nature computational science

ARTICLES

https://doi.org/10.1038/s43588-021-00083-2

A survival model generalized to regression learning algorithms

Yuanfang Guan[1,2] ✉, Hongyang Li[1], Daiyao Yi[3], Dongdong Zhang[4], Changchang Yin[5], Keyu Li[1] and Ping Zhang[4,5]

Survival prediction is an important problem that is encountered widely in industry and medicine. Despite the explosion of artificial intelligence technologies, no uniformed method allows the application of any type of regression learning algorithm to a survival prediction problem. Here, we present a statistical modeling method that is generalized to all types of regression learning algorithm, including deep learning. We present its empirical advantage when it is applied to traditional survival problems. We demonstrate its expanded applications in different types of regression learning algorithm, such as gradient boosted trees, convolutional neural networks and recurrent neural networks. Additionally, we demonstrate its application in clinical informatic data, pathological images and the hardware industry. We expect that this algorithm will be widely applicable for diverse types of survival data, including discrete data types and those suitable for deep learning such as those with time or spatial continuity.

Survival models are widely applicable in industry and clinical research. For example, one would like to predict which light bulb is likely to fail based on a range of monitoring measurements. In medical research, survival analyses are broadly applied to establish prognostic indices for the mortality of a disease and the outcome of a treatment. Irrespective of their specific application, survival models try to deal with a specific category of problem, where, with the observations available at one particular point or until a particular time point, we try to predict the likelihood of survival based on right-censored data. Right-censoring refers to the data type where the end outcome is a combination of observation time and a binary label: 1 for death/failure or 0 for alive (a censored data point).

The nature of right-censoring makes survival analysis unsuitable for most machine learning algorithms. In a typical machine learning problem, we have a single target to predict, which can be binary or continuous. For right-censored data, two targets (time and status) are concurrently present for each sample. To address this challenge, a proportional hazard model was developed over fifty years ago by Cox[1] to model the effect of multiple covariates on an individual through the hazard function. Since its invention, the Cox model has been the primary method used in survival analysis. In 2008, Ishwaran et al. invented the random survival forest (RSF)[2–4], which takes advantage of the splitting operation of tree-based algorithms to deal with the two targets involved in survival models. A number of other survival models have also been developed, such as the 'accelerated failure time' model[5,6] and exponential and Weibull models[7]. However, they, together with the Cox and RSF models, are all based on a single assumption: the censoring of a sample does not provide any information regarding the prospects of survival beyond the censoring time.

Recent advances in deep learning have allowed it to be built into survival modeling, but the above limitations remain. For example, DeepHit constructs a loss function that includes a binary entropy loss that only takes 1 and 0 binary labels and a ranking loss between 'acceptable pairs', that is, pairs for which the earlier time point individual is dead[8]. This method shares similarity with the Cox model. Another example is the attempt to build neural ordinary differential equations into survival modeling[8,9]. In this method, data from a chunk of time are taken, then this chunk is used to estimate the influence of each parameter by studying the patients that are still at risk. In this sense, it shares similarity with RSF. In both cases, the censoring of a sample will make it ineligible when paired with prospective samples.

We challenge the above practice of not considering an early censored sample when it is paired with another prospective sample, as we consider the time of censoring can be informative, and thus making the relative position of two censoring points or the prospects of survival beyond censoring informative. We also challenge the requirement to fix with a particular regression learning algorithm for survival models. For example, although the Cox model can be generalized to account for the time variability of the importance of features, that is, the 'general hazard rate model', it does not eliminate the assumption of a multiplicative relationship of the hazard ratios.

One open question is how to generalize survival models to datasets that are timewise or spatially continuous and are thus more suitable for either convolutional neural networks or recurrent neural networks (RNNs). For example, if we observe patients during a consecutive time, we would like to build a long–short-term memory (LSTM) to capture time-series information. In this case, neither the Cox nor RSF model, which rely on very specific modeling methods (hazard ratio and random forest, respectively) to predict survival, is suitable. Recent attempts to build deep learning with survival models either extract discrete features from deep learning and then feed into Cox models[10,11] or train with partial log-likelihood loss, as used in Cox[12,13]. Each of these approaches has limitations. By creating an intermediate step that extracts features, we do not fully maximize the potential of deep learning to extract high-level information. By training with a partial log-likelihood loss, we again assume a likelihood relationship and, due to the batch training in deep learning,

[1]Department of Computational Medicine and Bioinformatics, Michigan Medicine, University of Michigan, Ann Arbor, MI, USA. [2]Department of Internal Medicine, University of Michigan, Ann Arbor, MI, USA. [3]Department of Biomedical Engineering, Michigan Medicine, University of Michigan, Ann Arbor, MI, USA. [4]Department of Biomedical Informatics, The Ohio State University, Columbus, OH, USA. [5]Department of Computer Science and Engineering, The Ohio State University, Columbus, OH, USA. ✉e-mail: gyuanfan@umich.edu

the log-likelihood calculation could be questionable for a single batch. More broadly, how could we establish a statistical modeling that allows adaptation to any kind of regression learning algorithm (for example, linear, boosting trees, Gaussian process regression, support vector vachines and many others)?

In this study, we present a uniformed model that generalizes right-censored data to a standard regression problem, which allows the application of any type of regression learning algorithm to a survival prediction problem. We explain the theoretical basis of its advantage. We demonstrate its application in clinical informatics, image and time-series data in medical applications and industry cases, involving gradient boosted trees, convolutional neural networks and RNNs. We envision this method will become an important reference for survival analysis, particularly for datasets that are more suited to classifiers beyond hazard ratio and random forest.

Results

Generalization of right-censored data to a regression problem. For typical right-censored data, the first column is the sample/patient ID and the second column is the last date when the sample/patient is observed. The third column is the status of the sample/patient at its last observation date. If the status is 1, the sample failed (or the patient died) at that date. If the status is 0, the sample has not failed (or this patient is alive) on the last observation day and this data point is termed 'censored'. For a sample that is censored at a particular date, we have no information whether this sample will fail on the next day or 10 years later. Because not all samples have failed at the last observation date, generic regression methods cannot be used on this type of censored data. Our goal is to transform this two-target (status and time) data into a generic regression problem.

The goal of our model, which we refer to as a complete rank method, is to assign each training sample with a single value through a uniformed scheme. This single value will be the target in the regression, and any regression learning algorithm, including deep learning, can be integrated later. It is not intuitive how this can be done with two targets, time and status at the beginning, and direct ranking of any or a subset of any is erroneous. However, ranking is not only achievable by sorting a single array of numbers—it can also be achieved by thorough pairwise comparison among all samples. For example, if we have 100 balls, each of a specific size, we can acquire their ranking by ordering them by size. Alternately, we can acquire a specific ball's relative ranking by comparing it against every other ball along the survival curve (Fig. 1a,b). This property of ranking allows us to give a complete rank of samples in a censored dataset, by assigning the probabilities of one sample ranking ahead of another sample, when the absolute relative ranking is ambiguous due to censoring.

We now break down the right-censored dataset into different situations of pairwise comparisons. Let us order the samples so that those that are going to fail early will end up having a higher target value. We use T to denote last observation time, no matter whether the sample is censored or not, and use S to denote status ($S=1$ indicates a failed case and $S=0$ indicates a censored case). A pair of samples A and B may have the following four possibilities when $T_A < T_B$ (Fig. 1c):

$$S_A = 1 \text{ and } S_B = 0 \qquad \text{Case 1}$$

$$S_A = 1 \text{ and } S_B = 1 \qquad \text{Case 2}$$

$$S_A = 0 \text{ and } S_B = 1 \qquad \text{Case 3}$$

$$S_A = 0 \text{ and } S_B = 0 \qquad \text{Case 4}$$

When $T_A = T_B$:

$$S_A = S_B \qquad \text{Case 5}$$

$$(S_A = 1 \text{ and } S_B = 0) \text{ or } (S_A = 0 \text{ and } S_B = 1) \qquad \text{Case 6}$$

For case 1 and case 2, we add 1 to the rank of A and add zero to the rank of B, as we know A failed before B. Both case 1 and case 2 are sufficiently considered in the Cox model and RSF, but cases 3 and 4 are not considered in the maximal likelihood function in the Cox model or in the log-rank test in RSF or any other deep learning-based method discussed above.

We then calculate the Kaplan–Meier (K–M) curve. This curve gives an estimation of the survival function over time, $r(t)$, which is the proportion of non-failed cases at time t. For case 3, B failed at T_B. If A, which is censored at T_A, fails between T_A and T_B, A should have a higher rank than B; otherwise, B ranks above A. We cannot determine a binary relative ranking of this pair. However, we can derive the probability that A fails between T_A and T_B and thus rank above B using the K–M curve:

$$P = \frac{r(T_A) - r(T_B)}{r(T_A)} \tag{1}$$

which is the probability that A fails before B, and is the value added to the rank of A:

$$P = \frac{r(T_B)}{r(T_A)} \tag{2}$$

which is the probability that B fails before A, and is the value added to the rank of B.

For case 4, both samples are censored, and we first calculate the probability of A failing before the observation time of B (instead of before B fails, as we do not know when B will fail due to its censored status):

$$P^* = \frac{r(T_A) - r(T_B)}{r(T_A)} \tag{3}$$

The probability of A failing after the observation time of B would the be $1-P^*$. As we know that B fails after T_B, without any other information, the probability of B failing before A would be $0.5\times(1-P^*)$, which is added to the rank of B; that is, the chance is equal after T_B. The probability of A failing before B is $P^*+0.5\times(1-P^*)$, which is added to the rank of A, that is, the chance that A fails before T_B, and 50% chance after reaching time point B.

For case 5, as both the time and status are the same for samples A and B, we add 0.5, respectively, to their ranks. For case 6, as the time is the same but the status differs, we add 1 to the rank of the failed case.

Up to this point, we are able to complete all comparisons between all types of data point pairs in the censored data, and thus we are able to acquire a relative ranking of the likelihood of failing for all samples in the training set. By dividing this vector by the total number of examples, we acquire a vector with values between 0 and 1, representing the relative likelihood of a sample failing. This is our final target that we can build into any regression learning algorithm, be it random forest, gradient boosted trees, linear regression, neural network, Gaussian process regression or support vector machine, and it is no longer limited to datasets whose features are vectors, as will be demonstrated in the following.

Experiments with simulated survival data. We will demonstrate the robustness of the complete rank method, compared to the Cox and

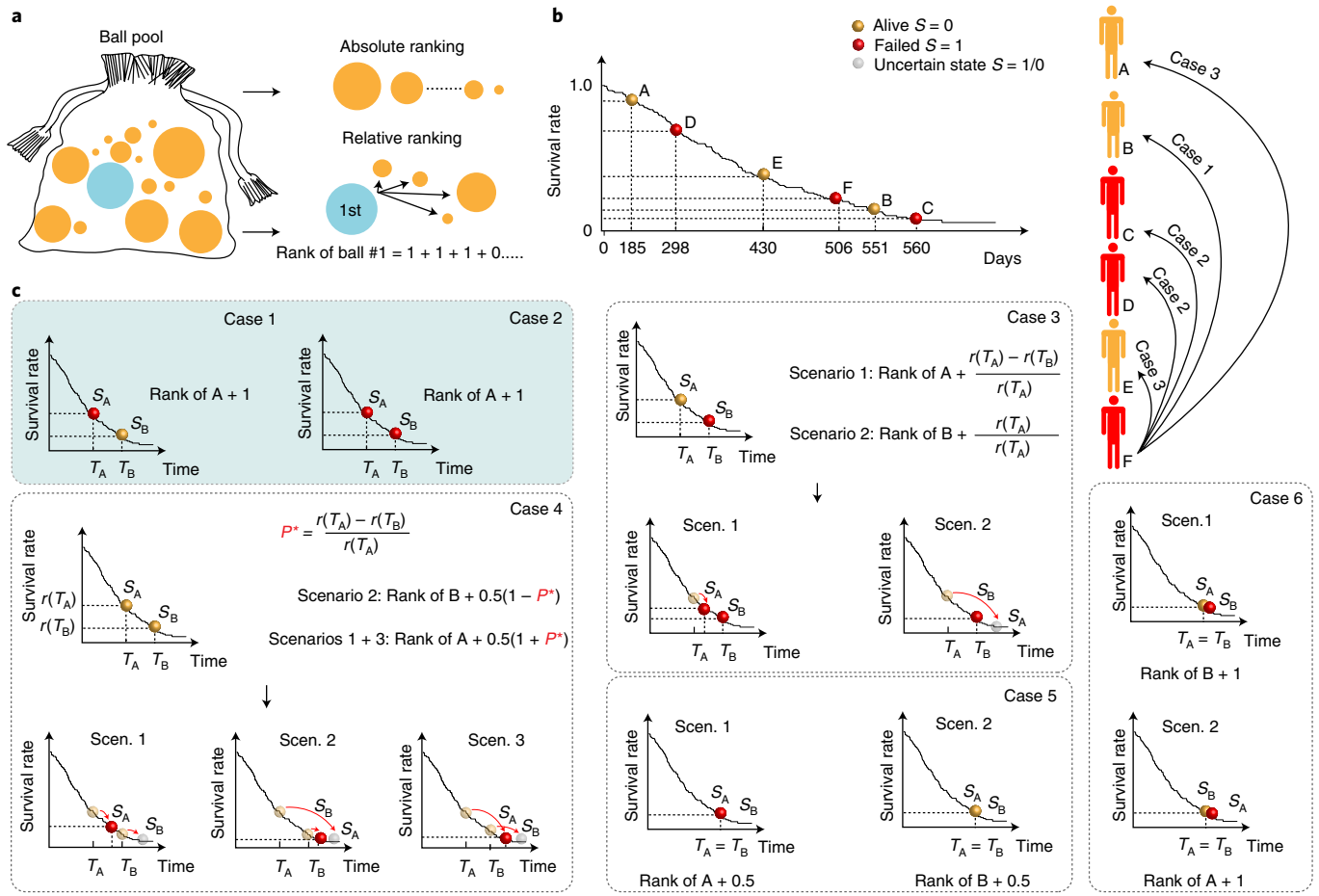

Fig. 1 | Generalizing right-censored data analysis to a regression problem by complete rank. a, Ranking can be achieved by either assigning each sample a unique value (absolute ranking) or comparing one example against every other example. **b**, Using a survival curve and individual F as an example, we illustrate three of the possible comparison scenarios in right-censored data: an early failed (F)–late censor pair (B); an early censor (A/E)–late failed pair (F); an early death (F)–late failed pair (C). **c**, Calculation of the ranking values added to each example in every paired case of censor and death combinations. Cases highlighted with a light cyan background indicate those already considered by existing survival models.

RSF models, using a simulated survival dataset, in which the probability that a sample fails on a particular day is correlated with a set of parameters (Fig. 2a). We assume a population with mean daily death rate ρ_{mean}, and standard deviation σ. The hidden risk of each individual r_i is sampled from $N(\rho_{\text{mean}}, \sigma)$, and we force the minimal risk to be zero. We assume a total of n examples, and m features that can be used to predict survival. Each feature j is parameterized by noise factor ε_j, uniformly drawn from $[0, \beta]$. Thus, the maximal level of noise of a simulated dataset is controlled by β and the noise level of an individual feature j is controlled by ε_j. The value of the jth feature of the ith sample, v_{ij}, is further parameterized by θ, which is uniformly distributed in $[-0.5, 0.5]$ and thus introduces a different noise value to each v_{ij}, to incorporate the noise factor to each example:

$$v_{ij} = r_i \times f_j(1 + \theta \times \varepsilon_j) \tag{4}$$

where j refers to the jth feature and f_j represents the scaling factor of a particular feature, uniformly drawn from $[0, \alpha]$. The above generated feature set will have the following properties. Each feature is correlated with the death rate, and the correlation is determined by ε_j, where the higher the ε_j, the less overall correlation between this feature and the death rate. The scale of a feature is driven by f_j, where the bigger the scaling factor, the bigger the deviation this feature will have across individual samples. Assuming that we start with a population of individuals that are all alive and create the death rate for each individual, we can generate an $n \times m$ matrix as features for these individuals, where n is the number of individuals and m is the number of features.

Next, we simulated the censoring status of the data. First, we assumed a maximal date of observation for all examples, $\delta_{\max}$. Between $[0, \delta_{\max}]$ days, we created a binomially distributed vector of length $\delta \sim B(\delta_{\max}, \rho_i)$, where ρ_i is the failing rate of sample i. The first occurrence of 1 in this binomially distributed vector defines the death date T_f. Next, we uniformly sampled, between $[0, \delta_{\max}]$, the censoring date T_c. If $T_c < T_f$, the sample is censored, with a status of 0 and censoring time T_c; otherwise, the sample failed, with a status of 1 and time to event of T_f.

We started with $\delta_{\max} = 1{,}000$, $\rho_{\text{mean}} = 0.0001$, $\sigma = 0.0002$, $\beta = 1$, $\alpha = 100$, $n = 1{,}000$ and $m = 100$. This created a dataset with 1,000 samples and 100 features, at a scale similar to that of commonly seen survival data. Approximately 12.5% of the samples failed, and the rest are either censored or did not fail even on day 1,000. Unless otherwise specified for testing model robustness, these are the base parameters we used. For the ranking method, we built in extreme gradient boosted trees, which cannot otherwise be used for survival models. With this starting point, we first checked several expected behaviors of the model. First, as σ increases, the performance of all models increases because the difference of risks among individuals increases (Supplementary Fig. 1). Second, the scaling factor α, which only changes the scale of the features, does not affect

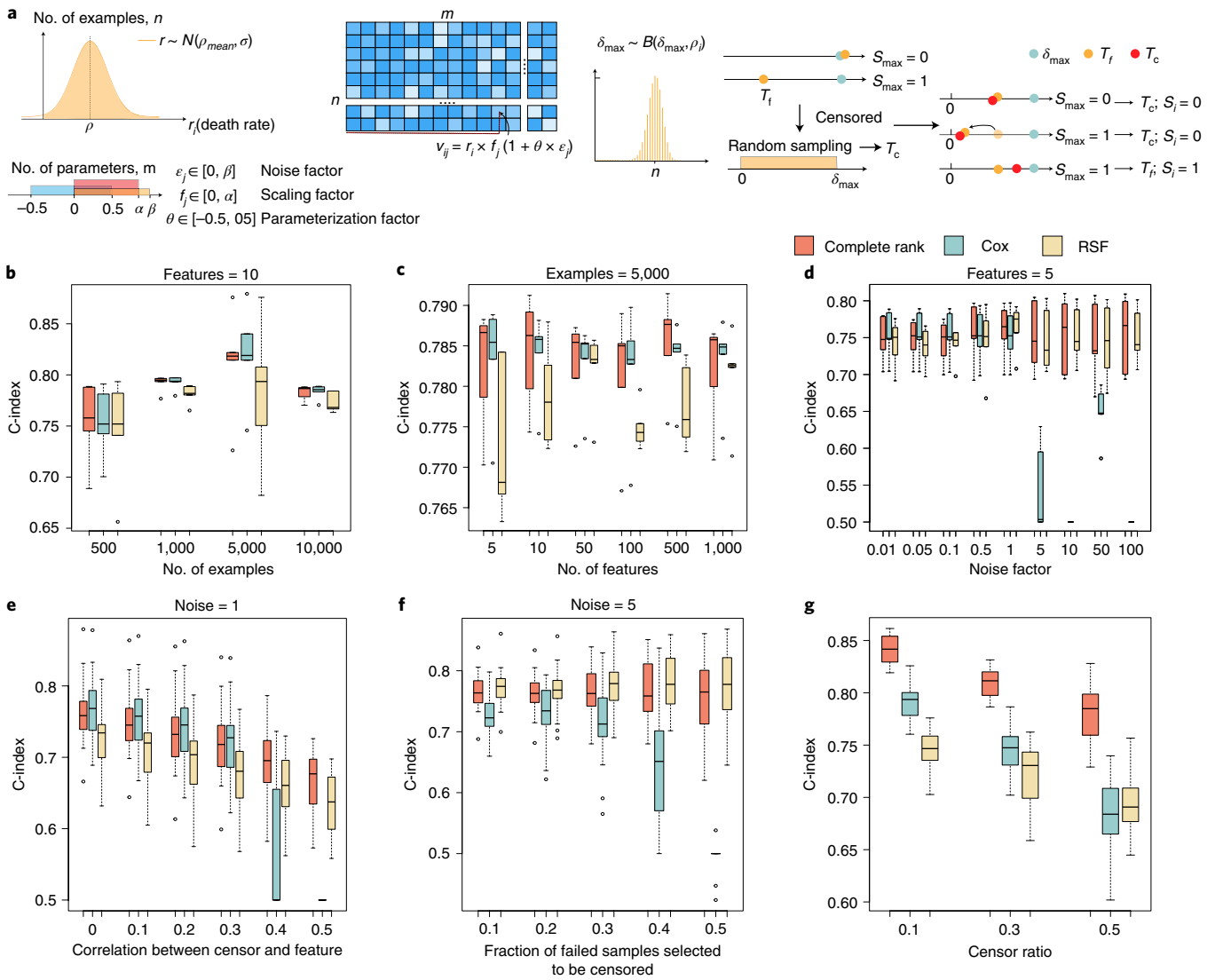

Fig. 2 | Simulation experiments for complete rank in different scenarios. a, Simulation scheme. We first randomly draw a death rate for each sample. Next, the feature matrix (blue) is created by incorporating the noise level, a random scaling factor and a random parameterization factor. We then simulate censoring by a uniformed selected censoring date, while the death date is binomially distributed. **b**, Performance comparison across different numbers of examples used in training. **c**, Performance comparison across different numbers of predictive features used for the training. **d**, Performance comparison across different ranges of noise factors, representing the noise levels of the data. **e**, Performance comparison across different levels of correlation between censoring rates and the values of a feature at a noise level of 1. A total of five features were used. **f**, Performance comparison across different levels of censoring (defined by the fraction of failed examples that are censored). **g**, Performance comparison across different censoring rates in Cox-specified simulation. The plots in **b**–**d** show fivefold cross-validation ($n=5$) and in **e**–**g** fivefold cross-validation repeated four times ($n=20$). All box plots represent the maximal, minimal, median, 25% and 75% of the distribution.

performance (Supplementary Fig. 2). Third, the performance in general increases as the number of examples increases until 5,000 individuals, but RSF seems to have a disadvantage when the number of examples is extremely large (Supplementary Fig. 3 and Fig. 2b,c).

We found that a major factor affecting model performance is the noise factor ε_j, uniformly drawn from $[0, \beta]$ (Supplementary Figs. 4 and 5 and Fig. 2d). Both the ranking method and the RSF performed robustly when β increased, until 100 (high noise in features), but the performance of the Cox model dropped towards random performance. This indicates that the Cox model is unable to pick up important features and upweight them when the data contain a lot of noise. Across all cases, the complete rank method performed well (Fig. 2 and Supplementary Figs. 1–8).

To evaluate the validity and robustness of the method when censoring depends on the features, we simulated cases where the censoring possibility of an individual is correlated to one input feature. We tested in the range between 0 correlation and 0.5 correlation of the censoring rate to one feature (among a total of five features). Although all models drop performance as the correlation increases, the complete rank method showed a substantial advantage in alleviating this confounding factor and demonstrated robust performance (Fig. 2e and Supplementary Fig. 9). We also tested the effect of censoring on the performance by allowing 10–50% of the failed examples to be censored, and found that the complete rank method is robust to different censoring levels (Fig. 2f and Supplementary Fig. 10).

To further evaluate the robustness of the method when the simulation is Cox-specified, we resimulated the survival data following a previous work that first specifies a base survival function and then applies hazard ratios of parameters on top of it[14]. In particular, we

restricted the simulation to proportional hazards. Complete rank demonstrated superior performance across the following parameters. First, complete rank performs well across all censoring rates, but showed a greater advantage at a high censoring rate (Fig. 2g). Second, although all performances increased as the data contained more examples, or more features, complete rank remained top across a large spectrum of parameters (Supplementary Figs. 11 and 12). Third, the performance of complete rank is robust when censoring depends on features (Supplementary Figs. 13 and 14). Fourth, the performance of complete rank is robust against how discriminative the features are (Supplementary Fig. 15). In this set of simulations, we first simulated the baseline hazard function and then simulated features according to proportional hazards, which is correctly specified to the Cox model. The performance of the complete rank method in this set of simulation supports its robustness across different scenarios.

Overall, the calculation of the rank function was swift, taking ~30 s for 10,000 samples (Supplementary Fig. 16), making it suitable for a wide range of computation tasks.

Prediction of cancer survival using histological images and clinical information. In this section, we demonstrate how the same training targets generated by the ranking method can be used to train two drastically different types of feature data to predict survival—medical images and clinical measurements—by using different regression learning algorithms. Here, we used hematoxylin and eosin (H&E) images from the The Cancer Genome Atlas (TCGA) database[15] using the Genomic Data Commons (GDC) Data Portal, including breast invasive carcinoma (BRCA), colon adenocarcinoma (COAD), kidney renal clear cell carcinoma (KIRC) and liver hepatocellular carcinoma (LIHC). We integrated the complete rank and cancer images into the deep learning models. Cancer cases with at least one histopathological image were included in the study. We also downloaded the clinical data used for computing the complete rank. A total of 2,453 individuals (BRCA, 1,084; COAD, 457; KIRC, 537; LIHC, 375) and 6,201 histopathological images (BRCA, 3,070; COAD, 983; KIRC, 1,656; LIHC, 492) were used in the study. Each individual could have multiple images. We carried out fivefold cross-validation, separated by individuals, to evaluate the performance of the method.

We used a typical deep learning architecture for training pathological section images (Fig. 3b), which consisted of a series of convolutional MaxPool blocks, connected to a final dense layer. The training target was the complete rank scores of the training set. Because each patient had more than one image in the database, the survival prediction for each patient was calculated by averaging predictions from each image. Through cross-validation, we demonstrated that pathological images can predict survival with an average C-index (concordance index) between 0.525 and 0.634. For liver cancer, pathological images are not very predictive of survival, but for the other three types, pathological images can provide information for survival. Although the performance, per se, was not high, which is expected as we are dealing with histological images, which may not directly provide survival information, this analysis supports the potential of integrating complete rank with images to establish survival models. It is now widely recognized that, due to spatial continuity, deep learning (convolutional neural networks) has a great advantage over other methods in analyzing image data. The complete rank method described above will allow seamless integration of survival models with deep learning to analyze images.

The rank scores are, in fact, flexible to being built with any regression learning algorithm and with any feature data. For demonstration, we also constructed survival models on the clinical data (gender, race, age, tumor stage and primary diagnosis) using these rank scores. The regression learning algorithm is LightGBM, a tree-based algorithm. We achieved C-index values above 0.7 in fivefold cross-validations for three out of four cancer types (not LIHC; Fig. 3c). This result is consistent with the trend of deep learning-based image models, where liver cancer is the hardest to predict, possibly because the number of samples is relatively small[15]. As expected, direct clinical observations resulted in better-performing survival models than histological images. We also combined the image and clinical models and observed further improvement. The average C-index values of fivefold cross-validation were 0.756, 0.714, 0.757 and 0.607 for BRCA, COAD, KIRC and LIHC, respectively. This example demonstrates the flexibility of the complete ranking when applied to both image and classical data types and built with diverse regression learning algorithms.

Prediction of disk failure using time-series data. To further demonstrate the flexibility of the algorithm when applied to a different, industrial, setting and time-series dataset, we examined its application to reliability data for hard disks, in comparison to binary labels of failure and alive. We downloaded 2013–2015 Backblaze disk failure data[16]. Specifically, the Backblaze data record the status of disks every day. Each hard disk is assigned a unique ID. If a disk fails on a particular date, the status will be labeled as 1 on that date, and this unique ID will disappear in all following days as the disk is replaced by a new one in the computer cluster. On each day, a total of 86 features are included to describe the physical characteristics of the disks, such as storage, type and the running status of the disks—all continuous features. If a disk ends on the last day of 2015 with a status of 0, it means the disk still runs fine on that date.

We are interested in this question: on a specific date, given all existing data from previous days, what is the risk of disk failure? We first created a relatively unbiased censored dataset from the data. A potential bias in the data is that a large proportion of the disks start with the date 10 April 2013, while others started at other dates in distinct batches. As hard disks tend to have different failing rates by batch, we do not want the model to know which batch/date the disks come from. We thus uniformly sampled from $[0, T_{\text{last}}]$ for each sample to create a new start date, T_{start}, where T_{last} is the last observation date or the failure date of a disk (the date where we have the survival status). Between T_{start} and T_{last}, we randomly select a date and designate it as T_{stop}, which is the date until which we have access to the observed features; that is, we make predictions at T_{stop}. Thus, the input feature contains the 86 features from T_{start} to T_{stop} and the output observation is time $T_{\text{last}} - T_{\text{stop}}$ and status. The output observations are used to create the rank scores in the training set, and the evaluation gold standard for the test set. Unlike the cancer study presented above, we not only have the baseline 86 features on T_{start}, but also all time-series features until T_{stop}, totaling $86 \times (T_{\text{stop}} - T_{\text{start}})$ features.

RNNs are typically used to extract information from such time-series data, and, in this case, the rank scores allow us to build in RNNs. We constructed the RNN model, building on the time-series features between T_{start} and T_{stop}. Briefly, this network has a bidirectional LSTM (or BiLSTM) layer[17] followed by two fully connected layers plus one rectified linear unit (ReLU) layer. We compared two strategies of training, one directly trained against the final status of the disk (failure as 1 and working as 0) and the other trained with the rank score created from the training set. Through cross-validation, we found that training with the rank improved the performance of predicting disk failure on the test set when the numbers of training samples were small (Fig. 4b). Of note, no existing survival model can build in these RNNs. Furthermore, one important advantage of the ranking method we present here is fully utilizing all training data, particularly the censored group. We observed a greater advantage of the ranking method over the binary training gold standard when the training set became smaller, corroborating the above argument. This experiment demonstrates the generalizability of the method presented in this study to time-series data.

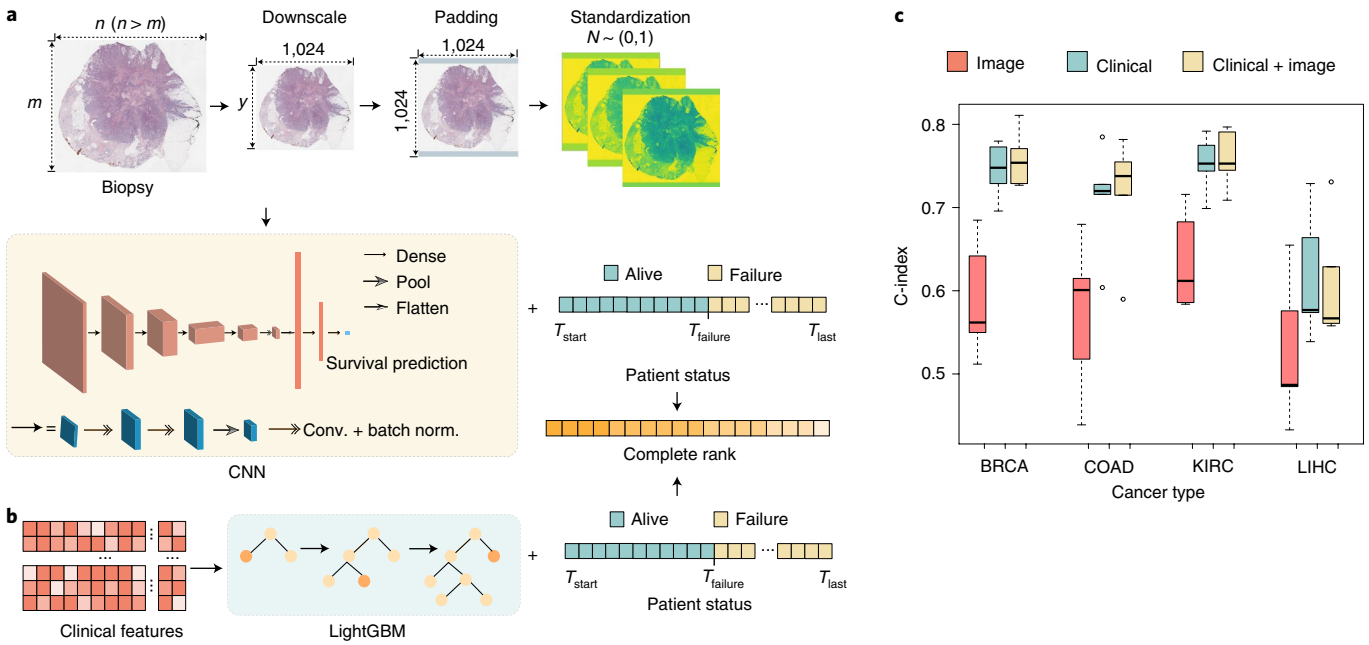

Fig. 3 | Building complete rank with deep learning and LightGBM to predict cancer survival using histological images and clinical information. a, Images are downsized or padded to 1,024 × 1,024 and standardized by each channel and fed into a convolutional neural network. This allows us to use the rank score inferred from patient status to train the survival models using deep learning. **b**, LightGBM models based on clinical features (orange and pink matrix). **c**, Performance in four cancer types: BRCA, COAD, KIRC and LIHC. All box plots represent the maximal, minimal, median, 25% and 75% of the distribution of fivefold cross-validation ($n = 5$). CNN, convolutional neural network.

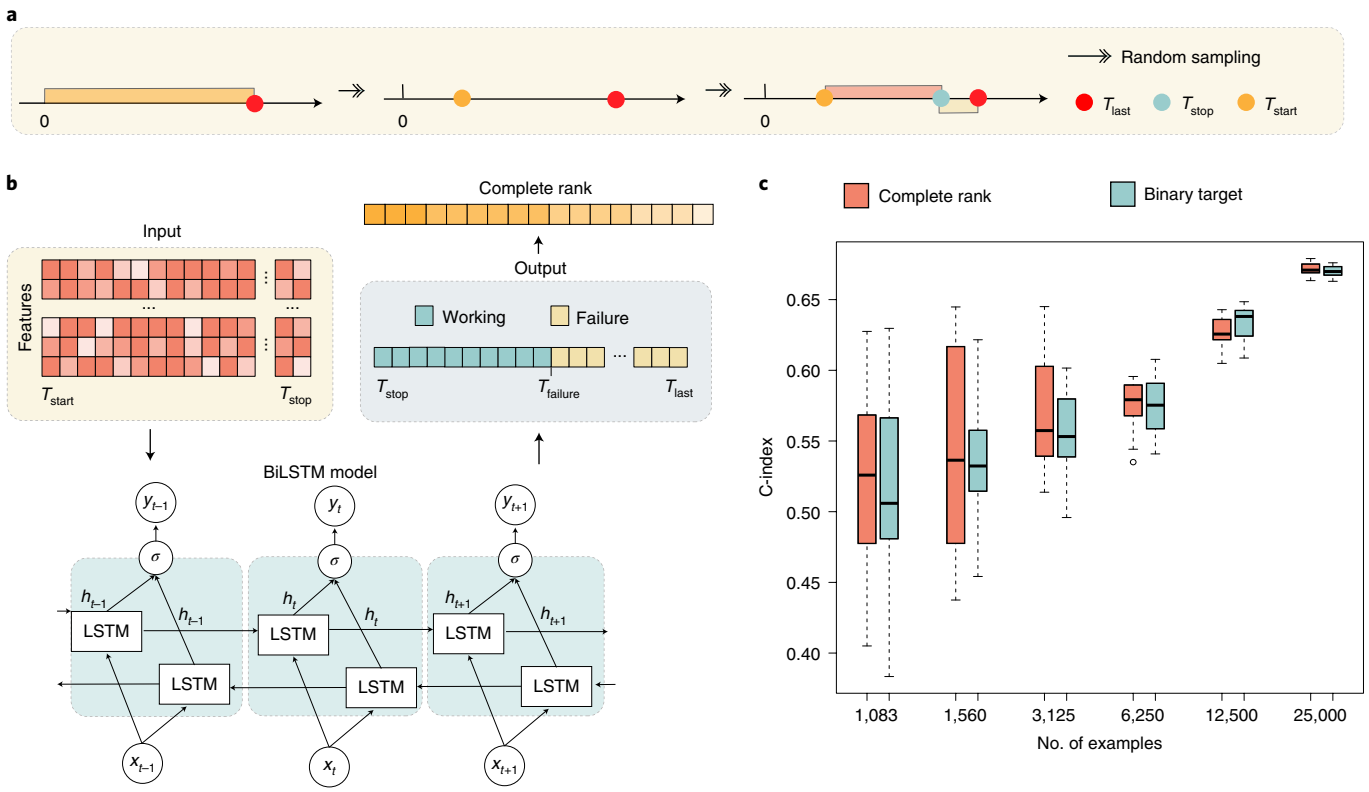

Fig. 4 | Building complete rank with RNNs (LSTM) to predict disk failure using time-series data. a, Sampling of time-series data and censored data. The record start date is selected randomly. The last date is either the failure date or a randomly selected end date (after the start date), whichever occurs first. A stop date is then selected between the start and last date to define the observation cutoff date. **b**, Time-series data are encoded into the BiLSTM model and the censored information is converted to the rank score for training. **c**, Comparison of use of rank scores and binary targets when the sample size is very small. All box plots represent maximal, minimal, median, 25% and 75% of the distribution of fivefold cross-validation repeated four times ($n = 20$).

Discussion

We consider that this approach addresses two limitations present in existing survival models. The first is the assumption of the relationship between the features and the outcome. For example, the Cox model assumes a multiplicative relationship of hazard ratios. The accelerated failure time model assumes that the effect of a covariate is to accelerate or decelerate over time by a constant[18]. Such modeling approaches are very different from machine learning approaches and thus prevent the integration of the majority of regression learning algorithms. The other example is RSF, in which random forest is forced to be the algorithm. The complete rank method allows flexible integration of any supervised machine learning algorithms, including deep learning. Using this method, we will be able to test all supervised regression learning algorithms.

The other assumption we question in existing survival models is the contribution of censored points, in which the early-censored–late-uncensored pairs and early-censored–late-censored pairs are mostly not considered in the modeling. We argue that the differences in censoring time, in combination with the K–M curve, can provide meaningful information to the model. We envision future wide application of this method in industry and medical sciences.

One limitation of this study is that the rank score does not directly predict time to event, which warrants future investigation. Furthermore, we have focused on individualized predictions of survival status. Often, we encounter in clinical and industrial settings a situation where a population-derived overall distribution of the expected survival status is expected. Estimating the distribution of a new population using an existing population's observations is of great use in practice, but not as yet well-formulated or explored in the literature. Future development of this and other methods to population-wise studies will be valuable.

Methods

Evaluation of cross-validation performance. Survival models are typically evaluated using the C-index. We used the survival package of R to calculate the C-index values. All evaluations were carried out by standard fivefold cross-validation in this study.

Cox-specified simulation. To test the robustness of the algorithm, we used sim.survdata in R to simulate a set of data that are specified to Cox. We started with 1,000 examples, 50 features, a discriminative level of 2.5 and a censoring rate of 0.5 and then scanned across all parameters to evaluate the relative performance of Cox, complete rank (linear regression as the regression learning algorithm) and RSF.

Training deep learning models for cancer images. Separate models were built for each type of cancer. The training, validation and testing tests were split based on cases (rather than images) in a ratio of 6:2:2. The validation set was used to call back the best model and prevent overfitting. We used a nested training strategy to improve the robustness of the model and make full usage of the training set. Specifically, for each model, five models were trained in parallel based on different training and validation data partitions. Finally, the predictions on the test set from five models were assembled as the final prediction of the model.

After retrieving whole-slide images from TCGA, several image preprocessing steps were conducted. We scaled the long edge of each image into 1,024 pixels with fixed aspect ratios and padded the entire image to 1,024×1,024. We normalized the pixel values for each channel.

We trained the network with the architecture presented in Fig. 3a. Specifically, we built a convolutional neural network with a total of 14 convolutional layers and four max-pooling layers. We first added a convolution-convolution-pooling block consisting of two convolutional layers with kernel size of 3 and one max-pooling layer with kernel size of 2. This convolution-convolution-pooling block reduced the image size by half and four blocks were used to gradually change the input size from 1,024×1,024 to 64×64. Six additional convolutional layers were added to gradually reduce the number of channels to 1. In each of the convolutional layers, ReLu activation was used to introduce nonlinearity. The last convolutional layer was flattened and two dense layers were used to generate the final output with 'sigmoid' activation. We used mean squared error loss, as the ranking value is continuous. We used the Adam optimizer and a batch size of 10. We first trained the neural network model with a relatively large learning rate of 1×10^{-3} for 20 epochs, then continued to train another 40 epochs with a smaller learning rate of 1×10^{-4}, resulting in a total of 60 epochs. To avoid potential overfitting, we further augmented the training data by multiplying all pixel values by a random number between 0.90 and 1.15. We also randomly flipped the training images horizontally and/or vertically. The model was implemented in Tensorflow.

Training LightGBM models for clinical informatics data. For the clinical features (gender, race, age, tumor stage and primary diagnosis), we built LightGBM models to predict survival. Specifically, we one-hot encoded all clinical features that were categorical, then trained LightGBM models with a maximum of 500 boosting rounds and applied the early stopping strategy if the loss did not drop further for 20 consecutive rounds. We set the number of leaves to 5 to control the complexity of the tree models, and the minimum number of data within a leaf was 3. We used a bagging fraction of 70%.

Training of LSTM models for disk failure data. For both the classification method with binary labels and the ranking method, we used a neural network with a BiLSTM layer[17] followed by two fully connected layers plus one ReLU layer. The BiLSTM layer concatenated the outputs from two hidden layers of opposite direction to the same output and learned bidirectional long-term dependencies of the time-series data. Two fully connected layers were used to produce the prediction and the input sizes were 512 and 100, respectively. We used a learning rate of 0.001, batch size of 32, 50 total epochs and used 20% of the training data as the validation set to call back the best model. The model was implemented in PyTorch and shared in the GitHub repository.

Data availability

Simulated data are available at from GitHub (https://github.com/GuanLab/GuanRank_All). TCGA data[15] are third party and downloadable from their websites using the Genomic Data Commons (GDC) Data Portal[19]. Backblaze disk failure data are third party and downloadable from the Backblaze harddrive data and stats website[16]. Source data are available with this paper.

Code availability

Source code is available at https://github.com/GuanLab/GuanRank_All (ref. [20]). No restriction is placed on access.

Received: 29 January 2021; Accepted: 13 May 2021;
Published online: 21 June 2021

References

1. Cox, D. R. Regression models and life-tables. *J. R. Stat. Soc. B* **34**, 187–202 (1972).
2. Ishwaran, H. The effect of splitting on random forests. *Mach. Learn.* **99**, 75–118 (2015).
3. Ishwaran, H., Kogalur, U. B., Blackstone, E. H. & Lauer, M. S. Random survival forests. *Ann. Appl. Stat.* **2**, 841–860 (2008).
4. Ishwaran, H., Kogalur, U. B., Chen, X. & Minn, A. J. Random survival forests for high-dimensional data. *Stat. Anal. Data Min.* **4**, 115–132 (2011).
5. Kalbfleisch, J. D. & Prentice, R. L. in *The Statistical Analysis of Failure Time Data* 328–374 (Wiley, 2011); https://doi.org/10.1002/9781118032985.ch11
6. Wei, L. J. The accelerated failure time model: a useful alternative to the Cox regression model in survival analysis. *Stat. Med.* **11**, 1871–1879 (1992).
7. Aitkin, M. & Clayton, D. The fitting of exponential, Weibull and extreme value distributions to complex censored survival data using GLIM. *J. R. Stat. Soc. C* **29**, 156–163 (1980).
8. Lee, C., Yoon, J. & van der Schaar, M. Dynamic-DeepHit: a deep learning approach for dynamic survival analysis with competing risks based on longitudinal data. *IEEE Trans. Biomed. Eng.* **67**, 122–133 (2020).
9. Quirós, A., de Prado, A. P., Montoya, N. & Hernández, J. Multi-state models for the analysis of survival studies in biomedical research: an alternative to composite endpoints. In *Proc. 13th International Joint Conference on Biomedical Engineering Systems and Technologies* (eds De Maria, E. et al.) 194-199 (BIOSTEC, 2020); https://doi.org/10.5220/0009105701940199
10. Cui, L. et al. A deep learning-based framework for lung cancer survival analysis with biomarker interpretation. *BMC Bioinf.* **21**, 1–14 (2020).
11. Ren, J., Singer, E. A., Sadimin, E., Foran, D. J. & Qi, X. Statistical analysis of survival models using feature quantification on prostate cancer histopathological images. *J. Pathol. Inform.* **10**, 30 (2019).
12. Li, H. et al. Deep convolutional neural networks for imaging data based survival analysis of rectal cancer. *Proc. IEEE Int. Symp. Biomed. Imaging* **2019**, 846–849 (2019).
13. Ching, T., Zhu, X. & Garmire, L. X. Cox-nnet: an artificial neural network method for prognosis prediction of high-throughput omics data. *PLoS Comput. Biol.* **14**, e1006076 (2018).
14. Harden, J. J. & Kropko, J. Simulating duration data for the Cox model. *Political Sci. Res. Methods* **7**, 921–928 (2019).
15. Weinstein, J. N. et al. The Cancer Genome Atlas Pan-Cancer analysis project. *Nat. Genet.* **45**, 1113–1120 (2013).
16. Backblaze. Hard Drive Data and Stats 2013–2015; https://www.backblaze.com/b2/hard-drive-test-data.html

17. Schuster, M. & Paliwal, K. K. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.* **45**, 2673–2681 (1997).
18. Swindell, W. R. Accelerated failure time models provide a useful statistical framework for aging research. *Exp. Gerontol.* **44**, 190–200 (2009).
19. National Cancer Institute. Genomic Data Commons Data Portal; https://portal.gdc.cancer.gov/
20. Guan, Y. GuanRank code (version 1.0.0) (Zenodo, 2021); https://doi.org/10.5281/zenodo.4751702

Acknowledgements

Y.G. is supported by the NIH (R35-GM133346) and the NSF (#1452656).

Author contributions

Y.G. conceived and implemented the complete rank algorithm, simulation and LSTM experiments and wrote the manuscript. D.Y. created the figures. H.L. and K.L. carried out cancer image experiments. D.Z., C.Y. and P.Z. performed LSTM experiments. All authors read and approved the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s43588-021-00083-2.

Correspondence and requests for materials should be addressed to Y.G.

Peer review information *Nature Computational Science* thanks the anonymous reviewers for their contribution to the peer review of this work. Handling editor: Fernando Chirigati, in collaboration with the *Nature Computational Science* team.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.