ddhazard

Benjamin Christoffersen

Intro

The `ddhazard` function estimates a binary regression model where the parameters are assumed to follow a pre-defined random walk. The function is implemented such that:

1) The time complexity of the computation is linear in the number of observations
2) The dimension of the observation equation can vary through time
3) It is fast due to the `C++` implementation and use of multithreading

We will breifly introduce the in model in the following paragraphs. Let $\boldsymbol{x}_{it}$ denote the co-variate vector for indvidual i at time t and let Y_{it} be the random variable for whether the i'th individual dies at time t. For given parameters at time t denoted by $\boldsymbol{\alpha}_t$ the probablity of death is:

$$\begin{aligned}\boldsymbol{y}_{it} = (y_{i1}, \dots, y_{it})^T, \quad \mathbf{X}_t = \left(\boldsymbol{x}_{i1}^T, \dots, \boldsymbol{x}_{it}^T\right)^T \\ P\left(Y_{it} = 1 | \, \boldsymbol{y}_{i,t-1}, \mathbf{X}_t, \boldsymbol{r}_{it}, \boldsymbol{\alpha}_t\right) = h(\boldsymbol{\alpha}_t^T \boldsymbol{x}_{it})\end{aligned}$$

where h is the link function. For example, this could be the logistic function such that $\exp(\eta)/(1 + \exp(\eta))$.

The models estimated in the `ddhazard` function are in the state space form:

$$\begin{array}{ll} \boldsymbol{y}_t = \boldsymbol{z}_t(\boldsymbol{\alpha}_t) + \boldsymbol{\epsilon}_t & \boldsymbol{\epsilon}_t \sim (\mathbf{0}, \mathbf{H}_t(\boldsymbol{\alpha}_t)) \\ \boldsymbol{\alpha}_{t+1} = \mathbf{F}\boldsymbol{\alpha}_t + \mathbf{R}\boldsymbol{\eta}_t & \boldsymbol{\eta}_t \sim N(\mathbf{0}, \mathbf{Q}) \end{array}, \qquad t = 1, \dots, n$$

$\boldsymbol{y}_t$ is the binary outcome and the associated equation is the *observational equation*. $\sim (a, b)$ denotes a random variable(s) with mean (vector) a and variance (co-variance matrix) b. It needs not be a normal distribution. $\boldsymbol{\alpha}_t$ is the state vector with the coresponding *state equation*

The mean $\boldsymbol{z}_t(\boldsymbol{\alpha}_t)$ and variance $\mathbf{H}(\boldsymbol{\alpha}_t)$ are state dependent with:

$$\begin{aligned} z_{it}(\boldsymbol{\alpha}_t) &= E\left(Y_{it} | \, \boldsymbol{\alpha}_t\right) = h(\boldsymbol{\alpha}_t^T \boldsymbol{x}_{it}) \\ H_{ijt}(\boldsymbol{\alpha}_t) &= \begin{cases} \text{Var}\left(Y_{it} | \, \boldsymbol{\alpha}_t\right) & i = j \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

The state equation is implemented with a 1. and 2. order random walk. For the first order random walk $\mathbf{F} = \mathbf{R} = \mathbf{I}_m$ where m is the dimension of the state vector and $\mathbf{I}_m$ is the identity matrix with dimension m. As for the second order random walk, we have:

$$\mathbf{F} = \begin{pmatrix} 2\mathbf{I}_m & -\mathbf{I}_m \\ \mathbf{0}_m & \mathbf{I}_m \end{pmatrix}, \qquad \mathbf{R} = \begin{pmatrix} \mathbf{I}_m \\ \mathbf{0}_m \end{pmatrix}$$

where $\mathbf{0}_m$ is a $m \times m$ matrix with zeros in all entries. The models are estimated by an Extended Kalman filter (EKF) or an Uncented Kalman filter (UKF). The method is chosen by passing a list to the `control` argument of `ddhazard` (the estimation function) with `list(method = "EKF", ...)` or `list(method = "UKF", ...)` respectivly

Either methods require that a vector $\boldsymbol{\alpha}_0$, co-variance matrix $\mathbf{Q}$ and co-variance matrix $\mathbf{Q}_0$ to start the filters. They can be estimated with an EM-algorithm as suggested in (Fahrmeir 1994). However, $\mathbf{Q}_0$ cannot be estimated and will tend towards zero. Hence, the default is not to estiamte the co-variance matrix $\mathbf{Q}_0$ and

only the state vector $\boldsymbol{\alpha}_0$. You can estimate $\mathbf{Q}_0$ by setting the `est_Q_0` element of the `control` to `TRUE` (`list(est_Q_0 = T, ...)`)

The rest of this note is structed as follows. The section 'Example of usage' will show how to quickly fit a model. This is followed by the section 'EM alogorithm' where the EM algorithm is explained. The sections 'Extended Kalman Filter' and 'Uncented Kalman Filter' respectivly covers the EKF and UKF used in the E-step of the EM algorithm. Finally, we end with the sections 'Logistic model' and 'Exponential model' which cover the two implemented link functions

Example of usage

EM algorithm

An EM algorithm is used to estiamte the initial state space vector $\boldsymbol{\alpha}_0$ and the co-variance matrix $\mathbf{Q}$. Optionally $\mathbf{Q}_0$ is also estimated if `control = list(est_Q_0 = T, ...)`. We will need a short hand for the conditional means and co-variances to ease the notation. Define

$$\boldsymbol{a}_{t|k} = E\left(\boldsymbol{\alpha}_t | \mathbf{Y}_k\right), \qquad \mathbf{V}_{t|T} = E\left(\mathbf{V}_t | \mathbf{Y}_k\right), \mathbf{Y}_k = (\boldsymbol{y}_{1k}, \dots, \boldsymbol{y}_{sk})$$

for the conditional mean and co-variance matrix where s is the number of observations. Notice that the letter 'a' is used for estimates while alpha is used for the unkown state. Further, these can both be filter estimates in the case where $k \leq t$ smoothed estimates when $k > t$. We supress the dependence of the co-variates here to simplfy the notation

The initial values for $\boldsymbol{\alpha}_0$, $\mathbf{Q}$ and $\mathbf{Q}_0$ can be set by passing vector to `a_0` argument of `ddhazard` and matricies to `Q_0` and `Q` argument of `ddhazard` for respectivly the $\mathbf{Q}_0$ and $\mathbf{Q}$

E-step

The outcome of the E-step is smoothed estimates:

$$\boldsymbol{a}_{i|T}^{(k)}, \quad \mathbf{V}_{i|d}^{(k)}, \quad \mathbf{B}_j^{(k)} = \mathbf{V}_{t-1|t-1}\mathbf{F}\mathbf{V}_{t|t-1}^{-1}, \quad i = 0, 1, \dots, T \wedge j = 1, 2, \dots, T$$

where T is the end of the last period we observe and supercripts $\cdot^{(k)}$ is used to destinguish the estimates from each iteration of the EM-algorithm. The required input to start the E-step is an initial mean vector $\widehat{\boldsymbol{a}}_0^{(k-1)}$ and co-variance matrices $\widehat{\mathbf{Q}}_0^{(k-1)}$ and $\widehat{\mathbf{Q}}_0^{(k-1)}$. Given these input, we compute the folowing estimates either using the EKF or UKF:

$$\boldsymbol{a}_{j|j-1}, \quad \boldsymbol{a}_{i|i}, \quad \mathbf{V}_{j|j-1}, \quad \mathbf{V}_{i|i}, \quad i = 0, 1, \dots, T \wedge j = 1, 2, \dots, T$$

Then the estimates are smoothed by computing:

$$\begin{aligned}
&\mathbf{B}_t^{(k)} = \mathbf{V}_{t-1|t-1}\mathbf{F}\mathbf{V}_{t|t-1}^{-1} \\
&\boldsymbol{a}_{t-1|d}^{(k)} = \boldsymbol{a}_{t-1|t-1} + \mathbf{B}_t(\boldsymbol{a}_{t|d}^{(k)} - \boldsymbol{a}_{t|t-1}) && t = T, T-1, \dots, 1 \\
&\mathbf{V}_{t-1|d}^{(k)} = \mathbf{V}_{t-1|t-1} + \mathbf{B}_t(\mathbf{V}_{t|d}^{(k)} - \mathbf{V}_{t|t-1})\mathbf{B}_t^T
\end{aligned}$$

M-step

The M-step is updates the mean $\widehat{\boldsymbol{a}}_0^{(k)}$ and co-variance matrices $\widehat{\mathbf{Q}}_0^{(k-1)}$ and $\widehat{\mathbf{Q}}_0^{(k-1)}$ (the latter being optional). These are computed by:

$$\widehat{\boldsymbol{\alpha}}_0^{(k)} = \boldsymbol{a}_{0|d}^{(k)}, \qquad \widehat{\mathbf{Q}}_0^{(k)} = \mathbf{V}_{0|d}^{(k)}$$

$$\widehat{\mathbf{Q}}^{(k)} = \frac{1}{d}\sum_{t=1}^{d} \mathbf{R}\left(\left(\boldsymbol{a}_{t|d}^{(k)} - \mathbf{F}\boldsymbol{a}_{t-1|d}^{(k)}\right)\left(\boldsymbol{a}_{t|d}^{(k)} - \mathbf{F}\boldsymbol{a}_{t-1|d}^{(k)}\right)^T + \mathbf{V}_{t|d}^{(k)} - \mathbf{F}\mathbf{B}_t^{(k)}\mathbf{V}_{t|d}^{(k)} - \left(\mathbf{F}\mathbf{B}_t^{(k)}\mathbf{V}_{t|d}^{(k)}\right)^T + \mathbf{F}\mathbf{V}_{t-1|d}^{(k)}\mathbf{F}^T\right)\mathbf{R}^T$$

Kalman Filter

The standard Kalman filter is carried out by recursively doing two steps. This also aplies for the EKF and UKF. Thus, this paragraph is included to introduce general notions. The first step is in the Kalman Filter is the *filter step* where we estimate $\boldsymbol{a}_{t|t-1}$ and $\mathbf{V}_{t|t-1}$ based on $\boldsymbol{a}_{t-1|t-1}$ and $\mathbf{V}_{t-1|t-1}$. Secondly, we cary out the *correction step* where we estimate $\boldsymbol{a}_{t|t}$ and $\mathbf{V}_{t|t}$ based on $\boldsymbol{a}_{t|t-1}$ and $\mathbf{V}_{t|t-1}$ and the observations. We then repeat the process

Extended Kalman Filter

The idea for the Extended Kalman filter in this application is to replace the observational equation with a first order Taylor expansion. This approximated model can then be estimated with a regular Kalman Filter. The EKF in the form presented here is orginally described in (Fahrmeir 1994) and (Fahrmeir 1992)

The formulation in (Fahrmeir 1994) differs from the standard Kalman Filter by re-writting the correction step using the Woodbury matrix identity. This has two computational advantages. The first one is that the time complexity is $O(p)$ instead of $O(p^3)$ where p denotes the dimension of the obsertion equation. Secondly, we do not have store an intermediate $p \times p$ matrix

The EKF starts with filter step where we compute:

$$\begin{aligned}\boldsymbol{a}_{t|t-1} &= \mathbf{F}\boldsymbol{a}_{t-1|t-1},\\ \mathbf{V}_{t|t-1} &= \mathbf{F}\mathbf{V}_{t-1|t-1}\mathbf{F}^T + \mathbf{R}\mathbf{Q}\mathbf{R}^T\end{aligned}$$

Secondly, we perform the correction step by:

$$\begin{aligned}\boldsymbol{a}_{t|t} &= \boldsymbol{a}_{t|t-1} + \mathbf{V}_{t|t}\boldsymbol{u}_t(\boldsymbol{a}_{t|t-1})\\ \mathbf{V}_{t|t} &= \left(\mathbf{V}_{t|t-1}^{-1} + \mathbf{U}_t(\boldsymbol{a}_{t|t-1})\right)^{-1}\end{aligned}$$

where $\boldsymbol{u}_t(\boldsymbol{a}_{t|t-1})$ and $\mathbf{U}_t(\boldsymbol{a}_{t|t-1})$ are given by:

$$\begin{aligned}\boldsymbol{u}_t(\boldsymbol{\alpha}_t) &= \sum_{i\in R_t}\boldsymbol{u}_{it}(\boldsymbol{\alpha}_t)\\ \boldsymbol{u}_{it}(\boldsymbol{\alpha}_t) &= \boldsymbol{z}_{it}\frac{\partial h(\eta)/\partial\eta}{H_{iit}(\boldsymbol{\alpha}_t)}\left(y_{it} - h(\eta)\right)\Big|_{\eta=\boldsymbol{z}_{it}^T\boldsymbol{\alpha}_t}\\ \mathbf{U}_t(\boldsymbol{\alpha}_t) &= \sum_{i\in R_t}\mathbf{U}_{it}(\boldsymbol{\alpha}_t)\\ \mathbf{U}_{it}(\boldsymbol{\alpha}_t) &= \boldsymbol{z}_{it}\boldsymbol{z}_{it}^T\frac{\left(\partial h(\eta)/\partial\eta\right)^2}{H_{iit}(\boldsymbol{\alpha}_t)}\Big|_{\eta=\boldsymbol{z}_{it}^T\boldsymbol{\alpha}_t}\end{aligned}$$

Divergence

Intial testing showed that the EKF had issues divergence for some data set. The cause of divergence is overstepping in the correction step where we update $\boldsymbol{a}_{t|t}$. In particular, the signs of the margins of $\boldsymbol{a}_{t|t}$ tended to alter between $t-1, t, t+1$ etc. and the absolute values tended to increase. The following section describes a solution to this issue

(Fahrmeir 1992) mentions that the filter step can be viewed as single Fischer Scoring step (and hence a step in a Newton Raphson method). This motivates:

1) To take multiple steps if $\boldsymbol{a}_{t|t}$ is far from $\boldsymbol{a}_{t|t-1}$
2) Introduce a learning rate

Simulation shows that the learning rate solves the issues with divergence. Let $l > 0$ denote the learning rate and ϵ_{NR} denote the tolerance of the for the Filter step. We then set $\boldsymbol{a} = \boldsymbol{a}_{t|t-1}$ and for compute:

$$
\begin{aligned}
&\boldsymbol{a}_{t|t} = \boldsymbol{a} + l \cdot \mathbf{V}_{t|t} \boldsymbol{u}_t(\boldsymbol{a}) \\
&\mathbf{V}_{t|t} = \left(\mathbf{V}_{t|t-1}^{-1} + \mathbf{U}_t(\boldsymbol{a}) \right)^{-1} \\
&\text{if } \|\boldsymbol{a}_{t|t} - \boldsymbol{a}\| / (\|\boldsymbol{a}\| + \delta) < \epsilon_{\text{NR}} \text{ then exit} \\
&\text{else set } \boldsymbol{a} = \boldsymbol{a}_{t|t} \text{ and repeat}
\end{aligned}
$$

where δ is small like 10^{-9}. The defaults are $l = 1$ and $\epsilon_{\text{NR}} = \infty$. Selecting $l < 1$ in case of divergence seems to help. Further, while (Fahrmeir 1992) does not observe improvments with multiple repitation, we find improvments in terms of mean sqare error of the state vector by taking multiple steps (setting $\epsilon_{\text{NR}} = 10^{-2}$ or lower)

l and ϵ_{NR} are set by respecivly setting the elements `LR` and `NR_eps` of the list passed to `control` argument of `ddhazard`. By default, `LR = 1` and `NR_eps = NULL` which yields a learning rate of 1 and single Fischer scoring step. These can be altered by setting `control = list(LR = .75, NR_eps = 0.001)` for a learning rate of 0.75 and a threshold in the Fischer Scoring of 10^{-3}

Parallel BLAS or LAPACK

Summary and recommandations

Choose low $\mathbf{Q}$ to start with Choose large $\mathbf{Q}_0$

Uncented Kalman Filter

The UKF selects state vectors called *sigma point* with given *sigma weigths* such that to match the moments of observational equation. Hence, the motivation to use the UKF in place of the EKF as we avoid linerization error in the EKF and match the moments of a given order. (Julier and Uhlmann 1997) the introduce UKF that match the first two moments and up to fourth moment in certain settings. (Julier and Uhlmann 2004) further develops the UKF and extend to what is later refered to as *the Scaled Unscented Transformation*. We will cover the the Scaled Unscented Transformation in a bit and the motivation for it. Firstly, we will cover the UKF and the implemented version

Usual UKF formulation

We start by introducing a common notation used in the UKF. Let:

$$\mathbf{P}_{\boldsymbol{a}_t,\boldsymbol{b}_t} = E\left((\boldsymbol{a}_t - \overline{\boldsymbol{a}}_t)(\boldsymbol{b}_t - \overline{\boldsymbol{b}}_t)^T \middle| \mathbf{Y}_t\right)$$

$\mathbf{P}_{\cdot,\cdot}$ is usefull short hand for the expected corelation matracies and expected co-variance matrix. Further, notice that $\mathbf{P}_{\boldsymbol{\alpha}_t,\boldsymbol{\alpha}_t} = \mathbf{V}_{t|t}$. The UKF method proceeds as follows: We are given estimates $\boldsymbol{a}_{t-|t-1}$ and $\boldsymbol{a}_{t-|t-1}$. We then select $2m+1$ *sigma points* (where m is the dimension of the state equation) denoted by $\boldsymbol{z}_0, \boldsymbol{z}_1, \dots, \boldsymbol{z}_{2m+1}$ according to:

$$\begin{aligned}
\widehat{\boldsymbol{a}}_0 &= \mathbf{F}\boldsymbol{a}_{t-1|t-1} \\
\widehat{\boldsymbol{a}}_i &= \mathbf{F}\left(\boldsymbol{a}_{t-1|t-1} + \sqrt{m+\lambda}\left(\sqrt{\mathbf{V}_{t-1|t-1}}\right)_i\right) \qquad i = 1, 2, \dots, n \\
\widehat{\boldsymbol{a}}_{i+m} &= \mathbf{F}\left(\boldsymbol{a}_{t-1|t-1} - \sqrt{m+\lambda}\left(\sqrt{\mathbf{V}_{t-1|t-1}}\right)_i\right)
\end{aligned}$$

where $\left(\sqrt{\mathbf{V}_{t-1|t-1}}\right)_i$ is the i'th column of the lower triangular matrix of the Cholesky decomposition of $\mathbf{V}_{t-1|t-1}$. We assign the following weights to each sigma point (we will cover selection of the scalaras α, β and κ shortly):

$$\begin{aligned}
W_0^{(m)} &= \frac{\lambda}{m+\lambda} \\
W_0^{(c)} &= \frac{\lambda}{m+\lambda} + 1 - \alpha^2 + \beta \\
W_i^{(m)} &= W_0^{(c)} = \frac{1}{2(m+\lambda)}, \qquad i = 1, \dots, 2m \\
\lambda &= \alpha^2(m+\kappa) - m
\end{aligned}$$

Let $\boldsymbol{W}^{(j)} = (W_0^{(j)}, \dots, W_{2m}^{(j)})^T$ and $\widehat{\mathbf{A}} = (\widehat{\boldsymbol{a}}_0, \dots, \widehat{\boldsymbol{a}}_{2m})$ The filter step given the sigma points and sigma weights is:

$$\begin{aligned}
\boldsymbol{a}_{t|t-1} &= \sum_{i=0}^{2m} W_i^{(m)} \widehat{\boldsymbol{a}}_i \\
\Delta\widehat{\mathbf{A}} &= \widehat{\mathbf{A}} - \boldsymbol{a}_{t|t-1}\mathbf{1}^T \\
\mathbf{V}_{t|t-1} &= \mathbf{R}\mathbf{Q}\mathbf{R}^T + \sum_{i=0}^{2m} W_i^{(c)}(\mathbf{F}\widehat{\boldsymbol{a}}_i - \boldsymbol{a}_{t|t-1})(\widehat{\boldsymbol{a}}_i - \boldsymbol{a}_{t|t-1})^T \\
&= \mathbf{R}\mathbf{Q}\mathbf{R}^T + \Delta\widehat{\mathbf{A}}\text{diag}(\boldsymbol{W}^{(c)})\Delta\widehat{\mathbf{A}}^T
\end{aligned}$$

where $\text{diag}(\cdot)$ returns a diagonal matrix with the passed vectors values in the diagonal and $\mathbf{1}$ is a vector with one in all the entries. We then proceed to the correction step. We start by defining the following intermediates:

$$
\begin{aligned}
\widehat{\boldsymbol{y}}_i &= \boldsymbol{z}_t\left(\widehat{\boldsymbol{a}}_i\right), \qquad i = 0, 1, \dots, 2m \\
\widehat{\mathbf{Y}} &= (\widehat{\boldsymbol{y}}_0, \dots, \widehat{\boldsymbol{y}}_{2m}) \\
\overline{\boldsymbol{y}} &= \sum_{i=0}^{2m} W_i^{(m)} \boldsymbol{y}_i, \qquad \Delta\widehat{\mathbf{Y}} = \widehat{\mathbf{Y}} - \overline{\boldsymbol{y}}\mathbf{1}^T, \qquad \widehat{\mathbf{H}} = \sum_{i=0}^{m} W_i^{(c)} \mathbf{H}_t(\widehat{\mathbf{a}}_i) \\
\mathbf{P}_{\boldsymbol{y}_t,\boldsymbol{y}_t} &= \sum_{i=0}^{m} W_i^{(c)} \left((\widehat{\boldsymbol{y}}_i - \overline{\boldsymbol{y}})(\widehat{\boldsymbol{y}}_i - \overline{\boldsymbol{y}})^T + \widehat{\mathbf{H}}\right) \\
&= \Delta\widehat{\mathbf{Y}} \mathrm{diag}(\boldsymbol{W}^{(c)}) \Delta\widehat{\mathbf{Y}}^T + \widehat{\mathbf{H}} \\
\mathbf{P}_{\boldsymbol{x}_t,\boldsymbol{y}_t} &= \sum_{i=0}^{m} W_i^{(c)} (\widehat{\boldsymbol{a}}_i - \boldsymbol{a}_{t|t-1})(\widehat{\boldsymbol{y}}_i - \overline{\boldsymbol{y}})^T \\
&= \Delta\widehat{\mathbf{A}} \mathrm{diag}(\boldsymbol{W}^{(c)}) \Delta\widehat{\mathbf{Y}}^T
\end{aligned}
$$

The final correction step is then:

$$
\begin{aligned}
\boldsymbol{a}_{t|t} &= \boldsymbol{a}_{t|t-1} + \mathbf{P}_{\boldsymbol{a},\boldsymbol{y}} \mathbf{P}_{\boldsymbol{y},\boldsymbol{y}}^{-1} (\boldsymbol{y}_t - \overline{\boldsymbol{y}}) \\
\mathbf{V}_{t|t} &= \mathbf{V}_{t|t} - \mathbf{P}_{\boldsymbol{a},\boldsymbol{y}} \mathbf{P}_{\boldsymbol{y},\boldsymbol{y}}^{-1} \mathbf{P}_{\boldsymbol{a},\boldsymbol{y}}^T
\end{aligned}
$$

Re-writting

The above formulation have the draw back that we have to invert $\mathbf{P}_{\boldsymbol{y},\boldsymbol{y}}^{-1}$ which requires that we store matrix with dimension equal to the number of observation in given interval and invert it. The later is an issue when the number of observation is large (say 10,000) while the case is an issue when the number of observation is even moderatly large (say greater than 200). We can though re-write the above using the Woodbury matrix identity to get algorithm $O(s_i)$ instead of $O(s_i^3)$ where s_i is the number of observations in the i'th interval

The proposed correction step can be computed as:

$$
\begin{aligned}
\tilde{\boldsymbol{y}} &= \Delta\widehat{\mathbf{Y}}^T \widehat{\mathbf{H}}^{-1} (\boldsymbol{y}_t - \overline{\boldsymbol{y}}) \\
\mathbf{G} &= \Delta\widehat{\mathbf{Y}}^T \widehat{\mathbf{H}}^{-1} \Delta\widehat{\mathbf{Y}} \\
\boldsymbol{c} &= \tilde{\boldsymbol{y}} - \mathbf{G} \left(\mathrm{diag}(\boldsymbol{W}^{(c)})^{-1} + \mathbf{G}\right)^{-1} \tilde{\boldsymbol{y}} \\
\mathbf{L} &= \mathbf{G} - \mathbf{G} \left(\mathrm{diag}(\boldsymbol{W}^{(c)})^{-1} + \mathbf{G}\right)^{-1} \mathbf{G} \\
\boldsymbol{a}_{t|t} &= \boldsymbol{a}_{t|t-1} + \Delta\widehat{\mathbf{X}} \mathrm{diag}(\boldsymbol{W}^{(c)}) \boldsymbol{c} \\
\mathbf{V}_{t|t} &= \mathbf{V}_{t|t-1} - \Delta\widehat{\mathbf{X}} \mathrm{diag}(\boldsymbol{W}^{(c)}) \mathbf{L} \mathrm{diag}(\boldsymbol{W}^{(c)}) \Delta\widehat{\mathbf{X}}^T
\end{aligned}
$$

where $\tilde{\boldsymbol{y}}$, $\mathbf{G}$, $\mathbf{L}$ and $\boldsymbol{c}$ are intermediates. The above algorithm is $O(s_i)$ since $\widehat{\mathbf{H}}$ is a diagonal matrix and all products involves at worst multiplication $m \times s_i$ and $s_i \times m$ matricies

Selecting hyperparameters

We still need to select the hyperparameters κ, α and β. We will cover these in the given order. κ is usually set to $\kappa = 0$ or $\kappa = 3 - m$ which (Julier and Uhlmann 1997) state is a "* useful heuristic*" when the state equation is Gaussian and $\alpha = 1$. The default is 0 and can be altered by setting the list element `kappa` passed as the `control` argument to `ddhazard`. For example, `control = list(kappa = 1, ...)` yields $\kappa = 1$.

$0 < \alpha \leq 1$ controls the spread of the sigma points. As an example, we can notice that $\lambda + m \to 0^+$, $w_0^{(c)}, w_0^{(m)} \to -\infty$ and $w_i^{(c)}, w_i^{(m)} \to \infty$ $(i > 0)$ as $\alpha \to 0^+$ with $\kappa = 0$. Hence, the lower the value, the lower

the spread but the higher the absolute weights. It is genrally suggsted to choose α small. The arguments hereof are provided in for example (Gustafsson and Hendeby 2012) and (Julier and Uhlmann 2004). The algorithm tend to have issues with divergence with $\alpha < 1$. In particular divergence seems to be linked to the choice of initial co-variance matrix $\mathbf{Q}$ and co-variance matrix $\mathbf{Q}_0$. For this reason, the default is $\alpha = 1$. The parameter can be altered through the `alpha` element of the list passed to the argument `control` of `ddhazard`.

Lastly, β is a correction term to match the fourth-order term in the Taylor series expansion of the covariance of the observational equation. (Julier and Uhlmann 2004) show in the appendix that the optimal value with a Gaussian state equation is $\beta = 2$. This is the default. It can be altered through the `beta` element of list passed to the argument `control` of `ddhazard`.

Selecting starting values

Exeperince with various data set and the UKF method have shown that the method is sensitive to the starting values of $\mathbf{Q}$ and $\mathbf{Q}_0$ (where the latter is may be fixed). The reason can be illustrated by the effect of $\mathbf{Q}_0$. We start the filter by setting $\mathbf{V}_{0|0} = \mathbf{Q}_0$. Say that we set $\mathbf{Q}_0 = \kappa \mathbf{I}_m$ and $\boldsymbol{a}_0 = \mathbf{0}$. Then the i'th column of the Cholesky decompositions $\sqrt{\mathbf{V}_{0|0}}$ is a vector with $\sqrt{\kappa}$ in the i'th entry and zero in the rest of the entries. Suppose that we set κ large. Then the linear predictors computed with the $k < m + 1$ sigma point is κx_{kjt} where x_{kj1} is the k'th entry of indviduals j's co-variate vector at time 1. This can be potentially quite large in aboslute terms if x_{kjt} is moderatly different from zero. This seems to lead to divergence in some cases for instance in the logistic model where we end with either zero or 1 estimates for the outcome

$\mathbf{Q}$ has a similar effect although it is harder to illustrate with a small example as it occours as an intermediate in the UKF. Question is then how to select $\mathbf{Q}$ and $\mathbf{Q}_0$. At this point, I can suggest to pick at diagonal matrix for plausible somewhat large values $\mathbf{Q}_0$ and $\mathbf{Q}_0$ to a diagonal matrix with small values. This is based on experince with various data sets. Though, what is plausible and what is small dependent on the data set

Summary and recommandations

Logistic model

The logistic model is fitted with by setting `model = "logit"` in the call to `ddhazard`. The link funtion h is defined as inverse logit function $h(\eta) = \exp(\eta)/(1 + \exp(\eta))$. The following paragraphs will cover the EKF and UKF notes. This is followed by some quick comment about the loss of information due to binning

EKF

From a numerical point of view, each individual computation are fairly stable with EKF method fot the logitistic model. The main reason is that:

$$\partial h(\eta)/\partial \eta|_{\eta = \boldsymbol{z}_{it}^T \boldsymbol{\alpha}_t} = H_{iit}(\boldsymbol{\alpha}_t)$$

Given the variance $\mathbf{H}_t(\boldsymbol{\alpha}_t)$ and expected mean $h(\eta)$ are bound this means that all the terms are on a resonably stable for all values of the linear predictor $\boldsymbol{\alpha}_t^T \boldsymbol{x}_{it}$

UKF

This is not the case for the UKF method and the logistic model. We scale by $\widehat{\mathbf{H}}_t^{-1}$ when computing $\mathbf{G}$ and $\tilde{\boldsymbol{y}}$ which will approach zero as linear predictor get large in abosolute value. For this reason, we truncate the

linear predictor $\eta_{it} = \boldsymbol{\alpha}_t^T \boldsymbol{x}_{it}$ the variance cannot be less than some pre-specified quantity δ. Effecitvly this means that we set:

$$\begin{aligned} & h(\eta)(1 - h(\eta) \geq \delta, \qquad \delta \in (0, 1/4) \\ \Leftrightarrow \quad & \log\left(\frac{1 - 2\delta - \sqrt{(1 - 4\delta)}}{2\delta}\right) \leq \eta \leq \log\left(\frac{1 - 2\delta + \sqrt{(1 - 4\delta)}}{2\delta}\right) \end{aligned}$$

In terms this implies that:

$$\frac{1 - 2\delta - \sqrt{1 - 4\delta}}{1 - \sqrt{1 - 4\delta}} \leq h(\eta) \leq \frac{1 - 2\delta + \sqrt{1 - 4\delta}}{1 + \sqrt{1 - 4\delta}}$$

The current implementation set $\delta = 10^{-4}$

Binning

This section will illustrate how binning is performed for the logistic model and how this can lead to loss of information. It is elementery but included to stress this point and motivate the exponential model. We will use figure **??** as the illustration. Each horizontal line represent an individual. A cross represents when the co-variate values change for the invidual and a circle represents the death of an invidual. Lines that ends with a cross are right censored

The vertical dashed lines represents the bin borders. The first vertical from the left is where we start our model, the second vertical line is where the first bin ends and the second start and the third vertical line is where the second bin ends. Thus, only have two bins in this example

We can now cover how the inviduals (horisontal lines) are used in the estimation:

a. Is a control in both bins. We use the co-variates from 0 in the first bin and the co-variates from 1 in the second bin
b. Is not included in any of the bins. We do not know the co-variates values at the start of the second bin so we cannot include him
c. Is a control in the first bin with the co-variates from 0. He will count as a death in the second bin with the co-variates from 1
d. Acts like a.
e. Is a death in the first bin with co-variates from 0
f. Is a control in the first bin with the co-variates from 0. He is a death in the second bin with the co-variates from 1
g. Is not included in any bins. We do not know if he survived the entire period of the first bin and thus we cannot include him

The example illustrates that:

1. We loose information about co-variates that are updated within bins. For instance, a., c., d. and f. all use the co-variates from 0 for the entire period of the first bin despite that the co-variates change at 1. Moreover, we never use the information at 2 from a., d. and f.
2. We loose information when we have right censoring. For instance, g. is not included at all since we only know that survives parts of the first bin
3. We loose information for observation that only occours within bins as is the case for b.

The above motivates the exponential model that will be covered in the next sections

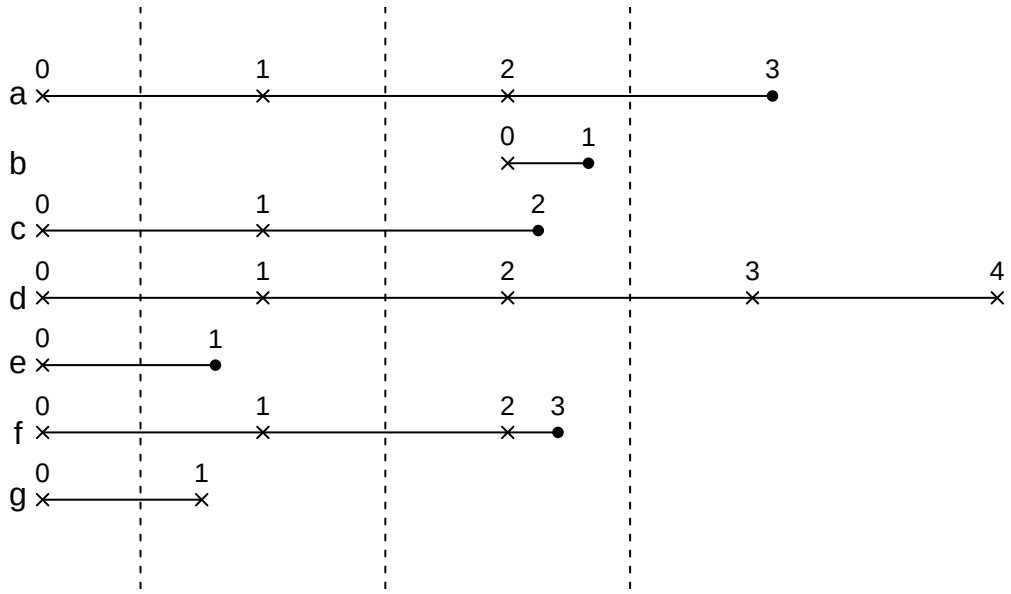

Figure 1: Illustration of binning. See text for explanation

Exponential model

We make the following assumption in the exponential model:

1. Parameters (i.e. state variables) change at time $1, 2, \dots, T$
2. The invidiuals co-variates change at discrete times
3. We have exponential distitubeted arrival times within bins. This means that we piecewise exponential constant destributed arival times where the instantaneous hazard change when either the invidual co-variates change or the parameters change

In terms, these assumption will resolve the issue we face with the binning for the logistic model. We make the following definition to formalize the above. Let $\boldsymbol{x}_{ij}$ denote the i'th individuals k'th co-variate vector. For each invidiual we observe $j = 1, 2, \dots, l_i$ co-variate vectors. Each co-variate vector is valid in a period $(t_{i,j-1}; t_{i,j}]$. Let T_i denote the random death time of the i'th invidual. Lastly, let $y_{ij} = 1_{\{T_i \in (t_{i,j-1}, t_{i,j}]\}}$ be the indicator for whether the i'th invidual dies in period $(t_{i,j-1}, t_{i,j}]$

The likelihood of observing what we do condtional on the state space variables are:

$$\begin{aligned} P\left(Y_{il_i} = 1 \middle| \boldsymbol{\alpha}_0, \boldsymbol{\alpha}_1, \dots\right) =& P\left(Y_{il_i} = 1 \middle| Y_{i,l_i-1} = 0 \wedge \boldsymbol{\alpha}_0, \boldsymbol{\alpha}_1, \dots\right) \\ & \cdot P\left(Y_{i,l_i-1} = 0 \middle| Y_{i,l_i-2} = 0 \wedge \boldsymbol{\alpha}_0, \boldsymbol{\alpha}_1, \dots\right) \\ & \vdots \\ & \cdot P\left(Y_{i,2} = 0 \middle| Y_{i,1} = 0 \wedge \boldsymbol{\alpha}_0, \boldsymbol{\alpha}_1, \dots\right) P\left(Y_{i,1} = 0 \middle| \boldsymbol{\alpha}_0, \boldsymbol{\alpha}_1, \dots\right) \end{aligned}$$

We can now use the memory less property of the exponetial distribution to conclude that each of the term above have:

$$P\left(Y_{i,s} = 1 \middle| Y_{i,s-1} = 0 \wedge \boldsymbol{\alpha}_0, \boldsymbol{\alpha}_1, \dots\right) = 1 - \prod_{z=\lfloor t_{i,s-1} \rfloor + 1}^{\lceil t_{i,s} \rceil} \exp\left(-\exp(\boldsymbol{x}_{is}^T \boldsymbol{\alpha}_z)\left(\min\{z, t_{i,s}\} - \max\{z-1, t_{i,s-1}\}\right)\right)$$

where $\lfloor \cdot \rfloor$ is the floor function and $\lceil \cdot \rceil$ is the ceiling function. The above we assume that $t_{i,s} - t_{i,s-1} \geq 1$ to simplify the product (if not we would have more factors in the product where the co-variate vectors change within a bin). This is for notional simplicifty

In order to get this into the state space model notation we further have to seperate $Y_{i,s}$ if ti crosses its time period crosses bins. That is, when the period $(t_{i,s-1}, t_{i,s}]$ cross a bin. Take for example $(0.5, 1.5])$. Here we add two binary observation: one with time period of $(0.5, 1]$ and another with $(1, 1.5]$. Notice that this also implies that an invidual who has different co-variate vectors in period $(0, 0.5]$ and $(0.5, 1]$ will yield to observation to the observation equation in the first interval

Computing the condtional mean the link can be done as follows h. Assume for simplicity of notation that the observation $Y_{i,s'}$ is inside an interval $(\lceil t_{i,s} \rceil - 1 \geq \lfloor t_{i,s-1} \rfloor)$ (this could be an observation we have introduced becuase the inital interval crossed a bin). The the link function in this case is:

Right censoring is not an issue in this setup if we assume independent censoring. In that case the *min* condition in $P\left(Y_{i,s} = 1 \middle| Y_{i,s-1} = 0 \wedge \boldsymbol{\alpha}_0, \boldsymbol{\alpha}_1, \dots\right)$ is valid

EKF

UKF

Implementation details

Computational issues

References

Fahrmeir, Ludwig. 1992. "Posterior Mode Estimation by Extended Kalman Filtering for Multivariate Dynamic Generalized Linear Models." *Journal of the American Statistical Association* 87 (418). Taylor & Francis Group: 501–9.

———. 1994. "Dynamic Modelling and Penalized Likelihood Estimation for Discrete Time Survival Data." *Biometrika* 81 (2). Biometrika Trust: 317–30.

Gustafsson, Fredrik, and Gustaf Hendeby. 2012. "Some Relations Between Extended and Unscented Kalman Filters." *IEEE Transactions on Signal Processing* 60 (2). IEEE: 545–55.

Julier, Simon J, and Jeffrey K Uhlmann. 1997. "New Extension of the Kalman Filter to Nonlinear Systems." In *AeroSense'97*, 182–93. International Society for Optics; Photonics.

———. 2004. "Unscented Filtering and Nonlinear Estimation." *Proceedings of the IEEE* 92 (3). IEEE: 401–22.