

# Rapport de projet OS13

## Analyse de politique de maintenance

TRAN QUOC NHAT HAN & ADRIEN WARTELLE

1<sup>er</sup> janvier 2019

### Sommaire

<b>1</b>	<b>Maintenance basant sur l'âge</b>	<b>1</b>
1.1	Rappel . . . . .	1
1.2	Modéliser la durée de vie du système . . . . .	2
<b>2</b>	<b>Annexe</b>	<b>4</b>
2.1	L'importation de données de pannes . . . . .	4
2.2	Le premier histogramme de distribution de pannes . . . . .	4
2.3	Estimer le mixage de la loi Exponentielle et Gamma . . . . .	5

### Résumé

Soient des données liées à la fonctionnement de système, nous déterminons un modèle approprié et puis choisir une politique de maintenance optimal.

## 1 Maintenance basant sur l'âge

### 1.1 Rappel

Considérons un système non maintenu. En l'observant, nous obtenons un liste des dates de panne, grâce auquel nous construirons une politique de remplacement systématique basée sur l'âge : *Nous remplaçons lorsque le système tombe en panne ou qu'il survit une durée  $t_0$ .*

Le but est de minimiser le coût moyen cumulé.

$$\mathbb{E}(C) = \frac{\mathbb{E}(C(S))}{\mathbb{E}(S)} \quad (1)$$

Où  $S$  est la variable aléatoire représentant la date de remplacement et  $C(S)$  est le coût de maintenance cumulé à l'instant  $S$  (sachant que  $C(S)$  est  $c_c$  si une maintenance corrective et  $c_p$  si préventive).

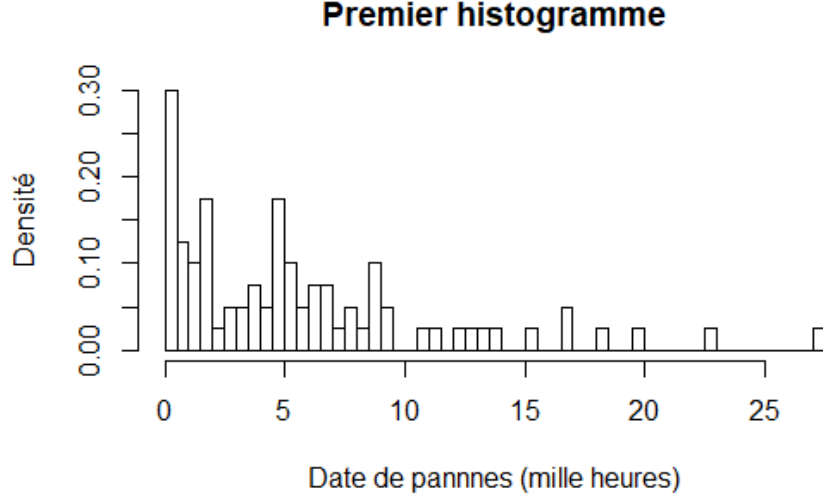


FIGURE 1 – Le premier histogramme de distribution de pannes

## 1.2 Modéliser la durée de vie du système

L'importation de données nous montre que les dates de pannes sont de l'ordre grandement variée (300 à 27000). Exponentiel des valeurs extrêmes résulteront *Inf*, ce qui est indésirable. Alors nous devons forcément les réduire en les divisant par un scalaire **scale**, prenons par exemple 1000. (Figure 1.2)

Les pannes se concentrent autour de 2 sommets, l'un à  $[0; 0, 5]$  et l'autre à  $[4, 5; 5]$ . Ceci nous fait penser naturellement à un mixage de deux lois.

Comme les valeurs sont positives, et que l'un sommet se situe auprès de zéro et l'autre à une valeur non nulle, nous essayons d'estimer un mixage de loi *Exponentielle* et *Gamma*.

La fonction de densité avec le paramètre  $\theta = (p_1, p_2, \lambda, \alpha, \beta)$  :

$$\begin{aligned} f_{\theta}(x) &= p_1 f_1(x) + p_2 f_2(x) \\ &= p_1 \lambda e^{-\lambda x} + p_2 \frac{\beta^{\alpha}}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x} \end{aligned} \quad (2)$$

Où  $f_1, f_2$  désignent respectivement  $\exp(\lambda)$  et  $\Gamma(\alpha, \beta)$  ;  $p_1, p_2 > 0$  ;  $p_1 + p_2 = 1$ .

Nous allons utiliser l'algorithme EM, la méthode la plus efficace pour estimer le MLE de mixage fini.

Soit  $\mathbf{X} = (x_1, \dots, x_N)$  le vecteur des données existants.

Soit la matrice de probabilité d'appartenance  $(\zeta_{ki})$ .  $\zeta_{ki}$  vaut la probabilité que  $x_i$  suive la loi  $f_k$ .

$$\zeta_{ki} = \frac{p_k f_k(x_i)}{p_1 f_1(x_i) + p_2 f_2(x_i)} \quad \forall k = \overline{1, 2} \quad \forall i = \overline{1, N}$$

La fonction de vraisemblance :

$$\ln \Lambda = \sum_{i=1}^N \ln f_{\theta}(x_i) = \sum_{i=1}^N \ln (p_1 f_1(x_i) + p_2 f_2(x_i)) \quad (3)$$

Nous cherchons à maximiser  $\ln \Lambda$  en la dérivant selon  $\lambda, \alpha, \beta$ .

Pour  $\lambda$  :

$$\begin{aligned} \frac{\partial}{\partial \lambda} \ln \Lambda &= \sum_{i=1}^N \frac{p_1 e^{-\lambda x_i} - p_1 \lambda x_i e^{-\lambda x_i}}{p_1 f_1(x_i) + p_2 f_2(x_i)} \\ &= \sum_{i=1}^N \frac{p_1 f_1(x_i)}{p_1 f_1(x_i) + p_2 f_2(x_i)} \left( \frac{1}{\lambda} - x_i \right) \\ &= \frac{1}{\lambda} \sum_{i=1}^N \zeta_{1i} - \sum_{i=1}^N \zeta_{1i} x_i = 0 \\ \Leftrightarrow \lambda &= \frac{\sum_{i=1}^N \zeta_{1i}}{\sum_{i=1}^N \zeta_{1i} x_i} \end{aligned} \quad (4)$$

Pour  $\beta$  :

$$\begin{aligned} \frac{\partial}{\partial \beta} \ln \Lambda &= \sum_{i=1}^N \frac{p_2 x_i^{\alpha-1}}{\Gamma(\alpha)} \frac{\alpha \beta^{\alpha-1} e^{-\beta x_i} - \beta^{\alpha} x_i e^{-\beta x_i}}{p_1 f_1(x_i) + p_2 f_2(x_i)} \\ &= \sum_{i=1}^N \frac{p_2 f_2(x_i)}{p_1 f_1(x_i) + p_2 f_2(x_i)} \left( \frac{\alpha}{\beta} - x_i \right) \\ &= \frac{\alpha}{\beta} \sum_{i=1}^N \zeta_{2i} - \sum_{i=1}^N \zeta_{2i} x_i = 0 \\ \Leftrightarrow \beta &= \alpha \frac{\sum_{i=1}^N \zeta_{2i}}{\sum_{i=1}^N \zeta_{2i} x_i} \end{aligned} \quad (5)$$

Pour  $\alpha$ , le calcul devient tellement difficile que la démonstration ici ne sera pas suffisante. Nous n'écrivons que le résultat approximatif de Newton-Rashphon : Fixer un entier non nul  $n_r$  indiquant le nombre d'itération :

$$\alpha_{r+1} = \alpha_r - \frac{\ln(\alpha_r) - \Psi(\alpha_r) - \ln \left( \frac{\sum_{i=1}^N \zeta_{2i} x_i}{\sum_{i=1}^N \zeta_{2i}} \right) + \frac{\sum_{i=1}^N \zeta_{2i} \ln(x_i)}{\sum_{i=1}^N \zeta_{2i}}}{\frac{1}{\alpha_r} - \Psi'(\alpha_r)} \quad (6)$$

Où  $\Psi$  et  $\Psi'$  sont les fonctions digamma et trigamma dans l'ordre.

Et puis, pour  $p_k$  :

$$p_k = \frac{\sum_{i=1}^N \zeta_{ki}}{N} \quad \forall k = \overline{1, 2} \quad (7)$$

Etant donné (4), (5), (6) et (7), nous définissons l'algorithme EM :

1. **Initialisation** : Choisir un  $\theta_{vieux}$ .
2. **Etape E** : Evaluer  $(\zeta_{ki})$  sachant  $\theta_{vieux}$ .
3. **Etape M** : Calculer  $\theta_{nouveau}$  à l'aide des équations (4), (5), (6) et (7).  
*Note* : Pour  $\alpha$ , l'itération se termine quand  $|\alpha_{r+1} - \alpha_r| < \varepsilon_\alpha$  avec  $\varepsilon_\alpha$  est un réel positif fixé à l'initialisation.
4. **Evaluation** : Si  $|\theta_{nouveau} - \theta_{vieux}| < \varepsilon_\theta$  ( $\varepsilon_\theta$  est un réel positif fixé à l'initialisation), l'algorithme finit et  $\theta = \theta_{vieux}$ .  
Sinon, reviens à l'étape E avec  $\theta_{vieux} \leftarrow \theta_{nouveau}$ .

Nous définissons ensuite la fonction de vraisemblance négative et utilisons la méthode MLE (implémentée dans le package *stats4*). Puisque les valeurs sont bornées ( $p \in ]0; 1[$ ;  $\lambda > 0$ ;  $\alpha > 0$ ;  $\beta > 0$ ), il nous fallait spécifier l'approche *L-BFGS-B*. Le code complet se trouve à l'annexe.

Malheureusement, le message d'erreur

```
Error in solve.default(oout$hessian) :  
Lapack routine dgesv: system is exactly singular: U[2,2] = 0
```

indique que les valeurs initiales, qui sont critiques pour cette méthode, étaient mal choisies en l'absence de théorèmes appropriés.

Nous devons passer à un mixage plus simple de deux lois Exponentielles. Prenons le cas où les deux coefficients d'intensité sont **différents**, car sinon nous retombe à la loi Exponentielle, qui est définitivement inappropriée.

La fonction de densité avec le paramètre  $\theta = (p, \lambda, \mu)$  :

$$f_\theta(x) = p\lambda e^{\lambda x} + (1-p)\mu e^{\mu x} \quad (8)$$

## 2 Annexe

### 2.1 L'importation de données de pannes

```
1 pannes = read.csv(  
2   file = "FailureTimes_5.csv",  
3   header = TRUE,  
4   sep = ",",  
5   dec = ".",  
6   colClasses = c("NULL", NA)  
7 )  
8 scale = 1000  
9 data = pannes$Heures / scale  
10 N = length(data)
```

### 2.2 Le premier histogramme de distribution de pannes

```
1 hist(  
2   data,  
3   breaks = 40,  
4   probability = TRUE,  
5   xlab = "Date de pannes (mille heures)",  
6   ylab = "Densité",  
7   main = "Premier histogramme"  
8 )
```

## 2.3 Estimer le mixage de la loi Exponentielle et Gamma

```

1 # Fitting mixture of Exp and Gamma
2 s = log(mean(data[data >= 2.5])) - mean(log(data[data >= 2.5]))
3 theta_melExpGamma = list(
4   p = 0.5,
5   lambda = 1 / sum(data[data <= 2.5]),
6   alpha = (3 - s + sqrt((s - 3)^2 + 24 * s)) / (12 * s),
7   beta = mean(data[data >= 2.5]) / theta_melExpGamma$alpha
8 )
9 # PDF
10 f_melExpGamma = function(x, theta = theta_melExpGamma) {
11   theta$p * dexp(data, rate = theta$p) + (1 - theta$p) * dgamma(data, shape
12     = theta$alpha, rate = theta$beta)
13 }
14 # CDF
15 F_melExpGamma = function(x, theta = theta_melExpGamma) {
16   theta$p * pexp(data, rate = theta$p) + (1 - theta$p) * pgamma(data, shape
17     = theta$alpha, rate = theta$beta)
18 }
19 # Fitting distribution
20 library(stats4)
21 nle_melExpGamma = function(
22   p,
23   lambda,
24   alpha,
25   beta
26 ) {
27   ## for debug purpose
28   theta = list(
29     p = p,
30     lambda = lambda,
31     alpha = alpha,
32     beta = beta
33   )
34   cat("p=", theta$p,
35     " lambda=", theta$lambda,
36     " alpha=", theta$alpha,
37     " beta=", theta$beta,
38     "\n")
39   - sum(log(f_melExpGamma(data, theta)))
40 }
41 fit_melExpGamma = mle(
42   nle_melExpGamma,
43   start = theta_melExpGamma,
44   method = "L-BFGS-B",
45   lower = c(
46     0,
47     1e-6,
48     1e-6,
49     1e-6
50   ),
51   upper = c(
52     1,
53     Inf,
54     Inf,
55     Inf
56   ),
57   control = list(
58     trace = 6,
59     maxit = 10000,
60     ndeps = c(
61       1e-7,
62       1e-7,
63       1e-7,
64       1e-7
65     )
66   )
67 )
68 summary(fit_melExpGamma)
69 theta_melExpGamma$p = fit_melExpGamma$coef["p"]
70 theta_melExpGamma$lambda = fit_melExpGamma$coef["lambda"]

```

```

70 theta_melExpGamma$alpha = fit_melExpGamma@coef["alpha"]
71 theta_melExpGamma$beta = fit_melExpGamma@coef["beta"]
72 h_melExpGamma = hist(
73   data,
74   breaks = 40,
75   probability = TRUE,
76   main = "Fitting mixture of Exp and Gamma"
77 )
78 curve(
79   f_melExpGamma(x),
80   add = TRUE,
81   col = "violet",
82   from = min(h_melExpGamma$mids),
83   to = max(h_melExpGamma$mids)
84 )

```