Survival SVM: a Practical Scalable Algorithm

V. Van Belle, K. Pelckmans, J.A.K. Suykens and S. Van Huffel *

Katholieke Universiteit Leuven - Dept. of Electrical Engineering (ESAT), SCD
Kasteelpark Arenberg 10 - B-3001 Leuven - Belgium
{vvanbell, kpelckma, johan.suykens, vanhuffe}@esat.kuleuven.be

Abstract. This work advances the Support Vector Machine (SVM) based approach for predictive modeling of failure time data as proposed in [1]. The main results concern a drastic reduction in computation time, an improved criterion for model selection, and the use of additive models for improved interpretability in this context. Particular attention is given towards the influence of right-censoring in the methods. The approach is illustrated on a case-study in prostate cancer.

1 Introduction

Traditional statistical techniques as based on Cox' proportional hazard model or the accelerated failure time model focus on explicitly modeling the underlying probabilistic mechanism of the phenomenon under study [2]. Machine learning techniques as based on SVM on the contrary take a rather different perspective; their main focus is merely to learn a predictive rule which will generalize well to unseen data [3]. The latter approach is formalized in the theory of statistical learning, integration of techniques from convex optimization, and nonlinear extensions were given using Mercer kernels, see e.g. [3, 4, 5] for an introduction.

Amongst other applications, Support Vector Machine (SVMs) have been successfully used for classification, segmentation and prognosis within cancer research [6]. SVMs have also been used for prognostic reasons by reformulation of the survival problem as a classification problem, dividing the time axis in a number of predefined intervals or classes [7], and as a rank regression problem in our previous work [1]. The algorithm proposed in the latter optimizes the concordance index between observed event times and estimated ranks of event occurrence. The relation between the concordance index and the area under the ROC curve permits the translation of advances in machine learning in a context of ordinal regression, ranking and information retrieval [8, 9].

In the present work we propose a practical alternative for the computationally demanding algorithm in [1]. Optimization of the concordance index implies the comparison of all data pairs in response and time domain. The number of comparisons is reduced by selecting appropriate pairs, resulting in a significant decrease of calculation time without notable loss of performance. Special attention is drawn towards the importance of the censoring mechanism on tuning algorithm and performance.

*We kindly acknowledge the support and constructive remarks of E. Biganzoli and P. Boracchi. Research supported by GOA-AMBioRICS, CoE EF/05/006, FWO G.0407.02 and G.0302.07, IUAP P6/04, BIOPATTERN (FP6-2002-IST 508803), eTUMOUR (FP6-2002-LIFESCIHEALTH 503094).

This paper is organized as follows. Section 2 describes the modification of support vector machines for survival data and illustrates the reduction of computational load. Section 3 summarizes results on artificial and cancer data.

2 Support Vector Machines for Survival Data

Throughout the paper the following notations are used for vectors and matrices: a lower case will denote a vector or a scalar (clear from context), an upper case will denote a matrix. The covariates x_i for each subject $i = 1, \ldots, N$ can be organized in a matrix $X \in \mathbb{R}^{d \times N}$ with $X_i = x_i$. The failure times $\{t\}_{i=1}^N$ are organized in a vector $t \in \mathbb{R}^{N \times 1}$. I_N indicates the identity matrix of size N.

Concordance Index

The concordance index (c-index) was proposed by Harrell *et al.* [10] as a measure of association between the predicted and observed failures in case of right censored data. The c-index equals the ratio of concordant to comparable pairs of data points. Two samples i and j are comparable if $t_i < t_j$ and $\delta_i = 1$, with δ a censoring variable equal to 1 for an observed event and 0 in case of censoring. A pair of samples i and j is concordant if they are comparable and $u(x_i) < u(x_j)$, with $u(x)$ the predicted health value corresponding to the sample x. Formally, the sample based c-index of a model generating predictions $u(x_i)$ for samples x_i from a dataset $\mathcal{D} = \{(x_i, t_i, \delta_i)\}_{i=1}^N$ can be expressed as

$$CI_N(u) \quad = \quad \frac{1}{N(N-1)} \sum_{i \neq j} I[(u(x_i) - u(x_j))(t(x_i) - t(x_j))] \,, \qquad (1)$$

where $I[z > 0] = 1$ if $z > 0$, and zero otherwise.

Pairwise Maximal Margin Machine

A learning strategy would try to find a mapping $u : \mathbb{R}^d \to \mathbb{R}$ which reconstructs the orders in the observed failure times measured by the corresponding CI_N. In order to overcome the combinatorial hardness which would result from directly optimizing over CI_N, [1] proposed to optimize a convex relaxation which can be written in terms of a hinge loss. Minimization of this upper bound results in an optimized empirical c-index. The optimal health function $u(x_i) = w^T \varphi(x_i) : \mathbb{R}^d \to \mathbb{R}$ can then be found as $(\hat{w}, \hat{\xi}) = \arg\min_{w,\xi} \frac{1}{2} w^T w + C \sum_{i<j, \delta_i = 1} V_{ij} \xi_{ij}$, subject to $w^T \varphi(x_j) - w^T \varphi(x_i) \geq 1 - \xi_{ij}$ and $\xi_{ij} \geq 0, \forall\, i,\ j = 1, \cdots, N$, with C a positive real constant and $V_{ij} = 1$ if the pair (x_i, x_j) is comparable, 0 otherwise. $\varphi : \mathbb{R}^d \to \mathbb{R}^{d_\varphi}$ is a feature map such that $K(x, x') = \varphi(x)^T \varphi(x')$ is a positive definite kernel function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. Taking the Lagrangian $\mathcal{L}(w, \xi; \alpha, \beta) = \frac{1}{2} w^T w + C \sum_{i<j} V_{ij} \xi_{ij} - \sum_{i<j} \beta_{ij} \xi_{ij} - \sum_{i<j} \alpha_{ij} (w^T \varphi(x_j) - w^T \varphi(x_i) - 1 + \xi_{ij})$ with multipliers $\alpha, \beta \in \mathbb{R}_+^N$ leads to the optimality conditions $w = \sum_{i<j} \alpha_{ij} (\varphi(x_j) - \varphi(x_i))$ and $CV_{ij} = \alpha_{ij} + \beta_{ij},\ \forall\ i < j$. The dual problem is obtained as $\min_\alpha \frac{1}{2} \sum_{i<j} \sum_{k<l} \alpha_{ij} \alpha_{kl} (\varphi(x_j) - \varphi(x_i))^T (\varphi(x_l) - \varphi(x_k)) - \sum_{i<j} \alpha_{ij}$, subject to $0 \leq \alpha_{ij} \leq CV_{ij},\ \forall\ i < j$. The estimated health function can be evaluated on a new point x^* as $\hat{u}(x^*) = \sum_{i<j} \alpha_{ij} (K(x_j, x^*) - K(x_i, x^*))$.

A major drawback of this algorithm is the large computational cost, making the method unapplicable for larger datasets. In the next section an approach to reduce the prohibitive computational cost is proposed.

A Scalable Nearest Neighbor Algorithm

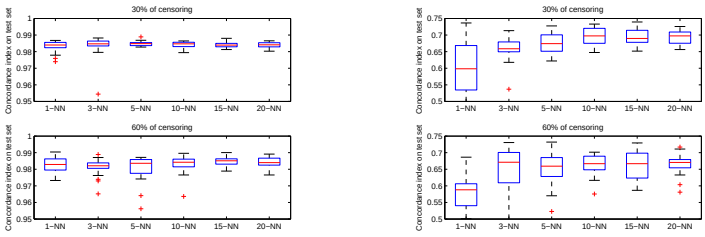
In this section we propose how the above method can be adapted to reduce computational load without considerable loss of performance. Ideas are illustrated on an artificial dataset which is made independent of the survival context since the proposed method can be used in any rank regression problem.

To find an optimal health function $u(x)$, $\mathcal{O}(qN^2/2)$ comparisons between time and health values are required, with q and N the percentage of non-censored and total number of samples in the training dataset respectively. The calculation time can be largely reduced by selecting a set $\mathcal{C}_i$ of k samples with a survival time nearest to the survival time of sample i (k-nearest neighbor (k-NN) algorithm):

$$\mathcal{C}_i = \{(i,j) : t_j \text{ is k} - \text{nearest to } t_i\}, \ \forall\ i = 1, \ldots, N\,. \tag{2}$$

This decreases the number of comparisons to $\mathcal{O}(qkN)$.

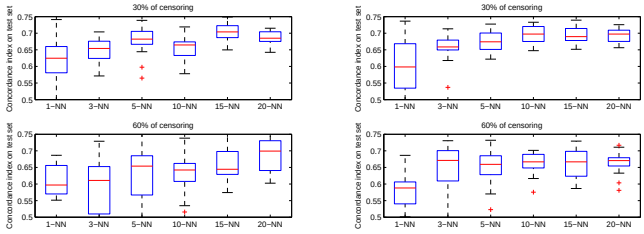
The effect of the number of neighbors in the training algorithm is illustrated in Figure 1. For a linear function (Figure 1(a)) the value of k in the k-NN algorithm has no influence on the performance. Therefore a very small number of neighbors suffices. For a non linear function the value of k accounts for a trade-off between computational load and performance. However, performance no longer increases above a certain value of k.



(a) $y_1 = x + \text{noise}$ (b) $y_2 = \text{sinc(x)} + \text{noise}$

Figure 1: Performance in function of the number of selected neighbors. For linear function a small value k suffices. For a non-linear function the performance saturates after a certain value k.

Figure 2 illustrates the importance of selecting the nearest neighbors. An artificial dataset with 100 training and test samples is created. The response depends on the covariate as $y = \text{sinc(x)} + \text{noise}$ (normally distributed). It is clearly seen that an increase in k results in an increase in performance when nearest neighbors are selected. This increase lowers for higher k. In case of randomly selected neighbors the variation in performance with increasing k is less predictive.



(a) random neighbours (b) nearest neighbours

Figure 2: Response $y = \text{sinc(x)} + \text{noise}$ (normally distributed). Concordance after selection of (a) random neighbors, (b) nearest neighbors. Only in the case of nearest neighbors a monotone relation between k and performance is found.

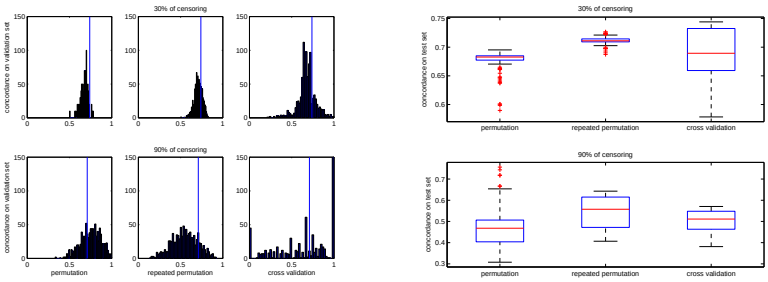
Model Selection Schemes

This subsection discusses shortly three model selection schemes in the context of censored data. The *first* scheme uses a single validation set of size $N/2$, with N the number of samples with known outcome. The *second* model selection criterion randomizes this scheme a number of times, such that one has in each iteration a disjunct training - test set. The classical 10-fold cross-validation criterion - the *third* scheme - imposes that the validation sets of size $N/10$ are disjunct over the folds.

The use if these schemes is illustrated on a small example. An artificial dataset with $y = \text{sinc}(x)+\text{noise}$ (normally distributed) was created. Figure 3 shows that for a dataset with 30% of censoring, cross-validation has a much wider spread in concordance. The second method achieves the best results. In a dataset with 90% of censoring the concordance on the validation sets ranges from 0 to 1 for tuning via cross-validation. This is explained by the very low number of events in a set of only $N/10$ samples. As a consequence there is no guarantee on the generalization performances of this model. Since the first method is dependent on the partition between training and validation samples, the results on the test set are disappointing. The second tuning algorithm forms a compromise, resulting in a better generalization ability. However, performance on high censored data are uncertain.

3 Case Study in Prostate Cancer

The performance of the k-NN model was compared with the Cox proportional hazard model on the prostate cancer dataset of Byar and Green [11]. The dataset can be found at http://lib.stat.cmu.edu/S/Harrell/data/descriptions/prostate.html. For the analysis the variables age, weight index, performance rating, history of cardiovascular disease, serum haemoglobin and Glesan stage/grade category were chosen. In this example only events due to the occurrence of prostate cancer are considered. From the 483 patients with complete observations 125



(a) validation set (b) test set

Figure 3: Concordance depending on tuning algorithm and censoring percentage q. The k-NN with k=10 is used. (a)-(b): left: single validation set; middle: 10 validation sets; right: 10-fold cross-validation; top: q=30%; bottom: q=60%. Repeated permutation has better generalization characteristics.

died of prostatic cancer, resulting in a censoring percentage of 74.12%. Since death is the endpoint of the study the data only contain right censoring. A Cox proportional hazard model trained on two third of the data resulted in a concordance index of 0.7635 on the test set (one third of the data). A nearest neighbor algorithm with 10 neighbors and a linear kernel gives a concordance of 0.7775. Figure 4 illustrates the influence of the number of neighbors on performance and calculation time. Performance does not increase between $k = 10$ and the original algorithm. Calculation time is drastically higher in the latter.

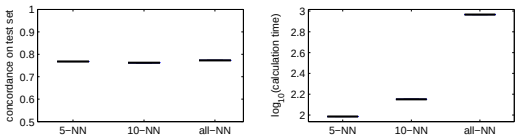
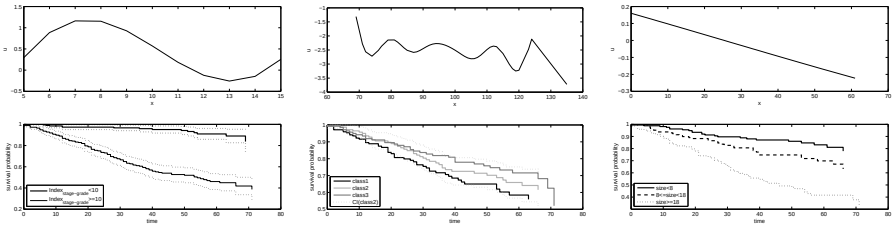


Figure 4: Performance and calculation time in function of k-NN.

Additive models can be used to improve interpretability. The relation between response and covariate x_d can be modelled as

$$u(X) \quad = \quad u_d(x_d) + u_{-d}(x_{-d}) = w_d^T \varphi_d(x_d) + w_{-d}^T \varphi(x_{-}d)\,, \tag{3}$$

where $u_d(x_d)$ is the contribution to the health function of the dth covariate, modelled by a non-linear function and $u_{-d}(x_{-d})$ the contribution of all other covariates, modelled in a linear way. Figure 5 illustrates this principle. The model predicts a difference in survival behavior for the stage/grade index below or above 10 (Figure 5(a)). Kaplan-Meier curves show a significant different observed survival in these strata, confirming the model estimation. The model suggests no relation between weight index and survival. The Kaplan-Meier curves stratified for three weight groups shows no significant difference in observed survival. An increasing tumor size results in decreasing estimated and observed survival.



(a) stage/grade index (b) weight index (c) tumor size

Figure 5: Illustration of the use of additive models for three covariates. Top: Estimated relation between covariates and survival. Bottom: Kaplan-Meier curves. Conclusions drawn from the model are confirmed by the observations.

4 Conclusions

This paper discussed the SVM for survival data as proposed in [1], and copes with issues in computational tractability, model selection and interpretability. As such, the approach is made practical for real world datasets as described in case of a retrospective prostate cancer dataset.

References

[1] V. Van Belle, K. Pelckmans, J.A.K. 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)*, 2007.

[2] J.D. Kalbfleisch and R.L. Prentice. *The Statistical Analysis of Failure Time Data.* Wiley series in probability and statistics. Wiley, 2002.

[3] V. Vapnik. *Statistical Learning Theory.* Wiley and Sons, 1998.

[4] J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle. *Least Squares Support Vector Machines.* World Scientific, 2002.

[5] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis.* Cambridge University Press, 2004.

[6] T. Furey, N. Cristianini, N. Duffy, D. Bednarski, M. Schummer, and D. Haussler. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16(10):906–914, 2000.

[7] E. Zafiropoulos, I. Maglogiannis, and I. Anagnostopoulos. *Artificial Intelligence Applications and Innovations*, volume 204 of *IFIP International Federation for Information Processing*, chapter A Support Vector Machine Approach to Breast Cancer Diagnosis and Prognosis, pages 500–507. Springer Boston, 2006.

[8] R. Herbrich, T. Graepel, and K. Obermayer. Large margin rank boundaries for ordinal regression. *Advances in Large Margin Classifiers*, pages 115–132, 2000.

[9] A. Rakotomamonjy. Optimizing auc with svms. In *Proceedings of the European Conference on Artificial Intelligence*, Workshop on ROC Analysis in AI, pages 71–79, Valencia (Spain), August 2004.

[10] F. Harrell Jr., K. Klee, R. Califf, D. Pryor, and R. Rosati. Regression modeling strategies for improved prognostic prediction. *Statistics in Medicine*, 95:634–635, 1984.

[11] D. Byar and S. Green. Prognostic variables for survival in a randomized comparison of treatments for prostatic cancer. *Bulletin du Cancer*, 67:477–490, 1980.