A neural-Bayesian approach to survival analysis

Bart Bakker and Tom Heskes *
Foundation for Neural Networks, Geert Grooteplein 21
6525 EZ Nijmegen, The Netherlands
{bartb,tom}@mbfys.kun.nl

Abstract

Standard survival analysis can be given a neural interpretation in terms of a multi-layered perceptron (MLP) with exponential transfer functions. More hidden units accommodate more complex relationships. The neural interpretation suggests a Bayesian analysis, which allows one to introduce sensible priors and to sample from the posterior. We also propose a method for computing p-values from the obtained ensemble of networks, because, in the end, this is the kind of information medical experts are familiar with. We apply our methods on a database regarding patients with ovarian cancer.

1 Introduction

The goal of survival analysis (in medical terms) is to estimate the chances of a patient's survival as a function of time, given the medical information available on this patient. A well-known way to conduct such an analysis, is the proportional hazards method designed by Cox [1]. In this method the hazard function $h(t;x)$, which estimates the probability density of death occurring at time t, consists of two independent parts. The first part is the proportional hazard, $h(x) = \exp(w^T x)$, which depends on patient information (x) only, the second part is a time-dependent baseline hazard $h_0(t)$.

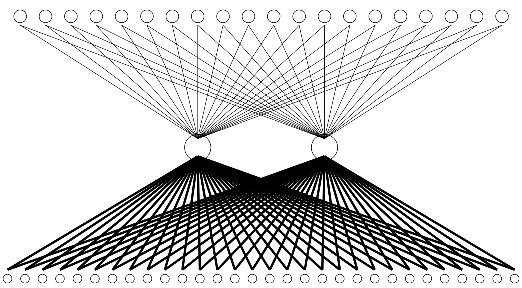
Cox's analysis has been successfully applied to real-world databases (see e.g. [2]) using a straightforward regression on w to estimate $h(t;x)$. The proportional hazards method can be implemented in the form of a multi-layer perceptron (MLP) with one hidden unit and exponential transfer functions, as will be shown in Section 2. By adding more hidden units, more complex relationships can be modeled. In the recent literature, other combinations between neural networks and survival analysis have been proposed and applied successfully (see e.g. [2, 3]).

As described in Section 3, our neural interpretation also suggests a Bayesian analysis to overcome some of the weaknesses of the standard approach. Sensible priors can be introduced, which, in combination with the available data, lead to a posterior distribution on the weights of the neural network. This posterior is intractable, but with sampling techniques such as Hybrid Markov Chain Monte Carlo (HMCMC, see e.g. [4]), one can sample from this posterior to obtain an ensemble of neural networks.

However, in practice, medical experts are not interested in ensembles of neural networks: they are raised with the concept of p-values. In the context of survival analysis, p-values are used to measure the relevance of patient characteristics. In Section 4, we propose a method for computing approximate or pseudo p-values from an ensemble of neural networks.

The proposed methods are tested on a medical database of 929 ovarian cancer patients, of whom (next to their medical information) the time of death or censoring (extraction from the research group for reasons other than ovarian cancer) has been recorded. More information about this database can be found in [3].

*This research was supported by the Technology Foundation STW, applied science division of NWO and the technology programme of the Ministry of Economic Affairs.



survival probabilities:

$$F(t_i; x) \equiv F_i = \exp\left[\sum_j v_{ij} y_j\right]$$

features:

$$y_j = \exp\left[\sum_k w_{jk} x_k\right]$$

patient characteristics:

$$x_k$$

Figure 1: Neural interpretation of survival analysis.

2 Neural survival analysis

Given the hazard function $h(t;x)$ the survivor function $F(t;x)$ indicating the probability to survive time t can be formulated as

$$F(t;x) = \exp\left[-\int_0^t dt' h(t';x)\right] . \qquad (1)$$

The probability density $f(t;x)$ for a patient to die at time t is then given by

$$f(t;x) = -\frac{\partial F(t;x)}{\partial t} = h(t;x)F(t;x) .$$

The likelihood function $P(D|w)$, expressing the probability to observe the data in database D given the model parameters w, then immediately follows from

$$P(D|w) = \prod_{\mu \in \text{uncensored}} f(t^\mu; x^\mu) \prod_{\nu \in \text{censored}} F(t^\nu; x^\nu) \qquad (2)$$

The first product is over the patients of whom the time of death is known. An element in the second product specifies the estimated probability of censored patient ν to be alive at time t^ν, the time patient ν was taken out of the study. Since in this case the time of death is not known, this is the strongest prediction that can be verified.

To find the optimal parameters w^{ML}, the likelihood function as given in (2) should be maximized. Maximum likelihood fitting has the advantage that it can be done sequentially: it can be shown that the maximum likelihood parameters w^{ML} only depend on the ordering of the times of deaths, not on their exact value. (All remaining time-dependent information can be modeled in the function $h_0(t)$.) Given w^{ML}, the maximum likelihood choice for $h_0(t)$ follows directly from a straightforward procedure similar to Kaplan-Meier estimation [1].

Standard Cox analysis can be implemented in a multi-layered perceptron with one hidden unit and T output units specifying the survivor function $F(t;x)$ at T discrete points in time (Figure 1). The input-to-hidden weights are denoted by w, the hidden-to-output weights by v. All units have exponential transfer functions, which makes this network different from standard MLP's with hyperbolic tangents. In this network $F(t_i;x)$ (the i^{th} network output) is given by

$$F(t_i;x) = \exp\left[v_i \exp(w^T x)\right] ,$$

which, using $v_i = -\int_0^{t_i} dt' h_0(t')$, yields (1).

Cox analysis, and thus the neural equivalent with one hidden unit, has the disadvantage that the impact of the data through the patient characteristics x is constant in time. By just adding more hidden units, i.e., choosing

$$F(t_i;x) = \exp\left[\sum_j v_{ij} \exp(w_j^T x)\right] ,$$

more complex relationships can be represented. Note that with more hidden units, the decoupling of ML estimation in two independent parts (first w, then $h_0(t)$ or v) is lost. A summary of this neural implementation is given in Figure 1.

Priors

- Smooth time-dependency:
$$P(v|\lambda_1) \propto \prod_j \exp\left[-\frac{\lambda_1}{2}\sum_i (v_{i+1,j} - 2v_{ij} + v_{i-1,j})^2\right]$$
with $P(\lambda_1) = \mathcal{G}(0.003, 20)$.

- Low hidden unit activity:
$$P(w|\lambda_2) \propto \prod_j \exp\left[-\frac{\lambda_2}{2} w_j^T C w_j\right]$$
with $C = \frac{1}{P}\sum_\mu x^\mu\, x^{\mu\,T}$ and $P(\lambda_2) = \mathcal{G}(0.01, 50)$.

- Decreasing survival probabilities: $v_{i+1,j} < v_{ij}$ for all i and j.

Likelihood $P(D|v,w) = \prod_{\mu \in \text{uncensored}} f(t^\mu; x^\mu) \prod_{\nu \in \text{censored}} F(t^\nu; x^\nu)$

Posterior $P(v,w|D) \propto \int d\lambda_1 d\lambda_2 P(D|v,w) P(v|\lambda_1) P(w|\lambda_2) P(\lambda_1) P(\lambda_2)$

Figure 2: Bayesian probability model.

3 Bayesian inference

In principle, by adding more hidden units, the risk of overfitting increases. However, even with a single hidden unit (i.e., standard Cox), the risk of overfitting is in general rather high: in most studies there is a tendency to consider quite a lot of different patient characteristics and, from a more technical point of view, especially the time-dependent part $h_0(t)$ or v is left completely free and has a tendency to become highly non-smooth.

The solution proposed in this paper is a Bayesian approach. Instead of merely searching for the maximum likelihood solution w^{ML}, we seek to construct a probability distribution over all possible values of the parameters in our network. This distribution will not only depend on the data in our database, but also on prior knowledge about the nature of the problem. Using Bayes' formula, the prior and the data likelihood can be transformed into a posterior distribution. First we will discuss how we choose our priors (see also Figure 2), then we describe how to sample from the posterior and the results that we obtained in this way.

Our first prior is actually a demand: since the probability to survive time t is always larger than the probability to survive time $t + \Delta t$ ($\Delta t > 0$), $v_{i+1,j}$ must always be smaller than v_{ij}. This constraint is met by defining $v_{ij} = -\sum_{i'=1}^{i} |\gamma_{i'j}|$, where γ_{ij} is a hidden network parameter. The prior $P(v)$ prevents the hazard from becoming too sharp as a function of time. It introduces a preference for survivor functions which decay exponentially. $P(w)$ prevents large activities of hidden units, i.e., prefers small weights. Incorporation of the covariance matrix C makes this preference independent of a (linear) scaling of the inputs x (see Figure 2 for the precise definitions).

The probability of the parameters w and v given the data and the hyperparameters follows from Bayes' formula:

$$P(w,v|D,\lambda) = \frac{P(D|w,v)P(w|\lambda_2)P(v|\lambda_1)}{P(D)},$$

with $P(D|w,v)$ the likelihood as in (2) and $P(D)$ an irrelevant normalizing constant. We choose gamma distributions for the hyperparameters λ_1 and λ_2. The posterior $P(w,v|D)$ follows by integrating out these hyperparameters (see Figure 2).

It is impossible to calculate $P(w,v|D)$ exactly, but one can draw a (large) set of samples from this posterior using sampling techniques such as Hybrid Markov Chain Monte Carlo [4]. Bayesian network inference now consists of drawing enough samples (each sample is in fact one realization of a neural network) to approach $P(w,v|D)$

sufficiently close. This then yields an ensemble of neural networks. An estimate of the survivor function can be obtained by averaging over the outputs of the networks in the ensemble.

Test results

We compared the Bayesian with the maximum likelihood approach for networks with one and two hidden units. The Bayesian networks are obtained through HMCMC sampling over the posterior $P(w,v|D)$ with parameters as in Figure 2. The corresponding maximum likelihood solutions are trained to maximize the likelihood function $P(D|w,v)$.

To test the different approaches, the input-output pairs in the database were randomly split into three parts: in each of the 3 different runs, 2 parts were used for training and the remaining part for testing. The results are summarized in Table 1.

	ML estimate	**Bayesian**
1 hidden	705	678
2 hidden	698	676

Table 1: Test errors for the two different approaches and network architectures.

The test errors in Table 1 are defined as minus the loglikelihood of independent test data not used for training and inference, averaged over 3 runs. For both maximum likelihood estimation and the Bayesian approach, the difference between one and two hidden units is not significant. However, the difference between the Bayesian approach and the maximum likelihood approach is significant for both architectures. Summarizing, the Bayesian approach seems to work well, although, for this database, the extra complexity introduced by more than one hidden unit is not rewarded.

4 Assigning p-values.

In the medical statistics literature, p-values are used to assign a measure of importance to each model input. Roughly speaking, the p-value measures the evidence in favor of the null hypothesis that the "true" model does not contain input k. For standard Cox analysis, there are several ways to estimate p-values more or less analytically, for example, by refitting a model with the particular input left out and using the likelihood-ratio statistic.

Here we will define a pseudo p-value which, at least in the limit of a large number of patterns P, coincides with the usual one in the case of one hidden unit and no priors, i.e., for standard Cox, but can also be used for other architectures. Furthermore, we can estimate this p-value from an ensemble of models as obtained, for example, through sampling a Bayesian posterior distribution. How this can be done is summarized in Figure 3.

First, we summarize the ensemble of network models by fitting a new network m to the average output of the ensemble (we use w and m to denote all parameters of the model, not just the input-to-hidden weights):

$$m = \operatorname*{argmin}_{w} \langle d(\bar{y}(x), y(w,x)) \rangle_x \ ,$$

where $\bar{y}(x)$ is the output on input x averaged over all ensemble networks, $y(w,x)$ denotes the output of the network with parameters w on input x, $\langle \ldots \rangle_x$ stands for an average over a set of inputs x, and $d(y,y')$ is a distance measure between two outputs. The distance measure used here is the Kullback-Leibler distance corresponding to the likelihood function in Figure 2. To compute the distance between any model with parameters w and the representative m, we can use the same distance function:

$$D(w,m) = \langle d(y(w,x), y(m,x)) \rangle_x \ .$$

To estimate the validity of the null hypothesis that input k is irrelevant, we need to compute the smallest distance $D_{\min}$ from

1. Summarize the ensemble by:
   - A representative network m, fitted to the average outputs of the ensemble.
   - A distribution $P(D)$ of distances from the ensemble networks to m.
2. To compute a pseudo p-value p^* for input k:
   - Estimate the minimal distance $D_{\min} = \min_{w; w_{\cdot k}=0} \frac{1}{2}(w-m)^T F(m)(w-m)$ of the constrained network ($w_{\cdot k} = 0$) to the unconstrained network m ($m_k \neq 0$).
   - Compare with the density $P(D)$: $p^* = \int_{D_{\min}}^{\infty} dD\, P(D)$

Figure 3: Computing p-values.

a network with no weights connected to input k ($w_{\cdot k} = 0$) to the representative m:

$$\begin{aligned} D_{\min} &= \min_{w; w_{\cdot k}=0} D(w,m) \\ &\approx \min_{w; w_{\cdot k}=0} \frac{1}{2}(w-m)^T F(m)(w-m) \end{aligned}$$

where the second step is based on a quadratic approximation of the distance close to m. If $D(w,m)$ can de derived from a loglikelihood or Kullback-Leibler distance, $F(m)$ corresponds up to irrelevant constants to the Fisher information metric. In this quadratic approximation, the minimal distance can be easily found to obey (see e.g. [5])

$$D_{\min} = \frac{1}{2} w_{\cdot k}^T [F_{kk}^{-1}]^{-1} w_{\cdot k}\ .$$

In the neural-network literature, this kind of "pruning" method is called Optimal Brain Surgeon [6].

Now that we have computed the minimal distance, we can define the p-value as the probability that the distance between the "true" model and the representative m is at least as large as this minimal distance:

$$p^* = \int_{D_{\min}}^{\infty} dD\, P(D)\ , \qquad (3)$$

where $P(D)$ is a distribution of distances.

The distribution $P(D)$ is unknown, but can be estimated from the distribution of distances from the ensemble networks to m. The underlying assumption is that these are "typical" distances, distances that correspond to the uncertainty that we still have about the true model. There are several ways to estimate this distribution. If the number of samples is sufficiently large, the integral (3) might be computed by simply counting the distances larger than $D_{\min}$. However, for smaller ensembles a more solid approach is to try and fit a parametric form. For our data, the gamma distribution

$$P(D) = \frac{\kappa^n}{\Gamma(n)} D^{n-1} \mathrm{e}^{-\kappa D}\ , \qquad (4)$$

with two free parameters κ and n yielded a reasonable fit. The parameters can be obtained through the method of moments:

$$\kappa = \frac{\langle D \rangle}{\langle D^2 \rangle - \langle D \rangle^2} \quad \text{and} \quad n = \frac{\langle D \rangle^2}{\langle D^2 \rangle - \langle D \rangle^2}\ ,$$

with averages taken over all ensemble networks. Substitution of (4) into (3) yields

$$p^* = \frac{1}{\Gamma(n)} \int_{\kappa D_{\min}}^{\infty} dt\, \mathrm{e}^{-t} t^{n-1} = 1 - P(n, \kappa D_{\min})\,,$$

with $P(n,x)$ the incomplete gamma function which can be found in any handbook of mathematical functions and is related to the chi-square probability distribution through $P(n,x) = P(2x|2n)$.

Results

We applied the proposed method to the ensemble of networks with one hidden unit obtained through HMCMC sampling on the posterior $P(w,v|D)$ as explained in Section 3. Straightforward computation of the p-values yielded that in the complete network none of the inputs are relevant ($p^* \approx 1$ for all inputs). This does not mean that the output of the network is completely independent of its input, but rather that in the complete network each input can sufficiently be replaced by a linear combination of the other inputs (at least on this data set). Therefore, we applied a backward elimination procedure ("input pruning") by successively removing the least relevant input (see e.g. [5]). The procedure was stopped when 21 of the 31 inputs were removed. The remaining inputs and their p-values can be found in Table 2.

Input	p**-value**
Patient's performance	0.000
# tumors after surgery	0.004
Presence of leucocytes	0.091
Tumor size after surgery	0.152
Cell type	0.259
Patient's length	0.272
Creatinine clearance	0.303
Presence of ascites	0.313
Type of treatment	0.412
Hexamemthylmelamine	0.944

Table 2: Remaining inputs and their p-values.

The p-values computed in this way show some similarities with the ones obtained in other studies on the same database using standard Cox and (a completely different) neural network approach [3]. In general, however, the p-values obtained here are somewhat higher, i.e., inputs are considered less relevant. One of the reasons might be that the ensemble of networks is not particularly well summarized by a single representative. In fact, this might well be one of the reasons why the Bayesian approach yields better results than a single maximum likelihood estimate (see Section 3). A method to obtain a better representation by first clustering the models and computing p-values based on the cluster centers, is described in [7].

5 Conclusions

The results of this paper show that a neural-Bayesian approach to survival analysis can be worthwhile, although the database under study does not benefit from the extra complexity introduced by adding hidden units. The Bayesian machinery, however, reduces the risk of overfitting by taking into account sensible prior information about the smoothness of the mappings. It seems that the solution obtained by averaging over the ensemble is more complex than any solution which can be obtained by a single model with one or even two hidden units, but without overfitting the data.

Being able to translate the complex ensemble to p-values, more information is acquired about the database, which is of fundamental importance for medical applications.

References

[1] D. Cox and D. Oakes. *Analysis of Survival Data.* Chapman Hall, London, 1984.

[2] C. Volinsky, D. Madigan, and A. Raftery. Bayesian model averaging in proportional hazards models: Assessing the risk of a stroke. *Applied Statistics*, 46:433–448, 1997.

[3] H. Kappen and J. Neijt. Neural network analysis to predict treatment outcome. *The Annals of Oncology*, 4:S31–S34, 1993.

[4] R. Neal. *Bayesian Learning for Neural Networks.* Springer-Verlag, New York, 1996.

[5] P. van de Laar and T. Heskes. Pruning using parameters and neuronal metrics. *Neural Computation*, 11(4), 1999.

[6] B. Hassibi and D. Stork. Second order derivatives for network pruning: optimal brain surgeon. In *NIPS 5*, pages 164–171, San Mateo, 1993. Morgan Kaufmann.

[7] B. Bakker and T. Heskes. Model clustering by deterministic annealing. In M. Verleysen, editor, *Proceedings of the European Symposium on Artificial Neural Networks '99*, pages 87–92, 27 rue du Laekenveld - B 1080 Brussels - Belgium, 1999. D-Facto.