BMC Bioinformatics

SOFTWARE

Open Access

AFFECT: an R package for accelerated functional failure time model with error-contaminated survival times and applications to gene expression data

Li-Pang Chen[1*] and Hsiao-Ting Huang[1]

*Correspondence:
lchen723@nccu.edu.tw

[1] Department of Statistics, National Chengchi University, Taipei, Taiwan, ROC

Abstract

Background: Survival analysis has been used to characterize the time-to-event data. In medical studies, a typical application is to analyze the survival time of specific cancers by using high-dimensional gene expressions. The main challenges include the involvement of non-informaive gene expressions and possibly nonlinear relationship between survival time and gene expressions. Moreover, due to possibly imprecise data collection or wrong record, measurement error might be ubiquitous in the survival time and its censoring status. Ignoring measurement error effects may incur biased estimator and wrong conclusion.

Results: To tackle those challenges and derive a reliable estimation with efficiently computational implementation, we develop the R package `AFFECT`, which is referred to Accelerated Functional Failure time model with Error-Contaminated survival Times.

Conclusions: This package aims to correct for measurement error effects in survival times and implements a boosting algorithm under corrected data to determine informative gene expressions as well as derive the corresponding nonlinear functions.

Keywords: Boosting, Gene expression, Measurement error, Survival analysis

Background

Survival analysis has been a useful tool to analyze time-to-event data. In applications of medical studies, researchers are interested in a specific cancer and wish to understand the failure time and survivor pattern of a specific cancer among all observations in a study. Typically, gene expressions from subjects are usually taken as covariates and are used to characterize the time-to-event responses (e.g., [20]). In the framework of survival analysis, the accelerated failure time (AFT) model is one of popular approaches, which is formulated in the parametric setting in most applications (e.g., [16]). In the literature, a large body of methods has been proposed to deal with the AFT model, such as [1, 23], and [36]. However, the covariates are possibly nonlinear with respect to the

survival time, then using conventional parametric AFT models may incur model misspecification. To relax the linear constraint, nonparametric approaches should be taken into account to address the nonlinear estimation between the survival time and the covariates. In recent years, a boosting method, one of statistical learning approaches, has been popular to address nonparametric estimation. The basic idea is to fit base learner many times on reweighed data to boost the performance, and the final estimator is formed by the linear combination of the multiple estimates. In the framework of survival analysis, several approaches have been proposed under various models, such as [3, 10, 12, 18, 19, 33].

The other challenging feature in datasets is measurement error, which indicates that the observed variables do not reflect what they should be. In applications, this phenomenon is usually caused by imprecise data collection or wrong records. In the existing literature, measurement error in covariates has been widely discussed and a large body of methods has been developed (e.g., [5–8]). However, as discussed in [25], the survival time might be subject to measurement error. To address this issue, [25] proposed the regression calibration method to correct for measurement error effects and developed the raking method to derive the estimator. However, their approach might ignore possibly misclassified censoring status and cannot be used to address nonlinear functions between the survival time and covariates.

From the perspective of the computational implementation, some R packages have been developed to fit the AFT model, including `aftgee` [13], `penAFT` [22], `spsurv` [26], and `survival` [29]. However, those approaches considered the simplest scenario without measurement error and nonlinear effects on covariates taken into account. In contrast, to deal with the complex structures, some R packages have been available to deal with either nonlinear functions or measurement error effects but not both. For example, the R packages [17, 24, 34], and [37] are used to deal with measurement error in covariates. However, those approaches rely on linear predictors, and except for [34], most packages can not handle survival outcome. On the other hand, the R packages [2, 11], and [32] implement boosting methods to deal with estimation of nonlinear functions, but they are not able to handle measurement error effects. With variable selection and measurement error effects taken into account simultaneously, [9] developed the R package `SIMEXBoost`. However, this package primarily focuses on parametric models and measurement error in covariates. To the best of our knowledge, rare computational software has been available to address nonlinear estimation and measurement error in the survival time under the survival model.

Consequently, to tackle those challenges simultaneously and provide potential users a conveniently computational implementation to derive a reliable estimate, we develop the R package `AFFECT`, which refers to Accelerated Functional Failure time model with Error-Contaminated survival Times. Here "functional" reflects nonlinear functions between the failure time and the covariates. We aim to correct for measurement error effects in the survival time and then employ the boosting algorithm to identify potentially important covariates and estimate their corresponding unknown functions. In the literature, a recent work [3] also considered the AFT model with nonlinear functions on covariates, but their approach is different from ours. For example, [3] primarily derived

the likelihood function under the given distribution for the noise term in the AFT model and then applied the package `xgboost` [11] to obtain the estimator; on the other hand, the package `AFFECT` is based on the estimation function derived by the Buckley-James method, which is different from [3] and does not require to specify the distribution of the noise term. In addition, the other obvious feature is that the package `AFFECT` is able to deal with measurement error effects but [3] does not take measurement error effects into account.

The remainder is organized as follows. In the section "Data structure and regression models", we introduce the data structure and relevant regression models. In the section "Methodology", we outline the estimation steps and the algorithm to derive the estimator. In the section "Illustration of the package `AFFECT`", we introduce the R package `AFFECT`, including the functions, the arguments, and the outputs. In the section "Simulation studies", we conduct simulation studies to assess the performance of the method in the package. In the section "Analysis of gene expression data", we apply the package to analyze a gene expression dataset. Finally, a general discussion is summarized in the section "Conclusion".

Data structure and regression models

Survival data

Let n denote the sample size. For subject $i = 1, ..., n$, let $\widetilde{T}_i$ and $\widetilde{C}_i$ be the non-negative failure and the censoring times of a specific cancer, respectively. Due to the purpose of analysis, we consider the log transformation: $T_i \triangleq \log(\widetilde{T}_i)$ and $C_i \triangleq \log(\widetilde{C}_i)$. Based on T_i and C_i, define $Y_i \triangleq \min\{T_i, C_i\}$ as the observed survival time and denote $\delta_i \triangleq \mathbb{I}(T_i < C_i)$ as the censoring indicator, where $\mathbb{I}(\cdot)$ is an indicator function. Moreover, let $\mathbf{X}_i = (X_{i1}, X_{i2}, ..., X_{ip})^\top$ be a p-dimensional vector of covariates or gene expressions. We impose the standard assumption that T_i and C_i are independent, given $\mathbf{X}_i$. Therefore, a typical survival data structure is given by $\{(Y_i, \delta_i, \mathbf{X}_i) : i = 1, 2, ..., n\}$.

The main interest in survival analysis is to characterize the relationship between the failure time and covariates. In our development, we consider the following accelerated failure time (AFT) model:

$$\begin{aligned} T_i =& F(\mathbf{X}_i) + \varepsilon_i \\ \triangleq& f_1(X_{i1}) + f_2(X_{i2}) + ... + f_q(X_{iq}) + \varepsilon_i, \end{aligned} \tag{1}$$

where ε_i is the noise term with $E(\varepsilon_i) = 0$ and has an unknown survivor function $S_\varepsilon(\cdot)$, and $f_j \in \mathcal{F}$ is a unknown function of interest with $\mathcal{F}$ being a class of continuous smooth functions. (1) shows that, among all p gene expressions, there are only q gene expressions informative to the failure time, where q is a positive integer and is smaller than p.

Ideally, if T_i is fully observed for all $i = 1, ..., n$, then one can consider the following least squares function

$$\sum_{i=1}^{n} \{T_i - F(\mathbf{X}_i)\}^2, \tag{2}$$

and $F(\cdot)$ can then be estimated by minimizing (2) via some nonparametric methods. However, in the presence of right-censoring, T_i is incomplete and one has Y_i in the dataset. Directly using Y_i in (2) may lead to biased estimator of $F(\cdot)$. Moreover, the other challenge is that dimension p in the gene expression data is usually larger than q, yielding that most gene expressions are possibly non-informative to the time-to-event response. As a result, detecting informative gene expressions is a crucial issue as well.

Measurement error models

In addition to the challenge from the complex regression model, measurement error is the other challenging and ubiquitous feature from the dataset, which is usually caused by imprecise measurement or wrong record. While we cannot examine whether variables are contaminated by measurement error, the key spirit is that we relax an "implicit" assumption that variables in the dataset are precisely measured. In most situations, measurement error in covariates has been widely explored. As commented by [25], however, survival times and censoring status are also possibly subject to measurement error. Specifically, let Y_i^* and δ_i^* denote the surrogate version of *unobserved* survival time Y_i and censoring status δ_i, respectively.

First, to characterize the error-prone survival time Y_i^* and the unobserved survival time Y_i, we modify the classical additive measurement error model (e.g.,[4, 35]) and follow an idea in [25] to consider the following measurement error model:

$$Y_i^* = Y_i + \gamma_0 + \boldsymbol{\gamma}_1^\top \mathbf{X}_i + \eta_i \triangleq Y_i + \boldsymbol{\omega}_i, \tag{3}$$

where η_i is assumed to follow a distribution with $E(\eta_i) = 0$ and $\text{var}(\eta_i) = \sigma_\eta^2$, and is independent of $\mathbf{X}_i$, γ_0 and $\boldsymbol{\gamma}_1$ are parameters.

Next, to characterize the misclassified censoring status, we let $\pi_{ikl} = P(\delta_i^* = k|\delta_i = l, \mathbf{X}_i)$ denote the conditional probability that links the observed censoring status k with the covariates and the unobserved censoring status l for $k, l \in \{0, 1\}$. By the law of total probability, one can express two probabilities $P(\delta_i^* = 1|\mathbf{X}_i)$ and $P(\delta_i^* = 0|\mathbf{X}_i)$ as

$$\begin{bmatrix} P(\delta_i^* = 1|\mathbf{X}_i) \\ P(\delta_i^* = 0|\mathbf{X}_i) \end{bmatrix} = \boldsymbol{\Pi}_i \begin{bmatrix} P(\delta_i = 1|\mathbf{X}_i) \\ P(\delta_i = 0|\mathbf{X}_i) \end{bmatrix} \tag{4}$$

with $\boldsymbol{\Pi}_i = \begin{bmatrix} \pi_{i11} & \pi_{i10} \\ \pi_{i01} & \pi_{i00} \end{bmatrix}$ being a 2×2 misclassification matrix. Moreover, as commented by [35] (Ch8), we impose the non-differentiable mechanism, which says that

$$\pi_{ikl} = P(\delta_i^* = k|\delta_i = l) \tag{5}$$

for $k, l \in \{0, 1\}$. As a result, in the following development, we will take (5) in our inference procedure. From now on, we respectively replace π_{ikl} and $\boldsymbol{\Pi}_i$ by π_{kl} and $\boldsymbol{\Pi}$ with the subscript i removed due to the assumption (5) and the independence of subject i.

Noting that parameters γ_0 and $\boldsymbol{\gamma}_1$ in (3) as well as $\boldsymbol{\Pi}$ (4) are usually unknown in applications. If the auxiliary information, such as the validation data, is available, then those parameters in (3) and (4) can be estimated. Otherwise, one may require prior knowledge and past experience for parameters γ_0, $\boldsymbol{\gamma}_1$, and $\boldsymbol{\Pi}$ or conduct sensitivity analyses, where the latter approach says that one can specify various values for those unknown

parameters based on background knowledge or under reasonable ranges to examine the impact of different magnitudes of measurement error effects and see whether the estimation method is robust with the change of parameter values in (3) and (4).

Methodology

Overview of the estimation procedure

In the presence of measurement error, the *observed* data are given by $\mathcal{D}^* \triangleq \{\{Y_i^*, \delta_i^*, \mathbf{X}_i\} : i = 1, \cdots, n\}$, and the goal is to estimate $F(\cdot)$ under the model (1). To tackle the challenges of measurement error and estimation of nonlinear functions, we propose the strategy that is summarized in Fig. 1.

According to the workflow in Fig. 1, we are first given a collected dataset $\mathcal{D}^*$ with relaxing an implicit assumption that the survival time in dataset is precisely measured, and we characterize error-prone survival data and the censoring status by two measurement error models (3) and (4), respectively. The next step in Fig. 1 is to correct for measurement error effects. We adopt the regression calibration method to create the corrected survival time, and employ the insertion method to obtain the corrected censoring status. Detailed discussions are deferred to the subsection "Correction of measurement error effects".

After correcting for measurement error effects, we then adopt the corrected survival time and censoring status to the boosting procedure, as shown in the last step in Fig. 1. To address the censoring effect, we employ the Buckley-James (BJ) estimator to create a corrected and pseudo response, such that its expectation can be recovered to the expectation of the failure time. Based on the corrected BJ response, we implement the boosting algorithm with the cubic spline estimation being the base learner. Through finite iterations, the potentially informative gene expressions as well as their estimated functions can be obtained simultaneously. Detailed descriptions are summarized in the subsection "Boosting for estimation of nonlinear functions".

Correction of measurement error effects

To correct for error-prone survival time, we first observe from (3) and take the conditional expectation, given $\mathbf{X}_i$, to obtain that

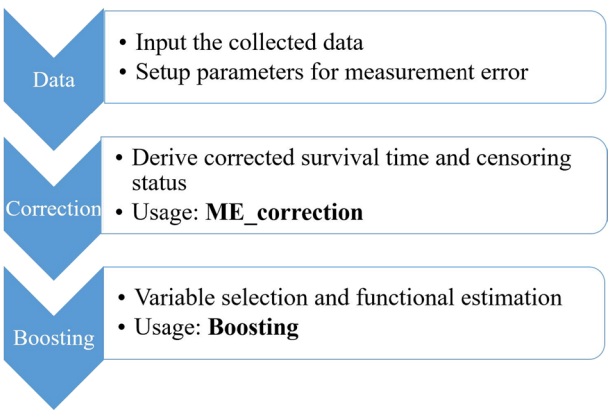

Fig. 1 A workflow of the package `AFFECT`

$$E(Y_i|\mathbf{X}_i) = E\{Y_i^* - E(\boldsymbol{\omega}_i|\mathbf{X}_i)|\mathbf{X}_i\}, \tag{6}$$

which implies that $Y_i^* - E(\boldsymbol{\omega}_i|\mathbf{X}_i)$ can be used to recover Y_i since they have the same expectation. To deal with the conditional expectation $E(\boldsymbol{\omega}_i|\mathbf{X}_i)$, we adopt regression calibration [5], which is given by

$$E(\boldsymbol{\omega}_i|\mathbf{X}_i) = \boldsymbol{\mu}_\omega + \boldsymbol{\Sigma}_{\omega\mathbf{X}}\boldsymbol{\Sigma}_{\mathbf{XX}}^{-1}(\mathbf{X}_i - \boldsymbol{\mu}_\mathbf{X}), \tag{7}$$

where $\boldsymbol{\mu}_\omega$ is the expectation of $\boldsymbol{\omega}_i$, $\boldsymbol{\Sigma}_{\omega\mathbf{X}}$ is the covariance matrix of $\boldsymbol{\omega}_i$ and $\mathbf{X}_i$, $\boldsymbol{\Sigma}_{\mathbf{XX}}$ is the covariance matrix of $\mathbf{X}_i$, and $\boldsymbol{\mu}_\mathbf{X}$ is the expectation of $\mathbf{X}_i$. When $\boldsymbol{\mu}_\omega$, $\boldsymbol{\Sigma}_{\omega\mathbf{X}}$, $\boldsymbol{\Sigma}_{\mathbf{XX}}$ and $\boldsymbol{\mu}_\mathbf{X}$ are estimated empirically, then (7) can be estimated by $\widehat{E(\boldsymbol{\omega}_i|\mathbf{X}_i)}$, yielding the "corrected" survival time

$$\widehat{Y}_i \triangleq Y_i^* - \widehat{E(\boldsymbol{\omega}_i|\mathbf{X}_i)}. \tag{8}$$

The validity of (8) can be justified by [25].

Next, we deal with measurement error in the censoring status. Provided that $\boldsymbol{\Pi}$ is invertible, (4) can be re-written as

$$\begin{bmatrix} P(\delta_i = 1|\mathbf{X}_i) \\ P(\delta_i = 0|\mathbf{X}_i) \end{bmatrix} = \boldsymbol{\Pi}^{-1} \begin{bmatrix} P(\delta_i^* = 1|\mathbf{X}_i) \\ P(\delta_i^* = 0|\mathbf{X}_i) \end{bmatrix}, \tag{9}$$

which gives that

$$P(\delta_i = 1|\mathbf{X}_i) = \frac{P(\delta_i^* = 1|\mathbf{X}_i) - \pi_{10}}{1 - \pi_{10} - \pi_{01}}.$$

Thus, the "corrected" censoring indicator status is defined as

$$\widehat{\delta}_i \triangleq \frac{\delta_i^* - \pi_{10}}{1 - \pi_{10} - \pi_{01}}, \tag{10}$$

which satisfies $E(\widehat{\delta}_i|\mathbf{X}_i) = E(\delta_i|\mathbf{X}_i)$. Therefore, (8) and (10) give the "corrected" survival data $\mathcal{D} \triangleq \left\{(\widehat{Y}_i, \widehat{\delta}_i, \mathbf{X}_i) : i = 1, 2, ..., n\right\}$.

Boosting for estimation of nonlinear functions

Given the corrected dataset $\mathcal{D}$, we present the boosting procedure that is summarized as the pseudo code in Algorithm 1.

Specifically, to adjust the censoring effect in the survival time, we implement the BJ estimator in (11) under the corrected data $\mathcal{D}$. To implement the boosting method and estimate $F(\cdot)$, we first set zero as the initial value for the function $F(\cdot)$. In each iteration, we take the cubic spline estimation as the weak learner for each covariate and select an index of the gene expression j^* that satisfies the smallest square error loss (12). Here the cubic spline estimation can be computed by the function `smooth.spline` in the existing R package `stats` [29]. After that, we compute the increment and update the estimated function in Steps 3 and 4, respectively. Finally, we continue the computations in Steps 1-4 with K times repetitions, and we can derive the final estimator $\widehat{F}(\cdot)$ and a set $\mathcal{S}^{(K)}$ containing all informative gene expressions.

Algorithm 1 AFFECT

Input: Given a corrected dataset $\mathcal{D} \triangleq \left\{(\widehat{Y}_i, \widehat{\delta}_i, \mathbf{X}_i) : i = 1, 2, ..., n\right\}$.

Step 0: Compute the BJ estimator

$$Y_i^{**} = \widehat{\delta}_i \widehat{Y}_i + (1 - \widehat{\delta}_i) F(\mathbf{X}_i) + (1 - \widehat{\delta}_i) \int_{\widehat{Y}_i - F(\mathbf{X}_i)}^{\infty} -u \times \frac{dS_\varepsilon(u)}{S_\varepsilon(\widehat{Y}_i - F(\mathbf{X}_i))} \tag{11}$$

for $i = 1, ..., n$, and take $F^{(0)}(\mathbf{X}_i) = 0$ as an initial value and specify $\mathcal{S}^{(0)} = \varnothing$ as an empty set.

for $k = 1, 2, ..., K$ **do**

Step 1: Let $r_i^{*(k)} = Y_i^{**(k-1)} - F^{(k-1)}(\mathbf{X}_i)$, where $Y_i^{**(k-1)}$ is (11) with F replaced by $F^{(k-1)}$.

Step 2: Select j^* by

$$j^* \triangleq \underset{j=\{1,2,...,p\}}{\arg\min} \sum_{i=1}^{n} \{r_i^{*(k)} - g_j^{(k)}(X_{ij})\}^2, \tag{12}$$

where $g_j^{(k)}$ is the cubic spline estimator for the jth gene expression under the kth iteration. Define $\mathcal{S}^{(k)} = \mathcal{S}^{(k-1)} \cup \{j^*\}$.

Step 3: Line search

Estimate the increment $\widehat{w}_k$ by solving $\widehat{w}_k = \underset{w_k \in \mathbb{R}}{\arg\min} \sum_{i=1}^{n} \{r_i^{*(k)} - w_k g_j^{(k)}(X_{ij})\}^2$.

Step 4: Update functions

(i) $F^{(k)}(\mathbf{X}_i) = F^{(k-1)}(\mathbf{X}_i) + \widehat{w}_k g_{j^*}^{(k)}(X_{ij})$.

(ii) Define $e_i^{(k)} = Y_i^{**(k-1)} - F^{(k)}(\mathbf{X}_i)$ and calculate survival function $S_\varepsilon^{(k)}(y)$ by calculating the Kaplan-Meier estimator.

(iii) replace $F(\mathbf{X}_i)$ in (11) by $F^{(k)}(\mathbf{X}_i)$ and denote it by $Y_i^{**(k)}$.

Output:

The final estimator is given by $\widehat{F}(\cdot) \triangleq F^{(K)}(\mathbf{X})$.

Illustration of the package `AFFECT`

To make the implementation of Algorithm 1 available for public use, we develop the R package `AFFECT`. Two functions in this package are used to implement the estimation method in section "Methodology". The first function `ME_correction` is used to do correction for error-prone response and misclassified censoring status, and the second function `Boosting` is used to estimate the function $F(\cdot)$ under the model (1).

`ME_correction`

This function aims to correct for measurement error in the survival time and misclassification in the censoring status. The key strategy in the function `ME_correction` includes regression calibration (8) for survival time under the model (3) and the unbiased conditional expectation approach for censoring status (10) under (4). With information of parameters in measurement error models implemented, this function will give outputs with corrected survival time and censoring status.

The implementation of `ME_correction` is given by

```
ME_correction(pi_10,pi_01,gamma0,gamma1,cor_covar,indicator,yast,covariate),
```

where the arguments include

- `pi_10`: Misclassifcation probability π_{i10} in (5).
- `pi_01`: Misclassifcation probability π_{i01} in (5).
- `gamma0`: A scalar γ_0 in the model (3).

- `gamma1`: A p-dimensional vector $\boldsymbol{\gamma}_1$ in the model (3).
- `cor_covar`: A $p \times p$ covariance matrix of a p-dimensional vector of covariates.
- `indicator`: A n-dimensional vector of censoring status.
- `yast`: A n-dimensional vector of survival times.
- `covariate`: A $n \times p$ matrix of covariates.

The first two arguments `pi_10` and `pi_01` refer to misclassification probabilities in (4) for characterizing misclassified censoring status. The middle three arguments `gamma0`, `gamma1`, and `cor_covar` are parameters in the measurement error model (3) for error-prone survival time. Finally, the last three arguments `indicator`, `yast`, and `covariate` are observed censoring status, survival time, and covariates, respectively. The function `ME_correction` provides a flexible implementation. If one believes that censoring status or survival time is free of measurement error, then arguments can be specified as `pi_10` $=$ `pi_01` $= 0$ or `gamma0` $=$ `gamma1` $=$ `cor_covar` $= 0$. Given those arguments, the function provides the corrected survival time and the corrected censoring status, which are given by

- `correction_data`: A $n \times 2$ data frame. This first column is the corrected survival time, and the second column is the corrected censoring indicator.

`Boosting`

With the function `smooth.spline` in existing R package `stats` [29] equipped, the function `Boosting` aims to implement Algorithm 1 to select informative covariates under the model (1) and estimate their corresponding functional forms with survival time. The implementation of `Boosting` is given by

```
Boosting(data, iter),
```

where the arguments include

- `data`: A $n \times (p + 2)$ dimension of data. The first column is survival time, the second column is censoring status, and the other columns are covariates.
- `iter`: The number of iterations K in Algorithm 1. The default value is 50 and the iteration will stop when the absolute value of increment of every estimated value is small than 0.01.

The first argument `data` is a dataset $\mathcal{D}$ with the first and second columns being the survival time and the censoring status, respectively, and the remaining columns are gene expressions. `iter` is a user-specific iteration number. If users do not input the value to the argument `iter`, then the algorithm will automatically run 50 iterations. On the other hand, larger value of `iter` may incur longer computation time. In this case, the function `Boosting` can make iteration stop early if the criterion

$$||F^{(k)}(\mathbf{X}_i) - F^{(k-1)}(\mathbf{X}_i)||_\infty \leq \tau \tag{13}$$

is satisfied, where $F^{(k)}(\mathbf{X}_i)$ is the updated function at the kth step, $\|F(\mathbf{x})\|_\infty \triangleq \max_{i=1,\cdots,n} |F(x_i)|$ is the infinity norm, and τ is a threshold value, which is specified as 0.01 in this function.

This function gives us the following outcome:

- `results`: A list that contains the informative covariates with respect to the failure time (`$covariates`) and their corresponding estimated functional curves (`$function_forms`). In addition, predicted failure time based on (1) as well as the estimated survivor curve are provided by using `$predict_failure_time` and `$survival_curve`, respectively.

Simulation studies

In this section, we conduct simulation studies to assess the performance of the proposed method and demonstrate the implementation of the package `AFFECT`.

Simulation setup

Let $n = 400$ denote the sample size, and let $p = 3, 10, 100$ denote the dimension of covariates. For $i = 1, ..., n$ and $j = 1, ..., p$, we independently generate the covariates X_{ij} from the uniform distribution with an interval $[-1, 1]$. Given $\mathbf{X}_i = (X_{i1}, ..., X_{ip})^\top$, we independently generate ε_i from the standard normal distribution $N(0, 1)$, and use (1) to independently generate $\widetilde{T}_i$ for $i = 1, 2, ..., n$, where $q = 3$ is considered and functions $f_j(\cdot)$ with $j = 1, 2, 3$ are specified as

$$f_1(X_{i1}) = 4X_{i1}^2, \ f_2(X_{i2}) = \cos(6X_{i2}), \ \text{ and } f_3(X_{i3}) = \arcsin(X_{i3}).$$

When $p = 3$, the model (1) says that all covariates are informative; otherwise, $F(\cdot)$ reflects that the first three covariates are nonlinearly informative to T_i and the remaining $p - 3$ covariates are irrelevant.

Next, we first generate the censoring status δ_i for $i = 1, ..., n$ by

$$P(\delta_i = 1|\mathbf{X}_i) = \frac{e^{\mathbf{X}_i}}{1 + e^{\mathbf{X}_i}}.$$

Given $\widetilde{T}_i$ and δ_i, the survival time is defined as $Y_i = \log \widetilde{T}_i$ if $\delta_i = 1$ and $Y_i = \log \widetilde{C}_i$ if $\delta_i = 0$, where the censoring time is defined as $\widetilde{C}_i = \widetilde{T}_i - \exp(0.003)$.

Finally, we generate error-prone data by treating (Y_i, δ_i) as unobserved survival times. For $i = 1, 2, ..., n$, Y_i^* is generated by (3), where $\gamma_0 = 1$, $\boldsymbol{\gamma}_1 = (1, \mathbf{0}_{p-1}^\top)^\top$ with $\mathbf{0}_p$ being the p-dimensional zero vector, and η_i is independently generated by the normal distribution $N(0, \sigma_\eta^2)$ with $\sigma_\eta^2 = 0.25, 0.5$ and 0.75 reflecting various magnitudes of measurement error effects. For the observed censoring status, we generate δ_i^* by (4) with $\pi_{10} = \pi_{10} = 0.1$ or $\pi_{10} = \pi_{10} = 0.9$. For each simulation setting, we run 100 repetitions. The following programming code demonstrates the data generation:

```
####### Generation of covariates and unobserved survival time
####### and censoring status

n = 400
p = 3
f1 <- function(x1){
  y <- 4*x1^2
  return(y)
}


f2 <- function(x2){
  y <- cos(6*x2)
  return(y)
}


f3 <- function(x3){
  y <- asin(x3)
  return(y)
}

X = NULL
for(i in 1:p) {

X = cbind(X, runif(n,-1,1))

}

T = f1(X[,1]) + f2(X[,2]) + f3(X[,3]) + rnorm(n,0,1)

pb = (exp(rowSums(X)))/(1+exp(rowSums(X)))
delta = (pb>0.5)*1
Y = T*(delta=1)*1 + (T - rexp(n,0.003))*(delta=0)*1

####### Measurement Error
sigmae = 0.25
Yast = Y + 1+ X[,1] + rnorm(n,0,sigmae)
pr = 0.9
P = matrix(c(pr,1-pr,1-pr,pr),2,2)
pbast = P %*% rbind(pb,1-pb)
deltaast = rbinom(n,1,pbast[1,])
```

Simulation results

Given a simulated dataset, we implement the proposed method to estimate $F(\cdot)$, derive the predicted failure time and estimate the survivor function. The implementation code is summarized as follows:

```
library(AFFECT)
library(MASS)

gamma_1 = matrix(0,ncol=p,nrow=1);gamma_1[1,1] = 1
corrected = ME_correction(pi_10 = 1-pr,pi_01 = 1-pr, gamma0 = 1,
                          gamma1 = gamma_1, cor_covar = cov(X),
                          y = Yast, indicator = deltaast,
                          covariate = X)

Ycorrect = corrected[,1]
deltacorrect = corrected[,2]

CorrectData = cbind(corrected, X)
estCorrect = Boosting(data = CorrectData)
```

To see the performance of variable selection, we use $\#X_1, \#X_2$ and $\#X_3$ to record the frequency of selecting X_i with $i = 1, 2, 3$ among the repetition of simulations, and we use $\mathcal{P}_a$ to represent the frequency that variables X_1-X_3 are simultaneously selected in each simulation. Numerical results are summarized in Table 1. Meanwhile, for the visualization, we display estimated curves by using the command `$function_forms` as well as the true functions for X_1-X_3 in Fig. 2. In general, all informative covariates X_1-X_3 can be selected regardless of values of p and σ_η^2. According to the observation in Fig. 2, we can see that the estimation method is able to capture the pattern of the true functions.

In addition, when the nonlinear functions of covariates are estimated, the function `Boosting` also produces the estimated survivor curve, which is obtained by the

Table 1 Simulation results: report of variable selection of the package `AFFECT` under various settings

p	Criteria	$\pi_{01} = \pi_{10} = 0.9$			$\pi_{01} = \pi_{10} = 0.1$		
		$\sigma_\eta^2 = 0.25$	$\sigma_\eta^2 = 0.5$	$\sigma_\eta^2 = 0.75$	$\sigma_\eta^2 = 0.25$	$\sigma_\eta^2 = 0.5$	$\sigma_\eta^2 = 0.75$
3	$\#X_1$	1.000	1.000	1.000	1.000	1.000	1.000
	$\#X_2$	1.000	1.000	1.000	1.000	1.000	1.000
	$\#X_3$	1.000	1.000	1.000	1.000	1.000	1.000
	$\#\mathcal{P}_a$	1.000	1.000	1.000	1.000	1.000	1.000
10	$\#X_1$	1.000	1.000	1.000	1.000	1.000	1.000
	$\#X_2$	1.000	1.000	1.000	1.000	1.000	1.000
	$\#X_3$	1.000	1.000	1.000	1.000	1.000	1.000
	$\#\mathcal{P}_a$	1.000	1.000	1.000	1.000	1.000	1.000
100	$\#X_1$	1.000	1.000	1.000	1.000	1.000	1.000
	$\#X_2$	1.000	1.000	1.000	1.000	1.000	1.000
	$\#X_3$	1.000	1.000	1.000	1.000	1.000	1.000
	$\#\mathcal{P}_a$	1.000	1.000	1.000	1.000	1.000	1.000

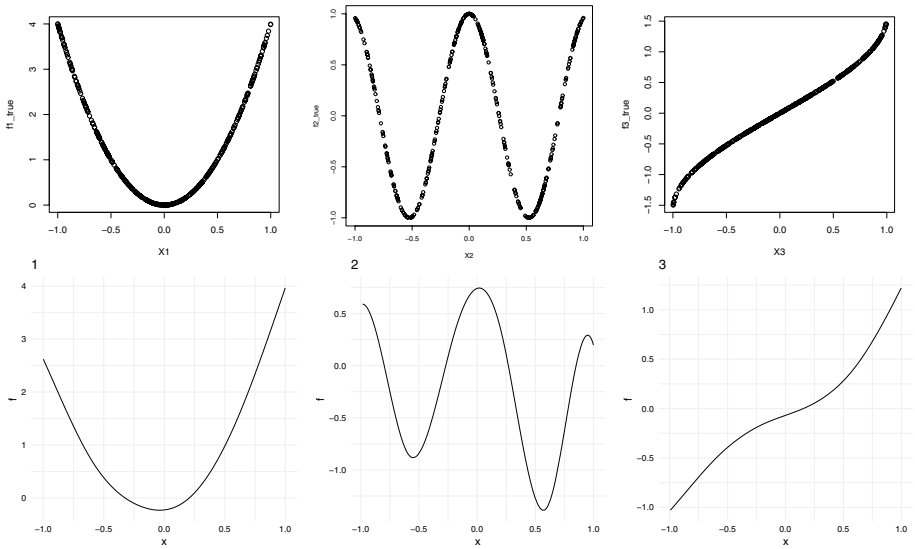

Fig. 2 Simulation results for estimating functions in the AFT model. The first row shows curves of the true functions for variables X_1, X_2, and X_3. The second row displays estimated curves for variables X_1, X_2, and X_3, which are printed by the command `$function_forms`

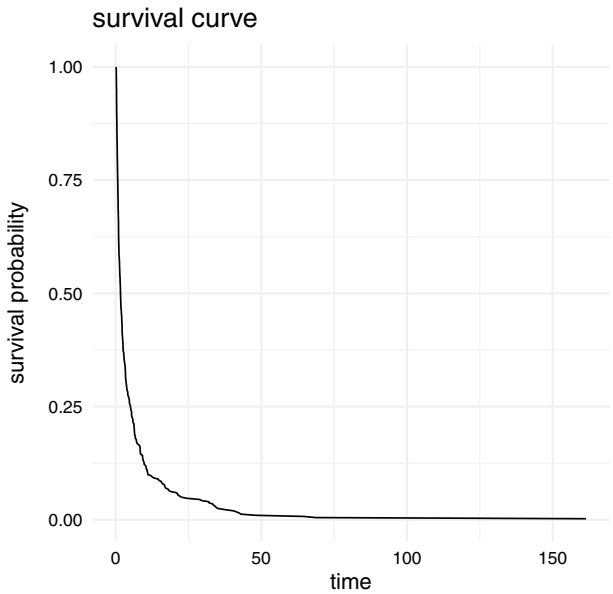

Fig. 3 Simulation results for estimating survivor curves based on the AFT model, which are printed by the command `$survival_curve`

command `$survival_curve` and is displayed in Fig. 3. We can see that the estimated survivor curve is sharply decreasing.

To assess the performance of predicted failure time obtained by the function `Boosting`, we apply the following commonly used criteria:

(a) The integrated Brier Score (IBS):

$$\text{IBS} = \{y_{\max}\}^{-1} \int_0^{y_{\max}} BS(t)dt,$$

where $y_{\max} = \max\{\widehat{Y}_i : i = 1, \cdots, n\}$ and

$$BS(t) = \frac{1}{n}\sum_{i=1}^{n}\left[\left\{\widehat{S}(t|\mathbf{X}_i)\right\}^2 \mathbb{I}\left(\widehat{Y}_i \le t, \widehat{\delta}_i = 1\right)\frac{1}{\widehat{G}(\widehat{Y}_i)} + \left\{1 - \widehat{S}(t|\mathbf{X}_i)\right\}^2 \mathbb{I}\left(\widehat{Y}_i > t\right)\frac{1}{\widehat{G}(t)}\right],$$

where $\widehat{G}(t)$ is the estimated survivor function for the censoring time and $\widehat{S}(t|\mathbf{X}_i)$ is the estimated survivor function of the failure time based on the model (1).

(b) The mean absolute error (MAE):

$$\sum_{i:\widehat{\delta}_i=1}\left|\widehat{F}(\mathbf{X}_i) - \widehat{Y}_i\right|.$$

(c) The Concordance index (C-index):

$$\frac{\sum_{j<k}\mathbb{I}(\widehat{Y}_j < \widehat{Y}_k)\mathbb{I}(\widehat{F}(\mathbf{X}_j) > \widehat{F}(\mathbf{X}_k))\widehat{\delta}_j + \mathbb{I}(\widehat{Y}_j > \widehat{Y}_k)\mathbb{I}(\widehat{F}(\mathbf{X}_j) < \widehat{F}(\mathbf{X}_k))\widehat{\delta}_k}{\sum_{j<k}\mathbb{I}(\widehat{Y}_j < \widehat{Y}_k)\widehat{\delta}_j + \mathbb{I}(\widehat{Y}_j > \widehat{Y}_k)\widehat{\delta}_k}.$$

To emphasize the advantage of the package `AFFECT` under the AFT model, we compare the performance of the package `AFFECT` with some existing packages: `aftgee` [13], `penAFT` [22], `spsurv` [26], `survival` [29], and `xgboost` [11]. After using those packages to obtain the estimators, we then apply the R package `SurvMetrics` to compute the criteria (a)–(c).

Table 2 reveals that the criteria (a)-(c) obtained by our package give reasonable values for the estimated failure time, suggesting that our estimation method provides satisfactory performance of deriving accurate prediction when the measurement error effects can be corrected. Compared with existing packages, we can see that IBS and MAE values from other packages are greater than ours, and C-index values from other packages are smaller than ours. In addition, the performance of existing packages seems worse as the dimension p becomes large. It might be due to that most existing packages are based on parametric models and may not be valid to address variable selection. While the package `xgboost` can handle the estimation of nonlinear functions in covariates, it cannot deal with measurement error effects; that is why the performance of [3] is slightly worse than our method, and its biases are induced by measurement error.

Analysis of gene expression data

In this section, our research interest lies on the survival data for the breast cancer, which is the most frequent cancer among women and causes the greatest number of cancer-related deaths among women. The motivating dataset comes from the Molecular Taxonomy of Breast Cancer International Consortium (METABRIC) database, which was a Canada-UK Project containing targeted sequencing data of primary breast cancer

Table 2 Simulation results under various settings: report of evaluating criteria under various R packages

p	Criteria	Method	$\pi_{01}=\pi_{10}=0.9$			$\pi_{01}=\pi_{10}=0.1$		
			$\sigma_\eta^2=0.25$	$\sigma_\eta^2=0.5$	$\sigma_\eta^2=0.75$	$\sigma_\eta^2=0.25$	$\sigma_\eta^2=0.5$	$\sigma_\eta^2=0.75$
3	C-index	AFFECT	0.712	0.704	0.708	0.710	0.795	0.764
		survival	0.471	0.396	0.530	0.610	0.609	0.633
		aftgee	0.514	0.501	0.389	0.570	0.629	0.633
		penAFT	0.488	0.387	0.403	0.561	0.625	0.621
		spsurv	0.584	0.499	0.528	0.615	0.694	0.665
		xgboost	0.327	0.309	0.311	0.387	0.394	0.458
	MAE	AFFECT	5.223	4.472	2.261	4.175	2.104	2.953
		survival	16.887	13.936	17.987	26.021	36.507	24.146
		aftgee	27.038	18.079	15.268	27.603	37.652	22.076
		penAFT	24.774	40.022	26.738	22.261	20.627	29.529
		spsurv	20.498	22.595	20.344	22.925	21.965	31.814
		xgboost	17.112	20.909	11.764	18.639	41.214	23.623
	IBS	AFFECT	0.119	0.134	0.170	0.117	0.168	0.183
		survival	0.383	0.393	0.346	0.352	0.358	0.336
		aftgee	0.358	0.372	0.378	0.344	0.353	0.350
		penAFT	0.385	0.399	0.392	0.374	0.348	0.344
		spsurv	0.346	0.360	0.382	0.353	0.338	0.317
		xgboost	0.459	0.447	0.481	0.418	0.435	0.341
p	Criteria	Method	$\pi_{01}=\pi_{10}=0.9$			$\pi_{01}=\pi_{10}=0.1$		
			$\sigma_\eta^2=0.25$	$\sigma_\eta^2=0.5$	$\sigma_\eta^2=0.75$	$\sigma_\eta^2=0.25$	$\sigma_\eta^2=0.5$	$\sigma_\eta^2=0.75$
10	C-index	AFFECT	0.732	0.719	0.718	0.706	0.729	0.798
		survival	0.303	0.312	0.313	0.613	0.627	0.646
		aftgee	0.323	0.347	0.353	0.567	0.572	0.596
		penAFT	0.304	0.311	0.323	0.548	0.607	0.569
		spsurv	0.464	0.350	0.362	0.502	0.604	0.545
		xgboost	0.305	0.333	0.293	0.332	0.299	0.326
	MAE	AFFECT	9.322	9.985	10.774	1.716	7.557	3.254
		survival	17.663	15.578	15.002	15.324	20.603	22.176
		aftgee	18.897	18.340	16.341	22.065	23.048	26.711
		penAFT	14.897	21.502	24.003	20.362	21.307	22.051
		spsurv	32.963	20.280	17.944	25.092	22.909	18.207
		xgboost	17.811	20.065	23.402	22.326	21.038	23.467
	IBS	AFFECT	0.079	0.102	0.119	0.088	0.112	0.129
		survival	0.412	0.406	0.391	0.329	0.340	0.337
		aftgee	0.387	0.395	0.403	0.367	0.368	0.368
		penAFT	0.394	0.407	0.394	0.359	0.355	0.344
		spsurv	0.364	0.395	0.371	0.372	0.349	0.371
		xgboost	0.453	0.424	0.424	0.439	0.444	0.409
p	Criteria	Method	$\pi_{01}=\pi_{10}=0.9$			$\pi_{01}=\pi_{10}=0.1$		
			$\sigma_\eta^2=0.25$	$\sigma_\eta^2=0.5$	$\sigma_\eta^2=0.75$	$\sigma_\eta^2=0.25$	$\sigma_\eta^2=0.5$	$\sigma_\eta^2=0.75$
100	C-index	AFFECT	0.754	0.723	0.725	0.773	0.706	0.782
		survival	0.240	0.242	0.230	0.525	0.539	0.602
		aftgee	0.212	0.231	0.208	0.554	0.510	0.573
		penAFT	0.318	0.310	0.270	0.465	0.459	0.467
		spsurv	0.241	0.275	0.303	0.590	0.560	0.544

Table 2 (continued)

p	Criteria	Method	$\pi_{01} = \pi_{10} = 0.9$			$\pi_{01} = \pi_{10} = 0.1$		
			$\sigma_\eta^2 = 0.25$	$\sigma_\eta^2 = 0.5$	$\sigma_\eta^2 = 0.75$	$\sigma_\eta^2 = 0.25$	$\sigma_\eta^2 = 0.5$	$\sigma_\eta^2 = 0.75$
		`xgboost`	0.274	0.301	0.271	0.356	0.329	0.298
	MAE	`AFFECT`	6.279	5.614	4.583	8.993	10.240	13.327
		`survival`	25.632	23.899	25.470	16.544	15.786	18.724
		`aftgee`	19.155	21.986	27.203	28.004	13.899	17.829
		`penAFT`	31.110	17.761	19.449	19.878	15.948	21.581
		`spsurv`	28.418	18.524	22.804	23.069	16.354	13.203
		`xgboost`	19.586	12.940	26.352	13.596	26.456	21.429
	IBS	`AFFECT`	0.128	0.141	0.152	0.166	0.116	0.145
		`survival`	0.424	0.422	0.413	0.377	0.367	0.329
		`aftgee`	0.461	0.439	0.426	0.355	0.390	0.346
		`penAFT`	0.405	0.424	0.432	0.386	0.400	0.381
		`spsurv`	0.457	0.413	0.403	0.346	0.334	0.359
		`xgboost`	0.462	0.457	0.435	0.388	0.457	0.444

samples and was collected by Cambridge Research Institute and the British Columbia Cancer Centre in Canada [27]. The full dataset and all variables' names are publicly available on the Kaggle website (https://www.kaggle.com/datasets/raghadalharbi/breast-cancer-gene-expression-profiles-metabric). The dataset has 1422 patients with censoring rate 74.965%. In addition, there are 331 gene expressions, which are continuous random variables and are recorded as m-RNA levels Z-score:

$$\frac{\text{expression in tumor sample} \ - \ \text{mean expression in reference sample}}{\text{standard deviation of expression in reference sample}}.$$

In our study, we take a variable "overall_survival_months" as the survival time, which is defined as duration from the time of the intervention to death. In addition, we take "overall_survival" as the censoring status because it reflects whether the patient is alive of dead. Moreover, gene expressions are taken as the covariates and are used to characterize the failure time. However, among 331 gene expressions, it is possible that a few of gene expressions are informative to the failure time, and the relationship between the gene expressions and the failure time is possibly nonlinear. On the other hand, as discussed in the section "Background" and existing literature (e.g., [15, 30]), survival time and the censoring status might be collected with error caused by wrong record or imprecise machines in laboratory. Consequently, taking measurement error effects and estimation of nonlinear functions into account is required.

To tackle the challenges and derive the reliable estimator, we adopt the package `AFFECT` to analyze this dataset. Since this dataset has no additional information to determine parameters in (3) and (4), to examine the impact of measurement error effects and see the robustness of the estimation result, we conduct sensitivity analyses. In our study, we consider three scenarios: Scenario I for minor effect ($\sigma_\eta^2 = 0.15$ and $\pi_{10} = \pi_{01} = 0.05$), Scenario II for moderate effect ($\sigma_\eta^2 = 0.3$ and $\pi_{10} = \pi_{01} = 0.1$), and Scenario III for severe effect ($\sigma_\eta^2 = 0.5$ and $\pi_{10} = \pi_{01} = 0.15$). For the parameters γ_0 and $\boldsymbol{\gamma}_1$, we specify $\gamma_0 = 0$ and $\boldsymbol{\gamma}_1 = (1, \mathbf{0}_{330}^\top)^\top$ as demonstrated in our package manual [14],

where $\mathbf{0}_p$ is a p-dimensional zero vector. We first implement those values to the function `ME_correction` to derive the corrected survival time and censoring status, and then implement them and gene expressions to the function `Boosting` to obtain the result. The demonstration of the programming code is given below:

```
library(AFFECT)

read.table("C://METABRIC.csv",sep=",") -> data

data = data[-1,]
n = dim(data)[1]
p = dim(data)[2]-2
y = log(as.numeric(data[,1]))
delta = as.numeric(data[,2])
X = matrix(as.numeric(unlist(data[,-c(1,2)])),n,p)


matrixa = diag(p)
gamma_0 = 1 ;
gamma_1 = matrix(0,ncol=p, nrow =1); gamma_1[1,1] = 1
corrected_data1 = ME_correction(pi_10 = 0.9, pi_01 = 0.9,
                                gamma0 = gamma_0, gamma1 = gamma_1,
                                cor_covar=matrixa, y=y,
                                indicator=delta, covariate = X)

##########################################

data_correct = cbind(corrected_data1,X)
Boosting(data_correct, iter=100)
```

The resulting estimated function and selected gene expression are displayed in Fig. 4. With measurement error correction accommodated, analysis results show that a gene expression "ccnd1" (known as Cyclin D1 and labelled as X30) is only selected regardless

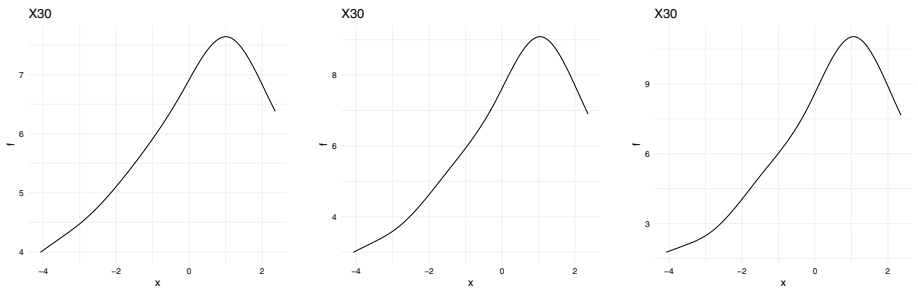

Fig. 4 Selected covariates and its estimated curves under Scenario I ($\sigma_\eta^2 = 0.15$ and $\pi_{10} = \pi_{01} = 0.05$), Scenario II ($\sigma_\eta^2 = 0.3$ and $\pi_{10} = \pi_{01} = 0.1$), and Scenario III ($\sigma_\eta^2 = 0.5$ and $\pi_{10} = \pi_{01} = 0.15$). X30 is a variable label, reflecting the gene expression "ccnd1"

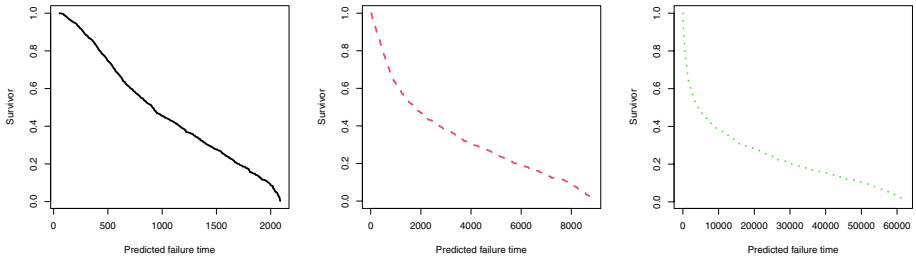

Fig. 5 Estimated survivor curves derived by the boosting method. A black solid curve is obtained under Scenario I ($\sigma_\eta^2 = 0.15$ and $\pi_{10} = \pi_{01} = 0.05$); a red dash curve is obtained under Scenario II ($\sigma_\eta^2 = 0.3$ and $\pi_{10} = \pi_{01} = 0.1$); a green dot curve is obtained under Scenario III ($\sigma_\eta^2 = 0.5$ and $\pi_{10} = \pi_{01} = 0.15$)

Table 3 Real data analysis result: evaluation criteria for estimation methods. Scenario I is the minor effect ($\sigma_\eta^2 = 0.15$ and $\pi_{10} = \pi_{01} = 0.05$); Scenario II represents the moderate effect ($\sigma_\eta^2 = 0.3$ and $\pi_{10} = \pi_{01} = 0.1$); and Scenario III reflects the severe effect ($\sigma_\eta^2 = 0.5$ and $\pi_{10} = \pi_{01} = 0.15$)

	C-index	MAE	IBS
`AFFECT`-Scenario I	0.679	5.023	0.333
`AFFECT`-Scenario II	0.679	9.076	0.333
`AFFECT`-Scenario III	0.679	14.785	0.333
`survival`	0.109	118.672	0.397
`aftgee`	0.120	118.674	0.394
`penAFT`	0.250	118.390	0.394
`spsurv`	–	–	–
`xgboost`	0.492	118.399	0.383

of different scenarios. The estimated curves of a gene expression "ccnd1" under different scenarios are nonlinear, which shows that the boosting procedure enables us to detect gene expressions with nonlinear relationship to the survival time. Interestingly, "ccnd1" was discussed to have the association with high histopathological grade, high proliferation, and Luminal B subtype (e.g., [21, 31]). Moreover, [28] also pointed out that "ccnd1" was associated with a good breast cancer prognosis. In general, the result shows that the gene expression selected by our package with measurement error correction accommodated is as important as findings in some scientific results, which justifies that taking measurement error into account seems necessary in this data analysis.

With the functions of selected gene expressions estimated, we further estimate survivor curves for the failure time under the AFT model (1), and we display the estimated survivor curves under three different scenarios in Fig. 5. We can see that the estimated curves look smooth and are (almost) strictly decreasing to zero.

Finally, when estimated failure times under the AFT model (1) are obtained, we follow the discussion in the section "Simulation results" to compute C-index, IBS, and MAE, and assess the performance of the estimation methods. In addition to the package `AFFECT`, we also examine the existing packages listed in the section "Simulation results", and then summarize numerical results in Table 3. We can see that IBS and C-index obtained by the package `AFFECT` are almost the same regardless of the specification of π_{10}, π_{01}, and σ_η^2, but MAE values become large when measurement error effect is more severe. For the implementation of the existing packages, the package `spsurv`

cannot produce the analysis result; while other packages provide comparable values of MAE and IBS, which are generally greater than values obtained by our package. In addition, the C-index value from our method is greater than values from other packages. This finding is consistent with simulation results in the section "Simulation results", and these results may be incurred by nonlinear functions of informative covariates as well as measurement error effects in survival times simultaneously.

Conclusion

In this paper, we introduce the R package that aims to deal with measurement error in survival times and simultaneously detect important covariates and nonparametrically estimate unknown functions by a boosting procedure. The function `smooth.spline` in the existing R package `stats` is only equipped to our package and is used to implement the nonparametric estimation, but the main contribution of our package is to deal with incomplete responses caused by the censoring effects and measurement error in the survival time simultaneously. Those complex features can not be addressed by the existing R packages, such as `stats` or `xgboost`. The output produced by the function includes a list of selected covariates and the corresponding estimated functional curves. The visualization enables users to easily see the estimation results. Based on the numerical comparisons with existing packages, we find that the estimation procedure in our package produces reliable estimation results, and it is expected that this package can be widely used to analyze gene expression survival data with measurement error effects accommodated.

From the methodological perspective, we primarily adopt the regression calibration method to adjust measurement error in survival times in our package. While this approach was similar to [25], the difference between our package and existing literature is that we adopt this technique to the AFT model with nonlinear covariates. While the regression calibration method is convenient and useful to correct for measurement error effects, a crucial concern is its application for error-prone covariates, and it simply induces approximately consistent estimator if the nonlinear pattern of covariates is not seriously oscillatory (e.g., [4], Section 4.8.2). It might be interesting to explore alternative correction methods to adjust measurement error in the survival time, and then extend our package and the computational algorithm accordingly.

Availability and requirements

• Project name: AFFECT • Project home page: https://cran.r-project.org/web/packages/AFFECT/index.html• Programming language: R • Other requirements: R 3.3.1 or higher • Operating system(s): Platform independent • License: GPL-3 • Any restrictions to use by non-academics: No.

Acknowledgements
The authors appreciate the editorial team for the careful review and useful comments that significantly improve the initial manuscript.

Author contributions
L.-P. Chen: paper preparation, writing, idea motivation, revision, supervision. H.-T. Huang: paper preparation, writing, coding, idea motivation.

Funding
National Science and Technology Council of Taiwan (L.-P. Chen).

Availability of data and materials
The full dataset is available on the Kaggle website: https://www.kaggle.com/datasets/raghadalharbi/breast-cancer-gene-expression-profiles-metabric.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interest
The authors declare that they have no conflict of interest

Received: 22 November 2023 Accepted: 10 June 2024
Published online: 13 August 2024

References

1. Alam TF, Rahman MS, Bari W. On estimation for accelerated failure time models with small or rare event survival data. BMC Med Res Methodol. 2022;22:169.
2. Alfaro E, Gamez M, Garcia L, Guo N, Albano A, Sciandra M, Plaia A. adabag: applies multiclass AdaBoost.M1, SAMME and Bagging, 2023; https://cran.r-project.org/package=adabag. R package version 5.0
3. Barnwal A, Cho H, Hocking T. Survival regression with accelerated failure time model in XGBoost. J Comput Graph Stat. 2022;31:1292–302.
4. Carroll RJ, Ruppert D, Stefanski LA, Crainiceanu CM. Measurement error in nonlinear model. Boca Raton, FL: Chapman and Hall; 2006.
5. Chen L-P, Yi GY. Semiparametric methods for left-truncated and right-censored survival data with covariate measurement error. Ann Inst Stat Math. 2021;73:481–517.
6. Chen L-P, Yi GY. Analysis of noisy survival data with graphical proportional hazards measurement error models. Biometrics. 2021;77:956–69.
7. Chen L-P, Yi GY. Sufficient dimension reduction for survival data analysis with error-prone variables. Electron J Stat. 2022;16:2082–123.
8. Chen L-P, Qiu B. Analysis of length-biased and partly interval-censored survival data with mismeasured covariates. Biometrics. 2023;79:3929–40.
9. Chen L-P, Qiu B. SIMEXBoost: an R package for analysis of high-dimensional error-prone data based on boosting method. R J. 2023;15:5–20.
10. Chen L-P, Yi GY. Unbiased boosting estimation for censored survival data. Stat Sin. 2024;34:439–58.
11. Chen T, He T, Benesty M, Khotilovich V, Tang Y, Cho H, Chen K, Mitchell R, Cano I, et al. xgboost: extreme Gradient Boosting, 2023; https://cran.r-project.org/package=xgboost. R package version 1.7.5.1
12. Chen Y, Jia Z, Mercola D, Xie X. A gradient boosting algorithm for survival analysis via direct optimization of concordance index. Comput Math Methods Med. 2013;873595:1–8.
13. Chiou S-H, Kang S, Yan J. aftgee: accelerated failure time model with generalized estimating equations, 2023; https://CRAN.R-project.org/package=aftgee. R package version 1.2.0.
14. Huang H-T, Chen L-P. AFFECT: accelerated functional failure time model with error-contaminated survival times. 2023; https://cran.r-project.org/web/packages/AFFECT/index.html. R package version 0.1.2.
15. Korn EL, Dodd LE, Freidlin B. Measurement error in the timing of events: effect on survival analyses in randomized clinical trials. Clin Trials. 2010;7:626–33.
16. Lawless JF. Statistical models and methods for lifetime data. New York: Wiley; 2003.
17. Lederer W, Seibold H, Küchenhoff H, Lawrence C, Brøndum RF. simex: SIMEX- and MCSIMEX-algorithm for measurement error models, 2019; https://cran.r-project.org/package=simex. R package version 1.8
18. Lee DK, Chen N, Ishwaran H. Boosted nonparametric hazards with time- dependent covariates. Ann Stat. 2021;49(4):2101–28.
19. Li H, Luan Y. Boosting proportional hazards models using smoothing splines, with applications to high-dimensional microarray data. Bioinformatics. 2005;21:2403–9.
20. Li J, Ma S. Survival analysis in medicine and genetics. Boca Raton: Chapman & Hall/CRC Press; 2013.
21. Mohammadizadeh F, Hani M, Ranaee M, Bagheri M. Role of cyclin D1 in breast carcinoma. J Res Med Sci. 2013;18:1021–5.
22. Molstad AH, Suder PM. penAFT: fit the regularized Gehan estimator with elastic net and sparse group lasso penalties, 2023; https://CRAN.R-project.org/package=penAFT. R package version 0.3.0.
23. Mustefa YA, Chen D-G. Accelerated failure-time model with weighted least-squares estimation: application on survival of HIV positives. Arch Public Health. 2021;79:88.
24. Nab L. mecor: measurement error correction in linear models with a continuous outcome. https://cran.r-project.org/package=mecor. R package version 1.0.0 , 2021;
25. Oh EJ, Shepherd BE, Lumley T, Shaw PA. Raking and regression calibration: methods to address bias from correlated covariate and time-to-event error. Stat Med. 2021;40:631–49.
26. Panaro R, Demarqui F, Mayrink V. spsurv: Bernstein polynomial based semiparametric survival analysis, 2020; https://CRAN.R-project.org/package=spsurv. R package version 1.0.0.

27. Pereira B, Chin SF, Rueda O, et al. The somatic mutation profiles of 2,433 breast cancers refine their genomic and transcriptomic landscapes. Nat Commun. 2016;7:11479.
28. Peurala E, Koivunen P, Haapasaari K-M, Bloigu R, Jukkola-Vuorinen A. The prognostic significance and value of cyclin D1, CDK4 and p16 in human breast cancer. Breast Cancer Res. 2013;15:R5.
29. R Core Team and contributors worldwide. stats: the R stats package, 2024. https://stat.ethz.ch/R-manual/R-devel/library/stats/html/00Index.html. R package version 4.4.0.
30. Sarfati D, Blakely T, Pearce N. Measuring cancer survival in populations: relative survival vs cancer-specific survival. Int J Epidemiol. 2010;39:598–610.
31. Valla M, Klæstad E, Ytterhus B, Bofin AM. CCND1 amplification in breast cancer-associations with oroliferation, histopathological grade, molecular subtype and prognosis. J Mammary Gland Biol Neoplasia. 2022;27:67–77.
32. Wang Z, Hothorn T. bst: Gradient Boosting, 2023; https://cran.r-project.org/package=bst. R package version 0.3-24
33. Wang Z, Wang CY. Buckley-James boosting for survival analysis with high- dimensional biomarker data. Stat Appl Genetics Mol Biol. 2010;9(1): 012008.
34. Xiong J, He W, Yi GY. simexaft: simexaft, 2019; https://cran.r-project.org/package=simexaft. R package version 1.0.7.1
35. Yi GY. Statistical analysis with measurement error and misclassication: strategy, method and application. New York: Springer; 2017.
36. Zeng D, Lin DY. Efficient estimation for the accelerated failure time model. J Am Stat Assoc. 2007;102:1387–96.
37. Zhang Q, Yi GY. augSIMEX: analysis of data with mixed measurement error and misclassification in covariates, 2020; https://cran.r-project.org/package=augSIMEX. R package version 3.7.4.

Publisher's Note