuralized event matrix unit, where N $t\bar{t}b\bar{b}$ events are grouped by their event numbers. The evaluation focuses on locating the unique row index of the additional b-jet pair across multiple event matrices. To achieve this goal, we reorganize the dataset into bundles of N $t\bar{t}b\bar{b}$ events that can be evaluated for matching efficiency. We now define a $t\bar{t}b\bar{b}$ event matrix, composed of c D -dimensional feature vectors arranged as rows, as $M_n \in \mathbb{R}^{c \times D}$. All b-jets constituting the n -th event matrix M_n are denoted as \mathbf{x} , and the row index of the additional b-jet pair vector corresponding to the event datum is defined as

$y_n \in \{1, \dots, c\}$:

$$M_n = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_c \end{bmatrix}, \quad y_n \in \{1, \dots, c\}, \quad n = 1, \dots, N. \quad (3)$$

Therefore, the N $t\bar{t}b\bar{b}$ event matrix data for supervised learning can be defined as

$$\mathcal{D} = \{M_n, y_n\}_{n=1}^N. \quad (4)$$

The dataset \mathcal{D} is restructured into $t\bar{t}b\bar{b}$ events, making it an appropriate data form for addressing Problem 1 and suitable for performance evaluation through matching efficiency, as introduced in Chapter 2.1.3.

Of course, the data can be restructured to have dimensions that are c times larger, represented as $X = [\mathbf{x}_1 \cdots \mathbf{x}_c] \in \mathbb{R}^{cD}$, $y \in \{1, \dots, c\}$. However, this approach is not advisable as it increases the dimensions of the data, thus invoking the curse of dimensionality [7].

2.1.3 Matching efficiency

We now focus on the evaluation metric for the problem of identifying additional b-jet pair data, as introduced earlier. For a given prediction model output \hat{y}_n and the actual additional b-jet pair index y_n in the n -th event matrix M_n , the matching efficiency ME can be defined using the Kronecker delta as follows:

$$\text{ME} = \frac{1}{N} \sum_{n=1}^N \delta_{y_n, \hat{y}_n}. \quad (5)$$

The Kronecker delta δ_{ij} returns 1 when i and j are identical and 0 otherwise. It is equivalent to the indicator function $\mathbb{I}(i = j)$ [26].

Our goal is now clear: given a matrix data $M \in \mathbb{R}^{c \times D}$ composed of c D -dimensional vectors, we aim to maximize Eq. (5) through training that aligns the index y of the vector data to be identified with the prediction \hat{y} .

2.2 Survival data reconstruction

This section demonstrates that the problem of survival analysis casts into the problem defined in Chapter 2.1. Using this approach, we can directly maximize the c-index used in the prediction model evaluation metric, which offers the advantage of utilizing right-censored data for training.

2.2.1 Concordance index

To introduce the c-index, we first explain the concept of acceptable pairs in survival data and the characteristics of survival data. The c-index proposed by Harrell [21] is a performance metric that indicates the degree of agreement between the observed and predicted survival orders. All pairs of data gathered from the test set are considered comparable regarding the magnitude relationship of survival time T and are thus called acceptable pairs. Subsequently, the c-index evaluates the performance of the prediction model in accurately determining the survival time order of these pairs.

Survival data comprises a D -dimensional random variable \mathbf{x} (covariate) that influences the occurrence of an event, an indicator $\delta \in \{0, 1\}$ that denotes whether the event occurred, and the time of event occurrence $T \in \mathbb{R} > 0$. If $\delta = 0$, no event occurred during the observation period, and for data where no event occurred, T is determined as the last confirmed survival time. The aforementioned right-censored data represents data for which no event occurred during the observation period and

is denoted as $\delta = 0$. Therefore, a set of N survival data can be expressed as:

$$\mathcal{D}^s = \{\mathbf{x}_i, \delta_i, T_i\}_{i=1}^N. \quad (6)$$

Next, the conditions for acceptable pairs, i.e., pairs for which survival time can be compared, are presented; all survival data are sorted in ascending order based on survival time. Consider the pairs $(\mathbf{x}_i, \mathbf{x}_j), i, j = 1, \dots, N$ that are sequentially bundled in order of increasing survival time from the sorted data. The first data point \mathbf{x}_i in acceptable pair must be uncensored, while the second data point \mathbf{x}_j can be either censored or uncensored and must always have a greater survival time than \mathbf{x}_i . In other words, because right-censored data informs us of the last observed minimum survival time, comparisons can only be made when one of the two has a greater survival time. Accordingly, acceptable pairs $(\mathbf{x}_i, \mathbf{x}_j)$ are satisfied with the following **conditions**: Condition 1) if $\delta_i = 1$ and $\delta_j \in \{0, 1\}$, $T_i < T_j$. Condition 2) In the case of $T_i = T_j$ with the same survival time values, both data must be uncensored. We define the set of acceptable pairs as the \mathcal{A} that meets these conditions and

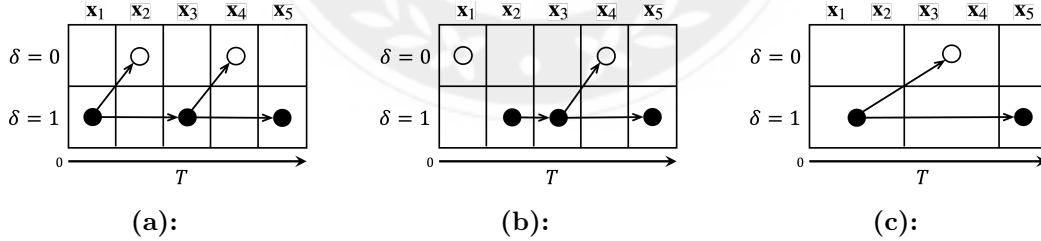


Figure 2.1: Order graph of acceptable pairs in terms of time for both complete and right-censored data. Solid black dots represent data where the event of interest occurred and are marked in the $\delta = 1$ column, while empty circles represent right-censored data where the event did not occur within the observation period and are marked in the $\delta = 0$ column.

for which survival times can be compared. We also represent the relationship of each data as graphically in Figure 2.1. In this figure, the data points are sorted by the time value T . Arrows representing comparison relationships indicate instances

where the survival times of the two data points can be compared; these arrows start from the lower cell and can only point to the right or upper-right diagonal. All nodes connected in the right direction from each node can form acceptable pairs. An example relationship involving data points with identical survival times is depicted in (c). As shown in Figure 2.1(a), the data point with the shortest survival time \mathbf{x}_1 can be compared with all nodes it can reach by following the arrows. The set of acceptable pairs among the data points $\mathbf{x}_1, \dots, \mathbf{x}_5$ in Figure 2.1 can be listed as follows:

- (a): $(\mathbf{x}_1, \mathbf{x}_2), (\mathbf{x}_1, \mathbf{x}_3), (\mathbf{x}_1, \mathbf{x}_4), (\mathbf{x}_1, \mathbf{x}_5), (\mathbf{x}_3, \mathbf{x}_4), (\mathbf{x}_3, \mathbf{x}_5)$.
- (b): $(\mathbf{x}_2, \mathbf{x}_3), (\mathbf{x}_2, \mathbf{x}_4), (\mathbf{x}_2, \mathbf{x}_5), (\mathbf{x}_3, \mathbf{x}_4), (\mathbf{x}_3, \mathbf{x}_5)$.
- (c): $(\mathbf{x}_1, \mathbf{x}_2), (\mathbf{x}_1, \mathbf{x}_3), (\mathbf{x}_1, \mathbf{x}_4), (\mathbf{x}_1, \mathbf{x}_5), (\mathbf{x}_2, \mathbf{x}_3), (\mathbf{x}_2, \mathbf{x}_4), (\mathbf{x}_2, \mathbf{x}_5)$.

To calculate the c-index, one can construct the set of acceptable pairs \mathcal{A} that meet the aforementioned conditions for all N data points. The **Concordance index** c for the survival time prediction model $f(\mathbf{x})$ with respect to \mathcal{A} is defined as follows [21, 36, 37]:

$$c(\mathbf{x}_i, T_i, \delta_i) = \frac{1}{|\mathcal{A}|} \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{A}} \mathbb{I}_{f(\mathbf{x}_i), f(\mathbf{x}_j)}. \quad (7)$$

where $|\mathcal{A}|$ is the number of elements in set \mathcal{A} , and the indicator function \mathbb{I} is defined as follows, reflecting the results of the prediction model $f(\mathbf{x}_i), f(\mathbf{x}_j)$:

$$\mathbb{I}_{f(\mathbf{x}_i), f(\mathbf{x}_j)} = \begin{cases} 1, & \text{if } T_i < T_j \text{ and } f(\mathbf{x}_i) < f(\mathbf{x}_j) \\ 0.5, & \text{if } T_i = T_j \text{ and } f(\mathbf{x}_i) = f(\mathbf{x}_j) \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

The indicator function \mathbb{I} returns 1 when the prediction model f correctly matches the order and 0 otherwise. However, in cases where the survival times are identical, determining an order is meaningless in the evaluation of the c-index. Therefore, pairs

with identical survival times are evaluated in a random order, and the indicator returns 0.5.

2.2.2 Data reconstruction

Upon examining conditions for an acceptable pair used in the c-index function, it becomes apparent that this metric inherently reflects how well the prediction model can order the survival time T of two already ordered data points [36]. The identification problem defined in Chapter 2.1., can also be approached as a Rank Problem (or ordering) [23], allowing the same approach to be applied without compromising its inherent meaning. Therefore, we transform the problem of ordering the survival times of two data points into the problem of identifying the data point with the largest survival time.

In order to leverage the suggested data reconstruction method for learning censored data, it is essential to transform not just the test data but also all the observed data into suitable pairs for utilization in the c-index. The problem of ordering T for survival analysis can thus be redefined as follows: Considering pairs that satisfy condition 1 from dataset \mathcal{D}^s , an acceptable pair composed of $(\mathbf{x}_i, \mathbf{x}_j), i, j = 1, \dots, N$ will have a target value $y_{i,j} = 2$ and can be transformed into a matrix $M_{i,j}$ as follows:

$$M_{i,j} = \begin{bmatrix} \mathbf{x}_i \\ \mathbf{x}_j \end{bmatrix}_{i,j \in \mathcal{A}}, y_{i,j} = 2, i, j = 1, \dots, N. \quad (9)$$

Eq.(9) transforms survival time ascending-ordered pair $(\mathbf{x}_i, \mathbf{x}_j)$ into a (2×1) -dimensional matrix. As per the initial assumption, the data point with the data of largest survival time value in all paired is \mathbf{x}_j ; thus, target value $y_{i,j}$ will always be 2 because \mathbf{x}_j always has largest survival time in pair. Therefore, the result of the survival model that processes data defined by Eq.(9) should always be 2. However, suppose we proceed to train on such reconstruction data. In that case, the prediction model will be

biased, losing the meaning of its ordering capability and not considering pairs with the same survival time. Thus, we include cases where the row order of the acceptable pair $(\mathbf{x}_i, \mathbf{x}_j)$ is switched in the data. We further propose a data transformation that accounts for pairs with the same survival time as follows:

$$M_n = \begin{bmatrix} \mathbf{x}_k \\ \frac{1}{2}(\mathbf{x}_i + \mathbf{x}_j) \\ \mathbf{x}_k \end{bmatrix}_{k \in \{i, j\}}, \quad y_n = \{1, 2, 3\}, \quad n = 1, \dots, N. \quad (10)$$

The reason (j, i) pairs can be reconstructed as matrices is possible because we redefine the survival analysis problem into predicting the index of the data row with the highest survival time value. If the survival times of the i, j pair are the same, the second index becomes the ground truth as $y_n = 1$. To represent the identical survival time of the two data points, we adopt a method that multiplies the sum of the two data points by 0.5, which can be modified and applied in different ways. Ultimately, the data set consisting of D -dimensional survival data in the matrix $M_n \in \mathbb{R}^{3 \times D}, n = 1, \dots, N$ is restructured as

$$\mathcal{D} = \{M_n, y_n\}_{n=1}^N. \quad (11)$$

The target value y_n is the row index of the data point with the largest survival time. With this problem transformation, the survival analysis problem can also be defined in the form of Eq. (3), where $c = 3$. Let us consider the set \mathcal{A} of acceptable pairs from the total data \mathbf{N} .

The proposed transformation method includes cases where the order of the acceptable pair i, j is switched. Therefore, the number of transformed matrix data becomes $2|\mathcal{A}|$. If all the data in \mathbf{N} of survival data \mathcal{D}^s form acceptable pairs, then

the number of elements in $|\mathcal{A}|_{\mathbf{N}}$ can be expressed by the following recurrence relation:

$$|\mathcal{A}|_{\mathbf{N}} = |\mathcal{A}|_{\mathbf{N}-1} + (\mathbf{N} - 1). \quad (12)$$

Where $|\mathcal{A}|_1 = 0$, $|\mathcal{A}|_2 = 1$. In other words, the number N of matrix data transformed from \mathbf{N} of data can be $2|\mathcal{A}|_{\mathbf{N}}$ if all data form acceptable pairs.

2.2.3 Concordance index for reconstructed data

The performance evaluation of the prediction model through the transformed survival data M_n cannot be directly calculated by the c-index in Eq.(7), which deals with vector-shaped data. However, if we transform the survival analysis problem into one that identifies the data point with the largest survival time among at least two data points, a performance evaluation with the same meaning can be defined. Therefore, we propose the following matrix c-index c^M for the reconstructed matrix data M_n , target value $y_n \in \{1, 2, 3\}$, and predicted result $\hat{y}_n \in \{1, 2, 3\}$:

$$c^M = \frac{1}{N} \sum_{n=1}^N \mathbb{I}_{y_n, \hat{y}_n}. \quad (13)$$

Where the indicator function is defined as

$$\mathbb{I}_{y_n \hat{y}_n} = \begin{cases} 1, & \text{if } T_i \neq T_j \text{ and } y_n = \hat{y}_n \\ 0.5, & \text{if } T_i = T_j \text{ and } y_n = \hat{y}_n, \\ 0, & \text{otherwise} \end{cases} \quad (14)$$

and has the same meaning as Eq.(8). If the survival times are different and the predicted index \hat{y} matches the target value y , it returns 1, and for incorrect results, it returns 0. Additionally, if the predicted results match two data with the same survival time, it returns 0.5.

The reconstructed data in Eq.(11), like in Eq.(4), is a set of matrices M composed of D -dimensional feature vectors. Furthermore, besides returning 0.5 when the survival times are the same, Eq.(13) can be used identically to Eq.(5). Therefore, the reconstruction of survival data can be defined in the same terms as the last paragraph of Chapter 2.1.3.



3 Methods

Although Chapters 2.1 and 2.2 address different tasks, they can be condensed into a single generalized objective as follows: to predict the row index of the unique data with $y = 1$ in a matrix $M \in \mathbb{R}^{c \times D}$ composed of c rows of D -dimensional vectors \mathbf{x} with a binary target value $y \in \{0, 1\}$. We propose a prediction model that can apply to this generalized problem and suggest a surrogate loss function as the objective function in neural network training.

3.1 Prediction model

Consider a prediction model that handles N sets of matrix data. The proposed neural network model takes the event matrix M as input, and returns predicted values for the indices of the rows composing the matrix. Therefore, to estimate the target value index $y_n \in \{1, \dots, c\}, n = 1, \dots, N$ for each matrix datum $M_n, n = 1, \dots, N$, the final output of the prediction model $f : \mathbb{R}^{c \times D} \rightarrow [0, 1]_1, \dots, [0, 1]_c$ is determined as the k -th index with the highest value among the c predicted values:

$$\hat{y}_n = \operatorname{argmax}_{k \in \{0, \dots, c\}} f_k(M_n), n = 1, \dots, N. \quad (15)$$

Hence, the n -th predicted value \hat{y}_n becomes the k -th index with the highest value among the c predicted values. The neural network prediction model f consists of L fully connected layers $\mathcal{F}_\ell = \sigma(\mathcal{F}_{\ell-1}\mathbf{w}_\ell + b_\ell), \ell = 1, \dots, L$. Where, $\mathcal{F}_0 = M_n, n = 1, \dots, N$ and $\mathbf{w}_1 \in \mathbb{R}^D, \mathbf{w}_\ell \in \mathbb{R}^h, b \in \mathbb{R}$ are the model parameters (h is the nodes of each layer), and $\sigma(\cdot)$ is defined as the nonlinear differentiable activation function for each layer. Thus, the neural network composed of L layers can be expressed as follows:

$$f(M_n) = \sigma(\mathcal{F}_L \mathbf{w}_{L+1} + b_{L+1}) \in \mathbb{R}^c. \quad (16)$$

Where the output hidden layer \mathbf{w}_{L+1} consists of dimensions of $D \times 1$ for c outputs. The softmax function [9] is used in the computation of the model's final layer $\mathcal{F}_L \mathbf{w}_{L+1} + b_{L+1}$, resulting in the sum $\sum_{k=0}^c f_k(M_n) = 1, n = 1, \dots, N$ of $f_1(M_n), \dots, f_c(M_n)$. Therefore, when the matrix data $M \in \mathbb{R}^{c \times D}$ is input into the prediction model, the output can be interpreted as probabilities corresponding to the c indices.

3.2 Surrogate loss

The evaluation functions of Eq.(5) and Eq.(13) of the prediction model are non-smooth step functions, making them unsuitable as loss functions for optimization via gradient methods [6]. Therefore, we consider a smoothed surrogate loss function that appropriately approximates these functions.

When the n -th data M_n is input and the y_n -th index value $f_{y_n}(M_n)$ is the largest among the c outputs from the proposed model, we can confirm that the function δ_{y_n, \hat{y}_n} or $\mathbb{I}_{y_n, \hat{y}_n}$ returns a score of 1. In such a case, let us consider a classifier where the ground truth y -th index result with respect to M_n , $f_y(M)$, is always 1. If the results for N sets of test data produce $f_{y_n}(M_n) = 1, n = 1, \dots, N$, then this prediction model becomes the optimal classifier. Therefore, instead of δ_{y_n, \hat{y}_n} and $\mathbb{I}_{y_n, \hat{y}_n}$ in Eq.(5, 13), one can choose a learning approach that estimates the parameters \mathbf{w} by maximizing $f_{y_n}(M_n)$ close to 1, i.e., $\arg\max_{\mathbf{w}} \frac{1}{N} \sum_{n=1}^N f_{y_n}(M_n)$. Furthermore, as objective functions are generally set as errors in machine learning, we minimize the transformed surrogate loss as follows:

$$\mathcal{L}_{sur} = \frac{1}{N} \sum_{n=1}^N (1 - f_{y_n}(M_n)). \quad (17)$$

4 Experiments

This chapter consisted of three total experiments. First, we discuss the differences between our approach and simple binary classification using synthetic data for the problem in Chapter 2.1. We then report the quantitative performance evaluation of the two approaches using intricately simulated $t\bar{t}b\bar{b}$ process data. Finally, we compare the effects of data reconstruction methods through public survival data experiments against seven state-of-the-art methods.

4.1 Synthetic data

This chapter implements the problem in Chapter 2.1, which serves as the motivation for our proposed method, using synthetic data. We then validate the effectiveness and utility of our approach through a comparison with the binary classification method of Eq.(2) and the proposed method of Eq.(17). The synthetic data experiments were designed to easily understand the structural features of this data while reproducing cases that are difficult to learn using a binary classifier. In contrast, our approach effectively learns and visually demonstrates the decision boundaries.

4.1.1 Details

The 2-dimensional data samples that follow the $t\bar{t}b\bar{b}$ event structure are synthetic data and are reproduced as follows. Let us consider N 2-dimensional signal data $\hat{\mathbf{x}}_n \sim \mathcal{N}(0, I), n = 1, \dots, N$ (where $I \in \mathbb{R}^{2 \times 2}$ is the identity matrix). The $(c - 1)$ sets of background data are sampled as $\hat{\mathbf{x}}_n + (-0.03, \epsilon_k) \in \mathbb{R}^2, k = 1, \dots, c - 1$ by shifting $\hat{\mathbf{x}}_n$ horizontally by -0.03 and injecting Gaussian noise ϵ_k vertically. Where,

$\epsilon_k \sim \mathcal{N}(0, 0.1)$, $k = 1, \dots, c - 1$ is defined. Eventually, one event matrix is

$$M_n = \begin{bmatrix} \hat{\mathbf{x}}_n \\ \mathbf{x}_1 = \hat{\mathbf{x}}_n + (-0.03, \epsilon_1) \\ \vdots \\ \mathbf{x}_{c-1} = \hat{\mathbf{x}}_n + (-0.03, \epsilon_{c-1}) \end{bmatrix} \in \mathbb{R}^{c \times 2}, n = 1, \dots, N, \quad (18)$$

In the experiment, c was set to 3, and $N = 1000$ event matrices M_1, \dots, M_{1000} were generated, as shown in Figure 4.1.1 (a) and (b). Of course, the row index of the signal data $\hat{\mathbf{x}}$ being identified is randomly set.

The simple binary linear classifier $f^{bce} : \mathbb{R}^2 \rightarrow [0, 1]$ using Eq.(2) for classifying data points in a 2-dimensional plane and the linear classifier $f^{sur} : \mathbb{R}^{c \times 2} \rightarrow [0, 1]_1, \dots, [0, 1]_c$ using Eq.(17) are defined as follows. First, the parameters of f^{bce} are defined as $\mathbf{w} \in \mathbb{R}^{2 \times 1}, b \in \mathbb{R}$, and the prediction for vector data $\mathbf{x} \in \mathbb{R}^2$ is defined as $f^{bce} = \sigma(\mathbf{x}\mathbf{w} + b) \in \mathbb{R}$. Here, the sigmoid function $\sigma(\cdot)$ is used to return a prediction value in $[0, 1]$. The prediction model f^{sur} that uses the softmax function to normalize the sum of c outcomes to 1 is the case when $L = 0$ in Eq.(16), and the same number of parameters $\mathbf{w} \in \mathbb{R}^{2 \times 1}, b \in \mathbb{R}$ are estimated as in binary classification. The prediction result becomes $f^{sur} = s(M\mathbf{w} + b) = \{f_1^{sur}, \dots, f_c^{sur}\} \in \mathbb{R}^c$.

4.1.2 Results

All data generated from synthetic data were constructed as shown in Figure 4.1.1 (a), with the left side being background data and right side being signal data. Therefore, the decision boundary of the prediction model should be formed vertically to classify all data correctly. However, the binary classification \mathcal{L}^{bce} minimizes only the prediction loss for overall signal data and background without considering structured features. Even when the boundary is vertical, the loss for the binary classifier remains high because the two entire clusters overlap, preventing optimal

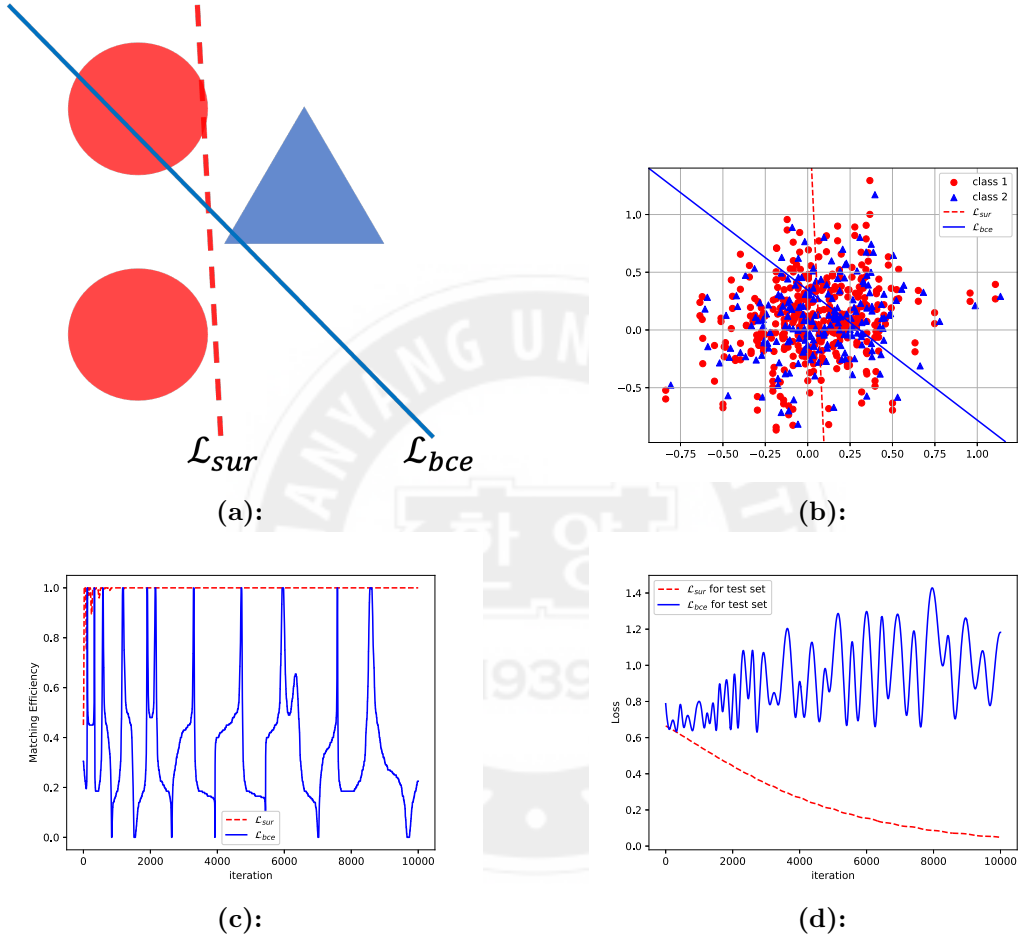


Figure 4.1: (a) A single test matrix of synthetic data and the decision boundaries for each method. (b) Decision boundaries of each linear classifier in the overall scattered test data. (c) and (d) Changes in matching efficiency and loss in Eq.(2) and Eq.(17) over 10,000 epochs, respectively.

convergence in a vertical direction. Regardless of how the boundary is drawn, the two classes remain substantially overlapped. This phenomenon can be observed in Figure 4.1.1 (c). As matching efficiency is not the objective of training, even when a peak is reached, learning continues to reduce the loss, altering the angle of the boundary. Meanwhile, the proposed method \mathcal{L}_{sur} focuses on a single matrix and classifies the classes well based on the horizontal axis values. Because the training objective directly maximizes matching efficiency, as seen in Figure 4.1.1 (c), matching efficiency converges to 1 while surrogate loss converges to 0.

4.2 Simulated $t\bar{t}b\bar{b}$ process data

This section presents a quantitative performance evaluation through $t\bar{t}b\bar{b}$ data that has been meticulously simulated, highlighting the performance difference between the proposed method and the binary classifier. In this experiment, the proposed method was evaluated by training a neural network model with L layers to minimize Eq.(17), which is identical to Eq.(16). The binary classifier also consists of a neural network with L layers. However, because it does not structure the data, the input data is set as $\mathcal{F}_0 = \mathbf{x}_n \in \mathbb{R}^D, n = 1, \dots, N$.

4.2.1 Details

The simulated data used for the experiments was generated identically to that in [14]. 31 million $t\bar{t}b\bar{b}$ samples were created through the MADGRAPH5 AMC@NLO (v2.6.6) [4] and PYTHIA (v8.240) [33]. Additionally, the W boson decayed through MadSpin [5], and the generated events were processed using the DELPHES package (v3.4.1) [18] designed for the CMS detector. These events were reconstructed through the particle-flow algorithm and particle flow jets were reconstructed using the anti- k_T algorithm [11] with a distance parameter of 0.5. Efficiencies for identifying electrons, muons, and jets were set using CMS data [18]. For more detailed

simulation information, refer to Chapters 2 and 3 of [14]. As a result, we generated 212,095 matchable events containing at least one additional b-jet pair. Each of these b-jet pairs comprises a feature vector of 78 random variables, as mentioned in Chapter 5 of [14].

We tuned the hyperparameters of the neural network on 161,676 event data using 40,419 validation events (the test event data used for reporting performance consists of 10,000 events: $161,676 + 40,419 + 10,000 = 212,095$). For the other construction of datasets, a total of 40,000 training data samples, along with 10,000 validation data samples, were employed for hyperparameter tuning. Subsequently, the training and validation set were integrated, and the performance was evaluated using 10,000 test data samples. Additionally, in any other construction with a smaller training data configuration, a neural network was trained using 10,000 data samples, while hyperparameters were tuned based on 2,500 validation data samples. The neural network, trained on the combined 12,500 data samples, was assessed for performance using 10,000 test data samples. Hyperparameter tuning occurred over a maximum of 1000 epochs with patience of 20 for early stopping. Early stopping is based on validation loss, and we explored the number of layers $\{2, 4, 6, 8\}$ in a fully connected neural network to find the one with the lowest validation loss. Furthermore, we also selected the number of nodes $\{25, 50, 75, 100, 125, 150\}$ in each layer that yields the lowest validation loss. This hyperparameter tuning was conducted for each random seed ranging from 0 to 9. We also tuned for each different size of the training data set to report the performance. Using the tuned hyperparameter values, we calculated the performance metrics on the test data (average performance across all 10 seeds is reported in each training data size).

Other hyperparameters that are not tuned include an Adam optimizer [24] with a learning rate of $1e-3$, a dropout rate [35] of 0.08, and the use of batch normalization [22]. All parameters for learning of the fully connected layers in the neural network were initialized using Glorot Initialization [20]. To train on a large volume of data, we used mini-batches sized according to the formula of total number of data/number of layers. We conducted all experiments using PyTorch [30] on an NVIDIA Quadro

RTX 8000 with 48GB GDDR6 memory, utilizing the CUDA [29], on Ubuntu [34].

4.2.2 Results

The Table 4.1’s second row represents varying amounts of training data, and the leftmost column distinguishes between the existing approach using different data structures and the proposed method. The table clearly illustrates that training with the proposed surrogate loss \mathcal{L}_{sur} outperforms the traditional approach using \mathcal{L}_{bce} , as demonstrated. This results confirms that learning from structured data with the proposed loss function is effective, as also evident in experiments with synthetic data. Furthermore, as the amount of training data increases, the performance difference can be observed to reach a maximum margin of 0.021. Conversely, when data is reduced, the performance gap decreases. Additionally, to ascertain statistical significance, we conducted t-tests on the results from 10 runs for each data configuration. The resulting p-value is $\{0.15e-01, 3.36e-06, 6.01e-10\}$ for training size $\{12,500, 50,000, 202,095\}$ indicating a statistically significant difference.

Table 4.1: Mean of Matching Efficiency (standard deviation) across all 10 trials in each training data size.

Loss	ME (std)	ME (std)	ME (std)
Training data size	202,095	50,000	12,500
\mathcal{L}_{bce}	0.61422 (0.0034)	0.61571 (0.0046)	0.60851 (0.0068)
\mathcal{L}_{sur}	0.63511 (0.0036)	0.62949 (0.0031)	0.61479 (0.004)

4.3 Survival data

In this chapter, we quantitatively evaluate the proposed method by comparing it with other approaches that have already performed well. In this study, the experiments evaluated performance by adding only the proposed data transformation method and loss function based on github code², which was made public in [25]. All

²https://github.com/Museau/Survival_Pytorch_EMD

experimental settings, including data preprocessing, hyperparameter tuning, and the neural network architecture used, were kept consistent. However, to achieve statistical significance, we report performance from 10-fold cross-validation instead of the initially set from previous research, 5-fold cross-validation. The loss functions and theoretical backgrounds of the other seven models compared can be confirmed through [37], an extension of [25].

4.3.1 Details

The experiments evaluated three survival datasets: SUPPORT2, AIDS3, and COLON DEATH³. The SUPPORT2 data, collected from the 1990 SUPPORT study, has the largest sample size among the three datasets. It also has the highest dimensionality at 98 and lowest rate of censored data. AIDS3 consists of Australian AIDS survival data and has the lowest dimensionality of covariates and highest rate of censored data. COLON DEATH is the smallest dataset, comprising survival analysis data for colorectal cancer treated with chemotherapy, and over half of the data is censored.

In our hyperparameters tuned via validation data, selecting the grid search to take a maximum of 1000 epochs with 20 patience.; learning rates [1e-4, 1e-3, 1e-2, 1e-1] and L2 Regularization [28] coefficient (λ) [1e-6, 1e-5, 1e-4, 1e-3, 1e-2]. The neural network consists of three layers of 100 nodes with activate function [27]. Furthermore, we used optimizer Adam [24] and normalization for mini-batch [22], drop-out[35] of 0.5 implemented by [30]. However, the drop-out used for input of the first layer is 0.2. The seven comparable models take a batch size of 512 in the SUPPORT2 data, 128 in the AIDS3, and 64 in the COLON DEATH data set. The proposed reconstruction method is training in 262144 mini-batch in SUPPORT2 data and set to 8192 in data in AIDS3 and 4096 in COLON DEATH data learning.

³<https://vincentarelbundock.github.io/Rdatasets/datasets.html>

Table 4.2: Table presents sample data count, right-censored data ratio, and random variable dimension from left to right.

Survival Data	N of data	$N(\%)$ of censored	N of features
SUPPORT2	9105	2904 (32.2)	98
AIDS3	3985	2223 (55.8)	19
COLON DEATH	929	477 (51.3)	48

4.3.2 Results

The evaluation of the classifier trained by the reconstruction method converts all the data into a matrix form. Then, it evaluates the prediction result Eq. (15) as a matrix c-index Eq. (13). To assess other methods, we calculate the original c-index Eq. (7).

The performance in Table 4.3 indicated in bold corresponds to achieving the best results in the respective datasets. The performance marked in bold at the bottom is the c-index average of all models. The Cox model and Cox Efron’s model are in the partial likelihood method, and the sigmoid, log-sigmoid, support vector machine, and ranking boost models belong to the ranking model. Finally, the classification approach includes two methods Wasserstein metric model and the proposed approach. And, we conduct all of the methods of test results for ten-fold cross-validations, as shown in tables in the Appendix: This section shows the t-test p-value of 10 trials and the total trial results. As a result, we can confirm that there is no statistically significant difference between the data reconstruction method and the seven methods believed to be state-of-the-art approaches in survival analysis.

Upon closer examination, Experimental results show that $\sigma(z)$ of the ranking method achieved the best on both data and that the classification method proposed in this paper performed best on SUPPORT2 data. Predictive models through the proposed classification method only perform best on specific data. In particular, the Wasserstein metric (WM) approach and the proposed method showed low performance on AIDS3 data. Furthermore, for AIDS3 data, ranking methods (Sigmoid et al.) showed the highest performance. This result indicates that a specific method

Table 4.3: Average of the c-index performance of other methods and our approach, and the standard deviation of the 10-fold validation results is bracketed.

Loss	SUPPORT2	AIDS3	COLON DEATH
Partial likelihood (Cox)	0.8487(0.0146)	0.5641(0.0233)	0.6428(0.0232)
Partial likelihood (Cox Efron's)	0.8495(0.0135)	0.5638(0.0269)	0.6471(0.0324)
Ranking (Sigmoid)	0.8545(0.0137)	0.5722 (0.0222)	0.6536 (0.0296)
Ranking (Log Sigmoid)	0.8538(0.0147)	0.5703(0.0251)	0.6496(0.0279)
Ranking (SVM)	0.8517(0.0135)	0.5604(0.027)	0.6453(0.0232)
Ranking (Boost)	0.8537(0.0146)	0.5714(0.0195)	0.6391(0.038)
Classification (WM)	0.8542(0.014)	0.5596(0.0336)	0.6475(0.0345)
Classification (Ours)	0.8548 (0.0135)	0.5555(0.0300)	0.6524(0.0247)
Mean of All Performance	0.8526(0.0022)	0.5647(0.0270)	0.647(0.004)

can be selected as a good strategy for AIDS3 data. Nonetheless, our classification approach is that the performance of other than AIDS3 data is above average and comparable to the highest-level methods.

5 Conclusion and Discussion

This paper proposed a surrogate loss function capable of directly maximizing evaluation metrics by generalizing two different data domains into a single problem. Unlike the previous approaches to the problem discussed in Chapter 1.1, which served as the motivation for the proposed method [14], this paper proposed a method that appropriately learns by considering the form of the data. Furthermore, we verified in Chapter 4.1 how our data restructuring method operates efficiently compared to simple binary classification and demonstrated its practical utility through Chapter 4.2. This study also shows how this method can be applied in survival analysis. Because the metrics used in survival analysis can be similarly defined as identifying a single data point from two data, as in Eq.(5), the proposed method was applicable. In this context, we proposed a matrix form of the concordance index and a corresponding surrogate loss function to optimize it. This loss function demonstrates competitive performance with other state-of-the-art methods, as shown through quantitative evaluations using publicly available survival data in Chapter 4.3.

This study effectively addressed a problem that can be defined as identifying a single data point from a collection comprising multiple data across two domains through a single approach. Such an approach could be potentially applied to problems like the U.S. presidential election, where one needs to identify a single data point from multiple groups. Additionally, as the identification problem can be interpreted as ranking a single data point highest, it has a close relationship with rank problems. It may be considered for application in recommendation models.

On the downside, applying the proposed method to data that fundamentally does not require grouping, as in Chapter 2.2, increases computational requirements for learning. As shown in Eq.(12), the number of restructured data points based on acceptable pairs \mathcal{A} could be redefined as up to $2|\mathcal{A}|_N$. Although modern computing capabilities may make learning feasible given the small amounts of survival analysis data typically available, the exponential increase in computational load when dealing

with large datasets remains a problem to be solved.



References

- [1] M. Aaboud, G. Aad, B. Abbott, B. Abeloos, D. Abhayasinghe, S. Abidi, O. AbouZeid, N. Abraham, H. Abramowicz, H. Abreu, et al. Observation of higgs boson production in association with a top quark pair at the lhc with the atlas detector. *Physics Letters B*, 784:173–191, 2018.
- [2] G. Aad, T. Abajyan, B. Abbott, J. Abdallah, S. A. Khalek, A. A. Abdelalim, R. Aben, B. Abi, M. Abolins, O. AbouZeid, et al. Observation of a new particle in the search for the standard model higgs boson with the atlas detector at the lhc. *Physics Letters B*, 716(1):1–29, 2012.
- [3] A. Alabdallah, M. Ohlsson, S. Pashami, and T. Rognvaldsson. The concordance index decomposition: a measure for a deeper understanding of survival prediction models. *arXiv preprint arXiv:2203.00144*, 2022.
- [4] J. Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, O. Mattelaer, H.-S. Shao, T. Stelzer, P. Torrielli, and M. Zaro. The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations. *Journal of High Energy Physics*, 2014(7):1–157, 2014.
- [5] P. Artoisenet, R. Frederix, O. Mattelaer, and R. Rietkerk. Automatic spin-entangled decays of heavy resonances in monte carlo simulations. *Journal of High Energy Physics*, 2013(3):1–19, 2013.
- [6] S. Ben-David, N. Eiron, and P. M. Long. On the difficulty of approximately maximizing agreements. *Journal of Computer and System Sciences*, 66(3):496–514, 2003.
- [7] C. M. Bishop and N. M. Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

- [8] A. R. Brentnall and J. Cuzick. Use of the concordance index for predictors of censored survival data. *Statistical methods in medical research*, 27(8):2359–2373, 2018.
- [9] J. Bridle. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. *Advances in neural information processing systems*, 2, 1989.
- [10] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pages 89–96, 2005.
- [11] M. Cacciari, G. P. Salam, and G. Soyez. The anti-kt jet clustering algorithm. *Journal of High Energy Physics*, 2008(04):063, 2008.
- [12] S. Chatrchyan, V. Khachatryan, A. M. Sirunyan, A. Tumasyan, W. Adam, E. Aguilo, T. Bergauer, M. Dragicevic, J. Erö, C. Fabjan, et al. Observation of a new boson at a mass of 125 gev with the cms experiment at the lhc. *Physics Letters B*, 716(1):30–61, 2012.
- [13] Y. Chen, Z. Jia, D. Mercola, X. Xie, et al. A gradient boosting algorithm for survival analysis via direct optimization of concordance index. *Computational and mathematical methods in medicine*, 2013, 2013.
- [14] J. Choi, T. J. Kim, J. Lim, J. Park, Y. Ryou, J. Song, and S. Yun. Identification of additional jets in the $t\bar{t}b\bar{b}$ events by using deep neural network. *Journal of the Korean Physical Society*, 77(12):1100–1106, 2020.
- [15] W. W. Cohen, R. E. Schapire, and Y. Singer. Learning to order things. *Advances in neural information processing systems*, 10, 1997.
- [16] C. collaboration et al. Measurement of $t\bar{t}$ production with additional jet activity, including b quark jets, in the dilepton decay channel using pp collisions at $\sqrt{s} = 8\text{TeV}$. 2016.

- [17] D. R. Cox. Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34(2):187–202, 1972.
- [18] J. De Favereau, C. Delaere, P. Demin, A. Giammanco, V. Lemaitre, A. Mertens, and M. Selvaggi. Delphes 3: a modular framework for fast simulation of a generic collider experiment. *Journal of High Energy Physics*, 2014(2):1–26, 2014.
- [19] J. Erdmann, T. Kallage, K. Kröninger, and O. Nackenhorst. From the bottom to the top—reconstruction of $t\bar{t}$ events with deep learning. *Journal of Instrumentation*, 14(11):P11015, 2019.
- [20] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.
- [21] 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.
- [22] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.
- [23] C. Jang, S.-K. Ko, J. Choi, J. Lim, Y.-K. Noh, and T. J. Kim. Learning to increase matching efficiency in identifying additional b-jets in the $t\bar{t}^- b\bar{b}^-$ process. *The European Physical Journal Plus*, 137(7):870, 2022.
- [24] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [25] M. Luck, T. Sylvain, J. P. Cohen, H. Cardinal, A. Lodi, and Y. Bengio. Learning to rank for censored survival data. *arXiv preprint arXiv:1806.01984*, 2018.
- [26] K. P. Murphy. *Probabilistic machine learning: an introduction*. MIT press, 2022.

- [27] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.
- [28] K. Nigam, J. Lafferty, and A. McCallum. Using maximum entropy for text classification. In *IJCAI-99 workshop on machine learning for information filtering*, volume 1, pages 61–67. Stockholom, Sweden, 1999.
- [29] NVIDIA, P. Vingelmann, and F. H. Fitzek. Cuda, release: 10.2.89, 2020. URL <https://developer.nvidia.com/cuda-toolkit>.
- [30] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.
- [31] G. Rodríguez. Chapter 7. survival analysis. lecture notes on generalized linear models. *Princeton University*, 2007. URL <https://grodri.github.io/glms/notes/>.
- [32] A. M. Sirunyan, A. Tumasyan, W. Adam, F. Ambroggi, E. Asilar, T. Bergauer, J. Brandstetter, M. Dragicevic, J. Erö, A. E. Del Valle, et al. Observation of $t\bar{t}H$ production. *Physical review letters*, 120(23):231801, 2018.
- [33] T. Sjöstrand, S. Ask, J. R. Christiansen, R. Corke, N. Desai, P. Ilten, S. Mrenna, S. Prestel, C. O. Rasmussen, and P. Z. Skands. An introduction to pythia 8.2. *Computer physics communications*, 191:159–177, 2015.
- [34] M. G. Sobell. *A practical guide to Ubuntu Linux*. Pearson Education, 2015.
- [35] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.
- [36] H. Steck, B. Krishnapuram, C. Dehing-Oberije, P. Lambin, and V. C. Raykar.

- On ranking in survival analysis: Bounds on the concordance index. *Advances in neural information processing systems*, 20, 2007.
- [37] T. Sylvain, M. Luck, J. Cohen, H. Cardinal, A. Lodi, and Y. Bengio. Exploring the wasserstein metric for time-to-event analysis. In *Survival Prediction- Algorithms, Challenges and Applications*, pages 194–206. PMLR, 2021.
- [38] L. Yan, D. Verbel, and O. Saidi. Predicting prostate cancer recurrence via maximizing the concordance index. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 479–485, 2004.



A Appendix

Table A.1: P-values by 10-trial t-test for each performance that compared the proposed methods and others.

Loss Type	Variant	SUPPORT2	AIDS3	COLON DEATH
Partial likelihood	Cox	0.3673	0.5038	0.4056
Partial likelihood	Cox Efron's	0.4163	0.544	0.6991
Ranking (Sigmoid)	$\sigma(z)$	0.9618	0.7868	0.9252
Ranking (Log Sigmoid)	Log-sigmoid	0.8769	0.2697	0.8208
Ranking (SVM)	$(z - 1)_+$	0.6257	0.1989	0.5351
Ranking (Boost)	$1 - (\exp - z)$	0.8692	0.7237	0.3888
Classification (WM)	$l = 1.5$	0.9262	0.1973	0.7322

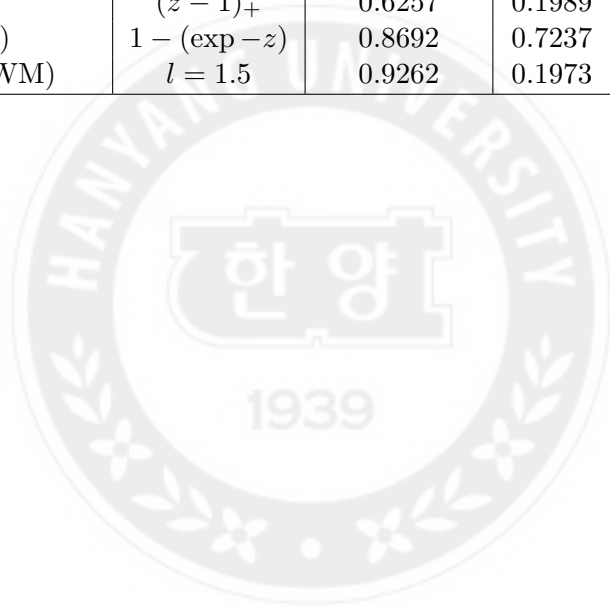


Table A.2: C-index results of 10-fold validation with SUPPORT2 data.

Loss Type	Variant	Mean(std)	0	1	2	3	4
Partial likelihood	Cox	0.8487(0.0146)	0.8287	0.8440	0.8328	0.8520	0.8710
Partial likelihood	Cox Efron's	0.8495(0.0135)	0.8279	0.8442	0.8347	0.8530	0.8703
Ranking (Sigmoid)	$\sigma(z)$	0.8545(0.0137)	0.8358	0.8467	0.8410	0.8572	0.8762
Ranking (Log Sigmoid)	Log-sigmoid	0.8538(0.0147)	0.8325	0.8504	0.8422	0.8588	0.8772
Ranking (SVM)	$(z - 1)_+$	0.8517(0.0135)	0.8300	0.8479	0.8398	0.8537	0.8731
Ranking (Boost)	$1 - (\exp - z)$	0.8537(0.0146)	0.8321	0.8466	0.8390	0.8562	0.8777
Classification (WM)	$l = 1.5$	0.8542(0.014)	0.8337	0.8462	0.8403	0.8563	0.8759
Classification (Ours)	Data Reconstruction	0.8548(0.0135)	0.8365	0.8471	0.8421	0.8609	0.8768
			5	6	7	8	9
			0.8763	0.8546	0.8455	0.8374	0.8443
			0.8731	0.8561	0.8452	0.8431	0.8473
			0.8808	0.8585	0.8512	0.8463	0.8510
			0.8813	0.8579	0.8485	0.8425	0.8460
			0.8756	0.8591	0.8459	0.8467	0.8445
			0.8801	0.8592	0.8511	0.8442	0.8508
			0.8803	0.8610	0.8522	0.8474	0.8485
			0.8789	0.8590	0.8536	0.8435	0.8495

Table A.3: C-index results of 10-fold validation with AIDS3 data.

Loss Type	Variant	Mean(std)	0	1	2	3	4
Partial likelihood	Cox	0.5641(0.0233)	0.5744	0.5328	0.5648	0.5451	0.5554
Partial likelihood	Cox Efron's	0.5638(0.0269)	0.5830	0.5386	0.5730	0.5447	0.5572
Ranking (Sigmoid)	$\sigma(z)$	0.5722(0.0222)	0.5955	0.5064	0.5922	0.5453	0.5685
Ranking (Log Sigmoid)	Log-sigmoid	0.5703(0.0251)	0.5958	0.5505	0.5851	0.5359	0.5565
Ranking (SVM)	$(z - 1)_+$	0.5604(0.027)	0.5961	0.5493	0.5662	0.5558	0.5601
Ranking (Boost)	$1 - (\exp - z)$	0.5714(0.0195)	0.5954	0.5563	0.5321	0.5489	0.5745
Classification (WM)	$l = 1.5$	0.5596(0.0336)	0.5947	0.5365	0.5793	0.5606	0.5631894
Classification (Ours)	Data Reconstruction	0.5555(0.0300)	0.5946	0.5458	0.5382	0.5192	0.5468
			5	6	7	8	9
			0.5672	0.5944	0.5876	0.5250	0.5946
			0.5588	0.6017	0.5901	0.5071	0.5839
			0.5752	0.5293	0.5865	0.5055	0.5918
			0.5686	0.6024	0.5805	0.5290	0.5992
			0.5626	0.6080	0.5932	0.5507	0.5720
			0.5409	0.5908	0.6044	0.5449	0.5158
			0.5575	0.6151	0.5933	0.5518	0.5697
			0.5592	0.6044	0.5776	0.5035	0.5656

Table A.4: C-index results of 10-fold validation with COLON DEATH data.

Loss Type	Variant	Mean(std)	0	1	2	3	4
Partial likelihood	Cox	0.6428(0.0232)	0.6497	0.6474	0.6533	0.6539	0.6480
Partial likelihood	Cox Efron's	0.6471(0.0324)	0.6762	0.6433	0.6437	0.6522	0.6511
Ranking (Sigmoid)	$\sigma(z)$	0.6536(0.0296)	0.6549	0.6474	0.6424	0.6525	0.6295
Ranking (Log Sigmoid)	Log-sigmoid	0.6496(0.0279)	0.6605	0.6294	0.6913	0.6305	0.6570
Ranking (SVM)	$(z - 1)_+$	0.6453(0.0232)	0.67405	0.6294	0.6631	0.6343	0.6301
Ranking (Boost)	$1 - (\exp - z)$	0.6391(0.038)	0.6115	0.6584	0.6195	0.6732	0.6103
Classification (WM)	$l = 1.5$	0.6475(0.0345)	0.6859	0.6442	0.6326	0.6197	0.6403
Classification (Ours)	Data Reconstruction	0.6524(0.0247)	0.6542	0.6412	0.6593	0.6532	0.6394
			5	6	7	8	9
			0.6298	0.6038	0.6881	0.6466	0.6068
			0.6206	0.6232	0.6232	0.7268	0.6100
			0.6282	0.6246	0.7144	0.7044	0.6376
			0.6291	0.6113	0.6593	0.7004	0.6263
			0.6045	0.6283	0.6777	0.6697	0.6413
			0.6321	0.6232	0.5948	0.7321	0.6354
			0.5963	0.6205	0.7101	0.6915	0.6335
			0.6263	0.6157	0.7	0.6898	0.6445

국문요지

평가 메트릭 직접 최대화 학습: Matching Efficiency와 Concordance Index를 중심으로

입자 물리학에서 힉스 입자의 비밀을 풀어내는 것은 매우 중요한 과제이다. 특히 $t\bar{t}b\bar{b}$ 과정(process)과 이와 관련된 문제들을 정확하게 이해하는 것은, $t\bar{t}H(b\bar{b})$ 과정과 연관되어 힉스 입자의 특성을 밝히는 것과 밀접한 관련이 있다. 따라서 $t\bar{t}b\bar{b}$ 과정에서 최상위 쿼크(top quark) 붕괴로 생성되는 b-jets들 중, 글루온 분열(gluon splitting)로 생성된 additional b-jets을 식별하는 것은 중요한 선행 연구이다. 이 문제를 위한 간단한 물리학적 식별 규칙이 없기 때문에, 선행 연구에서는 시뮬레이션 데이터를 기반으로 한 머신러닝 접근법이 사용되었다. 기존 접근법은 모든 b-jets과 additional b-jets에 대해 단순 이진 분류를 수행하여, 각 $t\bar{t}b\bar{b}$ 이벤트 단위로 구조화하지 않고 학습한다. 하지만 예측 모델의 성능 메트릭 **Matching Efficiency**는 성능을 평가할 때, 구조화된 $t\bar{t}b\bar{b}$ 데이터들에 각각 존재하는 additional b-jets을 식별해내는 능력을 평가한다. 따라서, 본 논문은 이 평가 함수의 목적을 감안하여 모든 데이터를 $t\bar{t}b\bar{b}$ 이벤트 구조로 묶어서 재구성하는 방법을 제안한다. 또한, 평가 지표를 직접 최대화하여 기존 접근 방식에 비해 향상된 성능을 달성하기 위한 대리 손실(surrogate loss) 함수를 제안한다. 기존 방법과 제안한 방법의 학습 방식을 비교하기 위해 두 가지 손실 함수를 통한 합성 데이터(synthetic data) 실험을 진행한다. 이 실험은 구조적 특성이 보존된 합성 데이터를 생성하여 기존의 이진 분류기에서는 학습이 불가능하지만, 제안한 방법에서는 효과적으로 학습할 수 있음을 보인다. 제안한 방법의 성능을 이진 분류의 성능과 비교하기 위해 정교하게 시뮬레이션된 $t\bar{t}b\bar{b}$ 과정 데이터를 사용하여 성능을 정량적으로 평가한다.

그 다음, 우리는 제안한 데이터 재구성 방법과 대리 손실 함수가 생존 분석에서도

적용될 수 있음을 주목한다. 생존 분석에서 모델 성능 평가로 사용되는 **Concordance Index** (c-index) 메트릭은 시험 데이터를 생존 시간이 비교 가능한 두 개의 데이터로 묶어서 구성하며, 그 쌍 데이터에서 생존 시간이 큰 데이터를 식별하는 능력을 평가하는 함수로 정의가 가능하다. 이러한 메트릭의 데이터 구성 방식은 앞서 언급된 Matching Efficiency와 유사하며, 이는 생존 분석 문제를 여러 데이터에서 하나의 데이터를 식별하는 문제로 재정의 할 수 있음을 내포한다. 따라서, 우리는 전체 데이터를 c-index에서 사용되는 데이터 쌍 형태로 재구성하고, 제안된 대리 손실 함수를 통해 이 메트릭을 직접 최대화 하는 방법을 제안한다. 데이터 재구성 접근 방법은 생존 분석에서 자주 발생하고, 불완전하여 일반적인 방법으로는 학습 데이터로 활용할 수 없는 우중도 절단(right-censored) 데이터를 활용할 수 있다. 우중도절단 데이터는 c-index에서 데이터의 순서화 평가에서만 비교 대상으로 사용되기 때문에, 학습용 데이터 집합 또한 메트릭에 사용되는 형태로 재구성 하면, 중도절단 데이터를 학습에 활용할 수 있기 때문이다. 제안된 방법은 생존 분석에서 입증된 Partial Likelihood, Rank Methods 그리고 Wasserstein Metric과 같은 최고 수준의 방법들과 정량적인 평가를 통해 경쟁적인 성능을 보이며, 통계적으로 유의미한 차이가 없음을 보인다.

감사의 글

많은 분들의 도움을 받아 석사 학위 논문을 마무리하고 감사의 글을 작성하는 순간이 되니 가슴이 벅차옵니다. 학위 과정 동안 정말 많은 분들의 도움을 받았기에 이렇게 글으로라도 감사의 마음을 전하고자 합니다.

가장 먼저, 첫 번째 제자로 저를 받아주시고 스스로 깨우칠 수 있는 지도를 해주신 노영균 교수님께 깊이 감사드립니다. 교수님께서 열정적으로 지도해주시고 조언해주셔서 연구의 참된 흥미를 느끼게 되었습니다. 교수님과 했던 chalk-talk들은 사진으로 간직하여 자주 찾아보며 여전히 많은 도움을 받습니다. 교수님께서 해주셨던 지도를 가슴 깊이 새겨서 독립된 연구자로 거듭나도록 항상 노력하겠습니다. 학자로서 존경하고 스승으로 감사하다는 말씀을 전합니다.

귀한 시간을 내주셔서 심사를 해주신 채동규 교수님께 감사드립니다. 바쁘신 와중에 심사위원장까지 맡아주시고 부족한 발표를 들어주셔서 거듭 감사의 말씀 전합니다. 더불어 논문 심사위원을 맡아주신 장청재 박사님께도 감사드립니다. 박사님과 함께 저널 논문을 작성해본 경험은 정말 소중한 기회였고 느낀 점이 많았던 시간이었습니다. 부족한 저를 이해 해주시고 아낌없는 조언을 해주셔서 감사드립니다. 논문 작성을 위해서 함께 실험할 때 말씀 해주셨던 ‘실수를 줄이기 위해 노력하는 몇 가지 팁’에 대한 조언은 항상 저에게 큰 도움이 됩니다.

함께 AAIS 연구실 생활을 했던 호준이, 진선님, 유빈님, 종현님에게도 감사드립니다. 모두 저보다 뛰어난 분들이어서 배운 점이 정말 많았습니다. 저 또한 도움을 많이 드렸어야 했는데 그러지 못한것 같아서 아쉬움이 많이 남습니다. 그리고 김봉재 교수님께 감사드립니다. 교수님 연구실에서 학부연구생 활동을 했던 경험은 저에게 큰 행운이었고 연구의 흥미를 처음으로 느끼게 해준 소중한 시간이었습니다. 더불어 학부 때부터 대학원 진학의 뜻을 품고 각자의 연구 방향을 찾아서 흠어졌던 보선이형, 현태형, 기철이에게 감사합니다. 형님들이 없었더라면 학업에 흥미를 느끼지 못했을거라고 생각합니다.

대학원 생활동안 원룸에서 동거동락하며, 각자의 연구를 하며 꿈을 꾸고 서로를 의지했던 반석이에게 감사드립니다. 항상 자신의 일처럼 걱정해주고 기뻐해줬기에 끝

까지 도전할 수 있었습니다. 학위 과정 동안 제 스스로에 대한 의심을 없애주고 용기를 주었던 세월이에게도 감사합니다. 먼저 사회로 나가서 풍부한 경험을 통해 해주었던 조언들은 저에게 깊은 도움이 되었습니다. 힘든 시기일 때마다 다양한 도움을 줬던 친구들 모두에게도 감사하다는 말씀을 드리고 싶습니다. 모두의 도움 덕에 의미있는 결실을 맺었습니다.

끝으로 석사 학위 과정동안 항상 저를 응원해준 가족들에게 감사의 말씀을 전합니다. 먼저 저와 달리 박학하고 다식한 누나에게 감사드립니다. 대학교의 입학부터 대학원의 졸업까지 누나의 지원과 도움이 없었더라면 해내지 못할 일들이 많았습니다. 그리고 천방지축 말 안듣던 아들을 끝까지 믿어주고 기다려주었던 어머님께 감사합니다. 홀로 부모의 몫을 다하시고 물심양면 지원해주신 덕분에 학사를 지나 석사까지 좋은 사람들을 많이 만났습니다. 모든게 다 어머님의 덕입니다. 부모님으로서 감사드리고 제 인생의 큰 스승으로 항상 존경합니다.

많은 분들에게 도움을 받았기에 글이 길어졌지만, 미처 감사의 글에 적지 못한 많은 분들에게도 감사하다는 말씀을 전합니다. 앞으로는 제가 도움을 드릴수 있는 사람이 되겠습니다.

고상균

Declaration of Ethical Conduct in Research

I, as a graduate student of Hanyang University, hereby declare that I have abided by the following Code of Research Ethics while writing this dissertation thesis, during my degree program.

"First, I have strived to be honest in my conduct, to produce valid and reliable research conforming with the guidance of my thesis supervisor, and I affirm that my thesis contains honest, fair and reasonable conclusions based on my own careful research under the guidance of my thesis supervisor.

Second, I have not committed any acts that may discredit or damage the credibility of my research. These include, but are not limited to : falsification, distortion of research findings or plagiarism.

Third, I need to go through with Copykiller Program(Internet-based Plagiarism-prevention service) before submitting a thesis."

NOVEMBER 07, 2023

Degree : Master

Department : DEPARTMENT OF COMPUTER SCIENCE

Thesis Supervisor : Noh, Yung-Kyun

Name : KO SANGKYUN


(Signature)

연구 윤리 서약서

본인은 한양대학교 대학원생으로서 이 학위논문 작성 과정에서 다음과 같이 연구 윤리의 기본 원칙을 준수하였음을 서약합니다.

첫째, 지도교수의 지도를 받아 정직하고 엄정한 연구를 수행하여 학위논문을 작성한다.

둘째, 논문 작성시 위조, 변조, 표절 등 학문적 진실성을 훼손하는 어떤 연구 부정행위도 하지 않는다.

셋째, 논문 작성시 논문유사도 검증시스템 "카피킬러"등을 거쳐야 한다.

2023년11월07일

학위명 : 석사

학과 : 컴퓨터·소프트웨어학과

지도교수 : 노영균

성명 : 고상균



한 양 대 학 교 대 학 원 장 귀 하