ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2013, i6doc.com publ., ISBN 978-2-87419-081-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100131010.

Ensembles of genetically trained artificial neural networks for survival analysis

Jonas Kalderstam, Patrik Edén and Mattias Ohlsson

Computational Biology and Biological Physics
Department of Astronomy and Theoretical Physics
Lund University - Sweden

Abstract. We have developed a prognostic index model for survival data based on an ensemble of artificial neural networks that optimizes directly on the concordance index. Approximations of the c-index are avoided with the use of a genetic algorithm, which does not require gradient information. The model is compared with Cox proportional hazards (COX) and three support vector machine (SVM) models by Van Belle et al. [10] on two clinical data sets, and only with COX on one artificial data set. Results indicate comparable performance to COX and SVM models on clinical data and superior performance compared to COX on non-linear data.

1 Introduction

In this paper we focus on models for survival analysis, designed to produce a prognostic index with the purpose of ordering data according to event times or dividing data into high and low risk groups. We use the concordance index (c-index) [1] to measure the performance of the survival models.

The proposed model directly optimizes the c-index and is based on ensembles of artificial neural networks (ANNs). Many machine learning techniques use gradients during training, and are therefore ill suited to maximize the rank-based c-index. Yan et al. [2] overcame this by introducing a smooth approximation to the step function. Van Belle et al. [3] have developed support vector machines (SVMs) for survival analysis, including c-index optimization. Another approach can be found by Raykar et al. [4] where bounds were derived for the c-index and used in the optimization. Our approach is to optimize on the c-index using a genetic algorithm, which does not require the computation of any gradients.

A similar model was introduced by us in [5] and in this study we have further developed the ensemble generation procedure and selection of optimal parameters. The purpose of this study is to evaluate our model on a selection of survival data sets and compare with other models.

2 Methods

To compare the performance of our model with existing results, we used clinical data sets from the study of Van Belle et al. [10]. A non-linear artificial data set was also used to illustrate the capabilities of the model and to compare with Cox proportional hazards (COX).

ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2013, i6doc.com publ., ISBN 978-2-87419-081-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100131010.

2.1 Clinical data

We used two publicly available clinical data sets[1]. The first one is the veteran's administration lung cancer trial (VLC) [8], a randomized trial of two treatment regimens for lung cancer, consisting of 137 patients where only 9 were alive at the end of the study. Information about treatment type, Karnofsky performance score, time from diagnosis to randomization, age, prior therapy and tumor histology is available.

The second data set originates from the North Central Cancer Treatment Group (NCCTG) [9]. This data set consists of 228 lung cancer patients where 63 were censored. Available information consists of age, sex, ECOG performance score, Karnofsky score rated by physician and patient, calories consumed at meals and weight loss in last six months. Van Belle et al. [10] also used these two data sets, but in the case of NCCTG only the uncensored cases were used. For comparative reasons, we also trained our model on this subset (denoted MLC as in [10]).

For all data sets, a training set of 2/3 and a test set of 1/3 stratified for censoring was used.

2.2 Artificial data

This data set (AD) is constructed in a way which makes it impossible to solve by a linear model. The data is highly non-monotonic and the expected result for a model such as COX is not better than random. 1000 training and 2000 test cases were generated using 10 covariates. Half of the data was censored and noise was uniformly added to both the covariates and the survival time. See [5] for further details.

2.3 The concordance index

To define the c-index we introduce the survival time t_j for patient j. In the case of a censored patient, t_j is the follow-up time.

A pair of patients are said to be *useable* if the patient with the shorter time is un-censored. Let p_j be the prognostic index for patient j, with the aim of sorting patients according to actual survival times. A useable pair is *in concordance* if the sample with shorter time t has a higher index p.

The c-index is simply the fraction of useable pairs in concordance. Thus, a c-index of 1.0 indicates a perfect ordering and a value of 0.5 is no better than random ordering.

[1]Both available in the survival package in the R-environment (last accessed 2012-12-05): http://cran.r-project.org/web/packages/survival/index.html

ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2013, i6doc.com publ., ISBN 978-2-87419-081-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100131010.

2.4 The prognostic index model

We model the prognostic index $p(\mathbf{x})$ using a multilayer perceptron with one hidden layer,

$$p(\mathbf{x}) = \sum_{j=1}^{J} \omega_j \cdot \varphi \left(\sum_{k=1}^{K} \tilde{\omega}_{jk} x_k + \tilde{\omega}_{j0} \right) + \omega_0,$$

given set of K covariates $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iK})$ for each patient i. A large index will indicate a high risk of an event. The activation function $\varphi()$ is here set to the hyperbolic tangent function tanh(). This model can easily become linear by setting $J = 1$ and $\varphi(x) = x$.

The weights $(\omega_j, \tilde{\omega}_{jk})$ are determined by minimizing an objective function. In our case the objective function is the c-index, which is not differentiable with respect to the weights, thereby limiting the number of minimization methods one can use.

2.5 Training using genetic algorithms

We have chosen to utilize a *genetic algorithm* which allows us to train directly on the c-index without requiring gradient information. Many possible implementations of genetic algorithms exist. The implementation used in this study is based on Montana and Davis [6].

Initialization of the population N_{pop} number of ANNs, are initialized with random weights from the exponential distribution

$$p(\omega) = \frac{2}{\sigma} \exp\left(-\frac{|\omega|}{\sigma} \right), \tag{1}$$

thus favoring smaller weights while allowing for larger weights in some cases. Appropriate values for both N_{pop} and σ are tuned before final training.

Creation of a new generation New ANNs are created by crossover where the child ANN inherits each weight randomly from one of its two parents. An ANN with rank k, when sorted by performance, is selected as parent with the probability $p(k) \propto (0.95)^{k-1}$. This results in a 90% probability to select a rank of 45 or less for a population of 100. In the mutation step, each weight ω is modified with probability P_μ according to $\omega = \omega + \Delta\omega$ where $\Delta\omega$ is a random number from the distribution in equation 1, where σ gradually decreases, as described below.

The child ANN is now evaluated and inserted into the population. Then, the ANN with the worst rank is deleted, thereby keeping the population size constant. A generation has elapsed when the number of generated children equals the population size.

The width of $\Delta\omega$ decreases over time as σ decreases linearly with each generation. σ reaches half its starting value at generation γ_{half} which, together with

ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2013, i6doc.com publ., ISBN 978-2-87419-081-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100131010.

the mutation probability P_μ and number of generations, are tuned before the final training.

2.6 Ensembles of prognostic models

To decrease the problem of over-fitting and thereby possibly increase the generalization performance, an ensemble approach using Bagging [7] was employed for the prognostic index model. The bagging process was additionally stratified for censoring.

With a rank-based objective function, the ensemble result cannot be generated by direct averaging of individual member outputs since they can be expected to differ wildly between ANNs, even if they are equivalent in terms of the c-index. To be able to average outputs, they will first be transformed into ranks. Let N_i

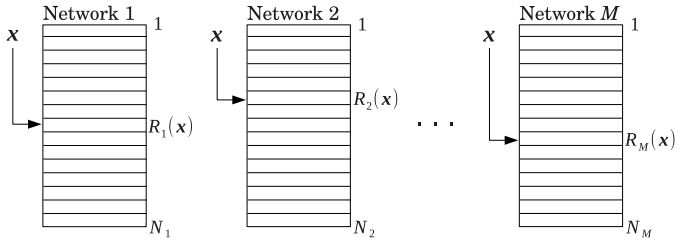

Fig. 1: A new patient $\mathbf{x}$ obtains a rank number $R_i(\mathbf{x})$ for each ANN i by comparing with the training data output list for each ANN.

be the number of training data that was used to train ensemble member i. The output $y_i(\mathbf{x})$ for ANN i and patient $\mathbf{x}$ will give rise to a rank $R_i(\mathbf{x})$. This rank is determined by inserting the output $y_i(\mathbf{x})$ into the sorted list of training data outputs for ANN i. The rank $R_i(\mathbf{x})$ is the position of $y_i(\mathbf{x})$ in the sorted list (see Fig. 1). To allow for ANNs trained with different sizes of training sets, the rank is divided by $N_i + 1$, yielding a number between 0 and 1 called the **normalized relative rank** $\tilde{R}_i(\mathbf{x})$. From a c-index point of view, ANN output $y_i(\mathbf{x})$ and $\tilde{R}_i(\mathbf{x})$ are completely equivalent. Computing an ensemble output $y_c(\mathbf{x})$ is now straightforward and is the average of normalized relative ranks,

$$y_c(\mathbf{x}) = \frac{1}{M} \sum_{i=1}^{M} \tilde{R}_i(\mathbf{x}) \tag{2}$$

where M is the size of the ensemble.

3 Results

The parameters for the genetic algorithm (see Table 1) were tuned to maximize the training performance over 10 runs, for each evaluated parameter value. The number of hidden nodes however, was selected based on 5x3-fold cross-validation

ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2013, i6doc.com publ., ISBN 978-2-87419-081-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100131010.

on the training set to avoid possible over-fitting. Cross-validation was also used to establish a suitable ensemble size and resulted in 30 ANNs for all cases. This model selection was done for all data sets and the parameters selected in each case are presented in Table 1.

Table 1: Number of hidden nodes was selected using 5x3-fold cross-validation on the training set. The other parameters were tuned to maximize training performance.

Data set	#Hidden	N_{pop}	P_μ	σ	γ_{half}	Generations
VLC	1	100	0.8	0.4	100	200
MLC	2	100	1.0	0.5	100	200
NCCTG	3	100	1.0	0.4	100	200
AD	14	100	0.4	0.25	100	200

The test results presented in Table 2 were based on 1000 bootstraps of the test set, for each individual data set. The statistical significance of the ANN results compared to the COX results was calculated using the Wilcoxon rank sum test.

Table 2: Median c-index for all models on the data sets. ANN and COX are based on 1000 bootstraps of the test set. P-values for the difference between ANN and COX results were calculated using the Wilcoxon rank sum test and resulted in $p < 0.005$ for all comparisons. The other models are, as reported in [10], based on 50 randomizations of the training and test sets.

Model	VLC	MLC	NCCTG	AD
ANN	0.65 ± 0.05	0.61 ± 0.04	0.63 ± 0.04	0.90 ± 0.00
COX	0.64 ± 0.04	0.59 ± 0.04	0.62 ± 0.04	0.50 ± 0.01
Results reported in [10] for comparison				
MODEL 1	0.61 ± 0.07	0.60 ± 0.05		
MODEL 2	0.69 ± 0.03	0.62 ± 0.05		
RANKSVMC	0.62 ± 0.08	0.59 ± 0.05		
$\text{PH}_{\text{linear}}$	0.68 ± 0.03	0.61 ± 0.04		

Also presented in Table 2 are the results reported by Van Belle et al. [10] for the SVM models. They used a slightly different methodology and their results are based on 50 randomizations between training and test sets. A slight discrepancy in results can thus be expected and to illustrate this we also include the result for Cox proportional hazards as reported by them ($\text{PH}_{\text{linear}}$).

4 Discussion & Conclusions

We have developed a prognostic index model for survival data based on an ensemble of ANNs that optimizes directly on the c-index.

ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2013, i6doc.com publ., ISBN 978-2-87419-081-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100131010.

Compared to the COX model we found a small but significant advantage for our ensemble based ANN model. The comparison with the three SVM models presented by Van Belle et al. [10] showed similar performance. The differences found can possibly be attributed to the different testing methodologies used, as indicted by the different results found for the equivalent models COX and $\mathrm{PH}_{\mathrm{linear}}$. The ANN model has major advantages over COX in the case of non-linear data.

Training directly on the c-index has the potential to introduce a bias; due to the bias of the c-index itself to overestimate model performance on censored data. This can be seen if one compares the results for MLC and NCCTG, which are the same data sets except MLC contains only the non-censored cases. The improvement on NCCTG could potentially be an overestimation of performance due to additional censored cases. This is however a problem that all models that are evaluated using the c-index are affected by. An advantage of our model is that it can be trained on any performance metric, not just the c-index.

References

[1] F. E. Harrell, K. L. Lee, and D. B. Mark. Multivariable prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors. *Statistics in medicine*, 15(4):361–87, 1996.

[2] L. Yan, D. Verbel, and O. Saidi. Predicting prostate cancer recurrence via maximizing the concordance index. In *Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '04*, page 479, New York, New York, USA, 2004. ACM Press.

[3] V. Van Belle, K. Pelckmans, J. A. K. Suykens, and S. Van Huffel. Support Vector Machines For Survival Analysis. In E Ifeachor and A Anastasiou, editors, *Proceedings of the third international conference on Computational Intelligence in Medicine and Healthcare (CIMED)*, pages 1–8, 2007.

[4] V. Raykar, H. Steck, B. Krishnapuram, C. Dehing-Oberije, and P. Lambin. On ranking in survival analysis: Bounds on the concordance index. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, volume 20, pages 1209–1216. MIT Press, Cambridge, MA, 2008.

[5] J. Kalderstam, P. Edén, P-O. Bendahl, C. Strand, M. Fernö, and M. Ohlsson. Training artificial neural networks directly on the concordance index for censored data using genetic algorithms. Submitted manuscript, 2012.

[6] D. J. Montana and L. Davis. Training feedforward neural networks using genetic algorithms. In *Proceedings of the 11th international joint conference on Artificial intelligence - Volume 1*, IJCAI'89, pages 762–767, San Francisco, CA, USA, 1989. Morgan Kaufmann Publishers Inc.

[7] L. Breiman. Bagging Predictors. *Machine Learning*, 24(2):123–140, 1996.

[8] J.D. Kalbfleisch and R.L. Prentice. *The statistical analysis of failure time data*. Wiley series in probability and mathematical statistics. Probability and mathematical statistics. J. Wiley, 2002.

[9] 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*, 12(3):601–607, 1994.

[10] 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.