A Cox model for component lifetimes with spatial interactions

F. Corset[1] M. Foularidad[2] C. Paroissin[3]
M. Anderhalt[4] E. Remy[4]

[1]Laboratoire Jean Kuntzmann - Université Grenoble Alpes

[2]UTT - Institut Charles delaunay

[3]Université de Pau et des Pays de l'Adour - LMAP

[4]EDF R & D

Fourth Project Workshop AMMSI, January 20-21, 2016,

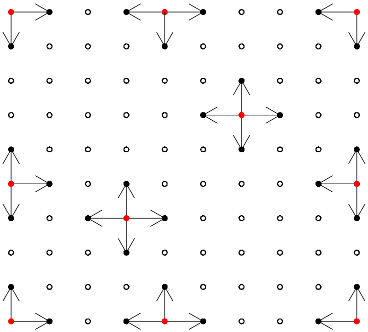
Context

- n components displayed on a structure, e.g. a set of crews on a steel plates.
- Components are located on a regular $\sqrt{n} \times \sqrt{n}$ grid.
- The failure of a component may lead to a higher stress on the components in its neighborhood.
- State of components are observed at some inspection times.

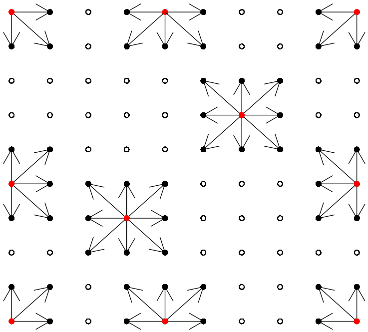
Model and Notation

- T_i: the lifetime of component i.
- n binary stochastic processes: $\forall i \in \{1, \ldots, n\}, \forall t \geq 0$, $Y_i(t) = \mathbf{1}_{T_i \leq t}$
- A failure of a component leads to an increase of the hazard function of this component.
- Let B_i denote the neighborhood of the i-th component.

Illustrations of neighborhood (4 neighbors)



Illustrations of neighborhood (8 neighbors)



Lifetime distribution

- Hazard function conditionnally to the past (the state of n components).
- Let $\mathcal{H}_{t^-}$ be the σ-field generated by the n binary stochatic processes defined above up to time t^-
- The hazard function of the i-th component, denoted λ_i:

$$\forall t \geq 0, \quad \lambda_i(t|\mathcal{H}_{t^-}) = \lambda(t) \exp\left(\alpha \sum_{j \in B_i} C_j(t) Y_j(t) + \beta' Z_i(t)\right),$$

where $\lambda(\cdot)$ is the baseline hazard function, $\alpha \in \mathbb{R}$, $\beta \in \mathbb{R}^p$ and where, for the i-th component, $C_i(\cdot)$ is a time-dependent covariate and $Z_i(\cdot)$ is a vector of p time-dependent covariates (temperature, constraints, etc.).

Survival function

The survival function of the component i is given by:

$$\forall t \geq 0, \quad S_i(t|\mathcal{H}_{t^-}) = \exp\left(-\int_0^t \lambda_i(u|\mathcal{H}_{u^-})\mathrm{d}u\right).$$

Assume λ is the hazard function of the Weibull distribution with scale parameter $a > 0$ and shape parameter $b > 0$:

$$\forall t \geq 0, \quad \lambda(t) = \frac{b}{a}\left(\frac{t}{a}\right)^{b-1}.$$

Survival function

The survival function of the component i is given by:

$$\forall t \geq 0, \quad S_i(t|\mathcal{H}_{t^-}) = \exp\left(-\int_0^t \lambda_i(u|\mathcal{H}_{u^-})\mathrm{d}u\right).$$

Assume λ is the hazard function of the Weibull distribution with scale parameter $a > 0$ and shape parameter $b > 0$:

$$\forall t \geq 0, \quad \lambda(t) = \frac{b}{a}\left(\frac{t}{a}\right)^{b-1}.$$

Simulation in independent case (without covariates)

- Weibull parameters: $a = 2$ and $b = 2$.
- square grid of 50×50 components.
- failed component are displayed in red.
- $\alpha = 0$ (no interaction)
- $\beta = 0$ (no covariates)

Plot of failed components (Independent case)

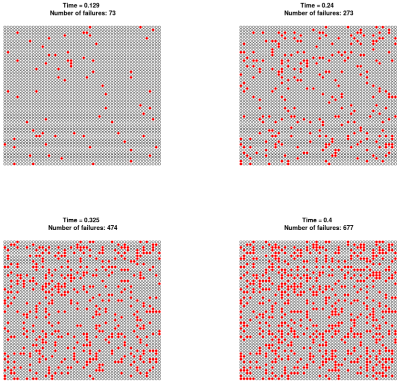


Figure: No interaction ($\alpha = 0$)

Simulation in dependent case

- Weibull parameters: $a = 2$ and $b = 2$.
- square grid of 50×50 components.
- failed component are displayed in red.
- $\alpha = 2$
- $\beta = 0$ (no covariates)

Plot of failed components (Dependent case)

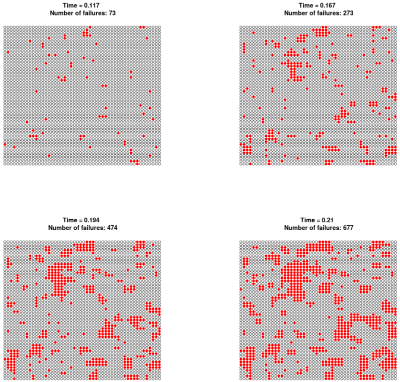


Figure: with interaction ($\alpha > 0$)

Simulation Algorithm

Require: $n, a, b, \alpha, \beta, Z_1(\cdot), \dots, Z_n(\cdot)$

$T \leftarrow \text{Vector}(\text{length} = n)$
$\tilde{T} \leftarrow \text{Vector}(\text{length} = n)$
draw $U_1, \dots, U_n$ i.i.d. from the uniform distribution over $[0; 1]$
{*Step* 1}
set $R \leftarrow \{1, \dots, n\}$ {set of unfailed components}
draw $\tilde{T}_1, \dots, \tilde{T}_n$: $\log(U_i) + \int_0^{t_i} \lambda(s) \exp(\beta' Z_i(s))\, ds = 0$
$i_1 \leftarrow \text{argmin}_{j \in R} \tilde{T}_j$ {select the next comp. to fail}
$T_{i_1} \leftarrow \tilde{T}_{i_1}$ {store the next failure times}
for $r \in \{2, \dots, n\}$ **do**
    {Step r}
    $R \leftarrow R \setminus \{i_{r-1}\}$ {update the set of non-failed components}
    for $j \in B_{i_{r-1}} \cap R$ **do**
        draw $\tilde{T}_j$: $\log(U_j) + \int_{T_{i_{r-1}}}^{t_j} \lambda(s) \exp\left(\beta Z(s) + \alpha \sum_{k \in B_j} C_k(s) Y_k(s)\right)\, ds = 0$
    end for
    $i_r \leftarrow \text{argmin}_{j \in R} \tilde{T}_j$ {select the next comp. to fail}
    $T_{i_r} \leftarrow \tilde{T}_{i_r}$ {store the next failure times}
end for
return $T_1, \dots, T_n$

Simulation Algorithm

Require: $n, a, b, \alpha, \beta, Z_1(\cdot), \ldots, Z_n(\cdot)$

$T \leftarrow$ Vector(length $= n$)
$\tilde{T} \leftarrow$ Vector(length $= n$)
draw $U_1, \ldots, U_n$ i.i.d. from the uniform distribution over $[0; 1]$
$\{Step\ 1\}$
set $R \leftarrow \{1, \ldots, n\}$ {set of unfailed components}
draw $\tilde{T}_1, \ldots, \tilde{T}_n$: $\log(U_i) + \int_0^{t_i} \lambda(s) \exp(\beta' Z_i(s))\, ds = 0$
$i_1 \leftarrow \mathrm{argmin}_{j \in R} \tilde{T}_j$ {select the next comp. to fail}
$T_{i_1} \leftarrow \tilde{T}_{i_1}$ {store the next failure times}
for $r \in \{2, \ldots, n\}$ **do**
    {Step r}
    $R \leftarrow R \setminus \{i_{r-1}\}$ {update the set of non-failed components}
    for $j \in B_{i_{r-1}} \cap R$ **do**
        draw $\tilde{T}_j$: $\log(U_j) + \int_{T_{i_{r-1}}}^{t_j} \lambda(s) \exp\left(\beta Z(s) + \alpha \sum_{k \in B_j} C_k(s) Y_k(s)\right) ds = 0$
    end for
    $i_r \leftarrow \mathrm{argmin}_{j \in R} \tilde{T}_j$ {select the next comp. to fail}
    $T_{i_r} \leftarrow \tilde{T}_{i_r}$ {store the next failure times}
end for
return $T_1, \ldots, T_n$

Simulation Algorithm

Require: n, a, b, α, β, $Z_1(\cdot), \ldots, Z_n(\cdot)$

$T \leftarrow$ Vector(length $= n$)
$\tilde{T} \leftarrow$ Vector(length $= n$)
draw $U_1, \ldots, U_n$ i.i.d. from the uniform distribution over [0; 1]
{*Step* 1}
set $R \leftarrow \{1, \ldots, n\}$ {set of unfailed components}
draw $\tilde{T}_1, \ldots, \tilde{T}_n$: $\log(U_i) + \int_0^{t_i} \lambda(s) \exp(\beta' Z_i(s))\, ds = 0$
$i_1 \leftarrow \text{argmin}_{j \in R} \tilde{T}_j$ {select the next comp. to fail}
$T_{i_1} \leftarrow \tilde{T}_{i_1}$ {store the next failure times}
for $r \in \{2, \ldots, n\}$ **do**
  {Step r}
  $R \leftarrow R \setminus \{i_{r-1}\}$ {update the set of non-failed components}
  for $j \in B_{i_{r-1}} \cap R$ **do**
    draw $\tilde{T}_j$: $\log(U_j) + \int_{T_{i_{r-1}}}^{t_j} \lambda(s) \exp\left(\beta Z(s) + \alpha \sum_{k \in B_j} C_k(s) Y_k(s)\right) ds = 0$
  end for
  $i_r \leftarrow \text{argmin}_{j \in R} \tilde{T}_j$ {select the next comp. to fail}
  $T_{i_r} \leftarrow \tilde{T}_{i_r}$ {store the next failure times}
end for
return $T_1, \ldots, T_n$

Simulation Algorithm

Require: $n, a, b, \alpha, \beta, Z_1(\cdot), \ldots, Z_n(\cdot)$
$T \leftarrow$ Vector(length $= n$)
$\tilde{T} \leftarrow$ Vector(length $= n$)
draw $U_1, \ldots, U_n$ i.i.d. from the uniform distribution over [0; 1]
{*Step* 1}
set $R \leftarrow \{1, \ldots, n\}$ {set of unfailed components}
draw $\tilde{T}_1, \ldots, \tilde{T}_n$: $\log(U_i) + \int_0^{t_i} \lambda(s) \exp(\beta' Z_i(s))\, ds = 0$
$i_1 \leftarrow \text{argmin}_{j \in R} \tilde{T}_j$ {select the next comp. to fail}
$T_{i_1} \leftarrow \tilde{T}_{i_1}$ {store the next failure times}
for $r \in \{2, \ldots, n\}$ **do**
  {Step r}
  $R \leftarrow R \setminus \{i_{r-1}\}$ {update the set of non-failed components}
  for $j \in B_{i_{r-1}} \cap R$ **do**
    draw $\tilde{T}_j$: $\log(U_j) + \int_{T_{i_{r-1}}}^{t_j} \lambda(s) \exp\left(\beta Z(s) + \alpha \sum_{k \in B_j} C_k(s) Y_k(s)\right) ds = 0$
  end for
  $i_r \leftarrow \text{argmin}_{j \in R} \tilde{T}_j$ {select the next comp. to fail}
  $T_{i_r} \leftarrow \tilde{T}_{i_r}$ {store the next failure times}
end for
return $T_1, \ldots, T_n$

Simulation Algorithm

Require: n, a, b, α, β, $Z_1(\cdot), \ldots, Z_n(\cdot)$

$T \leftarrow$ Vector(length $= n$)
$\tilde{T} \leftarrow$ Vector(length $= n$)
draw $U_1, \ldots, U_n$ i.i.d. from the uniform distribution over [0; 1]
{*Step* 1}
set $R \leftarrow \{1, \ldots, n\}$ {set of unfailed components}
draw $\tilde{T}_1, \ldots, \tilde{T}_n$: $\log(U_i) + \int_0^{t_i} \lambda(s) \exp(\beta' Z_i(s))\, ds = 0$
$i_1 \leftarrow \operatorname{argmin}_{j \in R} \tilde{T}_j$ {select the next comp. to fail}
$T_{i_1} \leftarrow \tilde{T}_{i_1}$ {store the next failure times}
for $r \in \{2, \ldots, n\}$ **do**
  {Step r}
  $R \leftarrow R \setminus \{i_{r-1}\}$ {update the set of non-failed components}
  for $j \in B_{i_{r-1}} \cap R$ **do**
    draw $\tilde{T}_j$: $\log(U_j) + \int_{T_{i_{r-1}}}^{t_j} \lambda(s) \exp\left(\beta Z(s) + \alpha \sum_{k \in B_j} C_k(s) Y_k(s)\right)\, ds = 0$
  end for
  $i_r \leftarrow \operatorname{argmin}_{j \in R} \tilde{T}_j$ {select the next comp. to fail}
  $T_{i_r} \leftarrow \tilde{T}_{i_r}$ {store the next failure times}
end for
return $T_1, \ldots, T_n$

Simulation Algorithm

Require: n, a, b, α, β, $Z_1(\cdot), \ldots, Z_n(\cdot)$

```
T ← Vector(length = n)
T̃ ← Vector(length = n)
draw U_1, ..., U_n i.i.d. from the uniform distribution over [0; 1]
{Step 1}
set R ← {1, ..., n} {set of unfailed components}
draw T̃_1, ..., T̃_n: log(U_i) + ∫_0^{t_i} λ(s) exp(β' Z_i(s)) ds = 0
i_1 ← argmin_{j∈R} T̃_j {select the next comp. to fail}
T_{i_1} ← T̃_{i_1} {store the next failure times}
for r ∈ {2, ..., n} do
    {Step r}
    R ← R \ {i_{r-1}} {update the set of non-failed components}
    for j ∈ B_{i_{r-1}} ∩ R do
        draw T̃_j: log(U_j) + ∫_{T_{i_{r-1}}}^{t_j} λ(s) exp( βZ(s) + α Σ_{k∈B_j} C_k(s) Y_k(s) ) ds = 0
    end for
    i_r ← argmin_{j∈R} T̃_j {select the next comp. to fail}
    T_{i_r} ← T̃_{i_r} {store the next failure times}
end for
return T_1, ..., T_n
```

Simulation Algorithm

Require: n, a, b, α, β, $Z_1(\cdot), \dots, Z_n(\cdot)$
$T \leftarrow$ Vector(length $= n$)
$\tilde{T} \leftarrow$ Vector(length $= n$)
draw $U_1, \dots, U_n$ i.i.d. from the uniform distribution over [0; 1]
{*Step* 1}
set $R \leftarrow \{1, \dots, n\}$ {set of unfailed components}
draw $\tilde{T}_1, \dots, \tilde{T}_n$: $\log(U_i) + \int_0^{t_i} \lambda(s) \exp(\beta' Z_i(s))\, \mathrm{d}s = 0$
$i_1 \leftarrow \operatorname{argmin}_{j \in R} \tilde{T}_j$ {select the next comp. to fail}
$T_{i_1} \leftarrow \tilde{T}_{i_1}$ {store the next failure times}
for $r \in \{2, \dots, n\}$ **do**
  {Step r}
  $R \leftarrow R \setminus \{i_{r-1}\}$ {update the set of non-failed components}
  for $j \in B_{i_{r-1}} \cap R$ **do**
    draw $\tilde{T}_j$: $\log(U_j) + \int_{T_{i_{r-1}}}^{t_j} \lambda(s) \exp\left(\beta Z(s) + \alpha \sum_{k \in B_j} C_k(s) Y_k(s)\right) \mathrm{d}s = 0$
  end for
  $i_r \leftarrow \operatorname{argmin}_{j \in R} \tilde{T}_j$ {select the next comp. to fail}
  $T_{i_r} \leftarrow \tilde{T}_{i_r}$ {store the next failure times}
end for
return $T_1, \dots, T_n$

Simulation Algorithm

Require: $n, a, b, \alpha, \beta, Z_1(\cdot), \ldots, Z_n(\cdot)$

```
T ← Vector(length = n)
T̃ ← Vector(length = n)
draw U_1, ..., U_n i.i.d. from the uniform distribution over [0; 1]
{Step 1}
set R ← {1, ..., n} {set of unfailed components}
draw T̃_1, ..., T̃_n: log(U_i) + ∫_0^{t_i} λ(s) exp(β' Z_i(s)) ds = 0
i_1 ← argmin_{j∈R} T̃_j {select the next comp. to fail}
T_{i_1} ← T̃_{i_1} {store the next failure times}
for r ∈ {2, ..., n} do
    {Step r}
    R ← R \ {i_{r-1}} {update the set of non-failed components}
    for j ∈ B_{i_{r-1}} ∩ R do
        draw T̃_j: log(U_j) + ∫_{T_{i_{r-1}}}^{t_j} λ(s) exp(βZ(s) + α Σ_{k∈B_j} C_k(s) Y_k(s)) ds = 0
    end for
    i_r ← argmin_{j∈R} T̃_j {select the next comp. to fail}
    T_{i_r} ← T̃_{i_r} {store the next failure times}
end for
return T_1, ..., T_n
```

Simulation Algorithm

Require: $n, a, b, \alpha, \beta, Z_1(\cdot), \ldots, Z_n(\cdot)$

$T \leftarrow$ Vector(length $= n$)
$\tilde{T} \leftarrow$ Vector(length $= n$)
draw $U_1, \ldots, U_n$ i.i.d. from the uniform distribution over [0; 1]
{*Step* 1}
set $R \leftarrow \{1, \ldots, n\}$ {set of unfailed components}
draw $\tilde{T}_1, \ldots, \tilde{T}_n$: $\log(U_i) + \int_0^{t_i} \lambda(s) \exp(\beta' Z_i(s))\, ds = 0$
$i_1 \leftarrow \text{argmin}_{j \in R} \tilde{T}_j$ {select the next comp. to fail}
$T_{i_1} \leftarrow \tilde{T}_{i_1}$ {store the next failure times}
for $r \in \{2, \ldots, n\}$ **do**
    {Step r}
    $R \leftarrow R \setminus \{i_{r-1}\}$ {update the set of non-failed components}
    for $j \in B_{i_{r-1}} \cap R$ **do**
        draw $\tilde{T}_j$: $\log(U_j) + \int_{T_{i_{r-1}}}^{t_j} \lambda(s) \exp\left(\beta Z(s) + \alpha \sum_{k \in B_j} C_k(s) Y_k(s)\right) ds = 0$
    end for
    $i_r \leftarrow \text{argmin}_{j \in R} \tilde{T}_j$ {select the next comp. to fail}
    $T_{i_r} \leftarrow \tilde{T}_{i_r}$ {store the next failure times}
end for
return $T_1, \ldots, T_n$

Simulation Algorithm

Require: n, a, b, α, β, $Z_1(\cdot), \ldots, Z_n(\cdot)$

$T \leftarrow$ Vector(length $= n$)
$\tilde{T} \leftarrow$ Vector(length $= n$)
draw $U_1, \ldots, U_n$ i.i.d. from the uniform distribution over [0; 1]
{*Step* 1}
set $R \leftarrow \{1, \ldots, n\}$ {set of unfailed components}
draw $\tilde{T}_1, \ldots, \tilde{T}_n$: $\log(U_i) + \int_0^{t_i} \lambda(s) \exp(\beta' Z_i(s))\, \mathrm{d}s = 0$
$i_1 \leftarrow \text{argmin}_{j \in R} \tilde{T}_j$ {select the next comp. to fail}
$T_{i_1} \leftarrow \tilde{T}_{i_1}$ {store the next failure times}
for $r \in \{2, \ldots, n\}$ **do**
  {Step r}
  $R \leftarrow R \setminus \{i_{r-1}\}$ {update the set of non-failed components}
  for $j \in B_{i_{r-1}} \cap R$ **do**
    draw $\tilde{T}_j$: $\log(U_j) + \int_{T_{i_{r-1}}}^{t_j} \lambda(s) \exp\left(\beta Z(s) + \alpha \sum_{k \in B_j} C_k(s) Y_k(s)\right) \mathrm{d}s = 0$
  end for
  $i_r \leftarrow \text{argmin}_{j \in R} \tilde{T}_j$ {select the next comp. to fail}
  $T_{i_r} \leftarrow \tilde{T}_{i_r}$ {store the next failure times}
end for
return $T_1, \ldots, T_n$

Simulation Algorithm

Require: n, a, b, α, β, $Z_1(\cdot), \ldots, Z_n(\cdot)$
$T \leftarrow$ Vector(length $= n$)
$\tilde{T} \leftarrow$ Vector(length $= n$)
draw $U_1, \ldots, U_n$ i.i.d. from the uniform distribution over [0; 1]
{*Step* 1}
set $R \leftarrow \{1, \ldots, n\}$ {set of unfailed components}
draw $\tilde{T}_1, \ldots, \tilde{T}_n$: $\log(U_i) + \int_0^{t_i} \lambda(s) \exp(\beta' Z_i(s))\, ds = 0$
$i_1 \leftarrow \operatorname{argmin}_{j \in R} \tilde{T}_j$ {select the next comp. to fail}
$T_{i_1} \leftarrow \tilde{T}_{i_1}$ {store the next failure times}
for $r \in \{2, \ldots, n\}$ **do**
  {Step r}
  $R \leftarrow R \setminus \{i_{r-1}\}$ {update the set of non-failed components}
  for $j \in B_{i_{r-1}} \cap R$ **do**
    draw $\tilde{T}_j$: $\log(U_j) + \int_{T_{i_{r-1}}}^{t_j} \lambda(s) \exp\left(\beta Z(s) + \alpha \sum_{k \in B_j} C_k(s) Y_k(s)\right) ds = 0$
  end for
  $i_r \leftarrow \operatorname{argmin}_{j \in R} \tilde{T}_j$ {select the next comp. to fail}
  $T_{i_r} \leftarrow \tilde{T}_{i_r}$ {store the next failure times}
end for
return $T_1, \ldots, T_n$

Simulation Algorithm

Require: $n, a, b, \alpha, \beta, Z_1(\cdot), \ldots, Z_n(\cdot)$

$T \leftarrow$ Vector(length $= n$)
$\tilde{T} \leftarrow$ Vector(length $= n$)
draw $U_1, \ldots, U_n$ i.i.d. from the uniform distribution over $[0; 1]$
$\{$ *Step* 1 $\}$
set $R \leftarrow \{1, \ldots, n\}$ {set of unfailed components}
draw $\tilde{T}_1, \ldots, \tilde{T}_n$: $\log(U_i) + \int_0^{t_i} \lambda(s) \exp(\beta' Z_i(s))\, \mathrm{d}s = 0$
$i_1 \leftarrow \operatorname{argmin}_{j \in R} \tilde{T}_j$ {select the next comp. to fail}
$T_{i_1} \leftarrow \tilde{T}_{i_1}$ {store the next failure times}
for $r \in \{2, \ldots, n\}$ **do**
    $\{$Step $r\}$
    $R \leftarrow R \setminus \{i_{r-1}\}$ {update the set of non-failed components}
    for $j \in B_{i_{r-1}} \cap R$ **do**
        draw $\tilde{T}_j$: $\log(U_j) + \int_{T_{i_{r-1}}}^{t_j} \lambda(s) \exp\left(\beta Z(s) + \alpha \sum_{k \in B_j} C_k(s) Y_k(s)\right) \mathrm{d}s = 0$
    end for
    $i_r \leftarrow \operatorname{argmin}_{j \in R} \tilde{T}_j$ {select the next comp. to fail}
    $T_{i_r} \leftarrow \tilde{T}_{i_r}$ {store the next failure times}
end for
return $T_1, \ldots, T_n$

Simulation Algorithm

Require: $n, a, b, \alpha, \beta, Z_1(\cdot), \ldots, Z_n(\cdot)$

$T \leftarrow$ Vector(length $= n$)
$\tilde{T} \leftarrow$ Vector(length $= n$)
draw $U_1, \ldots, U_n$ i.i.d. from the uniform distribution over [0; 1]
$\{$*Step* 1$\}$
set $R \leftarrow \{1, \ldots, n\}$ {set of unfailed components}
draw $\tilde{T}_1, \ldots, \tilde{T}_n$: $\log(U_i) + \int_0^{t_i} \lambda(s) \exp(\beta' Z_i(s))\, \mathrm{d}s = 0$
$i_1 \leftarrow \text{argmin}_{j \in R} \tilde{T}_j$ {select the next comp. to fail}
$T_{i_1} \leftarrow \tilde{T}_{i_1}$ {store the next failure times}
for $r \in \{2, \ldots, n\}$ **do**
    {Step r}
    $R \leftarrow R \setminus \{i_{r-1}\}$ {update the set of non-failed components}
    for $j \in B_{i_{r-1}} \cap R$ **do**
        draw $\tilde{T}_j$: $\log(U_j) + \int_{T_{i_{r-1}}}^{t_j} \lambda(s) \exp\left(\beta Z(s) + \alpha \sum_{k \in B_j} C_k(s) Y_k(s)\right) \mathrm{d}s = 0$
    end for
    $i_r \leftarrow \text{argmin}_{j \in R} \tilde{T}_j$ {select the next comp. to fail}
    $T_{i_r} \leftarrow \tilde{T}_{i_r}$ {store the next failure times}
end for
return $T_1, \ldots, T_n$

Simulation Algorithm

Require: $n, a, b, \alpha, \beta, Z_1(\cdot), \ldots, Z_n(\cdot)$

$T \leftarrow$ Vector(length $= n$)

$\tilde{T} \leftarrow$ Vector(length $= n$)

draw $U_1, \ldots, U_n$ i.i.d. from the uniform distribution over [0; 1]

{*Step* 1}

set $R \leftarrow \{1, \ldots, n\}$ {set of unfailed components}

draw $\tilde{T}_1, \ldots, \tilde{T}_n$: $\log(U_i) + \int_0^{t_i} \lambda(s) \exp(\beta' Z_i(s))\, \mathrm{d}s = 0$

$i_1 \leftarrow \operatorname{argmin}_{j \in R} \tilde{T}_j$ {select the next comp. to fail}

$T_{i_1} \leftarrow \tilde{T}_{i_1}$ {store the next failure times}

for $r \in \{2, \ldots, n\}$ **do**

  {Step r}

  $R \leftarrow R \setminus \{i_{r-1}\}$ {update the set of non-failed components}

  for $j \in B_{i_{r-1}} \cap R$ **do**

    draw $\tilde{T}_j$: $\log(U_j) + \int_{T_{i_{r-1}}}^{t_j} \lambda(s) \exp\left(\beta Z(s) + \alpha \sum_{k \in B_j} C_k(s) Y_k(s)\right) \mathrm{d}s = 0$

  end for

  $i_r \leftarrow \operatorname{argmin}_{j \in R} \tilde{T}_j$ {select the next comp. to fail}

  $T_{i_r} \leftarrow \tilde{T}_{i_r}$ {store the next failure times}

end for

return $T_1, \ldots, T_n$

Simulation Algorithm

Require: $n, a, b, \alpha, \beta, Z_1(\cdot), \dots, Z_n(\cdot)$

$T \leftarrow$ Vector(length $= n$)
$\tilde{T} \leftarrow$ Vector(length $= n$)
draw $U_1, \dots, U_n$ i.i.d. from the uniform distribution over [0; 1]
$\{$*Step* 1$\}$
set $R \leftarrow \{1, \dots, n\}$ $\{$set of unfailed components$\}$
draw $\tilde{T}_1, \dots, \tilde{T}_n$: $\log(U_i) + \int_0^{t_i} \lambda(s) \exp(\beta' Z_i(s))\, \mathrm{d}s = 0$
$i_1 \leftarrow \mathrm{argmin}_{j \in R} \tilde{T}_j$ $\{$select the next comp. to fail$\}$
$T_{i_1} \leftarrow \tilde{T}_{i_1}$ $\{$store the next failure times$\}$
for $r \in \{2, \dots, n\}$ **do**
    $\{$Step $r\}$
    $R \leftarrow R \setminus \{i_{r-1}\}$ $\{$update the set of non-failed components$\}$
    for $j \in B_{i_{r-1}} \cap R$ **do**
        draw $\tilde{T}_j$: $\log(U_j) + \int_{T_{i_{r-1}}}^{t_j} \lambda(s) \exp\left(\beta Z(s) + \alpha \sum_{k \in B_j} C_k(s) Y_k(s)\right)\, \mathrm{d}s = 0$
    end for
    $i_r \leftarrow \mathrm{argmin}_{j \in R} \tilde{T}_j$ $\{$select the next comp. to fail$\}$
    $T_{i_r} \leftarrow \tilde{T}_{i_r}$ $\{$store the next failure times$\}$
end for
return $T_1, \dots, T_n$

Simulation Algorithm

Require: $n, a, b, \alpha, \beta, Z_1(\cdot), \dots, Z_n(\cdot)$

$T \leftarrow$ Vector(length $= n$)
$\tilde{T} \leftarrow$ Vector(length $= n$)
draw $U_1, \dots, U_n$ i.i.d. from the uniform distribution over $[0; 1]$
$\{$*Step* 1$\}$
set $R \leftarrow \{1, \dots, n\}$ $\{$set of unfailed components$\}$
draw $\tilde{T}_1, \dots, \tilde{T}_n$: $\log(U_i) + \int_0^{t_i} \lambda(s) \exp(\beta' Z_i(s))\, \mathrm{d}s = 0$
$i_1 \leftarrow \operatorname{argmin}_{j \in R} \tilde{T}_j$ $\{$select the next comp. to fail$\}$
$T_{i_1} \leftarrow \tilde{T}_{i_1}$ $\{$store the next failure times$\}$
for $r \in \{2, \dots, n\}$ **do**
  $\{$Step $r\}$
  $R \leftarrow R \setminus \{i_{r-1}\}$ $\{$update the set of non-failed components$\}$
  for $j \in B_{i_{r-1}} \cap R$ **do**
    draw $\tilde{T}_j$: $\log(U_j) + \int_{T_{i_{r-1}}}^{t_j} \lambda(s) \exp\left(\beta Z(s) + \alpha \sum_{k \in B_j} C_k(s) Y_k(s)\right)\, \mathrm{d}s = 0$
  end for
  $i_r \leftarrow \operatorname{argmin}_{j \in R} \tilde{T}_j$ $\{$select the next comp. to fail$\}$
  $T_{i_r} \leftarrow \tilde{T}_{i_r}$ $\{$store the next failure times$\}$
end for
return $T_1, \dots, T_n$

Simulation Algorithm

Require: $n, a, b, \alpha, \beta, Z_1(\cdot), \ldots, Z_n(\cdot)$

$T \leftarrow$ Vector(length $= n$)
$\tilde{T} \leftarrow$ Vector(length $= n$)
draw $U_1, \ldots, U_n$ i.i.d. from the uniform distribution over [0; 1]
$\{$*Step* 1$\}$
set $R \leftarrow \{1, \ldots, n\}$ {set of unfailed components}
draw $\tilde{T}_1, \ldots, \tilde{T}_n$: $\log(U_i) + \int_0^{t_i} \lambda(s) \exp(\beta' Z_i(s))\, \mathrm{d}s = 0$
$i_1 \leftarrow \mathrm{argmin}_{j \in R} \tilde{T}_j$ {select the next comp. to fail}
$T_{i_1} \leftarrow \tilde{T}_{i_1}$ {store the next failure times}
for $r \in \{2, \ldots, n\}$ **do**
    $\{$Step $r\}$
    $R \leftarrow R \setminus \{i_{r-1}\}$ {update the set of non-failed components}
    for $j \in B_{i_{r-1}} \cap R$ **do**
        draw $\tilde{T}_j$: $\log(U_j) + \int_{T_{i_{r-1}}}^{t_j} \lambda(s) \exp\left(\beta Z(s) + \alpha \sum_{k \in B_j} C_k(s) Y_k(s))\right) \mathrm{d}s = 0$
    end for
    $i_r \leftarrow \mathrm{argmin}_{j \in R} \tilde{T}_j$ {select the next comp. to fail}
    $T_{i_r} \leftarrow \tilde{T}_{i_r}$ {store the next failure times}
end for
return $T_1, \ldots, T_n$

Available data

- Inspection times, $\tau_1 < \cdots < \tau_m$, are the same for all components.
- The available data at each inspection time is a binary information on whether the component is failed or not ($Y_i = 0$ or $Y_i = 1$).
- data are interval-censored.
- covariates $C_1, \ldots, C_n$ and $Z_1, \ldots, Z_n$ are observed continuously.

First case: Failure

- $\mathcal{N}_1$: the set of all failed components at the last inspection time τ_m
- For component i, let $j_i = \min\{j;\, Y_i(\tau_j) = 1\}$ be the inspection time s.t. the failure is observed for the first time on the i-th component
- $\mathcal{N}_{1,r} = \{i \in \mathcal{N}_1; j_i = r\}$.
- Failure appears between τ_{j_i-1} and τ_{j_i}.
- The probability of this event is given by:

$$\mathbb{P}\left[\tau_{j_i-1} < T_i \leq \tau_{j_i} | \mathcal{H}_{\tau_{j_i}^-}\right] = S_i\left(\tau_{j_i-1} | \mathcal{H}_{\tau_{j_i-1}^-}\right) - S_i\left(\tau_{j_i} | \mathcal{H}_{\tau_{j_i}^-}\right).$$

Second case: No failure

- $\mathcal{N}_0$: the set of all unfailed components at the last inspection time τ_m.
- We have $\mathbb{P}\left[T_i > \tau_m | \mathcal{H}_{\tau_m^-}\right] = S_i\left(\tau_m | \mathcal{H}_{\tau_m^-}\right)$.
- Notice that the subset $\mathcal{N}_0$ can be empty, but it cannot be the case for $\mathcal{N}_1$.

Pseudo-likelihood

- If $\alpha \neq 0$, the likelihood function can be difficulty written
- The pseudo-likelihood function (ignoring the dependency), pL is given by:

$$
\begin{aligned}
pL\left(\alpha,\beta,a,b|\mathcal{H}_{\tau_m^-}\right) &= \prod_{i\in\mathcal{N}_0} S_i\left(\tau_m|\mathcal{H}_{\tau_m^-}\right)\prod_{i\in\mathcal{N}_1}\left[S_i\left(\tau_{j_i-1}|\mathcal{H}_{\tau_{j_i}^-}\right)-S_i\left(\tau_{j_i}|\mathcal{H}_{\tau_{j_i}^-}\right)\right] \\
&= \prod_{i\in\mathcal{N}_0} S_i\left(\tau_m|\mathcal{H}_{\tau_m^-}\right)\prod_{r=1}^{m}\prod_{i\in\mathcal{N}_{1,r}}\left[S_i\left(\tau_{r-1}|\mathcal{H}_{\tau_{r-1}^-}\right)-S_i\left(\tau_r|\mathcal{H}_{\tau_r^-}\right)\right],
\end{aligned}
$$

- It could rather be useful to consider the pseudo-log-likelihood function $p\ell$ which is given by:

$$
p\ell\left(\alpha,\beta,a,b|\mathcal{H}_{\tau_m^-}\right) = \sum_{i\in\mathcal{N}_0}\log\left[S_i\left(\tau_m|\mathcal{H}_{\tau_m^-}\right)\right]+\sum_{r=1}^{m}\sum_{i\in\mathcal{N}_{1,r}}\log\left[S_i\left(\tau_{r-1}|\mathcal{H}_{\tau_{r-1}^-}\right)-S_i\left(\tau_r|\mathcal{H}_{\tau_r^-}\right)\right]
$$

Maximum Pseudo-Likelihood Estimator

- Maximum Pseudo-Likelihood Estimator of the parameters cannot be computed neither in a closed-form, nor numerically.
- hazard functions depend on binary stochastic processes $Y_1, \ldots, Y_n$ which are only known at inspection times.
- to compute the survival at time t, it is required to the value of the hazard function at any time between 0 and t.
- $\Rightarrow$ use of SEM algorithm

SEM Algorithm: Step 0

Step 0: parameter initialization.

For the covariates, set $\widehat{\alpha}^{(0)} = 0$ and $\widehat{\beta}^{(0)} = 0$ (no effect).

Parameters *a* and *b* can be thus estimated by maximizing the log-likelihood (since components are independent): $\widehat{a}^{(0)}$ and $\widehat{b}^{(0)}$.

SEM Algorithm: Step k

Step k: time-to-failures simulation and updating estimation.

1. simulate $T_1, \ldots, T_n$ using Sim. Algo. and considering the parameters $\widehat{\alpha}^{(k-1)}$, $\widehat{\beta}^{(k-1)}$, $\widehat{a}^{(k-1)}$ and $\widehat{b}^{(k-1)}$;
2. update the estimation by maximizing numerically the pseudo-log-likelihood function: $\widehat{\beta}^{(k)}$, $\widehat{\alpha}^{(k)}$, $\widehat{a}^{(k)}$ and $\widehat{b}^{(k)}$.

The final estimator is then given by considering the ergodic average of the estimators:

$$\widehat{\alpha} = \frac{1}{K}\sum_{k=1}^{K}\widehat{\alpha}^{(k)}, \quad \widehat{\beta} = \frac{1}{K}\sum_{k=1}^{K}\widehat{\beta}^{(k)}, \quad \widehat{a} = \frac{1}{K}\sum_{k=1}^{K}\widehat{a}^{(k)} \quad \text{and} \quad \widehat{b} = \frac{1}{K}\sum_{k=1}^{K}\widehat{b}^{(k)}.$$

This algorithm requires to simulate the missing data, which are the times-to-failure of the components.

SEM Algorithm: Step k

Step k: time-to-failures simulation and updating estimation.

1. simulate $T_1, \ldots, T_n$ using Sim. Algo. and considering the parameters $\widehat{\alpha}^{(k-1)}$, $\widehat{\beta}^{(k-1)}$, $\widehat{a}^{(k-1)}$ and $\widehat{b}^{(k-1)}$;
2. update the estimation by maximizing numerically the pseudo-log-likelihood function: $\widehat{\beta}^{(k)}$, $\widehat{\alpha}^{(k)}$, $\widehat{a}^{(k)}$ and $\widehat{b}^{(k)}$.

The final estimator is then given by considering the ergodic average of the estimators:

$$\widehat{\alpha} = \frac{1}{K}\sum_{k=1}^{K}\widehat{\alpha}^{(k)}, \quad \widehat{\beta} = \frac{1}{K}\sum_{k=1}^{K}\widehat{\beta}^{(k)}, \quad \widehat{a} = \frac{1}{K}\sum_{k=1}^{K}\widehat{a}^{(k)} \quad \text{and} \quad \widehat{b} = \frac{1}{K}\sum_{k=1}^{K}\widehat{b}^{(k)}.$$

This algorithm requires to simulate the missing data, which are the times-to-failure of the components.

SEM Algorithm: Step k

Step k: time-to-failures simulation and updating estimation.

1. simulate $T_1, \ldots, T_n$ using Sim. Algo. and considering the parameters $\widehat{\alpha}^{(k-1)}$, $\widehat{\beta}^{(k-1)}$, $\widehat{a}^{(k-1)}$ and $\widehat{b}^{(k-1)}$;
2. update the estimation by maximizing numerically the pseudo-log-likelihood function: $\widehat{\beta}^{(k)}$, $\widehat{\alpha}^{(k)}$, $\widehat{a}^{(k)}$ and $\widehat{b}^{(k)}$.

The final estimator is then given by considering the ergodic average of the estimators:

$$\widehat{\alpha} = \frac{1}{K}\sum_{k=1}^{K}\widehat{\alpha}^{(k)}, \quad \widehat{\beta} = \frac{1}{K}\sum_{k=1}^{K}\widehat{\beta}^{(k)}, \quad \widehat{a} = \frac{1}{K}\sum_{k=1}^{K}\widehat{a}^{(k)} \quad \text{and} \quad \widehat{b} = \frac{1}{K}\sum_{k=1}^{K}\widehat{b}^{(k)}.$$

This algorithm requires to simulate the missing data, which are the times-to-failure of the components.

SEM Algorithm: Step k

Step k: time-to-failures simulation and updating estimation.

1. simulate $T_1, \dots, T_n$ using Sim. Algo. and considering the parameters $\widehat{\alpha}^{(k-1)}$, $\widehat{\beta}^{(k-1)}$, $\widehat{a}^{(k-1)}$ and $\widehat{b}^{(k-1)}$;
2. update the estimation by maximizing numerically the pseudo-log-likelihood function: $\widehat{\beta}^{(k)}$, $\widehat{\alpha}^{(k)}$, $\widehat{a}^{(k)}$ and $\widehat{b}^{(k)}$.

The final estimator is then given by considering the ergodic average of the estimators:

$$\widehat{\alpha} = \frac{1}{K}\sum_{k=1}^{K}\widehat{\alpha}^{(k)}, \quad \widehat{\beta} = \frac{1}{K}\sum_{k=1}^{K}\widehat{\beta}^{(k)}, \quad \widehat{a} = \frac{1}{K}\sum_{k=1}^{K}\widehat{a}^{(k)} \quad \text{and} \quad \widehat{b} = \frac{1}{K}\sum_{k=1}^{K}\widehat{b}^{(k)}.$$

This algorithm requires to simulate the missing data, which are the times-to-failure of the components.

SEM Algorithm: Step k

Step k: time-to-failures simulation and updating estimation.

1. simulate $T_1, \ldots, T_n$ using Sim. Algo. and considering the parameters $\widehat{\alpha}^{(k-1)}$, $\widehat{\beta}^{(k-1)}$, $\widehat{a}^{(k-1)}$ and $\widehat{b}^{(k-1)}$;
2. update the estimation by maximizing numerically the pseudo-log-likelihood function: $\widehat{\beta}^{(k)}$, $\widehat{\alpha}^{(k)}$, $\widehat{a}^{(k)}$ and $\widehat{b}^{(k)}$.

The final estimator is then given by considering the ergodic average of the estimators:

$$\widehat{\alpha} = \frac{1}{K}\sum_{k=1}^{K}\widehat{\alpha}^{(k)}, \quad \widehat{\beta} = \frac{1}{K}\sum_{k=1}^{K}\widehat{\beta}^{(k)}, \quad \widehat{a} = \frac{1}{K}\sum_{k=1}^{K}\widehat{a}^{(k)} \quad \text{and} \quad \widehat{b} = \frac{1}{K}\sum_{k=1}^{K}\widehat{b}^{(k)}.$$

This algorithm requires to simulate the missing data, which are the times-to-failure of the components.

Simulation Algorithm based on observations 1

Require: n, a, b, α, β, $Z_1(\cdot), \dots, Z_n(\cdot)$, $\mathbf{D}_{obs}$

$T \leftarrow$ Vector(length $= n$)
$\tilde{T} \leftarrow$ Vector(length $= n$)
draw $U_1, \dots, U_n$ i.i.d. from the uniform distribution over $[0; 1]$
{*Interval* I_1}
$R \leftarrow I_1$ {set of unfailed components in I_1}
for $j \in I_1$ **do**
  draw $\tilde{T}_j$: $\log(U_j) + \int_0^{t_j} \lambda(s) \exp(\beta' Z_j(s))\, \mathrm{d}s = 0$
end for
$h \leftarrow \operatorname{argmin}_{j \in R} \tilde{T}_j$ {select the next comp. to fail}
$T_h \leftarrow \tilde{T}_h$ {store the next failure times}
if $n_1 > 1$ **then**
  for $r \in \{2, \dots, n_1\}$ **do**
    $R \leftarrow R \setminus \{h\}$ {update the set of non-failed components in I_1}
    for $j \in B_h \cap R$ **do**
      draw $\tilde{T}_j$: $\log(U_j) + \int_{T_h}^{t_j} \lambda(s) \exp\left(\beta Z(s) + \alpha \sum_{k \in B_j} C_k(s) Y_k(s)\right)\, \mathrm{d}s = 0$
    end for
    $h \leftarrow \operatorname{argmin}_{j \in R} \tilde{T}_j$ {select the next comp. to fail}
    $T_h \leftarrow \tilde{T}_h$ {store the next failure times}
  end for
end if

Simulation Algorithm based on observations 2

for $i \in \{2, \ldots, m\}$ **do**
  {Interval I_i}
  $R \leftarrow I_i$
  if $n_i > 0$ **then**
    for $j \in I_1$ **do**
      draw $\tilde{T}_j$: $\log(U_j) + \int_{\tau_{i-1}}^{t_j} \lambda(s) \exp\left(\beta Z(s) + \alpha \sum_{k \in B_j} C_k(s) Y_k(s)\right) \mathrm{d}s = 0$
    end for
    $h \leftarrow \text{argmin}_{j \in R} \tilde{T}_j$ {select the next comp. to fail}
    $T_h \leftarrow \tilde{T}_h$ {store the next failure times}
    if $n_1 > 1$ **then**
      for $r \in \{2, \ldots, n_1\}$ **do**
        $R \leftarrow R \setminus \{h\}$ {update the set of non-failed components in I_i}
        for $j \in B_h \cap R$ **do**
          draw $\tilde{T}_j$: $\log(U_j) + \int_{T_h}^{t_j} \lambda(s) \exp\left(\beta Z(s) + \alpha \sum_{k \in B_j} C_k(s) Y_k(s)\right) \mathrm{d}s = 0$
        end for
        $h \leftarrow \text{argmin}_{j \in R} \tilde{T}_j$ {select the next comp. to fail}
        $T_h \leftarrow \tilde{T}_h$ {store the next failure times}
      end for
    end if
  end if
end for

Numerical Illustration

- inspection times are periodic, say $\tau_j = j\delta$ for some $\delta > 0$.
- $n = 20 \times 20$ components.
- simulations with $a = 2$, $b = 3$, $\alpha = 1$, $K = 100$ and $\delta = 0.1$.
- no covariate/no constraint:

$$\forall t \geq 0, \quad \lambda_{i,j}(t|\mathcal{H}_{t^-}) = \lambda(t) \exp\left(\alpha \sum_{(i',j') \in B_{i,j}} Y_{i',j'}(t)\right).$$

$\hat{a}$	$\hat{b}$	$\hat{\alpha}$
2.206064	2.645539	1.019503

Numerical Illustration

- inspection times are periodic, say $\tau_j = j\delta$ for some $\delta > 0$.
- $n = 20 \times 20$ components.
- simulations with $a = 2$, $b = 3$, $\alpha = 1$, $K = 100$ and $\delta = 0.1$.
- no covariate/no constraint:

$$\forall t \geq 0, \quad \lambda_{i,j}(t|\mathcal{H}_{t^-}) = \lambda(t) \exp\left(\alpha \sum_{(i',j')\in B_{i,j}} Y_{i',j'}(t)\right).$$

$\hat{a}$	$\hat{b}$	$\hat{\alpha}$
2.206064	2.645539	1.019503

Future Work

- Constraint exerced at the center of the grid (e.g. $n = 2p + 1$) and diffused isotropically from the center as follows:

$$C_{i,j}(t) = \exp\left(-\frac{1}{\sigma^2}\left((i-p-1)^2 + (j-p-1)^2\right)\right).$$

- Application on real data