survnnet: Neural network survival models: Version 1.1

Ruth M. Ripley

April 16, 2004

1 Introduction

This software package contains functions to fit several different non-linear survival models. There are three main functions, each with predict, print, and summary methods.

The first function, survnnet, fits discrete time models of various sorts, and *parametric* models in continuous time. The second, phnnet, fits proportional hazards models in continuous time, and the third phtnnet, fits a time-varying version of phnnet, where the ratio of hazards varies over time.

This explanation is written in terms of analysing the time to relapse of breast cancer patients but the software is applicable in any survival context.

2 Background

Traditional survival models are linear in the predictors. Various non-linear methods have been suggested (splines, trees and local methods), but none has become widely used. Neural network models provide an alternative to these methods and offer a relatively parsimonious framework compared to that of splines.

Details of nnet, the neural network package on which these functions are based, may be found in Venables and Ripley (2002).

In our survival problem, we take our covariates as inputs to the net and the time to relapse as the output(s): neural network models can then be used to extend the various statistical models. We can fit unspecified non-linear functions of the

covariates and also allow the effect of the covariates to vary with time. In some models the time to relapse appears as an input rather than an output, the output being an indicator of relapse or not at that time.

The simplest method considers survival for some fixed number of months or years, and ignores patients censored before that time, thereby giving a standard two-class classification problem. Omitting censored patients may bias the result, however. If we can estimate the survival probability for these censored patients, we can include them and hope to reduce the bias.

There are several ways to use more than two intervals, but only one will be described here: this estimates the probabilities of relapse in the time periods *less than one year, one to two years, two to three years, three to five years, and greater than five years*. We can now include all patients for whom the outcome for at least the first time period is known, thus reducing the bias due to censoring. We fit a model which ignores the ordering of the outputs, since this can be done with a softmax network: an ordinal model is available in the package `nolr` developed by Mathieson (1996), and in our experience is a better model.

Using continuous time values allows the problem to be treated as regression rather than classification. The package allows fitting of four non-linear parametric models: exponential, Weibull, log-logistic and log-normal. It also includes functions to fit non-linear proportional hazards models, and a time-varying model based on the proportional hazards model where the ratio of hazards is allowed to vary over time.

Two new models have been added in the latest version: a log-normal model where the shape is also allowed to vary with the covariates, and a model directly fitting the log hazard as the output of a regression network with time as an additional input. The cumulative hazard is evaluated as an integral from the log hazard.

3 Models

We work in the framework of a non-zero random variable T representing the time to relapse of a patient, or in the discrete time case, a multinomial (or binomial) variable Y which takes the value 1 if a relapse occurs within a particular time period and 0 otherwise.

3.1 Discrete time

The simple model which predicts directly the probability of relapse within 5 years is a standard classification network with likelihood function

$$\prod_{\text{patients}} p_i^{t_i}(1-p_i)^{(1-t_i)}$$

and log likelihood

$$\sum_{\text{patients}} t_i \log p_i + (1-t_i)\log(1-p_i) \tag{1}$$

where p_i is the probability of relapse within five years for the ith patient and t_i, the target, is 1 if the patient relapsed within five years and 0 if not.

In the case of censored observations we do not know the outcome. If we wish to include these patients with an estimate of their outcomes, we would include them twice, once with target 1, with weight equal to their estimated probability of relapse before five years, $\hat{t}_i$ say, and once with target 0, with weight $1-\hat{t}_i$. This weights their contribution to the log likelihood, and leads to the same expression as (1), using $\hat{t}_i$ in place of t_i.

The model requires entropy likelihood fitting, one logistic output unit, and may include skip-layer connections if desired:

```
plearn <- survnnet(ti ~ ., data=X, decay = 0.1, size=2,
                   bias.decay=25, entropy = T, skip = T)
```

The other classification network estimates probabilities of relapse in the time periods *less than one year, one to two years, two to three years, three to five years, and greater than five years*. We include all patients for whom the outcome for at least the first time period is known and obtain the likelihood

$$\prod_{\text{patients}} \sum_{k=m_i+1}^{l_i} p_{ki} \tag{2}$$

where m_i is the last time period the ith patient is known to have survived without relapse, l_i is the final time period during which the patient may have relapsed, and p_{ki} is the probability that the ith patient relapses in time period k. For a patient known to relapse in the second year, say, $m_i + 1 = l_i = 2$, while for a patient lost to follow-up in the third time period without known relapse we would have $m_i = 2, l_i = 5$.

We ignore the ordering of the time periods, and fit the model

$$\log p_k - \log p_1 = \eta_k(\boldsymbol{x}) \qquad (k = 2, \ldots, 5)$$

(an $\eta_1(\boldsymbol{x})$ is not required: since the probabilities must add to 1 only four can vary independently). This model is fitted using a *softmax* neural network where we first obtain the outputs y_k from a network with five linear output units:

$$y_k = \sum_j w_{jk} x_j + \sum_h w_{hk} \ell\left(\sum_j w_{jh} x_j \right) \qquad (k = 1, \ldots, 5)$$

and then calculate the probabilities by

$$p_k = \frac{\exp(y_k)}{\sum_l \exp(y_l)}.$$

(Here we have set $\eta_k(\boldsymbol{x}) = y_k - y_1$, for $k = 2, \ldots, 5$. The output y_1 is not necessary, but a symmetric model is preferred when using weight decay.)

This can be fitted with the following

```
plearn <- survnnet(cat ~ .,data=X, decay = 0.1,  size = 2,
bias.decay=25, censored=T, skip = T)
```

Here the variable `cat` should be set up to be a matrix, the ith row corresponding to the ith patient, with value 1 for time periods between $m_i + 1$ and l_i and zero otherwise.

The output is the estimated absolute probability of relapse in each of the intervals: to predict prognosis, consider the cumulative probability over the intervals.

3.2 Continuous time

3.2.1 Parametric models

Details of the density functions and survivor functions are standard: they are quoted here to demonstrate the parametrisation used in the network. This differs from that used in `survreg`, for example, in having the coefficients of the model reversed in sign.

Exponential distribution

$$f(t) = \lambda \exp(-\lambda t), \qquad S(t) = \exp(-\lambda t).$$

Weibull distribution

$$f(t) = \lambda p(\lambda t)^{(p-1)} \exp(-(\lambda t)^p), \qquad S(t) = \exp(-(\lambda t)^p).$$

Log-logistic distribution

$$f(t) = \frac{\lambda p(\lambda t)^{(p-1)}}{[1+(\lambda t)^p]^2}, \qquad S(t) = \frac{1}{1+(\lambda t)^p}.$$

Log-normal distributtion

$$f(t) = 2(\pi)^{-1/2} p t^{-1} \exp\left(\frac{-p^2(\log(\lambda t))^2}{2}\right), \qquad S(t) = 1 - \Phi(p\log(\lambda t))$$

where Φ is the incomplete normal integral.

In all cases, we model $\log \lambda$ as a function of $\boldsymbol{x}$ by a neural network with a single linear output. Since p must be positive, we use $\alpha = \log p$ in the optimisation. The shape parameter does not depend on the inputs: a single value for the training data will be fitted. Again, skip-layer units may be used if desired.

For example, to fit a log-logistic model,

```
plearn <- survnnet(Surv(RFS,Relapse) ~ ., data = X,
  model = 'llog', decay = 0.1, bias.decay = 25, size = 2,
  skip = T, alpha =0.1)
```

Allowing the shape parameter to vary An additional model is available, which fits a log normal distribution allowing the shape parameter to vary with the covariates. It is selected as `model='lnormvar'`, and uses a neural network with two outputs. A parameter `varWt` is available to control the relative size of the outputs, as one would normally want the shape (variance) to vary more slowly than the mean. The neural network output corresponding to the shape parameter is divided by `varWt`, which has default value 10. (It should be possible to achieve more flexible control over this behaviour using variable weight decays, but the parameter `varWt` may suffice and is easier to use.)

Modelling the log hazard directly The option `model='hazard'` simply uses the output from the network as the log hazard, and calculates the cumulative hazard (which is required for the likelihood) by summing over a fixed number of intervals between zero and the survival time. Ths bins used are all the same width for a particular subject, but will vary between subjects. The number of intervals used is a parameter `Nintervals`, with default value 20. The larger the number of intervals the slower the model fit will be.

3.2.2 Proportional hazards

For this model we assume only that the ratio of the hazards for two patients is constant over time:

$$h(\boldsymbol{x}, t) = h_0(t) \exp \eta(\boldsymbol{x}).$$

where h_0, the *baseline hazard*, is unspecified.

We model $\eta(\boldsymbol{x})$ as the output from a neural network with one linear output unit. We omit the bias on the output unit since this is incorporated in the baseline hazard h_0.

The log partial likelihood is

$$\sum_r \left(\eta(\boldsymbol{x}_r) - \log \sum_a \exp \eta(\boldsymbol{x}_a) \right). \tag{3}$$

where r runs over the relapses only and a runs over all the patients at risk at the time of this event. The censored patients only occur in the denominator. The partial likelihood is invariant if a constant is added to each of the scores $\eta(\boldsymbol{x})$.

In practice relapse times may be tied: they may be recorded only to the nearest day. Various adjustments are possible for this case: we chose to break them arbitrarily, as the difference in partial likelihood will be small unless there are many ties.

This may be fitted using the following

```
plearn <- phnnet(Surv(RFS,Relapse) ~ ., data=X,  decay = 0.1,
 size = 2, skip=T, bias.decay=25)
plearn <- phnnet(Relapse ~ ., data=X,  decay = 0.1,  size = 2,
 skip=T, bias.decay=25)
```

The second form is acceptable if the data are sorted in decreasing order of RFS, otherwise the former will allow the sort to be performed by `phnnet`.

The baseline cumulative hazard (estimated by the Breslow estimator) can be obtained by using the option `dohaz=T`.

3.2.3 Time-varying

This model is similar to the proportional hazards one above but relaxes the restriction that the effect of covariates must be constant over time. We allow the function η to depend on time as well as the covariates $\boldsymbol{x}$. We want the model to vary smoothly with time, so we divide the whole range of follow-up times into a

small number of *zones*, and use the number of the zone as an input to the neural network, alongside the covariates. The model becomes

$$h(\boldsymbol{x}, t) = h_o(t) \exp \eta(\boldsymbol{x}, t).$$

This is a very flexible model, since each patient can have a different shaped hazard function. The amount of flexibility can be controlled by the number of zones used, and by the regularisation used in the fitting process. The log partial likelihood changes subtly:

$$E = -\sum_r \left(\eta(\boldsymbol{x}_r, t_r) - \log \sum_a \exp \eta(\boldsymbol{x}_a, t_r) \right)$$

The terms $\eta(\boldsymbol{x}_r, t_r)$ are calculated using the value of η at the time zone corresponding to t_r, the time of the event r, and the partial sums over the risk set must also be calculated using this time zone for all the patients. Thus as the time of the event changes, so does the contribution to the partial sums of each patient in the risk set. The same values of η are used in calculating the derivatives.

The survival probability at 5 years differs from the proportional hazard case since the cumulative hazard is different. The Breslow estimator can be used as before to obtain a baseline cumulative hazard estimate (using times as in the calculation of the partial likelihood and the derivatives), but the relationship of the patient's cumulative hazard $H(\boldsymbol{x}, t)$ and the baseline hazard $H_0(t)$ is different: if we define z_i to be the half-open interval $[z_{i1}, z_{i2})$, and let the z_i denote the zones into which we have divided t we have

$$\begin{aligned} H(\boldsymbol{x}, t) &= \int_0^t h_0(u) \exp \eta(\boldsymbol{x}, u) du \\ &= \sum_i \exp \eta(\boldsymbol{x}, z_i) \int_{z_i} h_0(u) du \\ &= \sum_i \exp \eta(\boldsymbol{x}, z_i) [H_0(z_{i2}) - H_0(z_{i1})]. \end{aligned}$$

From this formula we can derive the survival probability using the relationship $S(\boldsymbol{x}, t) = \exp(-H(\boldsymbol{x}, t))$.

The simplest way to use this model is

```
plearn <- phtnnet(Surv(RFS,Relapse) ~ ., data=X,  decay = 0.1,
 size = 2, skip=T, bias.decay=25, breakpts=c(0,733,1826,5000))
```

Other options for specifying the time zones are described in the help page.

Further details of these models and their use may be found in Ripley (1998).

References

Mathieson, M. J. (1996) Ordinal models for neural networks. In *Neural Networks in Financial Engineering. Proceedings of the Third International Conference on Neural Networks in the Capital Markets* (Eds A.-P. N. Refenes, Y. Abu-Mostafa, J. Moody and A. Weigend), pp. 523–536, Singapore. World Scientific.

Ripley, R. M. (1998) *Neural network models for breast cancer prognosis.* D.Phil. thesis, University of Oxford. Available at `http://www.stats.ox.ac.uk/~ruth/`.

Venables, W. N. and Ripley, B. D. (2002) *Modern Applied Statistics with S.* New York: Springer-Verlag, Fourth edition.