2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Learning Using Privileged Information (LUPI) for Modeling Survival Data

Han-Tai Shiao and Vladimir Cherkassky

Abstract— **Survival data is common in medical applications. The challenge in applying predictive data-analytic methods to survival data is in the treatment of censored observations, since the survival times for these observations are unknown. This paper presents formalization of the analysis of survival data as a binary classification problem. For this binary classification setting, we propose a strategy for encoding censored data, leading to the SVM+/LUPI formulations. Further, we present empirical comparison of the new method and the classical Cox modeling approach for predictive modeling of survival data. These comparisons suggest that for data sets with large amount of censored data, the proposed method consistently yields better predictive performance than classical statistical modeling.**

I. Introduction

A significant proportion of the medical data is a collection of time-to-event observations. Classical examples are the time from birth to cancer diagnosis, from disease onset to death, and from entry to a study to relapse. All these times are generally known as the *survival time*, even when the endpoint is something different from death. Methods for survival analysis developed in classical statistics have been used to model such data. Survival analysis focuses on the time elapsed from an initiating event to an event, or endpoint, of interest [1]. The models of classical survival analysis describe the occurrence of the event by means of survival curves and hazard rates and analyze the dependence (of this event) on covariates by means of regression [1]. One of the most popular survival curve estimation is the Cox modeling approach based on the proportional hazards model.

Most statistical methods aim to build a model that relates explanatory variables and the occurrences of the event. The field of machine learning is also targeting the same or similar goals. Learning is the process of estimating an unknown dependency between system's inputs and its output, based on a limited number of observations [2]. However, the machine learning techniques have not been widely used for survival analysis for two major reasons:

1) First, the survival time is not necessarily observed in all samples. For example, patients might not experience the occurrence of event (death or relapse) during the study, or they were lost to follow-up. Hence, the survival time is incomplete and only known "up-to-a-point." This is termed censoring in biostatistics, which is different from the notion of "missing data" in machine learning.

Han-Tai Shiao and Vladimir Cherkassky are with the Department of Electrical and Computer Engineering, University of Minnesota, Twin Cities, Minneapolis, Minnesota, U.S.A. (email: {shiao003, cherk001}@umn.edu).

2) The second reason is methodological. Machine learning techniques are usually developed and applied under predictive setting, where the main goal is the prediction accuracy for future (or test) samples. In contrast, classical statistical methods aim at estimating the true probabilistic model of the available data. So the prediction accuracy is just one of the several performance indices. The methodological assumption is that if an estimated model is "correct," then it should yield good predictions. Therefore, practitioners applying statistical methods often do not clearly differentiate between training (model estimation) and prediction (test) stages.

There have been many attempts to apply machine learning methods for modeling the survival data. Next we briefly comment on several studies applying Support Vector Machine (SVM) technology to survival data [3]–[5]. Most of these efforts formalize the problem under the regression setting. Specifically, the SVM regression was used to estimate a model that predicts the survival time. However, formalization using regression setting is intrinsically more difficult than classification. Further, practitioners generally use the modeling outputs as a reference and they are usually concerned with the status of a patient at a given time, such as six-month after surgery or two-year post-transplant.

Survival data sets represent a typical example of noisy high-dimensional biomedical data that describes complex phenomenon. Successful data-analytic modeling of such data sets requires development and/or creative application of new methodological approaches which represent an advance from simpler application of existing statistical or machine learning tools.

Most statistical and machine learning methods for modeling high-dimensional data focus on improvements to existing *inductive methods* (*i.e.*, Linear discriminant analysis, multi-layer perceptron neural networks, SVM) that try to incorporate a priori knowledge about the good models (*i.e.*, via specially designed kernels for SVM methods). Similarly, statistical methods focus on selecting a small number of informative inputs and their nonlinear combinations (selected by model building via a sequential process of progressive model refinement [6]). These approaches, however, are fundamentally constrained by the inductive learning setting itself. In contrast, *non-standard* learning methodologies focus on the most appropriate direct formulation of the learning problem. It can be argued that most recent advances in statistical learning (*i.e.*, transduction, semi-supervised learning, multi-task learning, *etc*.) reflect improved understanding of the learning problem setting [2], [7], [8].

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

This paper assumes a predictive setting, which is appropriate for many applications, and aims to develop new data-analytic methodology for predictive modeling of survival data. Under this predictive setting, the survival time is known for training data, but it is not available during the prediction (or testing) stage. Thus, modifications are required for applying existing machine learning approaches to survival data analysis. In this paper, we propose using a special classification formulation that addresses the issues of incomplete information in the survival time. Instead of predicting the survival time, we try to estimate a model that predicts a subject's status at a time point of interest.

This paper is organized as follows. Section II introduces necessary backgrounds on machine learning (LUPI) and on survival data analysis. Section III describes the proposed LUPI-based approach for survival analysis. The computational implementation of SVM+/LUPI is outlined in Section IV. Empirical comparisons for several synthetic and real-life data sets are presented in Section V and VI. Finally, the conclusions and discussion are given in Section VII.

II. Background

A. SVM+/LUPI

Learning Using Privileged Information (LUPI) [7], [9] is a general methodology for utilizing additional (privileged) information about training data (often available in our data-rich world). This information cannot be utilized by most standard supervised learning methods developed in statistics and machine learning, all of which assume standard inductive learning setting. Effective utilization of this privileged information during training often results in improved generalization [9].

Under the LUPI setting, the training data are a set of triplets

$$(\mathbf{x}_i, \mathbf{x}_i^*, y_i), \quad i = 1, \ldots, n \tag{1}$$

where $\mathbf{x}_i \in \mathbf{R}^d$, $\mathbf{x}_i^* \in \mathbf{R}^k$, and $y_i \in \{-1, +1\}$. The $(\mathbf{x}, y)$ is the 'usual' labeled training data and $(\mathbf{x}^*)$ denotes the additional *privileged* information available only for training data. This additional privileged information has two common properties:

- it is available only for training samples, and not known for test samples;
- it has an informative value for estimating a predictive model $\hat{y} = f(\mathbf{x})$.

These properties suggest another useful interpretation of the privileged information: it can be viewed as additional feedback from an expert teacher, provided during learning [7]. This feedback, or privileged information, is provided in a new feature space $\mathbf{x}^*$. In order to be useful, this privileged information should be relevant to errors made by a predictive model in the input (or decision) space $\mathbf{x}$.

According to Vapnik-Chervonenkis theory (VC theory), this new LUPI setting implements Structural Risk Minimization (SRM) approach via the construction of a new SRM structure on the training set. This task may appear similar to the development of new structures for non-standard learning formulations, where the new structures incorporate additional constraints, such as a large margin for test samples for transduction, or a large number of contradictions for Universum SVM [2], [7], [8]. The difference is that in earlier non-standard SVM-based formulations the appropriate structures have been defined in the same feature space ($\mathbf{x}$-space). In contrast, under LUPI setting, additional privileged information is specified in a different feature space, but this information is related to errors in the input feature space. Recently, a new technology called SVM+ has been developed for learning under LUPI setting [9]. This approach performs *learning in two different spaces*, as shown in Figure 1:

- *Decision* space $\mathcal{Z}$ (via the mapping $\Phi(\mathbf{x}) : \mathbf{x} \mapsto \mathbf{z}$), where the decision rule $\hat{y} = f(\mathbf{z})$ needs to be estimated. This is the same feature space as used in standard SVM;
- *Correcting* space $\mathcal{Z}^*$ (via the mapping $\Phi^*(\mathbf{x}) : \mathbf{x} \mapsto \mathbf{z}^*$), which reflects the privileged information about the training data. This information is encoded in the form of additional constraints on the training errors (*e.g.*, slack variables) in the decision space.

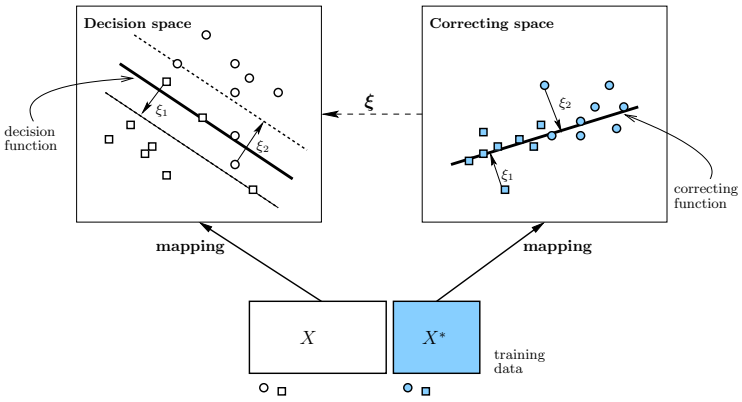

Fig. 1. SVM+ maps the training data into the decision space and the correcting space. Slack variables in the decision space are represented by the correcting function in the correcting space.

The decision and correcting spaces can use different kernels. The final performance of SVM+ models depends on the quality of the privileged information. Technically, the SVM+ approach estimates a decision function $(\mathbf{w}\cdot\mathbf{z})+b$ by using the correcting function $\xi(\mathbf{z}^*) = (\mathbf{w}^*\cdot\mathbf{z}^*)+b^* \geq 0$ as an additional constraint on the training errors (or slack variables) in the decision space. The SVM+ classifier is estimated from the training data in (1) by solving the following optimization problem:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + \frac{\gamma}{2}\|\mathbf{w}^*\|^2 + C\sum_{i=1}^{n}\xi_i \\ \text{subject to} \quad & \boldsymbol{\xi} \succeq 0 \\ & y_i((\mathbf{w}\cdot\mathbf{z}_i)+b) \geq 1-\xi_i, \quad i = 1, \ldots, n \\ & \xi_i = (\mathbf{w}^*\cdot\mathbf{z}_i^*)+b^*, \quad i = 1, \ldots, n \end{aligned} \tag{2}$$

with $\mathbf{w} \in \mathbf{R}^d$, $b \in \mathbf{R}$, $\mathbf{w}^* \in \mathbf{R}^k$, and $b^* \in \mathbf{R}$ as the variables. The symbol $\succeq$ denotes componentwise inequality.

Privileged information $\mathbf{x}^*$ often appears in modern complex clinical data sets. This may be a patient's medical history after

diagnosis or medical procedure. Clearly, this information is available in historical databases, but it cannot be included in the predictive model which use only patient's characteristics $\mathbf{x}$ known at the time when diagnosis/medical procedure is performed.

B. Survival Data Analysis

This section provides general background description of survival data analysis and its terminology. The survival data (or failure time data) are obtained by observing individuals from a certain initial time to either the occurrence of a predefined event or the end of the study. The predefined event is often the failure of a subject or the relapse of a disease. The major difference between survival data and other types of numerical data is the time to the event occurring is not necessarily observed in all individuals [1].

A common feature of these data sets is the possibility of *censored observations*. Censored data arise when an individual's life length is known to occur only during a certain period of time. Possible censoring schemes are *right censoring*, when all that is known is that the individual is still alive at a given time, and *left censoring* when all that is known is that the individual has experienced the event of interest prior to the start of the study, or *interval censoring*, where the only information is that the event occurs within some intervals. In this paper, we only consider the right censoring scheme.

The graphical representation of the survival data for a hypothetical study with six subjects is shown in Figure 2. In this study, subject 2 and 6 experienced the event of interest prior to the end of the study and they are called the *exact* observations. On the contrary, no events occur to subject 1, 3, and 5 before the end of the study. These subjects, who might experience the event after the end of the study, are only known to be alive at the end of the study. Subject 4 was included in the study for some time but further observation cannot be obtained. The data for subject 1, 3, 4, and 5 are called *censored* (right-censored) observations. Thus, for the censored observations, it is known that the survival time is greater than a certain value, but it is not known by how much.

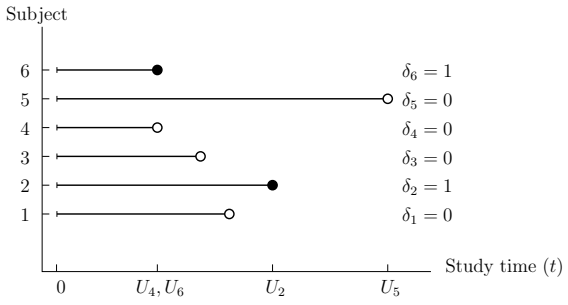

Fig. 2. Example of survival data in a study-time scale. The exact observations are indicated by solid dots, and the censored observations by hollow dots.

Let T denote the event time, such as death or lifetime; C denote the censoring time, *e.g.*, the end of study or loss to follow-up. The T's are assumed to be independent and identically distributed with probability density function $\varphi(t)$ and survival function $S(t)$. For right censoring scheme, we only know $T_i > C_i$ with observed C_i. Then the survival data can be represented by pairs of random variables (U_i, δ_i), $i = 1, \ldots, n$. The δ_i indicates whether the observed survival time U_i corresponds to an event ($\delta_i = 1$) or is censored ($\delta_i = 0$). The U_i is equal to T_i if the lifetime or event is observed, and to C_i if it is censored. Mathematically, U_i and δ_i are defined as

$$U_i = \min(T_i, C_i), \tag{3}$$

$$\delta_i = I(T_i \leq C_i) = \begin{cases} 0, & \text{for censored observation,} \\ 1, & \text{for exact observation.} \end{cases} \tag{4}$$

In Figure 2, subject 4 and 6 have the same observed survival time ($U_4 = U_6$), but their censoring indicators are different ($\delta_4 = 0$, $\delta_6 = 1$). Hence, in the survival analysis, we are given a set of data, $(\mathbf{x}_i, U_i, \delta_i)$, $i = 1, \ldots, n$, where $\mathbf{x}_i \in \mathbf{R}^d$, $U_i \in \mathbf{R}_+$ and $\delta_i \in \{0, 1\}$. The symbol $\mathbf{R}_+$ denotes non-negative real numbers. In contrast, under supervised learning setting, we are given a set of training data, $(\mathbf{x}_i, y_i)$, $i = 1, \ldots, n$, where $\mathbf{x}_i \in \mathbf{R}^d$ and $y_i \in \mathbf{R}$. The target values y_i's can be real-valued such as in standard regression, or binary class labels in classification.

Classical statistical approach for modeling survival data aims at estimating the survival function $S(t)$, which is the probability that the time of death is greater than a certain time t, or $\Pr(T > t)$. More generally, the goal is to estimate $S(t \mid \mathbf{x})$, or survival function conditioned on subject's characteristics, denoted as feature vector $\mathbf{x}$. Assuming that the probabilistic model $S(t \mid \mathbf{x})$ is known, or can be accurately estimated from the available data, this model provides complete statistical characterization of the data. In particular, it can be used for prediction and for explanation (*i.e.*, identifying input features that are strongly associated with an outcome, such as death).

III. Predictive Modeling of Survival Data via LUPI

In many applications, the goal is to estimate (or predict) survival at a certain pre-specified time point τ. Such time point, for example, could be the survival of cancer patients two years after initial diagnosis, or the survival status of patients one year after the bone marrow transplant procedure. Generally, τ can be about half of the maximum observed survival time. Next we describe a possible formalization of this problem under predictive setting, leading to a binary classification formulation.

Given the training survival data, $(\mathbf{x}_i, U_i, \delta_i, y_i)$, $i = 1, \ldots, n$, where $\mathbf{x}_i \in \mathbf{R}^d$, $U_i \in \mathbf{R}_+$, $\delta_i \in \{0, 1\}$, and $y_i \in \{-1, +1\}$, estimate a classification model $f(\mathbf{x})$ that predicts a subject's status at a pre-specified time τ based on the input (or covariates) $\mathbf{x}$. In the survival data, the status of subject i at time τ is a binary class label through the following encoding:

$$y_i = \begin{cases} +1, & \text{if } U_i < \tau, \\ -1, & \text{if } U_i \geq \tau. \end{cases} \tag{5}$$

where U_i is the observed survival time and δ_i is the corresponding event indicator. Note that U_i and δ_i are only available for training, not for prediction (or testing stage). So the challenge of predictive modeling is to develop novel classification formulations that incorporate the time information (U_i) and uncertain nature of the censored data.

In the hypothetical study shown in Figure 3, suppose a subject's status is given by (5), then there is no ambiguity in the statuses of subject 2 and 6. Likewise, the survival status of subject 5 is known, even though the observation is censored. However, the survival statuses of subjects 1, 3, and 4 are indeterminate since the observed survival times are shorter than τ.

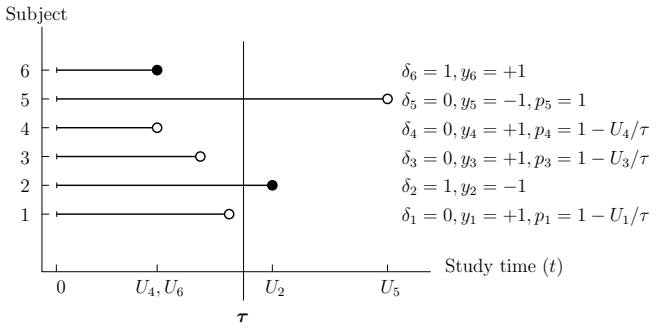

Fig. 3. Example of survival data under the predictive problem setting. The goal is to find a model $f(\mathbf{x})$ that predicts the subjects' statuses at time τ.

There are two simplistic ways to handle the censored data into standard classification formulation:

- Treat the censoring time as the actual event time, *i.e.*, replace T_i with C_i. This approach underestimates the actual event time because $T_i > C_i$.
- Discard the censored data and estimate a binary classifier using only exact observations. This approach is used in empirical comparisons presented later in Section V and VI (under the name standard SVM, or sSVM). It may yield sub-optimal models, as we ignore the information available in censored data.

This paper proposes a different strategy for incorporating censored data which leads to the SVM+/LUPI classifier. We assign a certainty measure p_i to reflect and quantify the uncertain nature of the censored data. One simple rule is to set the certainty of a subject being alive/dead at time τ inversely proportional to the (known) survival time, as indicated in Figure 3. That is, $p_i = (\tau - U_i)/\tau = 1 - U_i/\tau$.

The idea is that if U_i is small, it is more likely subject i will not survive at time τ. Or, subject i is dead at time τ with high certainty. On the other hand, if U_i is very close to τ, subject i will be alive/dead at time τ with low certainty. Therefore, the survival data $(\mathbf{x}_i, U_i, \delta_i, y_i)$, $i = 1, \ldots, n$, can be translated into $(\mathbf{x}_i, \tau - U_i, p_i, y_i)$, $i = 1, \ldots, n$. Further, the censoring information (available/known for training data, but not for test data) can be regarded as the privileged information under the LUPI paradigm (2), as follows:

The available survival data $(\mathbf{x}_i, \tau - U_i, p_i, y_i)$ can be represented as $(\mathbf{x}_i, \mathbf{x}_i^*, y_i)$, where $\mathbf{x}_i^* = (\tau - U_i, p_i)$ is the privileged information. Then the problem of survival analysis can be formalized and modeled using the SVM+/LUPI paradigm.

IV. Computational Implementation of LUPI

LUPI model selection is very difficult due to the fact that the kernelized version of SVM+ binary classifier has four tuning parameters. Hence, the computationally efficient solution of LUPI optimization becomes critical.

The process of training of standard SVM (or SVM+) involves solving a large Quadratic Programming (QP) problem. The computational complexity of solving the QP problem in SVM training grows as $\mathcal{O}(n^3)$, where n is the sample size [10]. To overcome these computational problems, many existing SVM implementations use Platt's Sequential Minimal Optimization (SMO) for training [11]. SMO is a fast iterative algorithm that breaks large QP problem into many QP subproblems of the smallest possible size (with only two variables), which can be solved analytically. This approach was implemented in the LIBSVM package (for standard SVM) and made this package popular in the machine learning community. Recently, a generalized SMO was developed for SVM+ [12]. However, this implementation is proprietary and not available in public domain.

Our initial implementations of SVM+ used a general-purpose convex optimization package CVX [13]. However, the scalability is an issue of using CVX as the solver of the QP problem. It takes more than 20 minutes to find the solution of the QP problem when the training size exceeds 1000 samples. Thus, current model selection strategies become impractical and infeasible. Notably, most recent academic papers seem to use general-purpose optimization for their LUPI implementations, as they show empirical comparisons only for small training sets (about 200 to 400 samples), and they do not address/describe the challenging issues of model selection. A typical quote from [14]: "On the data set of this size (a few thousand) we found it infeasible to run experiments using SVM+."

We opted to implement SVM+ using the *quadprog* package in Matlab Optimization Toolbox. The *quadprog* package was designed specifically for solving the QP problems, rather than general convex optimization problems. Our implementation involves the selection of the optimization option and also the stopping criterion (tolerance) optimally tuned for our LUPI models. Our experiments suggest that the *quadprog* implementation of SVM+ is capable of handling training data sets of size 1K-5K samples. That is, solving SVM+ optimization problem (for 1K-5K training samples) takes 2-12 seconds on a typical PC.

V. Empirical Comparisons for Synthetic Data

The empirical comparisons between the classical Cox regression and the proposed LUPI-based approach for modeling survival data are presented in this section. In these comparisons, the Cox modeling approach based on the proportional

hazards model is used under the predictive setting as follows. Once a survival function $S(t \mid \mathbf{x})$ is estimated (from training data), it is used for prediction via simple thresholding rule:

$$y_i = \begin{cases} +1, & \text{if } S(t \mid \mathbf{x}_i) < r, \\ -1, & \text{if } S(t \mid \mathbf{x}_i) \geq r, \end{cases} \tag{6}$$

where the threshold value r reflects the misclassification costs given *a priori*. All comparisons presented in Sections V and VI assume equal misclassification costs. So the threshold level is set to $r = 0.5$. Our implementation of the LUPI-based survival analysis model involves additional simplifications:

- For LUPI, the non-linearity is modeled only in the correcting space [15]. That is, in all experiments the decision space uses linear parametrization, and the correcting space is implemented via non-linear (RBF) kernels.
- For the standard SVM (sSVM), the survival times and event indicators are ignored. Both linear and non-linear mappings are investigated.

Consequently, sSVM with RBF kernel has two tuning parameters, C and σ (RBF width parameter), whereas LUPI has three tuning parameters, C, γ, and σ. Furthermore, sSVM with linear kernel has one tuning parameter (C). In contrast, there is no tunable parameter in the Cox modeling approach.

Empirical comparisons for the synthetic data are designed to understand relative advantages and limitations of SVM-based methods for modeling the survival data sets with various statistical characteristics, such as the number of training samples, the noise in the observed survival times, and the proportion of censoring. First, we consider the synthetic data set generated as follows [16]:

- Set the number of input features d to 30.
- Generate $\mathbf{x} \in \mathbf{R}^d$ with each element x_i being a random number uniformly distributed within $[-1, 1]$.
- For the coefficient vector

$$\boldsymbol{\beta} = [1, 1, 2, 3, 3, 1, 1, 1, 1, 0, 2, 0, 2, 2, 0, \\ 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],$$

  generate the event time T_i via exponential distribution $\mathrm{Exp}((\boldsymbol{\beta} \cdot \mathbf{x}) + 2)$. The Gaussian noise $\nu \sim \mathcal{N}(0, 0.2)$ is also added to the event time T_i. Generate the censoring time C_i via exponential distribution $\mathrm{Exp}(\lambda)$.
- The survival time U_i and event indicator δ_i are obtained according to (3) and (4). The rate of the exponential distribution, λ, is used to control the proportion of censoring in the training set.
- Assign class label to each data vector by the rule in (5). The time of interest, τ, is set to the median value among the survival times, so that the prior probability for each class is about the same.
- Generate 400 samples for training, 400 for validation, and 2000 for testing.

This data set conforms to the probabilistic assumptions (*i.e.*, exponential distribution) underlying the classical modeling approach. So the Cox modeling approach is expected to be very competitive for the synthetic data set. The following experimental procedure was used in all experiments:

- Estimate the classifier using the training data.
- Find optimal tuning parameters for each method using the validation data. For the Cox modeling approach, the validation data are not used.
- Estimate the test error of the final model using the test data.

Further, the experiment is performed ten times with different random realizations of the training, validation, and test data.

In this experiment, the average proportion of the censored observation is 16.7% (or about 67 observations in the training set are censored). The test errors for ten trials are shown in Table I. The average test errors in percentage (along with standard deviations) for the Cox model, sSVM with linear kernel, sSVM with RBF kernel, and LUPI are 27.9±1.5, 24.8±1.2, 27.9±1.0, and 23.8±1.3, respectively.

The LUPI achieves the lowest test error among the methods in eight trials. Comparing the sSVM method with different kernels, it is not surprising to find that sSVM with linear kernel performs better than that with RBF kernel. Because our synthetic data is generated from a nearly linear model and there is intrinsic linearity in the data, methods with linear kernel are expected to perform better than those with RBF kernel. The Cox model has the highest test error in all trials. The results indicate that LUPI method yields performance similar or superior to classical Cox models, even though this synthetic data set is generated using exponential distributions (for which the Cox method is known to be statistically optimal).

A. Number of Training Samples

To investigate the effect of training sample size on the test errors, the training sample size is reduced to 250 and 100. The validation sample sizes are changed accordingly. The results are reported in Table II and III.

For 250 training samples, the average test errors for the Cox model, sSVM with linear kernel, sSVM with RBF kernel, and LUPI are 28.5±2.1, 27.6±1.7, 31.1±1.7, and 27.6±2.8, respectively. The LUPI has the best performance in six trials, although the average test errors for LUPI and sSVM with linear kernel are roughly the same. Further, the performance gap between the Cox model and LUPI (or sSVM with linear kernel) is closing when the size of the training data is reduced. This observation is more evident when the sample size is reduced to 100. For 100 training samples, the sSVM with linear kernel has the lowest test error in four trials, and the Cox model has the best performance in three. However, LUPI gives the lowest test error in one trial only.

This can be explained by the fact that simpler model would have better generalization performance with small training sample size. Moreover, based on our problem setting described in Section III, the class labels (y) already carry the survival time information partially. That is why sSVM with linear kernel can achieve better performance with training sample size 100, even without the knowledge of the survival time.

TABLE I

THE TEST ERRORS (%) FOR THE SYNTHETIC DATA WITH 400 TRAINING SAMPLES.

Trial	1	2	3	4	5	6	7	8	9	10
Cox	26.9	27.9	28.7	27.1	30.6	28.6	25.1	28.0	26.8	29.1
sSVM linear	23.6	23.8	**24.6**	**23.7**	26.9	24.7	23.4	26.1	25.0	26.0
sSVM rbf	27.9	28.0	27.6	27.4	27.5	29.4	26.9	29.4	26.3	28.2
LUPI	**23.4**	**22.4**	26.4	24.5	**25.0**	**23.1**	**22.7**	**23.2**	**22.5**	**24.6**

TABLE II

THE TEST ERRORS (%) FOR THE SYNTHETIC DATA WITH 250 TRAINING SAMPLES.

Trial	1	2	3	4	5	6	7	8	9	10
Cox	27.0	27.6	25.8	31.3	29.6	29.4	27.4	26.9	32.4	**27.3**
sSVM linear	**25.5**	26.7	**25.3**	**26.5**	29.1	28.7	28.0	27.6	30.6	27.4
sSVM rbf	28.7	31.3	29.0	32.0	31.2	34.7	30.9	30.8	30.4	31.8
LUPI	32.0	**23.5**	27.0	26.8	**28.7**	**27.7**	**26.0**	**25.0**	**27.0**	32.5

TABLE III

THE TEST ERRORS (%) FOR THE SYNTHETIC DATA WITH 100 TRAINING SAMPLES.

Trial	1	2	3	4	5	6	7	8	9	10
Cox	31.6	31.8	**32.5**	32.1	32.1	36.5	**30.8**	31.4	**32.5**	30.4
sSVM linear	**31.4**	**31.3**	35.6	27.6	33.3	**32.8**	31.4	**29.8**	32.6	30.9
sSVM rbf	35.4	35.0	35.4	33.4	33.8	35.0	34.1	33.1	36.1	**30.3**
LUPI	32.8	36.1	35.2	**26.9**	**30.5**	36.8	37.1	30.9	36.2	31.5

Table IV summarizes the relative performance of the four methods, as a function of sample size. The LUPI outperforms all other methods when the training sample size is larger than 250. Nonetheless, the Cox model is more competitive for moderate training sample size (100), with which gives similar performance as the sSVM with linear kernel.

TABLE IV

TEST ERRORS AS A FUNCTION OF TRAINING SAMPLE SIZE (≤ 400).

Training size	100	250	400
Censoring	15.2%	16.9%	16.7%
Cox	32.2 $\pm$ 1.7	28.5 $\pm$ 2.1	27.9 $\pm$ 1.5
sSVM linear	**31.7 $\pm$ 2.1**	**27.6 $\pm$ 1.7**	24.8 $\pm$ 1.2
sSVM rbf	34.1 $\pm$ 1.6	31.1 $\pm$ 1.7	27.9 $\pm$ 1.0
LUPI	33.4 $\pm$ 3.4	**27.6 $\pm$ 2.8**	**23.8 $\pm$ 1.3**

TABLE V

TEST ERRORS AS A FUNCTION OF TRAINING SAMPLE SIZE (≥ 750).

Training size	750	1000	1200
Censoring	16.5%	16.2%	15.6%
Cox	27.7 $\pm$ 1.0	27.9 $\pm$ 0.5	26.9 $\pm$ 0.5
sSVM linear	24.0 $\pm$ 1.2	23.5 $\pm$ 0.7	22.6 $\pm$ 0.6
sSVM rbf	26.7 $\pm$ 1.0	25.4 $\pm$ 0.8	24.6 $\pm$ 1.0
LUPI	**23.7 $\pm$ 2.1**	**22.7 $\pm$ 0.7**	**21.9 $\pm$ 0.7**

To study the effectiveness of the LUPI method for large training size, we increase the training and validation sample size to 750, 1000, and 1200. In addition, the test sample size is set to 5000. The test errors are summarized in Table V. As expected, with the increasing size of training samples, the test errors of all methods are reduced, and the relative advantage of LUPI is more noticeable. This shows that mapping the privileged information into correcting space helps to estimate a better classifier.

B. Noise Level in the Survival Time

To examine the effect of noise level in the survival time on the test errors, noise with different variances are added to the survival time. The noise variance ranges from 0 to 0.5 and the training and validation sample sizes are set to 250. The proportion of censored observations is kept around 16%. The test errors of the four methods are summarized in Table VI as a function of noise levels.

TABLE VI

TEST ERRORS AS A FUNCTION OF NOISE LEVELS.

Noise level	0	0.1	0.25	0.5
Censoring	15.3%	16.0%	16.7%	19.3%
Cox	**10.9 $\pm$ 0.7**	23.0 $\pm$ 2.1	30.7 $\pm$ 1.9	34.8 $\pm$ 1.7
sSVM linear	13.8 $\pm$ 1.5	**21.7 $\pm$ 3.0**	28.9 $\pm$ 2.0	34.3 $\pm$ 1.9
sSVM rbf	16.5 $\pm$ 1.2	24.0 $\pm$ 2.6	30.9 $\pm$ 2.3	35.7 $\pm$ 3.1
LUPI	14.8 $\pm$ 1.9	22.1 $\pm$ 2.7	**26.7 $\pm$ 1.7**	**32.1 $\pm$ 1.9**

It is evident that the test errors are reduced in all methods when the noise variance is decreased. When there is no noise in the survival time, the data are generated from a distribution that follows exactly the Cox modeling assumption. It is expected that the Cox model achieves the lowest test error under the zero-noise scenario. However, the increasing of noise level has much larger negative effect in the Cox modeling approach. The

test error is increased from 11% to 35% when the noise level is raised from 0 to 0.5. Meanwhile, for the same changes in the noise level, the test error of LUPI is raised from 15% to 32%.

Apart from the zero-noise scenario, the sSVM with linear kernel achieves the lowest average test error when the noise variance is 0.1. The LUPI, however, has the best performance when the noise level is higher than 0.25. It can be concluded that the SVM-based methods are more suitable for noisy data or data deviated much more from the Cox modeling assumption.

C. Proportion of Censoring

We also adjust the proportion of censoring in the training data to investigate the effect of censoring on the test errors. The percentage of censored observations in the training data varies from 6% to 45% in our experiment. The noise variance is set to 0.2 and the training and validation sample sizes are kept at 250. The experiment results are summarized in Table VII.

TABLE VII
TEST ERRORS AS A FUNCTION OF CENSORING RATES.

Censoring	6.2%	16.6%	30.6%	45.4%
Cox	27.4 ± 1.3	28.8 ± 1.2	33.0 ± 1.9	41.2 ± 1.5
sSVM linear	**25.8 ± 2.4**	26.8 ± 2.6	31.9 ± 1.7	39.2 ± 1.3
sSVM rbf	27.4 ± 1.8	28.4 ± 2.5	33.7 ± 1.6	41.6 ± 1.9
LUPI	27.0 ± 1.2	**25.7 ± 1.9**	**30.6 ± 1.6**	**37.8 ± 2.5**

When about 6% of the training data are censored, the sSVM with linear kernel gives the lowest test error. A low censoring rate means that most of the observed survival times are exact. Through our class label encoding scheme, most of training samples can be associated with well-defined class labels and the survival time information can be completely embedded in the class membership. Thus, the sSVM with linear kernel is expected to perform better than the LUPI since the parameter tuning (model selection) is easier for the sSVM.

On the contrary, if a large portion of the observations are censored (about 16% or more), the LUPI outperforms all other methods. With more censored observations in the training set, more observed survival times are obtained by the non-linear operator in (3). Hence, there exists non-linearity within the survival time information, and methods with non-linear parametrization (kernel) are expected to achieve better performances.

VI. Real-Life Data Sets

This section describes empirical comparisons using three real-life data sets from the *Survival* package in R [17]. For all comparisons, the common decision space for SVM+/LUPI uses the linear kernel while the correction space uses the RBF kernel. For the sSVM method, both linear and the RBF kernels are investigated. In all experiments, the time of interest τ was set to the median of the observed survival times. Our experiments for the three medical data sets follow the procedure [2], [15]:

- Use five-fold cross-validation to estimate the test errors.
- Within each training fold, the parameter tuning (model selection) is performed through a five-fold resampling.

Our experimental set-up uses double resampling procedure [2]. One level of resampling is used for estimating the test error of the learning method, and the second level is for tuning the model parameters. Since there is no well-defined class labels for the censored observations with $U_i < \tau$, the test errors are reported based on samples with definite labels, *i.e.*, exact observations and censored observations with $U_i \geq \tau$. Further, model parameters are selected based on the performance with those samples linked to well-defined labels.

The *Veteran* data set is from the Veterans' Administration Lung Cancer Study which is a randomized trial of two treatment regimens for lung cancer [17]. In the *Veteran* data set, there are 137 instances (observations) and each instance has 10 attributes. Less than 7% of the instances are censored. Among the nine censored instances, one has the observed survival time less than the time of interest. In other words, only one instance is associated with the uncertain class label in the *Veteran* data set.

The *Lung* data set studied the survival and usual daily activities in patients with advanced lung cancer by the North Central Cancer Treatment Group (NCCTG) [17]. There are 167 instances in this data set, and each instance has 8 attributes. About 28% of the instances are censored, and 21 censored instances are linked to uncertain class labels.

The *PBC* data set is from the Mayo Clinic trial in primary biliary cirrhosis (PBC) of the liver conducted between 1974 and 1984 [17]. The *PBC* data set contains 258 instances and each instance has 22 attributes. More than half of the instances are censored, and 54 censored instances do not have the definite class labels.

The descriptions of the data sets are summarized in Table VIII. The number of censored observation within each data is listed in the row of '$\delta = 0$'. The row of 'Censored %' indicates the proportions of censored observations in the data sets. The 'Uncertain %' row shows the percentage of data with uncertain class labels. Table VIII also shows the test errors from different methods applied to the three data sets.

TABLE VIII
SUMMARY OF THE *Survival* DATA SETS AND THE EXPERIMENT RESULTS.

Data set	Veteran	Lung	PBC
Size	137	167	258
Attributes	10	8	22
$\delta = 0$	9	47	147
Censored %	6.57	28.14	56.98
Uncertain %	0.7	12.6	20.9
Cox	**23.4 ± 4.6**	43.3 ± 5.6	34.3 ± 7.1
sSVM linear	26.1 ± 7.3	40.8 ± 8.2	32.2 ± 6.4
sSVM rbf	29.9 ± 4.9	**37.7 ± 7.4**	33.0 ± 4.9
LUPI	30.4 ± 4.5	38.3 ± 9.9	**25.3 ± 10.6**

These comparisons suggest that SVM-based modeling achieves lower test error than the Cox model in two data sets that have significant amount of censored/uncertain data. These results also show large variability of estimated test errors, due to partitioning of the available data into five (training, test)

folds. This variability is reflected in large standard deviations (of test error rates).

Another reason for variability of the SVM-based model estimates is due to its model selection via resampling. Notably, standard deviations of error rates for LUPI shown in Table VIII are generally higher than standard deviations for the Cox model (which has no tunable parameters). This underscores the importance of robust model selection strategies for the SVM-based approaches. Further, the LUPI approach can easily model the non-linearity in the data, even though our comparisons use linear parametrization in the decision space, in order to make fair comparisons with the Cox regression.

We observe the effect of censoring rate on the generalization performance from these results. For the *Veteran* data set that has very small amount of censored/uncertain data, the Cox model gives the lowest test error. However, LUPI shows its advantage when the proportion of censoring is increased. This is especially true for the *PBC* data set, within which more than half instances are censored. Since the high censoring rate brings some level of non-linearity to the privileged information, LUPI performs better.

VII. Conclusions and Discussion

This paper proposes predictive modeling of survival data as a binary classification problem. We apply the SVM+ formulation to solve the problem. The SVM+ approach incorporate the information about survival time to estimate an SVM classifier. We have illustrated the advantages and limitations of these modeling approaches using several synthetic and real-life data sets. We also improved the scalability of the SVM+ implementation by using the *quadprog* as the QP solver. This implementation is capable of handling 1K-5K training samples. Likewise, it solves the SVM+ optimization problem within reasonable time so that the model selection strategies are feasible.

Advanced SVM-based methods appear very effective when the proportion of censoring in the training data is large, or the observed survival time does not follow the classical probabilistic assumptions, *e.g.*, the exponential distribution [1], [16]. With more training data available, LUPI can estimate a classifier with higher accuracy. On the other hand, when the proportion of censored data is small, then the best strategy is to apply the standard SVM classifier.

The LUPI paradigm maps all privileged information onto the same correcting space. However, in real-life data (such as clinical data or survival data) different training samples (patients) often have different privileged information. One possible strategy for handling such heterogeneous training data is to map different types of privileged information onto different correcting spaces. This new approach is called the Multiple-Space Privileged Information (MSPI), an extension of the SVM+/LUPI framework [9]. The survival data have two different types of hidden information (due to exact observations and due to censoring). Under MSPI approach, these two types of hidden information can be modeled in two different correcting spaces. It has never been tested empirically; therefore, finding strategies to map the privileged information into multiple spaces would be the focus of our future work.

The equal misclassification cost is assumed throughout this paper; however, realistic medical applications use unequal costs, *i.e.* the costs for false-positive and false-negative are different. We will incorporate different misclassification costs into the SVM+/LUPI formulations. Incorporating different costs can be easily done for standard SVM+ model following the same approach as in standard cost-sensitive SVM [2]. However, special attention may be needed for handling censored data which is often encountered in medical prediction applications. Further, our methodology for predictive modeling of survival data can be readily extended to other (non-medical) applications, such as predicting business failure (aka bankruptcy) or predicting marriage failure (aka divorce).

References

[1] O. Aalen, Ø. Borgan, H. Gjessing, and S. Gjessing, *Survival and Event History Analysis: A Process Point of View*, ser. Statistics for Biology and Health. Springer-Verlag New York, 2008.

[2] V. Cherkassky and F. Mulier, *Learning from data: concepts, theory, and methods*. Wiley, 2007.

[3] F. Khan and V. Zubek, "Support Vector Regression for censored data (SVRc): A novel tool for survival analysis," in *Data Mining, 2008. ICDM '08. Eighth IEEE International Conference on*, Dec. 2008, pp. 863–868.

[4] J. Shim and C. Hwang, "Support vector censored quantile regression under random censoring," *Comput. Stat. Data Anal.*, vol. 53, no. 4, pp. 912–919, Feb. 2009.

[5] P. K. Shivaswamy, W. Chu, and M. Jansche, "A support vector approach to censored targets," in *Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, ser. ICDM '07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 655–660.

[6] D. J. Hand, "Classifier technology and the illusion of progress," *Statistical Science*, vol. 21, no. math.ST/0606441. IMS-STS-STS-155. 1, pp. 1–15, June 2006.

[7] V. N. Vapnik, *Estimation of dependences based on empirical data, Empirical inference science: afterword of 2006*. Springer, 2006.

[8] V. Cherkassky, *Predictive Learning*. VCtextbook.com, 2013.

[9] V. Vapnik and A. Vashist, "A new learning paradigm: Learning using privileged information," *Neural Networks*, vol. 22, no. 5-6, pp. 544–557, July-August 2009.

[10] I. W. Tsang, J. T. Kwok, and P.-M. Cheung, "Core vector machines: Fast svm training on very large data sets," *Journal of Machine Learning Research*, vol. 6, pp. 363–392, Dec. 2005.

[11] J. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines," Microsoft Research, Seattle, WA, TechReport MSR-TR-98-14, April 1998.

[12] D. Pechyony, R. Izmailov, A. Vashist, and V. Vapnik, "SMO-style algorithms for learning using privileged information," in *Proceedings of the 2010 International Conference on Data Mining (DMIN'10)*, 2010.

[13] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 1.22," August 2012. [Online]. Available: http://cvxr.com/cvx/

[14] S. Fouad, P. Tino, S. Raychaudhury, and P. Schneider, "Learning using privileged information in prototype based models," in *Artificial Neural Networks and Machine Learning - ICANN 2012*, ser. Lecture Notes in Computer Science, A. Villa, W. Duch, P. Érdi, F. Masulli, and G. Palm, Eds. Springer Berlin Heidelberg, 2012, vol. 7553, pp. 322–329.

[15] L. Liang, F. Cai, and V. Cherkassky, "Predictive learning with structured (grouped) data," *Neural Networks*, vol. 22, no. 5-6, pp. 766–773, 2009.

[16] M. Zhou. Use software R to do survival analysis and simulation. a tutorial. [Online]. Available: http://www.ms.uky.edu/~mai/Rsurv.pdf

[17] T. M. Therneau, *A Package for Survival Analysis in R*, 2013, r package version 2.37-4. [Online]. Available: http://CRAN.R-project.org/package=survival