

Support Vector Machines for Survival Analysis with R

by Césaire J. K. Fouodo, Inke R. König, Claus Weihs, Andreas Ziegler and Marvin N. Wright

Abstract This article introduces the R package `survivalsvm`, implementing support vector machines for survival analysis. Three approaches are available in the package: The regression approach takes censoring into account when formulating the inequality constraints of the support vector problem. In the ranking approach, the inequality constraints set the objective to maximize the concordance index for comparable pairs of observations. The hybrid approach combines the regression and ranking constraints in a single model. We describe survival support vector machines and their implementation, provide examples and compare the prediction performance with the Cox proportional hazards model, random survival forests and gradient boosting using several real datasets. On these datasets, survival support vector machines perform on par with the reference methods.

Introduction

Survival analysis considers time to an event as the dependent variable. For example, in the veteran's administration study (Kalbfleisch and Prentice, 2002), a clinical trial of lung cancer treatments, the dependent variable is time to death. The particularity of such a survival outcome is censoring, indicating that no event occurred during the study. For example, in the veteran's lung cancer study, some patients stayed alive until the end of the study such that time to death is censored. Because the time to event is unknown for censored observations, standard regression techniques cannot be used. The censoring indicator δ is set to be 0 or 1 for censored and not censored individuals, respectively. The main interest is to analyze the time $T \in [0, \infty]$ until an event occurs, given covariates $X \in \mathbb{R}^d$.

The most popular statistical approach for survival analysis is the Cox proportional hazards (PH) model, which is described in detail in standard textbooks (Kleinbaum and Klein, 2012; Lee and Wang, 2003). The most important advantage of the PH model is that it does not assume a particular statistical distribution of the survival time. However, the crucial assumption is that the effect of covariates on the survival variable is time independent. The hazards of two individuals are thus assumed to be proportional over time (proportional hazards assumption). The general form of the PH model is given by

$$h(t|X) = h_0(t) \exp(\beta' X), \quad (1)$$

where $\beta' = (\beta_1, \beta_2, \dots, \beta_d) \in \mathbb{R}^d$ are parameters to be estimated and $h_0(t)$ is the baseline hazard function, which is independent of the covariates. It does not need to be pre-specified or estimated in the model.

The proportional hazards assumption can easily be checked in one dimension ($X \in \mathbb{R}$), but is difficult to verify when working in higher dimensions. Another difficulty when fitting a PH model occurs when the number of covariates d exceeds the number of individuals n , since it uses the partial likelihood for parameter estimation.

One alternative approach to the PH model is to use support vector machines (SVMs). SVMs were first proposed by Vapnik (1995) as a learning method for dichotomous outcomes. They are known for their good theoretic foundations and high classification accuracy on high dimensional classification problems (Cervantes et al., 2008; Hsu et al., 2003). The formulation of an SVM supposes a target variable $Y \in \{-1, 1\}$ and covariates $X \in \mathbb{R}^d$. Assuming that the two target classes are linearly separable, there exists a linear function $f(x) = \psi x + b$ such that $y f(x) > 0$. The SVM task is to find the separating hyperplane $H(\psi, b) = \{x \mid \langle x, \psi \rangle + b = 0\}$ for the two classes with the maximal margin. The margin is defined as the smallest distance between any data point and the separating hyperplane. Data points lying at this distance from the separating hyperplane are called support vectors. They determine the margin, and must verify either $f(x) = 1$ or $f(x) = -1$. If the two classes are not linearly separable, misclassifications can be allowed. This is done by introducing slack variables $\xi_i \geq 0$, allowing but penalizing misclassifications. The slack variable for an individual i is defined as $\xi_i = |y_i - f(x_i)|$. Hence, it is $\xi_i = 0$ if i is correctly classified, $\xi_i < 1$ if it is inside the margin and on the correct side of the hyperplane, $\xi_i > 1$ if the data point lies on the wrong side of the hyperplane, and $\xi_i = 1$ if observation i lies exactly on the hyperplane. Figure 1 presents an illustration of the SVM approach for a two dimensional data set. Data points are grouped into the classes $y = 1$ (circles) and $y = -1$ (triangles). The filled and darkened data points are the support vectors defining the margin. The two red data points lying inside the margin are misclassified. They are penalized depending on their locations on the wrong or the correct side of the hyperplane, represented by the dashed line. All the other data points are correctly classified and therefore not penalized.

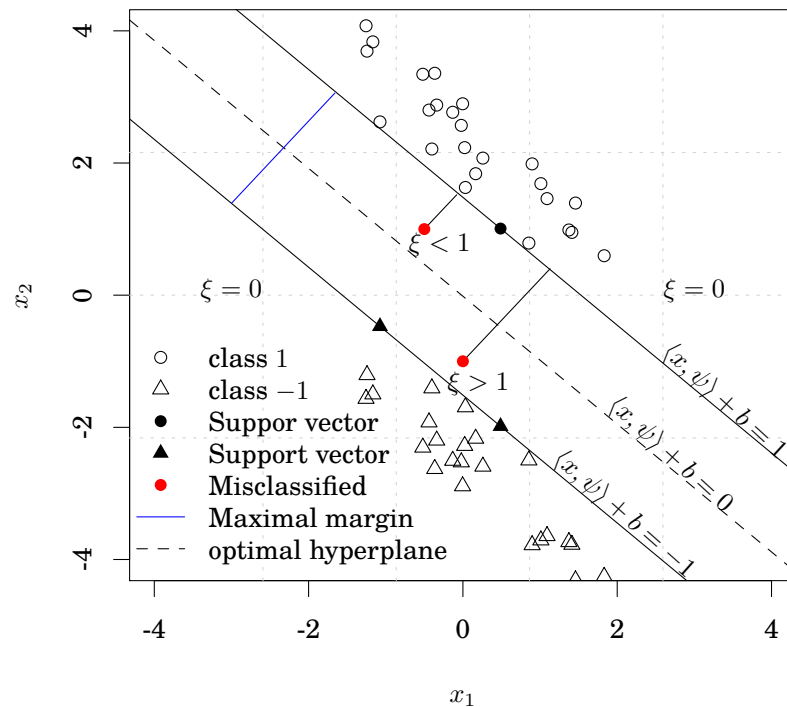


Figure 1: Illustration of the SVM approach with a two dimensional data set, split into two classes $y = 1$ (circles) and $y = -1$ (triangles). The filled data points are the support vectors defining the margin. Data points lying outside of the margin are correctly classified and not penalized ($\xi = 0$), contrary to the two misclassified and penalized red points lying inside the margin. These data points are penalized with the slack variables $\xi < 1$ or $\xi > 1$, depending on whether they lie on the correct or on the wrong side of the separating hyperplane (dashed line). This illustration is inspired by Rhode (2012) and the SVM problem was solved with the [kernlab](#) package (Karatzoglou et al., 2004).

To achieve the goal of the SVM approach described above, the following optimization problem is posed in primal space:

$$\begin{aligned} \min_{\psi, b, \xi} \quad & \frac{1}{2} \|\psi\|^2 + \gamma \sum_{i=1}^n \xi_i \\ \text{subject to} \quad & -(\mathbf{y}_i \langle \mathbf{x}_i, \psi \rangle + b) + \xi_i - 1 \leq 0 \\ \text{and} \quad & \xi_i \geq 0, i = 1, \dots, n, \end{aligned} \quad (2)$$

where ψ , b and the slack variables ξ_i are unknown, n is the number of individuals, and $\gamma > 0$ is a regularization parameter controlling the maximal margin and misclassification penalties. Instead of solving the optimization problem (2) in the primal space, the optimization problem is transformed to the dual problem, and the Lagrange function is optimized in the dual space. Details can be found in Bishop (2007). Eitrich and Lang (2006) recommends tuning the model to find the best parameter for regularization. In the SVM problem posed in (2), it is assumed that the two classes are linearly separable, but this is usually not the case. Russell and Norvig (2010) demonstrates that a set of n data points is always separable in an $n - 1$ dimensional space, which guarantees the existence of a higher dimensional space in which the two classes are linearly separable. This concept is described in more detail in the next section.

SVMs were extended to support vector regression, a variant for continuous outcomes by Vapnik (1998). More recently, several extensions to survival analysis were proposed. Van Belle et al. (2007) and Evers and Messow (2008) extended the formulation of the SVM problem with the aim to maximize the concordance index for comparable pairs of observations. This approach, also known as the ranking approach, was modified in Van Belle et al. (2008) to improve computational performance. Shivaswamy

et al. (2007) introduced a regression approach to survival analysis, based on the idea of support vector regression (Vapnik, 1998). Van Belle et al. (2011) combined the ranking and regression approaches in a single model to build the hybrid approach of SVMs for survival outcomes. The ranking approach, as proposed by Evers and Messow (2008), is implemented in the R package **survpack** available on the authors' website (Evers, 2009). The approaches of Van Belle et al. (2008) and Van Belle et al. (2011) are available in a Matlab toolbox (Yang and Pelckmans, 2014).

In the next section, we describe the three approaches for survival SVMs in detail. After that, we present the implementation of these methods in the R package **survivalsvm**. Finally, an application of survival SVMs on real data sets compares their prediction performance and runtime with established reference methods and other available implementations.

Survival support vector machines

Three approaches have been proposed to solve survival problems using SVMs: the regression (Shivaswamy et al., 2007), the ranking (Van Belle et al., 2007; Evers and Messow, 2008; Van Belle et al., 2008) and the hybrid approach (Van Belle et al., 2011). The regression approach is based on the support vector regression (SVR) (Vapnik, 1998) idea and aims at finding a function that estimates observed survival times as continuous outcome values y_i using covariates x_i . A naïve variant of this approach is to ignore all censored observations and just solve the resulting SVR problem. Unfortunately, such a formulation implies a loss of information, since it does not take the particularity of survival data into account. Shivaswamy et al. (2007) improved this formulation by including the censoring in the SVM problem. For censored observations, the time to event after censoring is unknown and thus predictions greater than the censoring time do not have to be penalized. However, all survival predictions lower than the censoring time are penalized as usual. For not censored data, the exact survival times are known and, as in standard SVR, all survival predictions lower or greater than the observed survival time are penalized. Alternatively, survival predictions from censored and not censored data can be penalized differently, as proposed by Khan and Zubek (2008). However, this implies more parameters of regularization when formulating the survival SVR problem, increasing computation times (Van Belle et al., 2011). In the present work, we implement the survival SVR as proposed by Shivaswamy et al. (2007) and formulate the problem as follows:

$$\begin{aligned} \min_{\psi, b, \xi, \xi^*} \quad & \frac{1}{2} \|\psi\|^2 + \gamma \sum_{i=1}^n (\xi_i + \xi_i^*) \\ \text{subject to} \quad & y_i - \langle \psi, F(x_i) \rangle - b \leq \xi_i, \\ & \delta_i (\langle \psi, F(x_i) \rangle + b - y_i) \leq \xi_i^* \\ \text{and} \quad & \xi_i, \xi_i^* \geq 0, \end{aligned} \quad (3)$$

where $i = 1, \dots, n$. δ is the censoring indicator and F is a function that maps observed covariates to the feature space. The feature space, compared with the original space in which data is observed, is a higher dimensional space in which the training data points are projected to be separated more easily by a hyperplane. More details about this concept can be found in Vapnik (1995). Since the feature space implies a higher dimensional space, the inner product $\langle \psi, F(x_i) \rangle$ is calculated using a kernel function to reduce runtime (Vapnik, 1995). In the case of no censoring, i.e., $\delta_i = 1$, the inequality constraints in (3) are the same as in the classical SVR problem. However, if censoring occurs, the second constraint is reduced to $\xi_i^* \geq 0$.

The ranking approach considers survival analysis based on SVMs as a classification problem with an ordinal target variable (Herbrich et al., 1999). It aims at predicting risk ranks between individuals instead of estimating survival times (Van Belle et al., 2007; Evers and Messow, 2008). Suppose two individuals i and j at time t , where an event occurs for j but not necessarily for i . Their predictions based on the classical SVM idea are $\langle \psi, F(x_i) \rangle + b$ and $\langle \psi, F(x_j) \rangle + b$, respectively. The ranking approach aims at predicting the correct ranking of y_i and y_j , meaning $(y_i - y_j)(\langle \psi, F(x_i) \rangle - \langle \psi, F(x_j) \rangle) > 0$. If we further suppose $y_j < y_i$, predictions for i and j have to verify $\langle \psi, F(x_i) \rangle - \langle \psi, F(x_j) \rangle > 0$. Therefore, the ranking approach, as proposed by Van Belle et al. (2007), is equivalent to solving the optimization problem

$$\begin{aligned} \min_{\psi, \xi} \quad & \frac{1}{2} \|\psi\|^2 + \gamma \sum_{\substack{j < i \\ \delta_i = 1}} v_{ij} \xi_{ij} \\ \text{subject to} \quad & \langle \psi, F(x_i) \rangle - \langle \psi, F(x_j) \rangle \geq 1 - \xi_{ij} \\ \text{and} \quad & \xi_{ij} \geq 0, i, j = 1, \dots, n, \end{aligned} \quad (4)$$

where

$$v_{ij} = \begin{cases} 1 & \text{if } i \text{ and } j \text{ are comparable} \\ 0 & \text{else} \end{cases}$$

is the comparison function for observations i and j . Assuming that individuals are sorted increasingly by their survival times, two observations i and j ($i < j$) are comparable if their survival times t_i and t_j verify $t_i < t_j$ and $\delta_i = 1$, i.e., the shorter observed time is not censored. If we further suppose that the observation with the shortest survival time is not censored, each observation $i = 2, \dots, n$ will be comparable to at least one other individual. The problem posed in (4) is equivalent to maximizing the concordance index (*C-index*) defined by Van Belle et al. (2007) over comparable pairs for a given prediction function u as

$$CI_n(u) = \frac{1}{n(n-1)} \sum_{v_{ij}=1} I \left[\left(u(x_i) - u(x_j) \right) \left(t_i - t_j \right) \right],$$

where $I(a) = 1$ if $a > 0$, and $I(a) = 0$, otherwise. This definition is inspired from those proposed by Harrell et al. (1984). Given an observation i , we only consider one comparable observation $\bar{j}(i) = i - 1$. We denote the first variant of the ranking approach presented in (4) as *vanbelle1*. In an alternative formulation, Van Belle et al. (2011) restrict the comparability effect to the inequality constraints as follows:

$$\begin{aligned} \min_{\psi, \tilde{\zeta}} \quad & \frac{1}{2} \|\psi\|^2 + \gamma \sum_{i=1}^n \tilde{\zeta}_i \\ \text{subject to} \quad & \langle \psi, F(x_i) \rangle - \langle \psi, F(x_{\bar{j}(i)}) \rangle \geq y_i - y_{\bar{j}(i)} - \tilde{\zeta}_i \\ \text{and} \quad & \tilde{\zeta}_i \geq 0, \end{aligned} \tag{5}$$

where $i = 1, \dots, n$. Thereby, the value 1 on the right-hand side of the first constraint in *vanbelle1* (4) is replaced by the difference $y_i - y_{\bar{j}(i)}$ between the survival times y_i and $y_{\bar{j}(i)}$ of data points i and its nearest comparable neighbor $\bar{j}(i)$. We denote this second variant of the ranking approach as *vanbelle2*. Another variant of the ranking approach was developed by Evers and Messow (2008) and implemented in the R package **survpack**, offering the possibility to fit survival SVM models using either linear or radial basis kernels. In this variant, for a data point i , all comparable pairs are considered, instead of only the nearest neighbor. The denotation *evers* is used to refer to this variant.

The hybrid approach (Van Belle et al., 2011) combines the regression and ranking approaches in the survival SVMs problem. Thus, the constraints of (3) and (5) are included in the optimization problem

$$\begin{aligned} \min_{\psi, b, \varepsilon_i, \tilde{\zeta}_i, \tilde{\zeta}_i^*} \quad & \frac{1}{2} \|\psi\|^2 + \gamma \sum_{i=1}^n \varepsilon_i + \mu \sum_{i=1}^n (\tilde{\zeta}_i + \tilde{\zeta}_i^*) \\ \text{subject to} \quad & \langle \psi, F(x_i) \rangle - \langle \psi, F(x_{\bar{j}(i)}) \rangle \geq y_i - y_{\bar{j}(i)} - \varepsilon_i, \\ & y_i - \langle \psi, F(x_i) \rangle - b \leq \tilde{\zeta}_i, \\ & \delta_i (\langle \psi, F(x_i) \rangle + b - y_i) \leq \tilde{\zeta}_i^*, \\ \text{and} \quad & \varepsilon_i, \tilde{\zeta}_i, \tilde{\zeta}_i^* \geq 0, \end{aligned} \tag{6}$$

where $i = 1, \dots, n$. To solve the optimization problems (3), (4), (5) and (6) we consider the corresponding Lagrange function in dual space and solve the quadratic optimization problem. The next section presents our implementation of these models in R.

Implementation in R

We implemented the four models presented in (3), (4), (5) and (6) in the **survivalsvm** package. The function `survivalsvm` fits a new survival model, and risk ranks are predicted using the generic R function `predict`. Common to the four models is that a quadratic optimization problem is solved when moving to the dual space. In the function `survivalsvm`, we solve this optimization problem using two quadratic programming solvers: `ipop` from the package **kernlab** (Karatzoglou et al., 2004) and `pracma` from the package **pracma** (Borchers, 2016). The function `pracma` wraps `quadprog` from the package **quadprog** (Berwin A., 2013), which is implemented in **Fortran** to solve quadratic problems as described by Goldfarb and Idnani (1982). Thereby, the kernel matrix is assumed to be positive semi-definite. If this is not the case, the function `nearPD` from the **Matrix** package (Bates and Maechler, 2016) is used to adjust the kernel matrix to the nearest positive semi-definite matrix. In contrast to

quadprog, ipop is written in pure R. Hence, the runtime of ipop is expected to be greater than that of quadprog for solving the same optimization problem. However, an advantage of ipop is that the kernel matrix does not need to be modified when solving the optimization problem. The user of `survivalsvm` can choose which solver is used for solving the quadratic optimization problem.

As in the ranking formulations (4) and (5), the hybrid formulation (6) calculates differences between comparable data points. Three options to define comparable pairs are available in `survivalsvm`: `makediff1` removes the assumption that the first data point is not censored, `makediff2` computes differences over not censored data points only and `makediff3` uses the definition described above.

The R package **survivalsvm** allows the user to choose one of the four kernels linear, additive (Damen and De Moor, 2009), radial basis, and polynomial, labeled `lin_kernel`, `add_kernel`, `rbf_kernel` and `poly_kernel`, respectively. They can be passed to the `survivalsvm` function using the `kernel` parameter.

Example of usage

To exemplify the usage, the implementation is applied to the data set *veteran* available in the package **survival** (Therneau, 2015). The function `Surv` from the package **survival** serves to construct a survival target object.

```
R> library(survivalsvm)
R> library(survival)
R> data(veteran, package = "survival")
```

First, we split the data into a training and a test data set

```
R> set.seed(123)
R> n <- nrow(veteran)
R> train.index <- sample(1:n, 0.7 * n, replace = FALSE)
R> test.index <- setdiff(1:n, train.index)
```

and next fit a survival support vector regression model

```
R> survsvm.reg <- survivalsvm(Surv(diagtime, status) ~ .,
+   subset = train.index, data = veteran,
+   type = "regression", gamma.mu = 1,
+   opt.meth = "quadprog", kernel = "add_kernel")
```

The regularization parameter is passed using the argument `gamma.mu`. For each of the models (3), (4) and (5), only one value is required, while two values are needed when fitting a hybrid model. Calling the `print` function on the output gives

```
R> print(survsvm.reg)
survivalsvm result
Call:
survivalsvm(Surv(diagtime, status) ~ ., subset = train.index, data = veteran,
  type = "regression", gamma.mu = 1, opt.meth = "quadprog", kernel = "add_kernel")

Survival svm approach           : regression
Type of Kernel                  : add_kernel
Optimization solver used        : quadprog
Number of support vectors retained : 39
survivalsvm version             : 0.0.4
```

We can now make the predictions for the observations given by `test.index`:

```
R> pred.survsvm.reg <- predict(object = survsvm.reg,
+   newdata = veteran, subset = test.index)
```

and print the prediction object:

```
R> print(pred.survsvm.reg)
survivalsvm prediction

Type of survivalsvm           : regression
Type of kernel                 : add_kernel
Optimization solver used in model : quadprog
Predicted risk ranks           : 13.89 14.95 11.12 15.6 10.7 ...
survivalsvm version           : 0.0.4
```

Real data application and comparison to other survival models

To evaluate the survival SVM models and our implementation, four publicly available survival data sets were used. The first is the data set *veteran* from the Veteran's lung cancer trial study (Kalbfleisch and Prentice, 2002), available in the package **survival**. It includes 137 individuals and 5 covariates. Second, we utilized data from the *Interpretation of a Trial Stopped Early* study (Ermerson and Banks, 1994) in which 130 individuals participated. Two survival outcomes were of interest in this study, complete remission and death. The corresponding data sets are labeled *leuk_cr* and *leuk_death*. For both data sets, 10 covariates are provided. Third, we used the Germany Breast Cancer Study Group 2 (GBSG2) data (Schumacher et al., 1994) which consists in 686 samples and 8 covariates. Finally, we considered the *Mayo Clinic Lung Cancer (MCLC)* study (Loprinzi et al., 1994) comprising 168 individuals and 8 covariates. Table 1 provides a brief summary of the data sets used.

Data set	Sample size	#Covariates	Status	Survival time
veteran	137	5	status	time
leuk_cr	130	10	complete_rem	data_cr
leuk_death	130	10	status_last_fol_up	data_last_fol_up
GBSG2	686	8	cens	time
MCLC	168	8	status	time

Table 1: Data sets used to compare prediction performance and runtime. The data sets *leuk_cr* and *leuk_death* differ only in the event considered. In *leuk_cr*, the event is a complete remission while in *leuk_death* the event of interest is death. The two last columns give the names of the censoring status and survival time variables.

For each data set, we fitted the four survival SVM models (3), (4), (5) and (6) using linear, additive and radial basis function (RBF) kernels. The ranking approach of survival SVMs implemented in the R package **survpack** is applied using linear and RBF kernels, the two kernels offered by the package. The Cox PH model, random survival forest (RSF) (Ishwaran et al., 2008) and gradient boosting (Gboost) for survival analysis Ridgeway (1999) served as reference models.

For RSF, the package **randomForestSRC** (Ishwaran and Kogalur, 2016) was used. The number of random variables for splitting and the minimal number of events in the terminal nodes were tuned when building survival trees. **randomForestSRC** refers to these parameters as `mtry` and `nodesize`, respectively. For gradient boosting models implemented in **mboost** (Hothorn et al., 2016), we fitted a PH model as base learner. The number of boosting iterations and the regression coefficient were tuned. In **mboost**, these parameters are named `mstop` and `nu`, respectively.

Tuning was conducted using 5×10 -fold nested cross validation: Data sets were randomly divided into 5 almost equally sized subsamples, and in each iteration, one of the 5 groups was used as test data set. On the remaining groups, the models were trained after tuning the model parameters with a 10-fold cross validation. The best models were chosen based on the C-index. The **mlr** package (Bischl et al., 2016a,b) was employed for parameter tuning.

Experiments were run on a high performance computing platform using 6 cores of Intel Xeon 3.33 GHz CPUs and an available memory of 183 GB. Table 2 summarizes the empirical mean runtime required by each survival SVM model to run a single resampling operation. As depicted, the runtimes are influenced by the kernel function used. Compared with linear and additive kernel functions, which required approximatively equal runtimes, the runtimes of the radial basis function (RBF) are higher. The reason is that the RBF requires one additional parameter, which has to be optimized via tuning. The effect of the number of parameters is also observed on the hybrid survival SVM approach, which required more runtime than the other approaches. Finally, the runtime of the ranking approaches *vanbelle1* and *vanbelle2*, implemented in **survivalsvm**, was considerably lower than that of the *evers* approach, implemented in the **survpack** package.

Table 3 and Figure 2 present the performance estimates of the compared models, based on the C-index. Of the 17 models, the best model received the rank 1, the worst model the rank 17. In case of ties, all tied models were assigned the mean rank value. On the *veteran* data set, the hybrid approach with the additive kernel performed best of all survival SVM models, with only slight differences to the PH model and SSVR approach. The models based on the ranking approach (*vanbelle1* and *vanbelle2*)

Data set	Kernel	Mean runtime in minutes				
		vanbelle1	vanbelle2	SSVR	hybrid	evers
veteran	linear	0.45	0.44	1.33	16.12	6.57
	additive	0.58	0.60	1.41	15.97	
	RBF	4.79	5.06	14.02	144.03	43.33
leuk_cr	linear	0.87	0.90	1.38	26.24	3.56
	additive	0.98	0.99	1.54	30.75	
	RBF	2.98	3.21	8.57	238.97	21.13
leuk_death	linear	0.29	0.30	0.96	28.08	4.82
	additive	0.33	0.36	0.96	30.84	
	RBF	2.99	3.10	8.52	269.54	20.04
GBSG2	linear	2.90	3.01	41.89	1064.53	1005.39
	additive	3.65	4.21	65.57	374.68	
	RBF	30.91	35.23	597.77	NA [†]	NA [†]
MCLC	linear	0.47	0.46	1.83	48.07	14.92
	additive	0.64	0.62	2.10	17.18	
	RBF	4.86	5.11	17.98	585.94	81.34

Table 2: Mean runtime for each survival SVM model to run a single resampling operation. This operation includes tuning of the parameters of regularization for the ranking and regression based models. Since the RBF, in comparison to linear and additive kernel functions, requires one additional parameter to be computed, its runtimes were higher. Furthermore, because the hybrid approach uses two parameters of regularization, it also needs more time to find the best parameters of regularization. The implementation of the *evers* approach in **survpack** does not include the additive kernel.

[†] Interrupted after 10 days of computation.

performed worse than the other models. The differences between the reference models were small. For the *leuk_cr* data set, the *evers* approach with the RBF kernel was the best SVM approach, followed by the hybrid approach with the additive kernel. The best reference models were the PH model and GBoost. For *leuk_death*, the hybrid approach with the additive kernel performed on par with RSF. The RSF model also performed better than the other reference models on the *GBSG2* data set, while *evers* performed worse. The hybrid model was still the best survival SVM model, with almost the same results for the linear and additive kernel. No results could be obtained for the RBF kernel in 10 days of computation. For the *MCLC* data set, the survival SVM with the hybrid approach and RBF kernel performed best, while the PH model was the best reference model. The differences were small, except for *evers*, which performed worse. In summary, there are only slight differences between the best survival SVM models and the best reference models. However, the differences between the SVM approaches and kernels were substantial.

Conclusion

We presented the R package **survivalsvm** for fitting survival SVMs. Three approaches are available in the package. First, in the regression approach, the classical SVR was extended to censored survival outcomes. Second, in the ranking approach, the ranks between model predictions and the observed survival times are maximized based on the C-index. The third approach, called the hybrid approach, combines the two first approaches into a single model. We implemented these three approaches in the **survivalsvm** package and used 5 data sets to compare the prediction performance and runtime of our implementation with the Cox PH model, RSF and gradient boosting. Furthermore, we included an implementation of a variant of the ranking approach (Evers and Messow, 2008) in the comparison. Of the survival SVM models, the hybrid approach generally performed best in terms of prediction performance but was slowest to compute. We observed only small differences between the best SVM models and the best reference models and could not determine a clear winner.

Comparing the ranking and regression based models, the *evers* approach always required more runtime than the approaches implemented in **survivalsvm**. Although the hybrid approach performed better than the others survival SVM approaches, its runtime was considerably increased. This was due to the fact that the formulation of the hybrid approach needs two parameters of regularization, while the ranking and regression approaches require only one parameter.

Method	Kernel	veteran		leuk_cr		leuk_death		GBSG2		MCLC	
		C-index (SD)	Rank	C-index (SD)	Rank	C-index (SD)	Rank	C-index (SD)	Rank	C-index (SD)	Rank
vanbelle1	linear	0.68 (0.05)	11	0.61 (0.05)	13	0.62 (0.06)	15	0.60 (0.05)	9	0.61 (0.05)	9.5
	additive	0.57 (0.07)	17	0.63 (0.06)	12	0.66 (0.08)	11	0.59 (0.03)	11	0.62 (0.05)	5.5
	RBF	0.60 (0.10)	14	0.53 (0.19)	17	0.59 (0.12)	17	0.60 (0.07)	10	0.61 (0.03)	7
vanbelle2	linear	0.59 (0.15)	16	0.59 (0.06)	14.5	0.61 (0.08)	16	0.59 (0.06)	12	0.61 (0.06)	12
	additive	0.59 (0.07)	15	0.57 (0.03)	16	0.64 (0.10)	13	0.58 (0.04)	13	0.62 (0.05)	5.5
	RBF	0.64 (0.05)	12	0.59 (0.06)	14.5	0.63 (0.11)	14	0.58 (0.07)	14	0.61 (0.05)	9.5
SSVR	linear	0.69 (0.03)	8.5	0.66 (0.05)	10	0.68 (0.08)	10	0.67 (0.04)	6.5	0.63 (0.09)	3
	additive	0.71 (0.05)	3	0.70 (0.03)	7	0.71 (0.09)	3	0.67 (0.04)	6.5	0.64 (0.08)	2
	RBF	0.70 (0.04)	4	0.68 (0.07)	9	0.70 (0.09)	4	0.67 (0.06)	8	0.61 (0.08)	14
hybrid	linear	0.69 (0.04)	10	0.71 (0.05)	5	0.68 (0.05)	9	0.68 (0.04)	4	0.62 (0.03)	4
	additive	0.71 (0.02)	1	0.72 (0.02)	4	0.72 (0.09)	1.5	0.68 (0.05)	5	0.61 (0.05)	9.5
	RBF	0.69 (0.03)	8.5	0.71 (0.06)	6	0.69 (0.11)	8	NA [†]		0.64 (0.05)	1
evers	linear	0.62 (0.08)	13	0.64 (0.11)	11	0.65 (0.09)	12	0.56 (0.03)	15	0.58 (0.13)	16
	RBF	0.70 (0.08)	6	0.73 (0.06)	3	0.70 (0.12)	6	NA [†]		0.56 (0.06)	17
PH Model		0.71 (0.04)	2	0.73 (0.03)	1.5	0.70 (0.10)	5	0.68 (0.03)	2.5	0.61 (0.05)	9.5
RSF		0.70 (0.05)	5	0.70 (0.06)	8	0.72 (0.09)	1.5	0.69 (0.03)	1	0.60 (0.02)	15
GBoost		0.70 (0.09)	7	0.73 (0.03)	1.5	0.69 (0.09)	7	0.68 (0.03)	2.5	0.61 (0.06)	13

Table 3: Prediction performance of the five survival SVMs (*vanbelle1*, *vanbelle2*, *SSVR*, *hybrid* and *evers*) and three reference methods (PH, RSF and GBoost) on 5 data sets. The C-index was computed for each method using 5×10 -fold nested cross validation. Mean values and standard deviations (SD) are shown. The C-index and a rank is presented for each model. Rank 1 was assigned to the best performing model, rank 17 to the worst performing one. In case of ties, all tied models were assigned the mean rank value. The quadprog optimizer was used in the package **survivalsvm**. The best SVM models and the best reference models are marked in light and dark gray colors, respectively. The implementation of the *evers* approach in **survpack** does not include the additive kernel.

[†] Interrupted after 10 days of computation.

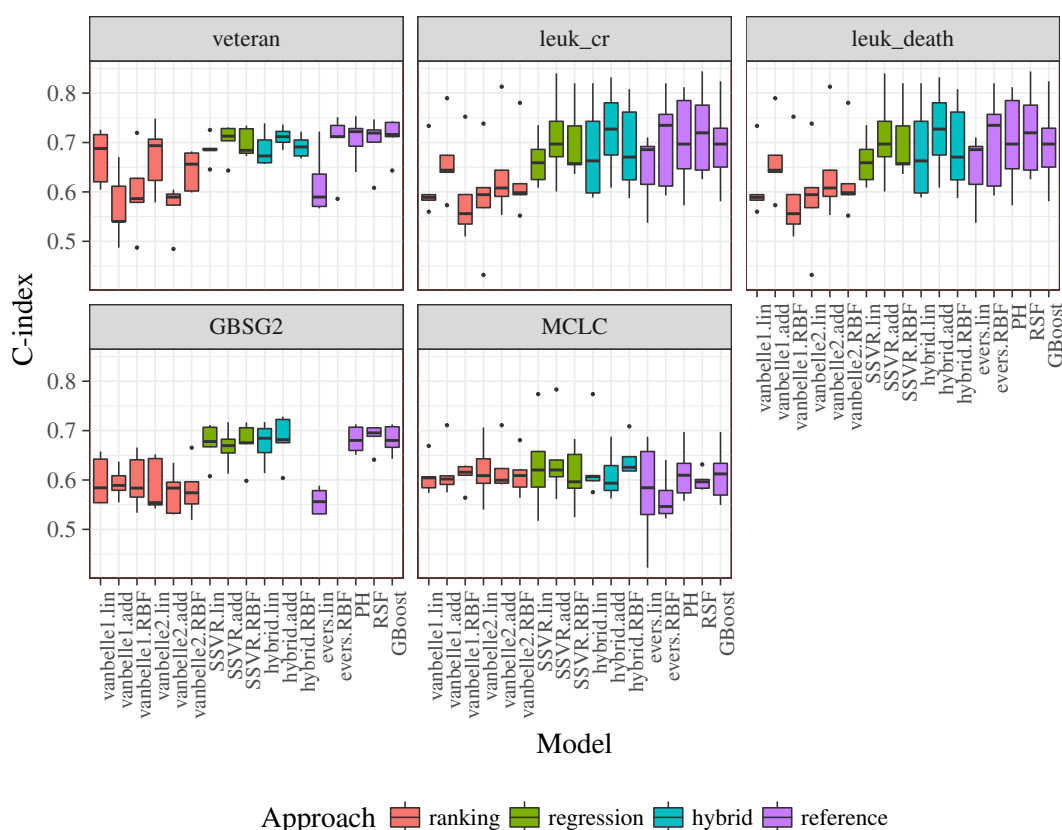


Figure 2: Prediction performance for the five survival support vector models (*vanbelle1*, *vanbelle2*, *SSVR*, *hybrid* and *evers*) and three reference methods (PH, RSF and GBoost) on 5 data sets. The C-index was computed for each method using nested 5×10 cross validation. The quadprog optimizer was used in the package **survivalsvm**. Plots were generated using the **ggplot2** (Wickham, 2009) and **tikzDevice** (Sharpsteen et al., 2016) packages.

The best performing kernel functions depended on the data set and the chosen SVM model. For the ranking approaches, the differences were larger than the regression and hybrid approaches. For the hybrid approach, the additive and RBF kernels achieved the best results. However, the runtimes for the RBF kernel were substantially larger. Again, this was due to the tuning of an additional parameter.

Our implementation utilizes quadratic programming and an interior point optimizer to solve the quadratic optimization problem derived from the primal support vector optimization problem. When the quadratic programming is used for a non positive semi-definite kernel matrix, this matrix is slightly modified to the nearest positive semi-definite matrix. Calling the interior point optimizer does not make any modification on the original matrix, but is computationally slower since the software is fully implemented in R. Pölsterl et al. (2015) proposed a fast algorithm to train survival SVMs in primal space. This algorithm is fast in low dimensions, but for high dimensional problems the authors recommended reducing the dimensions before applying an SVM algorithm. However, some special and fast algorithms, such as the sequential minimal optimization (SMO) (Platt, 1998), which are available for classical SVM optimization problems, were shown to be more accurate (Horn et al., 2016). The implementation for survival SVMs could possibly be improved by an extension of the SMO optimization procedure.

Having restrictions on only the nearest neighbor in the ranking approach, as formulated in *vanbelle1* and *vanbelle2*, can considerably improve the computational performance, but can also reduce prediction performance. In principle, the number of nearest neighbors is not limited to the choices of Evers and Messow (2008) and Van Belle et al. (2008). Since the optimal number of neighbors depends on the dataset and the availability of computational resources, it may be included as a further tuning parameter.

In conclusion, we have shown that SVMs are a useful alternative for survival prediction. The package **survivalsvm** provides a fast and easy-to-use implementation of the available approaches of survival SVMs. Our results show that the choice of the SVM model and the kernel function is crucial. In addition to prediction performance, runtime is an important aspect for large data sets. We

recommend to conduct benchmark experiments using several approaches and available kernels before analyzing a data set.

Availability

The package **survivalsvm** is available on CRAN and a development version at <https://github.com/imbs-hl/survivalsvm>. The presented examples and R code to reproduce all results in this paper are available at <https://github.com/imbs-hl/survivalsvm-paper>.

Bibliography

- D. Bates and M. Maechler. *Matrix: Sparse and Dense Matrix Classes and Methods*, 2016. URL <https://CRAN.R-project.org/package=Matrix>. R package version 1.2-6. [p415]
- T. Berwin A. *Quadprog: Functions to Solve Quadratic Programming Problems.*, 2013. URL <https://CRAN.R-project.org/package=quadprog>. R package version 1.5-5. [p415]
- B. Bischl, M. Lang, L. Kotthoff, J. Schiffner, J. Richter, Z. Jones, and G. Casalicchio. *Mlr: Machine Learning in R*, 2016a. URL <https://CRAN.R-project.org/package=mlr>. R package version 2.9. [p417]
- B. Bischl, M. Lang, L. Kotthoff, J. Schiffner, J. Richter, E. Studerus, G. Casalicchio, and Z. M. Jones. mlr: Machine learning in R. *Journal of Machine Learning Research*, 17(170):1–5, 2016b. URL <http://jmlr.org/papers/v17/15-066.html>. [p417]
- C. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag, 2nd edition, 2007. ISBN 978-0-387-31073-2. [p413]
- H. W. Borchers. *Pracma: Practical Numerical Math Functions*, 2016. URL <https://CRAN.R-project.org/package=pracma>. R package version 1.9.3. [p415]
- J. Cervantes, X. Li, W. Yu, and K. Li. Support vector machine classification for large data sets via minimum enclosing ball clustering. *Neurocomputing*, 71(4–6):611–619, 2008. URL <https://dx.doi.org/10.1016/j.neucom.2007.07.028>. [p412]
- A. Daemen and B. De Moor. Development of a kernel function for clinical data. In *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 5913–5917, 2009. [p416]
- T. Eitrich and B. Lang. Efficient optimization of support vector machine learning parameters for unbalanced datasets. *Journal of Computational and Applied Mathematics*, 196(2):425–436, 2006. URL <http://dx.doi.org/10.1016/j.cam.2005.09.009>. [p413]
- S. Ermerson and P. Banks. Interpretation of a leukemia trial stopped early. In N. Lange, editor, *Case Studies in Biometry*, chapter 14, pages 275 – 99. John Wiley & Sons, 1994. [p417]
- L. Evers. *survpack: Methods for Fitting High-Dimensional Survival Models*, 2009. URL <http://www.stats.gla.ac.uk/~levers/software.html>. R package version 0.1-4. [p414]
- L. Evers and C.-M. Messow. Sparse kernel methods for high-dimensional survival data. *Bioinformatics*, 24(14):1632–1638, 2008. URL <https://dx.doi.org/10.1093/bioinformatics/btn253>. [p413, 414, 415, 418, 420]
- D. Goldfarb and A. Idnani. Dual and primal-dual methods for solving strictly convex quadratic programs. In J. P. Hennart, editor, *Numerical Analysis*, number 909 in Lecture Notes in Mathematics, pages 226–239. Springer-Verlag, 1982. URL <https://dx.doi.org/10.1007/BFb0092976>. [p415]
- F. E. Harrell, K. L. Lee, R. M. Califf, D. B. Pryor, and R. A. Rosati. Regression modelling strategies for improved prognostic prediction. *Statistics in Medicine*, 3(2):143–152, 1984. URL <https://dx.doi.org/10.1002/sim.4780030207>. [p415]
- R. Herbrich, T. Graepel, and K. Obermayer. Support vector learning for ordinal regression. In *9th International Conference on Artificial Neural Networks (ICANN99)*, pages 97–102, 1999. URL <https://dx.doi.org/10.1049/cp:19991091>. [p414]

- D. Horn, A. Demircioğlu, B. Bischl, T. Glasmachers, and C. Weihs. A comparative study on large scale kernelized support vector machines. *Advances in Data Analysis and Classification*, pages 1–17, 2016. URL <https://dx.doi.org/10.1007/s11634-016-0265-7>. [p420]
- T. Hothorn, P. Buehlmann, T. Kneib, M. Schmid, and B. Hofner. *mboost: Model-Based Boosting*, 2016. URL <http://CRAN.R-project.org/package=mboost>. R package version 2.6-0. [p417]
- C.-W. Hsu, C.-C. Chang, and C.-J. Lin. A practical guide to support vector classification. Technical report, Department of Computer Science, National Taiwan University, 2003. URL <http://www.csie.ntu.edu.tw/~cjlin/papers.html>. Date of access: April 4, 2017. [p412]
- H. Ishwaran and U. B. Kogalur. *Random Forests for Survival, Regression and Classification (RF-SRC)*, 2016. URL <https://cran.r-project.org/package=randomForestSRC>. R package version 2.3.0. [p417]
- H. Ishwaran, U. B. Kogalur, E. H. Blackstone, and M. S. Lauer. Random survival forests. *The Annals of Applied Statistics*, 2(3):841–860, 2008. URL <https://dx.doi.org/10.1214/08-AOAS169>. [p417]
- J. D. Kalbfleisch and R. L. Prentice. *The Statistical Analysis of Failure Time Data*. John Wiley & Sons, 2nd edition, 2002. ISBN 978-0-471-36357-6. [p412, 417]
- A. Karatzoglou, A. Smola, K. Hornik, and A. Zeileis. Kernlab – an S4 package for kernel methods in R. *Journal of Statistical Software*, 11(9):1–20, 2004. URL <https://dx.doi.org/10.18637/jss.v011.i09>. [p413, 415]
- F. M. Khan and V. B. Zubek. Support vector regression for censored data (SVRC): A novel tool for survival analysis. In *Eighth IEEE International Conference on Data Mining (ICDM08)*, pages 863–868, 2008. URL <https://dx.doi.org/10.1109/ICDM.2008.50>. [p414]
- D. G. Kleinbaum and M. Klein. *Survival Analysis, a Self-Learning Text*. Springer-Verlag, 3rd edition, 2012. ISBN 978-1-4419-6646-9. [p412]
- E. T. Lee and J. W. Wang. *Statistical Methods for Survival Data Analysis*. John Wiley & Sons, 2003. ISBN 978-0-471-45854-8. [p412]
- C. L. Loprinzi, J. A. Laurie, H. S. Wieand, J. E. Krook, P. J. Novotny, J. W. Kugler, J. Bartel, M. Law, M. Bateman, and N. E. Klatt. Prospective evaluation of prognostic variables from patient-completed questionnaires. North Central Cancer Treatment Group. *Journal of Clinical Oncology: Official Journal of the American Society of Clinical Oncology*, 12(3):601–607, 1994. URL <https://dx.doi.org/10.1200/JCO.1994.12.3.601>. [p417]
- J. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. Technical report, Microsoft Research, 1998. URL <https://www.microsoft.com/en-us/research/publication/sequential-minimal-optimization-a-fast-algorithm-for-training-support-vector-machines>. Date of access: April 4, 2017. [p420]
- S. Pölsterl, N. Navab, and A. Katouzian. Fast training of support vector machines for survival analysis. In A. Appice, P. P. Rodrigues, V. Santos Costa, J. Gama, A. Jorge, and C. Soares, editors, *Machine Learning and Knowledge Discovery in Databases, ECML PKDD 2015*, pages 243–259. Springer-Verlag, 2015. URL https://dx.doi.org/10.1007/978-3-319-23525-7_15. [p420]
- A. Rhode. Learning Kernels SVM, 2012. URL <https://quantsignals.wordpress.com/2012/09/25/kernel-learning-svm>. Accessed on Jan 03, 2018. [p413]
- G. Ridgeway. The state of boosting. *Computing Science and Statistics*, 31:172–181, 1999. [p417]
- S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson, Boston, 3rd edition, 2010. ISBN 978-0-13-207148-2. [p413]
- M. Schumacher, B. G., H. Bojar, K. Huebner, M. Olschewski, W. Sauerbrei, C. Schmoor, C. Beyerle, R. L. A. Neumann, and H. F. Rauschecker. Randomized 2×2 trial evaluating hormonal treatment and the duration of chemotherapy in node-positive breast cancer patients. *Journal of Clinical Oncology*, 12(10):2086 – 2093, 1994. URL <https://dx.doi.org/10.1200/JCO.1994.12.10.2086>. [p417]
- C. Sharpsteen, C. Bracken, K. Müller, and Y. Xie. *tikzDevice: R Graphics Output in LaTeX Format*, 2016. URL <https://CRAN.R-project.org/package=tikzDevice>. R package version 0.10-1. [p420]
- P. K. Shivaswamy, W. Chu, and M. Jansche. A support vector approach to censored targets. In *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pages 655–660, 2007. URL <https://dx.doi.org/10.1109/ICDM.2007.93>. [p413, 414]

- T. M. Therneau. *A Package for Survival Analysis in S*, 2015. URL <https://CRAN.R-project.org/package=survival>. version 2.38. [p416]
- V. Van Belle, K. Pelckmans, J. Suykens, and S. Van Huffel. Support vector machines for survival analysis. In *Proceedings of the Third International Conference on Computational Intelligence in Medicine and Healthcare (CIMED2007)*, pages 1–8, 2007. [p413, 414, 415]
- V. Van Belle, K. Pelckmans, J. Suykens, and S. Van Huffel. Survival SVM : a practical scalable algorithm. In *European Symposium on Artificial Neural Networks (ESANN)*, pages 89–94, 2008. [p413, 414, 420]
- V. Van Belle, K. Pelckmans, S. Van Huffel, and J. A. K. Suykens. Support vector methods for survival analysis: a comparison between ranking and regression approaches. *Artificial Intelligence in Medicine*, 53(2):107–118, 2011. URL <https://dx.doi.org/10.1016/j.artmed.2011.06.006>. [p414, 415]
- V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995. ISBN 978-0-387-94559-0. [p412, 414]
- V. N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, New York, 1998. ISBN 978-0-471-03003-4. [p413, 414]
- H. Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag, 2009. ISBN 978-0-387-98140-6. URL <http://ggplot2.org>. [p420]
- L. Yang and K. Pelckmans. *Survlab: A Survival Analysis Toolbox*, 2014. URL <http://user.it.uu.se/~kripe367/survlab/index.html>. version 2.0. [p414]

Césaire J. K. Fouodo
Institut für Medizinische Biometrie und Statistik
Universität zu Lübeck
Universitätsklinikum Schleswig-Holstein, Campus Lübeck
Lübeck, Germany
fouodo@imbs.uni-luebeck.de

Inke R. König
Institut für Medizinische Biometrie und Statistik
Universität zu Lübeck
Universitätsklinikum Schleswig-Holstein, Campus Lübeck
Lübeck, Germany
inke.koenig@imbs.uni-luebeck.de

Claus Weihs
Fakultät Statistik
Technische Universität Dortmund
Dortmund, Germany
weihs@statistik.uni-dortmund.de

Andreas Ziegler
StatSol
Lübeck, Germany
ziegler@statsol.de

Marvin N. Wright
Institut für Medizinische Biometrie und Statistik
Universität zu Lübeck, Universitätsklinikum Schleswig-Holstein, Campus Lübeck
Lübeck, Germany
Leibniz Institute for Prevention Research and Epidemiology – BIPS
Bremen, Germany
wright@leibniz-bips.de