

# Package: SurvMA (via r-universe)

February 21, 2025

**Title** Model Averaging Prediction of Personalized Survival Probabilities

**Version** 1.6.8

**Author** Mengyu Li [aut, cre] (ORCID: <https://orcid.org/0009-0008-4024-7573>), Jie Ding [aut] (ORCID: <https://orcid.org/0000-0002-6083-7529>), Xiaoguang Wang [aut] (ORCID: <https://orcid.org/0000-0001-7391-9788>)

**Maintainer** Mengyu Li <mylilucky@163.com>

**Description** Provide methods for model averaging prediction of personalized survival probabilities.

**License** GPL (>= 2)

**Encoding** UTF-8

**Depends** R (>= 3.5.0)

**Imports** survival, maxLik, pec, quadprog, splines, methods

**LazyData** TRUE

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.2.3

**NeedsCompilation** no

**Config/pak/sysreqs** make libicu-dev

**Repository** <https://stat-wangxg.r-universe.dev>

**RemoteUrl** <https://github.com/stat-wangxg/survma>

**RemoteRef** HEAD

**RemoteSha** a6a6ac69913fb6a8891b37887b0210640cdf24ac

## Contents

RealData.ROT . . . . .	2
SimData.APL . . . . .	2
SimData.TVC . . . . .	3
SurvMA . . . . .	3
SurvMA.Fit . . . . .	4
SurvMA.Predict . . . . .	7

**Index****9**


---

RealData.ROT	<i>RealData.ROT: A simulated dataset based on a pre-specified time-varying coefficients Cox model.</i>
--------------	--

---

**Description**

RealData.ROT

**Usage**

RealData.ROT

**Format**

An object of class data.frame with 444 rows and 8 columns.

**Examples**

```
# An example of illustrating this dataset can be found in the help page of our
# function \code{SurvMA.Fit()} by typing \code{?SurvMA.Fit()}.

# It was originally extracted from the dataset rotterdam in R package survival
# The specific extractions can be done using the following R commands
library(survival)
RealData.ROT <- na.omit(rotterdam[
  rotterdam$year %in% c(1992,1993),-c(1,2,5,6,7,12,13)
])
rownames(RealData.ROT) <- NULL
colnames(RealData.ROT)[c(7,8)] <- c("time","delta")
```

---

SimData.APL	<i>SimData.APL: A simulated dataset based on a pre-specified partly linear additive Cox model.</i>
-------------	--

---

**Description**

SimData.APL: A simulated dataset based on a pre-specified partly linear additive Cox model.

**Usage**

SimData.APL

**Format**

An object of class `data.frame` with 200 rows and 9 columns.

**Examples**

```
# An example of illustrating this dataset can be found in the help page of our  
# function \code{SurvMA.Fit()} by typing \code{?SurvMA.Fit()}.
```

---

SimData.TVC	<i>SimData.TVC: A simulated dataset based on a pre-specified time-varying coefficients Cox model.</i>
-------------	---

---

**Description**

SimData.TVC: A simulated dataset based on a pre-specified time-varying coefficients Cox model.

**Usage**

```
SimData.TVC
```

**Format**

An object of class `data.frame` with 150 rows and 8 columns.

**Examples**

```
# An example of illustrating this dataset can be found in the help page of our  
# function \code{SurvMA.Fit()} by typing \code{?SurvMA.Fit()}.
```

---

SurvMA	<i>SurvMA: Model Averaging Prediction of Personalized Survival Probabilities using R Package SurvMA.</i>
--------	--

---

**Description**

This package provides model averaging-based approaches that can be used to predict personalized survival probabilities.

---

SurvMA.Fit	<i>Model averaging prediction of personalized survival probabilities (model fitting)</i>
------------	--

---

### Description

Model averaging prediction of personalized survival probabilities (model fitting)

### Usage

```
SurvMA.Fit(
  formula,
  sdata,
  submodel = c("PL", "TVC"),
  continuous = NULL,
  control = list(K.set = c(5:10), criterion = "AIC", method = "KM")
)
```

### Arguments

formula	a formula expression, of the form response ~ predictors. The response is a Surv object (from R package "survival") with right censoring. It is used to specify the included covariates (risk factors). See the documentation for survreg and Surv in R package survival for details. The expression to the right of the "~" specifies the covariates.
sdata	a survival dataset (dataframe) in which to interpret the variables named in the formula and the cureform.
submodel	a character string defining the groups of candidate submodels, as introduced. It can be "PL" for partial linear Cox submodels or "TVC" for time varying coefficient Cox submodels.
continuous	a vector of integers representing the positions of continuous covariates within predictors specified in formula. If submodel="TVC" is set, this argument is redundant and the default value NULL is sufficient.
control	indicates more detailed control of the underlying model averaging fitting procedures. It is a list of the following three arguments: K.set specifies the range of the number of spline basis functions, with the default being K.set=c(5:10); criterion is a character string that specifies the information criterion for choosing the optimal number of B-spline basis functions and it can be either the default Akaike Information Criterion (criterion="AIC") or the Bayesian Information Criterion (criterion = "BIC"); method determines the approach to estimate the survival function of censoring time, which can be method="KM" to estimate it via the Kaplan-Meier estimator or method = "Cox" to estimate it via the Cox proportional hazards model.

## Details

This is a function used to conduct model averaging prediction (model fitting) of personalized survival probabilities. For obtaining specific predictions of personalized survival probabilities, see another function `SurvMA.Predict()`. The underlying methods are based on the paper titled "Semi-parametric model averaging method for survival probability predictions of patients", which has been published in Mengyu Li and Xiaoguang Wang (2023) [doi:10.1016/j.csda.2023.107759](https://doi.org/10.1016/j.csda.2023.107759).

## Value

A list of fitted results that contain not only parameter estimates for all candidate submodels, but also optimal averaging weights (weights).

## Examples

```
#-----#
# Basic preparations before running subsequent examples ####
#-----#

rm(list=ls(all=TRUE))

## library necessary packages
library(SurvMA)
library(survival)
#'
#-----#
# Simulated dataset: from partial linear additive Cox model ####
#-----#

## Pre-process the dataset

# - load the dataset
data(SimData.APL)
head(SimData.APL,2)

# - split the data into training and test datasets
set.seed(1)
train.index <- sort(sample(1:200,0.75*200))
sdata.train <- SimData.APL[train.index,]
sdata.test  <- SimData.APL[-train.index,]

## Fit the dataset via our model averaging method

# - fit the data using provided R function SurvMA.Fit
set.seed(1)
sol.SurvMA.PL <- SurvMA.Fit(
  formula = Surv(time,delta) ~ X + U1 + U2 + U3 + U4 + U5 + U6,
  sdata = SimData.APL, submodel = "PL", continuous = 2:7
)
print(sol.SurvMA.PL$weights)
```

```

# - do prediction using provided R function SurvMA.Predict
predict.SurvMA.PL <- SurvMA.Predict(
  object = sol.SurvMA.PL,
  covariates = sdata.test[,-c(1,2)],
  times = round(quantile(sdata.test$time,c(0.25,0.50,0.75)),2)
)
head(predict.SurvMA.PL$sprobs,2)

#-----#
# Real dataset: using time-varying coefficient Cox model ####
# - the breast cancer data originally from survival package
#-----#

## Pre-process the dataset

# - load the dataset
data(RealData.ROT)
summary(RealData.ROT$time)
table(RealData.ROT$delta)

# - plot the Kaplan-Meier curve
plot(
  survfit(Surv(time,delta) ~ 1, data = RealData.ROT),
  mark.time = TRUE, conf.int = TRUE, lwd=2,
  xlim = c(0,3200), ylim=c(0.4,1),
  xlab="Time (in Days)", ylab="Estimated Survival Probability"
)

# - test time-varying effects
TVC.Test <- cox.zph(coxph(Surv(time, delta)~., data = RealData.ROT))
print(TVC.Test)
par(mfrow=c(2,3))
plot(
  TVC.Test, resid = FALSE, lwd = 2,
  xlab = "Time (in Days)",
  ylab = paste("Coefficient for",colnames(RealData.ROT)[1:6])
)

# - split the data into training and test datasets
set.seed(1)
n <- nrow(RealData.ROT)
train.index <- sort(sample(1:n,0.75*n))
sdata.train <- RealData.ROT[train.index,]
sdata.test <- RealData.ROT[-train.index,]

## Fit the dataset via our model averaging method

# - fit the data using provided R function SurvMA.Fit
set.seed(1)
sol.SurvMA.ROT <- SurvMA.Fit(
  formula = Surv(time, delta) ~ age + meno + pgr + er + hormon + chemo,
  sdata = sdata.train, submodel = "TVC", continuous = NULL
)

```

```

)
print(sol.SurvMA.ROT$weights)

# - do prediction using provided R function SurvMA.Predict
predict.SurvMA.ROT <- SurvMA.Predict(
  object = sol.SurvMA.ROT, covariates =
    sdata.test[,!(colnames(sdata.test) %in% c("time","delta"))],
  times = round(quantile(sdata.test$time,c(0.25,0.50,0.75)))
)
head(predict.SurvMA.ROT$sprobs,2)

#-----#
# Simulated dataset: from time-varying coefficients Cox model ####
#-----#

## Pre-process the dataset

# - load the dataset
data(SimData.TVC)
head(SimData.TVC,2)

# - split the data into training and test datasets
set.seed(1)
train.index <- sort(sample(1:150,0.75*150))
sdata.train <- SimData.TVC[train.index,]
sdata.test <- SimData.TVC[-train.index,]

## Fit the dataset via our model averaging method

# - fit the data using provided R function SurvMA.Fit
set.seed(1)
sol.SurvMA.TVC <- SurvMA.Fit(
  formula = Surv(time,delta) ~ Z1 + Z2 + Z3 + Z4 + Z5 + Z6,
  sdata = sdata.train, submodel = "TVC", continuous = NULL
)
print(sol.SurvMA.TVC$weights)

# - do prediction using provided R function SurvMA.Predict
predict.SurvMA.TVC <- SurvMA.Predict(
  object = sol.SurvMA.TVC,
  covariates = sdata.test[, -c(1,2)],
  times = round(quantile(sdata.test$time,c(0.25,0.50,0.75)),2)
)
head(predict.SurvMA.TVC$sprobs,2)

```

**Description**

Model averaging prediction of personalized survival probabilities (prediction)

**Usage**

```
SurvMA.Predict(object, covariates, times)
```

**Arguments**

object	a list of all outputted results from another main function named <code>SurvMA.Fit()</code> .
covariates	is a <code>data.frame</code> with rows representing individuals and columns containing the necessary covariates used in the formula argument of <code>SurvMA.Fit()</code> .
times	specifies the time points at which survival probabilities will be calculated.

**Details**

This is a function used to conduct model averaging prediction (prediction) of personalized survival probabilities. For preliminary model fitting process, see another function `SurvMA.Fit()`.

**Value**

A list of fitted results that contain, for example, predicted values of personalized survival probabilities.

**Examples**

```
# Examples of illustrating the usages of this function can be found in the help
# page of our another function SurvMA.Fit() by typing ?SurvMA.Fit().
```



# Index

## \* datasets

RealData.ROT, [2](#)

SimData.APL, [2](#)

SimData.TVC, [3](#)

RealData.ROT, [2](#)

SimData.APL, [2](#)

SimData.TVC, [3](#)

SurvMA, [3](#)

SurvMA.Fit, [4](#)

SurvMA.Predict, [7](#)