Supplementary Material
CASPAR: A Hierarchical Bayesian Approach to predict Survival Times in Cancer from Gene Expression Data

L. Kaderali, T. Zander, U. Faigle, J. Wolf, J.L. Schultze, R. Schrader

Bioinformatics, 2006, to appear

1 Bayesian Cox Model - Details

For readers not familiar with the Cox regression model, a brief summary is given in the following.

1.1 The Cox Regression Model

Let a clinical study with L patients be given. Consider the survival time of patient j, $t^{(j)}$, a realization of a random variable $T^{(j)}$. In addition to the observed survival times $t^{(j)}$ of patient j, $j = 1, ..., L$, one is given explanatory variables $x^{(j)} = (x_1^{(j)}, ..., x_n^{(j)})$ for each patient j, assumed to correlate to survival of the patient in some way. These variables are, in our setting, the gene expression values as measured by a DNA microarray. The data for individual j in the study is thus given as

$$\left\{t^{(j)}, x_1^{(j)}, ..., x_n^{(j)}\right\},$$

where $x_i^{(j)}$ is the i–th gene of the j–th individual. A functional dependence $h_\theta(x^{(j)}) = t^{(j)} + \xi$ is assumed, where θ is a vector of regression parameters, and the observed survival times are corrupted by noise ξ. Hence, it is assumed that there is a functional relationship linking the clinical outcome $t^{(j)}$ to the explanatory variables $x^{(j)}$.

In the *proportional hazards model*, also known as *Cox regression model*, one assumes that the hazard for a given patient with observed characteristics

x, $\lambda(t|x)$, is given by

$$\lambda(t|x,\theta) := \lambda_0(t)e^{\theta x}, \tag{1}$$

where $\lambda_0(t)$ is an arbitrary base-line hazard function, and $\theta = (\theta_1, ..., \theta_n)$ is the vector of regression parameters. The base-line hazard $\lambda_0(t)$ describes the hazard when $x = 0$, and is assumed to depend on time t.

The probability density function $f(t)$ of the survival time T, conditioned on the explanatory variables x and the regression parameters θ, is then given by

$$f(t|x,\theta) = \lambda_0(t)e^{\theta x} \exp\left[-e^{\theta x}\int_0^t \lambda_0(u)\,du\right], \tag{2}$$

and, by simple integration, the survivor function $F(t|x,\theta) = \int_t^\infty f(t'|x,\theta)\,dt'$ can be shown to be

$$F(t|x,\theta) = \left(\exp\left[-\int_0^t \lambda_0(u)\,du\right]\right)^{\exp[\theta x]}. \tag{3}$$

1.2 Estimating Cox' θ under censoring

We will now try to estimate one individual distribution for each patient j, depending on the patients gene expression values, stored in the vector $x^{(j)}$. For a regression model such as the Cox model (2), the question is how the model parameters θ can be estimated when censoring is present in the data. In order to do so, it becomes necessary to explicitly model the censoring process.

Let the censoring time $C^{(j)}$ for patient j be a random variable with survivor function

$$G(t) = P\left\{C^{(j)} \geq t\right\}$$

and density function $g(t)$. Furthermore, let $C^{(1)}, ..., C^{(L)}$ be independent of one another and of the failure times $T^{(1)}_{true}, ..., T^{(L)}_{true}$. Note, that this model includes the relevant case where a study ends at some prespecified time, and patients enter the study randomly over time.

Under these assumptions, what is the probability of observing a survival time $t^{(j)}$ in the interval $[t, t+dt]$? If no censoring acts on $T^{(j)}$ ($\delta^{(j)} = 0$),

$$\begin{aligned} P\left\{t^{(j)} \in [t,t+dt], \delta^{(j)} = 0 | x^{(j)}, \theta\right\} &= P\left\{t^{(j)} \in [t,t+dt], C^{(j)} > t | x^{(j)}, \theta\right\} \\ &= f(t|x^{(j)},\theta)G(t)dt. \end{aligned} \tag{4}$$

Equivalently, the probability of censored survival time $t^{(j)}$ in the interval $[t, t+dt]$, with censoring taking place at $t^{(j)}$ ($\delta^{(j)} = 1$), is

$$P\left\{t^{(j)} \in [t,t+dt], \delta^{(j)} = 1 | x^{(j)}, \theta\right\} = g(t)F(t|x^{(j)},\theta)dt. \tag{5}$$

Hence, when the censoring is not dependent on θ, the likelihood of the data can be written as a function of θ:

$$\begin{aligned} L_D(\theta) &= \prod_{j=1}^{L} \begin{cases} f(t^{(j)}|x^{(j)},\theta)G(t^{(j)}) & \text{if } \delta^{(j)}=0 \text{ (no censoring)} \\ g(t^{(j)})F(t^{(j)}|x^{(j)},\theta) & \text{if } \delta^{(j)}=1 \text{ (censored time)} \end{cases} \\ &= \prod_{j=1}^{L} \left[f(t^{(j)}|x^{(j)},\theta)G(t^{(j)})\right]^{1-\delta^{(j)}} \left[g(t^{(j)})F(t^{(j)}|x^{(j)},\theta)\right]^{\delta^{(j)}} \\ &\propto \prod_{j=1}^{L} f(t^{(j)}|x^{(j)},\theta)^{1-\delta^{(j)}} F(t^{(j)}|x^{(j)},\theta)^{\delta^{(j)}}. \end{aligned} \tag{6}$$

The maximum likelihood solution to the problem of estimating θ now is a parameter θ^* maximizing (6). Note that this solution is not necessarily unique. Employing the density function $f(t|x,\theta)$ and the survivor function $F(t|x,\theta)$ from the Cox model,

$$f(t|x,\theta) = \lambda_{Cox}(t)e^{\theta x}e^{-\lambda_{Cox}(t)t\exp[\theta x]} \tag{7}$$

and

$$F(t|x,\theta) = \exp\left[-\lambda_{Cox}(t)t\right]^{\exp[\theta x]}, \tag{8}$$

where $\lambda_{Cox}(t)$ is an assumed baseline hazard function, a maximum likelihood solution for θ from the Cox model can be computed using standard optimization techniques.

2 Details on Gradient Descent Computation

In our work, we use the Cox regression model (2), with constant baseline hazard function $\lambda_0(t) = \lambda_{Cox}$. Hence,

$$f(t|x,\theta) = \lambda_{Cox}e^{\theta x}e^{-\lambda_{Cox}t\exp[\theta x]} \tag{9}$$

and

$$F(t|x,\theta) = \exp\left[-\lambda_{Cox}t\right]^{\exp[\theta x]}. \tag{10}$$

To account for censoring, we need to specify a distribution over the censoring time. Assuming that a clinical study is run for a predetermined time τ_{total}, and that patients enter the study randomly over time as they are diagnosed, it may appear reasonable to assume a uniform distribution $g(t)$ over the total study time τ_{total}, with survivor function $G(t) = \int_t^\infty g(u)\,du$.

The probability distribution over the observed data given the model parameters, $p(D|\theta)$, is then given by

$$p(D|\theta) = \prod_{j=1}^{L} \left[f(t^{(j)}|x^{(j)}, \theta) G(t^{(j)}) \right]^{1-\delta^{(j)}} \left[g(t^{(j)}) F(t^{(j)}|x^{(j)}, \theta) \right]^{\delta^{(j)}}, \quad (11)$$

see equation (6). Maximizing this probability with respect to the parameter θ yields a parameter vector that is most likely given the data, provided one has no a-priori expectations on θ. However, this estimate of θ is based on very few data points – in fact, frequently, far less data points than θ has components.

The problem of dataset sparsity can be remedied by assuming a strong prior distribution on θ, which includes additional knowledge such as the expectation that most features will be irrelevant, and drives the solution towards corresponding "sparse" weight vectors θ, where most components θ_i are in the proximity of zero. Instead of the likelihood (11), we then maximize the posterior $p(\theta|D)$. The expectation that most features will be irrelevant can be encoded by the ARD prior

$$p(\theta) = \frac{a^{nr}}{\Gamma(\gamma)^n (2\pi)^{n/2}} \prod_{i=1}^{n} \int_0^\infty \sigma_{\theta_i}^{r-2} \exp\left[-\frac{1}{2} \frac{\theta_i^2}{\sigma_{\theta_i}^2} - a\sigma_{\theta_i} \right] d\sigma_{\theta_i}, \quad (12)$$

as described in the paper. This prior will not only keep the overall length of the weight vector θ small, in addition, it will guarantee that most of the weights are close to zero, and only few weights are allowed to be significantly non-zero – and hence to have a relevant impact on the prediction.

Using Bayes' theorem, the posterior distribution is given by

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)}, \quad (13)$$

and this term should be maximized with respect to θ.

When carrying out this maximization, the denominator $p(D)$ can be neglected, since it is independent of θ. The same holds for the factors $g(t)$ and $G(t)$ in $p(D|\theta)$, compare equation (11). Inserting (11) and (12) into equation (13), taking the negative logarithm and dropping terms independent of θ, maximization of $p(\theta|D)$ with respect to θ is equivalent to minimizing

$$MP = -\sum_{i=1}^{n} \ln \int_0^\infty \sigma_{\theta_i}^{r-2} \exp\left[-\frac{1}{2} \frac{\theta_i^2}{\sigma_{\theta_i}^2} - a\sigma_{\theta_i} \right] d\sigma_{\theta_i}$$

$$-\sum_{j=1}^{L}\left[(1-\delta^{(j)})\left(\theta x^{(j)}-\lambda_{Cox}t^{(j)}e^{\theta x^{(j)}}\right)-\delta^{(j)}\left(\lambda_{Cox}t^{(j)}e^{\theta x^{(j)}}\right)\right]. \tag{14}$$

The first derivative of (14) with respect to θ_i is given by

$$\begin{aligned} g_i &= \frac{\int_0^\infty \sigma_{\theta_i}^{r-2}\exp\left[-\frac{1}{2}\frac{\theta_i^2}{\sigma_{\theta_i}^2}-a\sigma_{\theta_i}\right]\frac{\theta_i}{\sigma_{\theta_i}^2}\,d\sigma_{\theta_i}}{\int_0^\infty \sigma_{\theta_i}^{r-2}\exp\left[-\frac{1}{2}\frac{\theta_i^2}{\sigma_{\theta_i}^2}-a\sigma_{\theta_i}\right]\,d\sigma_{\theta_i}} \\ &\quad -\sum_{j=1}^{L}\left[(1-\delta^{(j)})\left(x_i^{(j)}-\lambda_{Cox}t^{(j)}x_i^{(j)}e^{\theta x^{(j)}}\right)-\delta^{(j)}\left(\lambda_{Cox}t^{(j)}x_i^{(j)}e^{\theta x^{(j)}}\right)\right]. \end{aligned} \tag{15}$$

This derivative is required for gradient descent optimization, the integrals in equations (14) and (15) are not analytically tractable, but can be approximated numerically using Gauss-Laguerre quadrature.

3 Convergence of the Algorithm and Choice of Starting Points for Gradient Descent

The choice of starting point for the gradient descent may have a significant impact on results, in particular, when the function (14) minimized has many local optima. The usual solution to this problem is to start the gradient descent from multiple points, and select the best result from these iterations, or to use randomization techniques such as simulated annealing to avoid local minima.

In the special case where we expect a solution with most of the components θ_i in the proximity of zero, as discussed in the paper, one can actually make use of this expectation when choosing the starting point for the gradient descent. Why choose a starting point very far from the origin, when we expect the solution to be in the proximity of the origin for most components θ_i anyway? This would only require the gradient descent algorithm to traverse the long way back to the origin, with the danger of getting stuck in local optima on the way.

It is for this reason that in most computations presented in this work, the gradient descent algorithm was started at the origin. A comparison with randomly chosen starting points showed that this is feasible, and usually

yields superior results. However, one problem with this choice of starting point is that the prior distribution over the weights, $p(\theta)$, has its peak at the origin, and thus the posterior distribution $p(\theta|\mathcal{D})$ potentially has a local optimum here. To avoid this local optimum, we usually carried out one or two steps of the gradient descent algorithm on the likelihood $p(\mathcal{D}|\theta)$ only, before continuing on the posterior distribution $p(\theta|\mathcal{D}) \propto p(\mathcal{D}|\theta)p(\theta)$. This makes sure the algorithm does not remain in the local optimum at the origin.

4 Running Time

All calculations were carried out on a 3 GHz Pentium IV machine with 2 GB of main memory and Linux 2.6.8. Computer programs were implemented in C++ and compiled using the GNU gcc compiler, version 3.3.5.

Running times depend largely on the size of the dataset analyzed, in particular, on the number of genes. For the DLBCL runs reported in the paper, the running time per run is approximately 5 minutes and 6 seconds. On the BC dataset, each single run of the crossvalidation analysis requires approximately 6 minutes and 11 seconds, the full computation over the 125 runs thus requires about 13 hours.