Deep survival algorithm based on nuclear norm

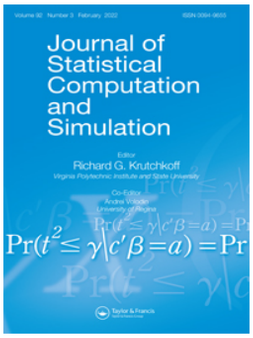
Article *in* Journal of Statistical Computation and Simulation · January 2022

DOI: 10.1080/00949655.2021.2015770

2 authors, including:

Xuejing Zhao
Lanzhou University
32 PUBLICATIONS **389** CITATIONS

SEE PROFILE



Taylor & Francis
Taylor & Francis Group

Journal of Statistical Computation and Simulation

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/gscs20

Deep survival algorithm based on nuclear norm

Jianyang Tong & Xuejing Zhao

To cite this article: Jianyang Tong & Xuejing Zhao (2022): Deep survival algorithm based on nuclear norm, Journal of Statistical Computation and Simulation, DOI: 10.1080/00949655.2021.2015770

To link to this article: https://doi.org/10.1080/00949655.2021.2015770

Published online: 21 Jan 2022.

JOURNAL OF STATISTICAL COMPUTATION AND SIMULATION
https://doi.org/10.1080/00949655.2021.2015770

Taylor & Francis
Taylor & Francis Group

Deep survival algorithm based on nuclear norm

Jianyang Tong [a,b] and Xuejing Zhao [a]

[a]School of Mathematics and Statistics, Center for Data Science, Lanzhou University, Lanzhou, People's Republic of China; [b]School of Mathematics and Statistics, Yunnan University, Kunming, People's Republic of China

ABSTRACT

This paper devotes to propose a nuclear-norm-based deep survival algorithm (NN-DeepSurv), to study the regression problem of survival data with right censoring. The nuclear norm method is used to impute missing covariates, and it's combined with DeepSurv algorithm to train the regression model. We compare our algorithm with other state-of-the-art methods: Cox proportional hazards regression model (Coxph), Cox proportional hazards model with lasso regression (Cox-lasso), random survival forests (RSF), DeepSurv, and Xgboost algorithm, on 2 simulated datasets and 6 clinical datasets, to show the superiority of the performance of our algorithm.

ARTICLE HISTORY
Received 1 January 2021
Accepted 5 December 2021

KEYWORDS
Survival analysis; deep learning; missing data complement; nuclear norm; NN-DeepSurv; concordance index

2000 MATHEMATICS SUBJECT CLASSIFICATION
62N01

1. Introduction

The purpose of survival analysis in medical statistics is mainly to explore the relationship between patients' covariates and lifetimes, Cox proportional hazards regression model and accelerated lifetime model are always the common practical models. Nowadays, the survival data has become more and more complicated in structures and dimensions. The covariates may be heterogeneous, for example, some covariates of the data can be expressed by image, while other covariates are all numerical. Zhou et al. [1] applied tensor regression to train the regression model for heterogeneous data. More commonly, there may be categorical variables in features. Whether they're numerical or not, they need to be converted for better interpretability. Usually, the dummy variable method is chosen to express categorical variables. Dummy variables are limited to two specific values, 1 or 0. This kind of representation is convenient and effective. However, it's not suitable for every training algorithm. For example, when the decision tree algorithm is used to train the model, the dummy variable is not a good choice since this representation may reduce the model accuracy, especially when the categorical variable is of high-dimension [2]. Micci-Barreca [3] proposed a new representation method for high-cardinality categorical attributes, which can transform the values of categorical attributes into the forms of probability and expectation. In addition, the existence of missing value in survival data is also a

CONTACT Xuejing Zhao zhaoxj@lzu.edu.cn School of Mathematics and Statistics, Center for Data Science, Lanzhou University, 730000 Lanzhou, People's Republic of China

common problem. Khosla et al. [4] introduced four kinds of methods to accomplish missing data imputation: filling-in missing entries with column mean, filling-in missing entries with column median, imputation through linear regression and regularized expectation maximization (EM).

Nowadays, high-dimension data is also a great challenge in survival analysis and has been paid extensive attention [2,5,6]. Principal component analysis (PCA) is a practical method in dimensionality reduction. However, classical PCA, based on the L_2 norm, which relies on the Euclidean Distance, is sensitive to outliers and noises [7]. Galpin and Hawkins [8] utilized covariance estimation method based on the L_1 norm, which has better robustness than the L_2 norm. Ke and Kanade [9] tried to solve the L_1 norm optimization with semidefinite programming (SDP). Ding et al. [10] combined the L_2 norm with L_1 norm to extract features of robust subspace. However, all these algorithms above have two same drawbacks: the non-unique solutions of optimization and the complexity in calculation of the matrix rank. Candès and Recht [11] replaced matrix rank with matrix nuclear norm in dimension reduction to avoid problems above.

Training survival model with machine learning algorithms becomes a hot subject in recent years, especially with the application of deep learning technology. There are many branches in deep learning and the most classical one is deep neural network (DNN). But the performance of DNN in training survival data was not as good as random survival forests (RSF), a state-of-the-art method to train survival regression model [12]. Hinton et al. [13] came up with a new method called Dropout to prevent neural networks from overfitting. Many kinds of Dropout methods were proposed thereafter. Klambauer et al. [14] put forward a new kind of activation function: scaled exponential linear units (SELUs), which induces self-normalizing properties. To update the weights of network efficiently, [15] proposed an Adam algorithm for stochastic optimization, learning rate decay technology can be used for an Adam optimizer, which is helpful for raising the accuracy of the model.

This paper's main contribution is applying the nuclear norm optimization to complete missing survival covariates and constrain dimensions of survival data, and combining the nuclear norm optimization with DeepSurv algorithm [16], with the superiority of needn't require an a priori selection of covariates but learns them adaptively, to train the regression model for survival datasets. DropConnect is used as a substitute for standard Dropout method, which works better in smaller networks than standard Dropout [17]. Before training the model, random search [18] is used for hyper-parameters' optimization. And SELUs are used in networks as the activation function, Adam algorithm and learning rate decay method are exploited to update parameters.

The paper is organized as follows. Section 2 gives a detailed introduction about how to use the nuclear norm, dummy variables method and transformation scheme based on empirical Bayes in the preprocessing step. Details of networks training and models are given in Section 3. Then Section 4 illustrates the performance of proposed algorithm by comparing it with other five state-of-the-art models on 2 simulated datasets and 6 clinical datasets. Section 5 and 6 give some discussions and conclusions respectively.

2. Preprocessing schemes for survival data

There are many problems to deal with the raw survival datasets. This paper focuses on two main issues: data missing and the representation of categorical attributes. For

high-dimension data, some ordinary methods may be used to process it at first, such as removing features with high missing rates or with low variance, high correlation filter and so on. After that, we pay attention to two main problems next.

2.1. Preprocessing schemes for categorical attributes

For a survival analysis covariates matrix, there may well be categorical attributes, sometimes even non-numerical. It is necessary to convert the values of categorical attributes to numerical values before training. In most cases dummy variables are introduced into original features. For a categorical attribute which contains k different values, usually $k-1$ dummy variables are explored. If the value of the categorical attribute is one of the k values, the corresponding dummy variable can be 1, and all other $k-1$ dummy variables are 0. This kind of method is easy to operate and useful, but it can also easily increase the dimensions of data matrices, especially for high-dimension categorical attributes. Moreover, it works not well in tree-based algorithms [2].

In case of the application in tree-based methods, this paper recommends a representation method based on empirical Bayes. For a categorical variable X, we need to convert each value X_i into a continuous numerical scalar ψ_i, and ψ_i is based on the following conditional probability:

$$X_i \longrightarrow \psi_i \cong P(Y|X = X_i).$$

The dependent variables of survival datasets we found are all continuous variables. For this situation, ψ_i can be calculated as the form of expectation [3]:

$$\psi_i = \lambda(n_i)\frac{\sum_{k\in L_i} Y_k}{n_i} + [1-\lambda(n_i)]\frac{\sum_{k=1}^{n_{TR}} Y_k}{n_{TR}}, \tag{1}$$

where n_i denotes the amount of subjects which meet the condition $X = X_i$, n_{TR} is the number of subjects in the training set, L_i is set of k where $X_k = X_i$ $(1 < k < n_{TR})$, Y_k is the corresponding value of Y, $\lambda(n_i)$ is usually called the shrinkage factor $(0 < \lambda(n_i) < 1)$, which can be calculated as [3]:

$$\lambda(n_i) = \frac{n_i\tau^2}{\sigma^2 + n_i\tau^2},$$

where τ^2 is the variance of the dataset where $X = X_i$, σ^2 denotes the variance of the whole dataset.

2.2. Missing data completion

The nuclear norm, the highest convex surrogate of the matrix rank, has been proved to result in sparsity in high-dimensional data analysis, so it has the superiority over other imputation methods in dealing with high-dimensional input data. Also the optimization problem can be effectively solved by an alternative direction method of the multipliers (ADMM) algorithm. This paper uses the nuclear norm optimization to complete missing

entries in data matrices. The objective is to replace the matrix X (with missing entries) by Z, the imputation of X. The nuclear norm optimization is:

$$\begin{aligned} &\text{minimize} \quad \|Z\|_* \\ &\text{subject to} \quad \sum_{(i,j)\in\Omega} (Z_{ij} - X_{ij})^2 \leq \varepsilon, \end{aligned} \tag{2}$$

where $\|Z\|_*$ is the nuclear norm of Z, Z_{ij} and X_{ij} denote respectively the entries of corresponding matrices, ε is an arbitrarily small positive number, and Ω denotes the point set which records the coordinates of non-missing points in the original matrix.

The nuclear norm of a matrix Z is described as [11]:

$$\|Z\|_* = tr\left(\sqrt{Z^T Z}\right).$$

For a matrix Z there exists the singular value decomposition $Z = U\Sigma V^T$, then we have:

$$\begin{aligned} tr\left(\sqrt{Z^T Z}\right) &= tr\left(\sqrt{(U\Sigma V^T)^T U\Sigma V^T}\right) \\ &= tr\left(\sqrt{V\Sigma^T U^T U\Sigma V^T}\right) \\ &= tr\left(\sqrt{V\Sigma^2 V^T}\right) \qquad \left(\Sigma^T = \Sigma\right) \\ &= tr\left(\sqrt{V^T V\Sigma^2}\right) \\ &= tr\left(\Sigma\right), \end{aligned} \tag{3}$$

which means the nuclear norm of the matrix is equivalent to the sum of eigenvalues of the matrix.

It can be proved that the nuclear norm of a matrix is convex (see the appendix). Because of the convexity, the existence of optimal solution can be confirmed. To the nuclear norm optimization problem (2), gradient method is adopted. As mentioned before, we have:

$$\frac{\partial \|Z\|_*}{\partial Z} = \frac{\partial tr(\Sigma)}{\partial Z} = \frac{\partial tr(U^T Z V)}{\partial Z} = \frac{\partial tr(V U^T Z)}{\partial Z}. \tag{4}$$

On using $\frac{\partial tr(AB)}{\partial B} = A^T$, where A and B are matrices [19], we have:

$$\frac{\partial \|Z\|_*}{\partial Z} = \frac{\partial tr(V U^T Z)}{\partial Z} = (VU^T)^T = UV^T. \tag{5}$$

Equation (5) illustrates how to easily calculate the gradient of matrix nuclear norm, and we can solve the optimization problem by iteration:

$$Z^{\text{new}} = Z^{\text{old}} - \eta \cdot \nabla_{Z^{\text{old}}} J(Z^{\text{old}}) = Z^{\text{old}} - \eta U^{\text{old}} (V^{\text{old}})^T, \tag{6}$$

where η is the learning rate, J is the objective function. The iteration stops when it converges or reaches the max iteration step.

3. Nuclear-norm-based deep Cox proportional hazards network (NN-DeepSurv)

In this Section, the survival regression model is exploited and model parameters are optimized with DNN algorithms. This paper combines DNN with the Cox proportional hazards model and applies some newly proposed and efficient deep learning tricks in optimization, named as NN-DeepSurv.

For survival data $\{(X_i, \delta_i, Y_i) : X_i \in R^p, \delta_i \in \{0, 1\}, Y_i \in R, i = 1, \ldots, n\}$, here $Y_i = min\{T_i, C_i\}$ is the corresponding observed lifetime, where T_i and C_i are respectively the survival time and censoring time for the ith subject, X_i is the observed covariates of the ith subject, $\delta_i = I(T_i \leq C_i)$ is the corresponding censoring indicator. The objective function is usually composed of two parts: cost function and regularization term. The cost function is based on Cox partial likelihood function, the Cox partial likelihood function is:

$$L_c(\beta) = \prod_{i:\delta_i=1} \frac{\exp\left(\hat{h}_\beta(X_i)\right)}{\sum_{j\in\Re(T_i)} \exp\left(\hat{h}_\beta(X_j)\right)}, \tag{7}$$

where $\hat{h}_\beta(X_i) = X_i^T\beta$, $\Re(T) = \{i : T_i \geq t\}$ is the risk set, which is the set of subjects who are still alive at time t.

When the survival data doesn't satisfy the linear proportional hazards condition, the linear combination of features $\hat{h}_\beta(X)$ in Equation (7) can be replaced with the output of the network $\hat{h}_\theta(X)$, where θ is the weights of the network. The regularization item is chosen to be L_2 norm, so the objective function will be:

$$l(\theta) = -\frac{1}{N_{\delta=1}} \sum_{i:\delta_i=1} \left(\hat{h}_\theta(X_i) - \log \sum_{j\in\Re(T_i)} e^{\hat{h}_\theta(X_j)} \right) + \lambda \cdot \|\theta\|_2^2, \tag{8}$$

where $N_{\delta=1}$ is the number of subjects who die during observation, λ is the L_2 regularization parameter. In consideration of the nuclear-norm-type imputation, the objective function can also be:

$$\begin{aligned} l(\theta) = &-\frac{1}{N_{\delta=1}} \sum_{i:\delta_i=1} (\hat{h}_\theta(Z_i) - \log \sum_{j\in\Re(T_i)} e^{\hat{h}_\theta(Z_j)}) + \lambda_1 \cdot \|\theta\|_2^2 \\ &+ \lambda_2 \|Z\|_* + \lambda_3 \sum_{(i,j)\in\Omega} (Z_{ij} - X_{ij})^2. \end{aligned} \tag{9}$$

Combined with the nuclear norm imputation, the objective function (9) has the advantage of sparsity in covariates, therefore reduce the complexity of the network thereafter and prevent the overfitting.

The network we construct here is based on a fully-connected network. There are many modern deep learning techniques used to optimize the training of the network, such as DropConnect, SELU activation function, Adam optimization algorithm and learning rate scheduling. The hyper-parameters in the network are determined by random search method.

To prevent our regression model from overfitting, the Dropout technique is used in the network. Compared to the initial classical Dropout method, the DropConnect method is chosen, which performs better in small-scale networks [17]. In the initial Dropout method, every network node may be dropped out temporarily from the network with a certain probability. To generalize and improve this method, DropConnect was proposed to drop some weights in the network with a certain probability instead of dropping nodes. During the network training, the output of a certain layer can be denoted as:

$$y = f\left((W \circ M)x\right), \quad m_{ij} \sim Bernoulli(p), \tag{10}$$

where f is the activation function, x is the input of this layer, W is the weight matrix between this layer and previous layer, M is the dropout mask matrix, where the entry m_{ij} obeys the Bernoulli distribution with a probability p.

Also the SELUs is chosen as the activation function. The SELU activation function is defined as:

$$selu(x) = \lambda \begin{cases} x, & x > 0 \\ \alpha e^x - \alpha, & x \leq 0 \end{cases}, \tag{11}$$

where $\lambda = 1.0507$, $\alpha = 1.6733$ [14]. The SELUs function is a great development in deep neural network, with the property of self-normalization, and helps avoid the problem of gradient vanishing and gradient exploding.

As for the optimization algorithm of the objective function, the Adam algorithm is selected compared to traditional gradient descent algorithms. Different from the constant learning rate of the stochastic gradient descent method, the Adam method designs a kind of adaptive learning rate by calculating the first-order and second-order moment estimation of gradient. And the learning rate decay technique can be combined with Adam algorithm to improve the accuracy. At each step of iteration, the learning rate can be updated as follows [20]:

$$decayed\_LR = \frac{LR}{1 + epoch \cdot lr\_decay\_rate}, \tag{12}$$

where LR is the initial learning rate, $epoch$ is the step number of iterations, lr_decay_rate is the decay rate of learning rate, and $decayed\_LR$ is the learning rate after decay.

The input of the neural network is the survival covariates X, and the output is the log-risk function $h(X)$. To sum up, the specific procedure of the deep survival algorithm based on the nuclear norm is shown in Algorithm 1, the hyper-parameters (ε, learning rate in nuclear norm optimization η, L_2 regularization parameter λ, number of hidden layers, number of nodes in each layer, DropConnect rate p, initial learning rate in network training LR and decay rate of learning rate lr_decay_rate) are determined from random search.

4. Experiments

In this Section the performance of our algorithm is compared with the Cox proportional hazards model, Cox proportional hazards model with lasso regularization (Cox-lasso), random survival forests (RSF), Deep learning survival algorithm (DeepSurv) and Xgboost algorithm, on two artificial survival datasets and six clinical survival datasets.

Algorithm 1 NN-DeepSurv Algorithm

Input:

Training data $\{(X_i, \delta_i, Y_i)$: $X_i \in R^p, \delta_i \in \{0, 1\}, Y_i \in R, i = 1, \cdots, n\}$.

Output:

The regression function $h(X)$.

Algorithmic:

1: *Data preprocessing*: Convert categorical attributes to numerical attributes by the dummy variable method and empirical Bayes method (1), complete missing data with nuclear norm optimization (2), normalize the features;
2: *Optimization*: Optimize the objective function (9) by the Adam algorithm, to get the values of network weights;
3: Obtain the resulted regressor $h(X)$, we can get the value of $h(X)$ through the network according to the input X.

To measure the performances of different algorithms, the concordance index (C-index) is adopted, which is widely used in survival analysis [21,22]. It's mainly used to evaluate the difference between predicted values and actual values, which is an index to measure the prediction accuracy of the model. The value of concordance index could be accomplished by the Python function *concordance_index*(*event_times*, *predicted_scores*, *event_observed*), where *event_times* and *event_observed* are respectively the observed survival times and censoring indicators, *predicted_scores* is the predicted scores (these could be survival times, hazards, et al), which is the output of network here.

4.1. Performance on simulated data

Two different kinds of datasets are simulated to test the performance of algorithms, one's log-risk function with linear form and the other one with nonlinear form of Gaussian function. Two datasets both consist of 6000 subjects and 10 covariates, and are divided into the training set, validation set and test set at the proportion of 3:1:1. The values of every covariate obey the uniform distribution on $(-1, 1)$, the survival time T of each subject is obtained from a function with respect to covariates by using the exponential Cox model [23]:

$$T \sim Exp\left(\lambda(t; x)\right) = Exp\left(\lambda_0 \cdot e^{h(x)}\right). \tag{13}$$

The censoring time C is simulated such that 90% of subjects' survival time can be observed. Also there are 5% missing in each of the covariates intentionally.

For subjects with linear log-risk functions, they fit the logarithmic linear hypothesis of Cox proportional hazards model. Suppose the log-risk function is:

$$h(x) = \sum_{i=1}^{n} a_i x_i, \tag{14}$$

where $a_i \in (0, 1)$ is a random number, n is the number of covariates.

Table 1. Comparison of different algorithms on simulated datasets.

Regressor	Simulated linear	Simulated nonlinear
Coxph	0.782365 (0.762,0.801)	0.512746 (0.499,0.526)
Cox-lasso	0.784328 (0.764,0.803)	0.523658 (0.511,0.536)
RSF	0.756769 (0.737,0.776)	0.627865 (0.612,0.644)
Xgboost	**0.867795 (0.846,0.878)**	0.694262 (0.675,0.713)
DeepSurv	0.761269 (0.742,0.780)	0.642352 (0.626,0.658)
NN-DeepSurv	0.785827 (0.766,0.804)	**0.734798 (0.717,0.753)**

Note: Boldface numbers indicate the best performance.

For subjects with nonlinear log-risk functions, we assume the log-risk function of them is a Gaussian function of covariates:

$$h(x) = \ln(\lambda_{\max}) \exp\left(-\frac{\sum_{i=1}^{n} x_i^2}{2\sigma^2} \right), \tag{15}$$

where $\lambda_{\max} = 5, \sigma = 0.5, n = 10$. After getting two simulated datasets, the min-max normalization is used to normalize data, and then the simulated data is trained with different methods. The result can be seen from Table 1, where all the models' parameters have been optimized.

In linear dataset where the hypothesis of linearity between the input covariates and the risk of the death is satisfied, all models can capture the linear relationships, including the Cox-type ones. Results in Table 1 show that the confidence intervals of the algorithms are overlapped, which means that all algorithms have informative performances in linear datasets except Xgboost, slightly higher than others. Even through Xgboost archives the best results, the others also have a good performance.

Whereas on nonlinear dataset where the linearity assumption can no longer be satisfied, the Cox-type algorithms (Coxph, Cox-Lasso) have a poor performance, less than that of linear ones. Even the tree-based algorithms demonstrate a better performance, the C-index of NN-DeepSurv is the largest in all methods. Compared to the original DeepSurv algorithm, the C-indexes of NN-DeepSurv on two datasets are both improved, owing to the utility of the nuclear-norm-type imputation of the missing covariates and the DropConnect in network. Generally speaking, our improved algorithm has better performance over the most other algorithms on the two simulated datasets.

4.2. Performance on real data

In this section, the proposed algorithm and other statistical machine learning algorithms are applied to six clinical real datasets. Here six different clinical survival datasets are WHAS, SUPPORT, METABRIC, Rotterdam&GBSG, MLC and MIMIC-III:

(1) The Worcester Heart Attack Study (WHAS). This dataset is mainly used to explore the factors which influence the death rate of acute myocardial infarction (AMI) patients. And this dataset contains 1638 patients, each patient has 5 features: age, sex, body mass index (BMI), left heart failure complications (CHF), and order of MI (MIORD).

In the dataset, 42% of patients died during observation, and the median death time is 516 days.

(2) Study to Understand Prognoses Preferences Outcomes and Risks of Treatment (SUPPORT). The dataset consists of 9105 patients and 14 features: age, sex, rage, number of comorbidities, presence of diabetes, presence of dementia, heart rate, respiration rate, temperature and the others. 68% of patients died during the observation, and the median death time is 58 days.

(3) Molecular Taxonomy of Breast Cancer International Consortium (METABRIC). This group of data is used to find the expression of gene and protein leading to the breast cancer, to help patients make a definite diagnosis in advance and provide better treatment plans. The dataset consists of 1980 patients and 9 features: hormone treatment indicator, radiotherapy indicator, chemotherapy indicator, ER-positive indicator, age at diagnosis and the other 4 gene indicators (MKI67, EGFR, PGR, and ERBB2). In the dataset, there are 57% of patients who died during observation, with the median death time of 116 months.

(4) Rotterdam&German Breast Cancer Study Group(Rotterdam&GBSG). This dataset is from the Rotterdam Tumor Bank, aims to provide the recommendation system of treatment for German breast cancer study group. The group of data consists of 1546 patients, 90% of patients died during observation.

(5) The Mayo clinic lung cancer data(MLC). This group of data consists of 228 patients with the lung cancer, the features cover age, sex, 2 clinic performance scores from experts, one clinic performance score from patients, loss of calories and loss of weights during the last 6 months.

(6) MIMIC-III [24]. This dataset is a group of free and open-source big-scale data widely used in the medical statistics field. This dataset includes health information of 58,976 patients from the Beth Israel Deaconess medical centre, with a time span from 2001 to 2012. The data contains not only many patients but also a large number of features. We select 19 features from them: SUBJECT_ID, HADM_ID, icustay_id, ADMITTIME, DISCHTIME, los_total, DEATHTIME and the other 12 covariates.

All these datasets above can be found either on http://lib.stat.cmu.edu/datasets or R-package 'survival' or 'cgdsr'. For the six datasets above, we extract 60% of subjects as the training set, 20% of subjects as the validation set and the rest as the testing set.

The results of training can be seen from Table 2, where all models' parameters have been optimized. Compare to the original DeepSurv algorithm, NN-DeepSurv performs better on all the six datasets, especially on MIMIC-III. Generally speaking, the C-indexes of NN-DeepSurv are the largest on five groups of data except the WHAS dataset; for the WHAS dataset, the NN-DeepSurv algorithm also performs well, which has a comparatively large C-index. Thus the proposed NN-DeepSurv model performs better than other five algorithms in most cases.

5. Discussion

NN-DeepSurv is an effective method in survival analysis with high-dimensional covariates with possible missing values, where the utility of the nuclear norm in imputation can archive a sparse covariates matrix, therefore reducing the complexity of the covariates used

Table 2. Comparison of NN-DeepSurv, DeepSurv, Coxph, Cox-lasso, RSF and Xgboost on six clinical datasets.

Dataset	Regressor	C-index	Dataset	Regressor	C-index
WHAS	Coxph	0.816025 (0.813,0.819)	Rotterdam &GBSG	Coxph	0.658773 (0.655,0.662)
	Cox-lasso	0.821376 (0.781,0.861)		Cox-lasso	0.665471 (0.637,0.693)
	RSF	**0.892884 (0.890,0.895)**		RSF	0.647924 (0.644,0.651)
	DeepSurv	0.866723 (0.863,0.870)		DeepSurv	0.676349 (0.673,0.679)
	Xgboost	0.871598 (0.829,0.915)		Xgboost	0.616929 (0.590,0.643)
	NN-DeepSurv	0.870008 (0.827,0.913)		NN-DeepSurv	**0.686052 (0.657,0.715)**
SUPPORT	Coxph	0.583076 (0.581,0.585)	MLC	Coxph	0.577925 (0.567,0.588)
	Cox-lasso	0.572634 (0.561,0.585)		Cox-lasso	0.577423 (0.565,0.590)
	RSF	0.619302 (0.618,0.621)		RSF	0.572537 (0.560,0.584)
	DeepSurv	0.618907 (0.617,0.621)		DeepSurv	0.575024 (0.501,0.650)
	Xgboost	0.587118 (0.575,0.599)		Xgboost	0.568526 (0.494,0.642)
	NN-DepSurv	**0.620708 (0.608,0.634)**		NN-DeepSurv	**0.634076 (0.551,0.717)**
METABRIC	Coxph	0.631674 (0.627,0.636)	MIMIC -III	Coxph	0.770325 (0.770,0.771)
	Cox-lasso	0.639241 (0.611,0.667)		Cox-lasso	0.801045 (0.773,0.845)
	RSF	0.619517 (0.615,0.624)		RSF	0.769523 (0.768,0.772)
	DeepSurv	0.654452 (0.650,0.659)		DeepSurv	0.720025 (0.718,0.722)
	Xgboost	0.575994 (0.549,0.601)		Xgboost	0.741258 (0.739,0.743)
	NN-DeepSurv	**0.656192 (0.627,0.685)**		NN-DeepSurv	**0.828837 (0.827,0.829)**

Note: Boldface numbers indicate the best performance.

in network and enhancing the ability of the DeepSurv algorithm, the algorithm can be easily solved by ADMM. The combination strategy of DropConnect, SELU activation function and Adam optimization algorithm can effectively enhance the accuracy of the optimization problem on utilizing the deep network.

6. Conclusion

NN-DeepSurv method, based on the nuclear norm and the DeepSurv, was proposed to process survival data with right censoring. The nuclear norm optimization was used to impute missing values in the DeepSurv algorithm, and some new deep learning techniques were applied to get more improvement. The NN-DeepSurv method preserves the advantages of DeepSurv and performs well on large-scale datasets. The performance of the proposed method has been illustrated by comparing our method with the Cox proportional hazards model (Copxph), Cox proportional hazards model with lasso regularization(Cox-lasso), random survival forests(RSF), Xgboost and DeepSurv algorithm on two simulated datasets and six clinical datasets. The results of experiments show that the proposed algorithm is competitive on survival data.

Acknowledgments

The authors would also like to thank Editor-in chief and the referees for their suggestions to improve the article.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

The project was supported by National Natural Science Foundation of China [grant numbers 11971214, 81960309], sponsored by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, Ministry of Education of China, and supported by Cooperation Project of Chunhui Plan of the Ministry of Education of China 2018.

ORCID

Jianyang Tong http://orcid.org/0000-0002-3345-1241
Xuejing Zhao http://orcid.org/0000-0002-0959-2933

References

[1] Zhou H, Li L, Zhu H, Tensor regression with applications in neuroimaging data analysis. J Am Stat Assoc. 2013;108(502):540–552. PMID: 24791032.
[2] Hastie T, Tibshirani R, Friedman J. The elements of statistical learning: data mining, inference, and prediction. 2nd ed. New York: Springer; 2009.
[3] Micci-Barreca D. A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems. SIGKDD Explor. 2001;3(1):27–32.
[4] Khosla A, Cao Y, Lin CC-Y, et al. An integrated machine learning approach to stroke prediction. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '10; New York, NY, USA; 2010. p. 183–192.
[5] Ma G, Zhao X. Regression of survival data via twin support vector regression. Commun Stat Simul Comput. 2020. DOI:10.1080/03610918.2020.1757710
[6] Zhao X, Su J. Variable selection for semiparametric proportional hazards model under progressive type-II censoring. Commun Stat Simul Computat. 2017;46(6):4367–4376.
[7] Jolliffe I. Principal component analysis. New York: Springer; 2002.
[8] Galpin JS, Hawkins DM. Methods of L_1 estimation of a covariance matrix. Comput Stat Data Anal. 1987;5(4):305–319.
[9] Ke Q, Kanade T. Robust L_1 norm factorization in the presence of outliers and missing data by alternative convex programming. In: Proceedings of (CVPR) Computer Vision and Pattern Recognition. Vol. 1; 2005. p. 739–746.
[10] Ding C, Zhou D, He X, et al. R_1-PCA: rotational invariant L_1-norm principal component analysis for robust subspace factorization. In: ICML 2006 – Proceedings of the 23rd International Conference on Machine Learning; 2006. p. 281–288.
[11] Candès EJ, Recht B. Exact matrix completion via convex optimization. Foundat Comput Math. 2009;9(6):717–772.
[12] Ishwaran H, Kogalur UB, Blackstone EH, et al. Random survival forests. Ann Appl Stat. 2008;2(3):841–860.
[13] Hinton G, Srivastava N, Krizhevsky A, et al. Improving neural networks by preventing co-adaptation of feature detectors. Preprint 2012. Available from: arXiv:1207.0580.
[14] Klambauer G, Unterthiner T, Mayr A, et al. Self-normalizing neural networks. In: 31st Conference on Neural Information Processing Systems (NIPS 2017); Long Beach, CA, USA; 2017. p. 972–981.
[15] Kingma D, Ba J. Adam: A method for stochastic optimization. The 3rd International Conference for Learning Representations; San Diego, CA, USA; 2015.
[16] Katzman J, Shaham U, Cloninger A, et al. Deepsurv: personalized treatment recommender system using a cox proportional hazards deep neural network. BMC Med Res Methodol. 2018;18:24.
[17] Wan L, Zeiler M, Zhang S, et al. Regularization of neural networks using dropconnect. In: Proceedings of the 30th International Conference on International Conference on Machine Learning, ICML'13. Vol. 28; 2013. p. 1058–1066.

[18] Bergstra J, Bengio Y. Random search for hyper-parameter optimization. J Mach Learn Res. 2012;13(10):281–305.
[19] Horn RA, Johnson CR. Matrix analysis. 2nd ed. New York: Cambridge University Press; 2012.
[20] Senior A, Heigold G, Ranzato M, et al. An empirical study of learning rates in deep neural networks for speech recognition. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing; 2013. p. 6724–6728.
[21] Harrell FE Jr, Mark DB. Multivariable prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors. Stat Med. 1996;15(4):361–387.
[22] Harrell FE Jr, Pryor DB, Lee KL et al. Evaluating the yield of medical tests. J Am Med Assoc. 1982;247(18):2543–2546.
[23] Bender R, Augustin T, Blettner M. Generating survival times to simulate cox proportional hazards models. Stat Med. 2005;24(11):1713–1723.
[24] Johnson A, Pollard T, Mark R. MIMIC-III, a freely accessible critical care database. Sci Data. 2016 May;24(3):160035.

Appendix

First we need to prove that the L_p norm of a vector x is a convex function. Suppose that the L_p norm of x is $f(x)$, we have $f(x) = (\sum_{i=1}^{n} x_i^p)^{\frac{1}{p}}$. Then we can get the gradient of $f(x)$:

$$\nabla f(x) = \left(\sum_{j=1}^{n} x_j^p\right)^{\frac{1-p}{p}} \left(x_1^{p-1} x_2^{p-1} \cdots x_n^{p-1}\right)^T. \tag{A1}$$

Furthermore, we can get the second-order partial derivative of $f(x)$:

$$\begin{cases} \frac{\partial^2 f}{\partial x_i^2} = (1-p)\left(\sum_{j=1}^{n} x_j^p\right)^{\frac{1}{p}} \frac{1}{x_i^2}\left(\frac{x_i^p}{\sum_{j=1}^{n} x_j^p} - 1\right) \frac{x_i^p}{\sum_{j=1}^{n} x_j^p} \\ \frac{\partial^2 f}{\partial x_i \partial x_k} = (1-p)(\sum_{j=1}^{n} x_j^p)^{\frac{1}{p}} \frac{1}{x_i x_k}\left(\frac{x_i^p}{\sum_{j=1}^{n} x_j^p}\right)\left(\frac{x_k^p}{\sum_{j=1}^{n} x_j^p}\right) \end{cases} \tag{A2}$$

Let $z \triangleq (x_1^p, x_2^p, \ldots, x_n^p)^T$, $A \triangleq diag(\frac{1}{x_1}, \frac{1}{x_2}, \ldots, \frac{1}{x_n})$, it's easy to express the second-order gradient of $f(x)$ as:

$$\nabla^2 f = (1-p) f A^T \left(zz^T - (1^T z)\mathrm{diag}(z)\right) A \frac{1}{(1^T z)^2}. \tag{A3}$$

For L_p norms with $p \geq 1$, to prove the convexity of $f(x)$, we only need to prove that $\forall y \in R^n$, $y^T \nabla^2 f y \geq 0$. This is the truth since:

$$\begin{aligned} y^T \nabla^2 f y &= \frac{(1-p)f}{(1^T z)^2} (yA)^T \left(zz^T - (1^T z) diag(z)\right) Ay \qquad (Let\ q = Ay) \\ &= \frac{(1-p)f}{(1^T z)^2}\left[q^T z z^T q - (1^T z) q^T diag(z) q\right] \\ &= \frac{(1-p)f}{(1^T z)^2}\left[\left(\sum_{i=1}^{n} q_i z_i\right)^2 - \left(\sum_{i=1}^{n} z_i\right)\left(\sum_{i=1}^{n} q_i^2 z_i\right)\right]. \end{aligned} \tag{A4}$$

According to the Cauchy inequality, we have $y^T \nabla^2 f y \geq 0$, which means when $p \geq 1$, the L_p norm of a vector is convex. On the contrary, the L_p norm of a vector is concave when $p < 1$.

For the L_p norm of a matrix, it belongs to the operator norm, which means we can transform the matrix into the vector by multiplying the matrix by a vector, and then we can get the L_p norm of the matrix by calculating the norm of vectors. Specifically, let $\|B\|_p$ be the L_p norm of the matrix B, it's defined as:

$$\|B\|_p = \sup_{x, \|x\|_p = 1} \frac{\|Bx\|_p}{\|x\|_p}. \tag{A5}$$

From the equation (A5), the L_p norm of the matrix is also convex. Specially, $\|B\|_2$ is convex. Because the nuclear norm $\|B\|_*$ is the dual norm of $\|B\|_2$, the nuclear norm $\|B\|_*$ is also convex.