

Rapport de projet OS13

Analyse de politique de maintenance

TRAN QUOC NHAT HAN & ADRIEN WARTELLE

2 janvier 2019

Sommaire

1	Maintenance basant sur l'âge	1
1.1	Rappel	1
1.2	Modéliser la durée de vie du système	2
1.3	La politique de maintenance basée sur l'âge	5
2	Annexe	5
2.1	L'importation de données de pannes	5
2.2	Le premier histogramme de distribution de pannes	5
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).

1.2 Modéliser la durée de vie du système

L'importation de données (l'annexe 2.1) nous montre que les dates de pannes sont de l'ordre grandement variée (300 à 27000) (l'annexe 2.2). 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)

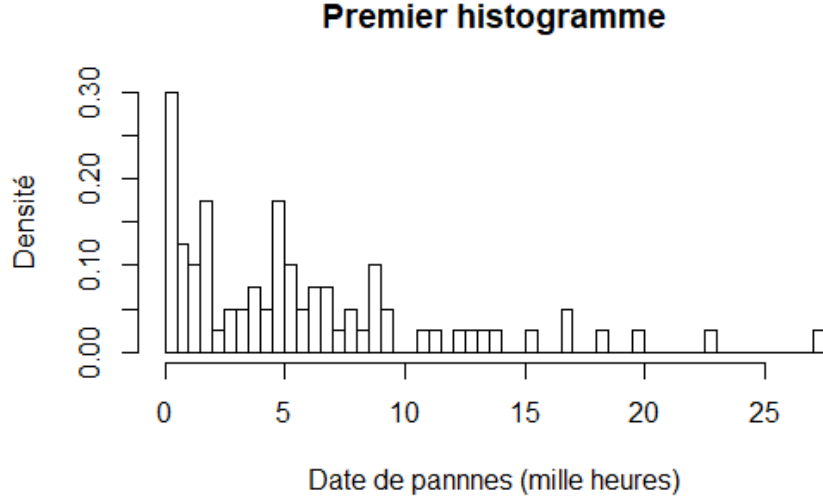


FIGURE 1 – Le premier histogramme de distribution de pannes

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 (ζ_{ki}) : ζ_{ki} vaut la probabilité que x_i suive la loi f_k .

$$\zeta_{ki} = \frac{p_k f_k(x_i)}{p_1 f_1(x_i) + p_2 f_2(x_i)} \quad \forall k = \overline{1, 2} \quad \forall i = \overline{1, N} \quad (3)$$

La fonction de vraisemblance :

$$\ln \Lambda = \sum_{i=1}^N \ln f_{\theta}(x_i) = \sum_{i=1}^N \ln (p_1 f_1(x_i) + p_2 f_2(x_i)) \quad (4)$$

Nous cherchons à maximiser $\ln \Lambda$ en la dérivant selon λ, α, β .
Pour λ :

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

Pour β :

$$\begin{aligned} \frac{\partial}{\partial \beta} \ln \Lambda &= \sum_{i=1}^N \frac{p_2 x_i^{\alpha-1}}{\Gamma(\alpha)} \frac{\alpha \beta^{\alpha-1} e^{-\beta x_i} - \beta^{\alpha} x_i e^{-\beta x_i}}{p_1 f_1(x_i) + p_2 f_2(x_i)} \\ &= \sum_{i=1}^N \frac{p_2 f_2(x_i)}{p_1 f_1(x_i) + p_2 f_2(x_i)} \left(\frac{\alpha}{\beta} - x_i \right) \\ &= \frac{\alpha}{\beta} \sum_{i=1}^N \zeta_{2i} - \sum_{i=1}^N \zeta_{2i} x_i = 0 \\ \Leftrightarrow \beta &= \alpha \frac{\sum_{i=1}^N \zeta_{2i}}{\sum_{i=1}^N \zeta_{2i} x_i} \end{aligned} \quad (6)$$

Pour α :

$$\begin{aligned}
\frac{\partial}{\partial \alpha} \ln \Lambda &= \sum_{i=1}^N \frac{p_2 e^{-\beta x_i}}{p_1 f_1(x) + p_2 f_2(x)} \left(\frac{\beta (\ln \beta + \ln x_i) (\beta x_i)^{\alpha-1}}{\Gamma(\alpha)} - \beta^\alpha x_i^{\alpha-1} \frac{\Psi(\alpha)}{\Gamma(\alpha)} \right) \\
&= \sum_{i=1}^N \frac{p_2 f_2(x_i)}{p_1 f_1(x) + p_2 f_2(x)} (\ln \beta + \ln x_i - \Psi(\alpha)) \\
&= \left(\sum_{i=1}^N \zeta_{2i} \right) \ln \beta + \sum_{i=1}^N \zeta_{2i} \ln x_i - \Psi(\alpha) \left(\sum_{i=1}^N \zeta_{2i} \right) = 0 \\
\Leftrightarrow 0 &= \ln \alpha + \ln \frac{\sum_{i=1}^N \zeta_{2i}}{\sum_{i=1}^N \zeta_{2i} x_i} + \frac{\sum_{i=1}^N \zeta_{2i} \ln x_i}{\sum_{i=1}^N \zeta_{2i}} - \Psi(\alpha) \quad (\text{substitué par (6)}) \\
\Leftrightarrow 0 &= \ln \alpha - \Psi(\alpha) - c
\end{aligned}$$

Où $c = \ln \left(\frac{\sum_{i=1}^N \zeta_{2i} x_i}{\sum_{i=1}^N \zeta_{2i}} \right) - \frac{\sum_{i=1}^N \zeta_{2i} \ln(x_i)}{\sum_{i=1}^N \zeta_{2i}}$; Ψ est la fonction digamma.

Selon la méthode de Newton-Rashphon, nous pouvons résoudre α numériquement avec ce formul itératif :

$$\alpha_{r+1} = \alpha_r - \frac{\ln \alpha_r + \Psi(\alpha_r) - c}{\frac{1}{\alpha_r} - \Psi'(\alpha_r)}$$

[1] propose un autre formule convergeant plus vite :

$$\frac{1}{\alpha_{r+1}} = \frac{1}{\alpha_r} + \frac{\ln(\alpha_r) - \Psi(\alpha_r) - c}{a_r^2 \left(\frac{1}{\alpha_r} - \Psi'(\alpha_r) \right)} \quad (7)$$

Avec Ψ' la fonction trigamma. L'itération part avec $\alpha_0 = \frac{0.5}{c}$.

Au final, pour p_k :

$$p_k = \frac{\sum_{i=1}^N \zeta_{ki}}{N} \forall k = \overline{1, 2} \quad (8)$$

Etant donné (3), (5), (6), (7) et (8), nous définissons l'algorithme EM :

1. **Initialisation** : Choisir un θ_{vieux} .
2. **Etape E** : Evaluer (ζ_{ki}) sachant θ_{vieux} en utilisant (3).
3. **Etape M** : Calculer $\theta_{nouveau}$ à l'aide des équations (5), (6), (7) et (8).
Note : Pour α , l'itération se termine quand $|\alpha_{r+1} - \alpha_r| < \varepsilon_\alpha$ où ε_α est un réel positif fixé à l'initialisation.
4. **Evaluation** : Si $\|\theta_{c+1} - \theta_c\| < \varepsilon_\theta$ (ε_θ est un réel positif fixé à l'initialisation), l'algorithme s'arrête et $\theta = \theta_{vieux}$.
Sinon, reviens à l'étape E avec $\theta_{vieux} \leftarrow \theta_{nouveau}$.

Basé sur le résultat obtenu, nous traçons la fonction de densité f_θ trouvé (figure 2) et réalisons un test de Kolmogorov-Smirnov qui donne $p - value = 0,9663111$ signifiant 96,63% de nous tromper si nous rejetons ce modèle. Nous l'acceptons alors, quoiqu'il ne génère pas 2 sommets comme la remarque initiale. Le code est trouvable à l'annexe 2.3.

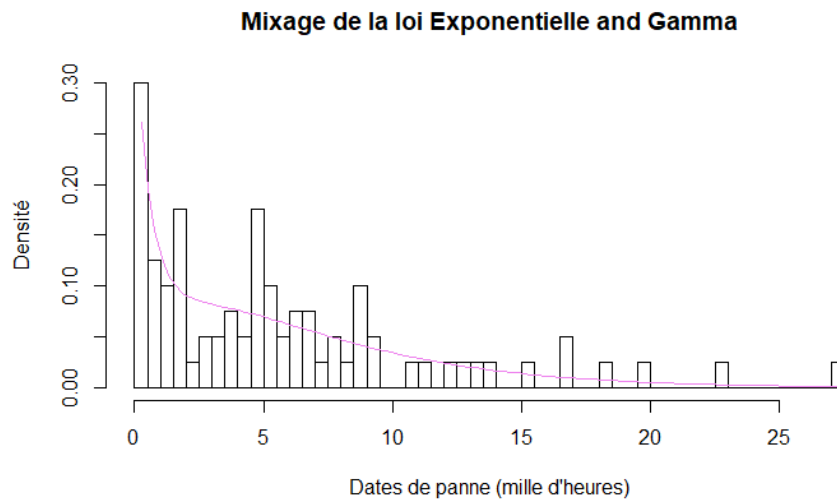


FIGURE 2 – Mixage de la loi Exponentielle et Gamma

1.3 La politique de maintenance basée sur l'âge

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 # Algorithm EM
3 # Initialisation

```

```

4 k = 2 # number of components
5 p = c(0.5, 0.5)
6 lambda = 1
7 alpha = 5
8 beta = 1
9 f = list(
10   '1' = function(x) {
11     dexp(x, rate = lambda)
12   },
13   '2' = function(x) {
14     dgamma(x, shape = alpha, rate = beta)
15   }
16 )
17 epsilon = list(
18   alpha = 1e-4,
19   theta = 1e-4
20 )
21 zeta = matrix(
22   0,
23   nrow = k,
24   ncol = N
25 )
26 # Norm
27 normVec = function(x) sqrt(sum(x^2))
28 # New value
29 p_new = p
30 alpha_new = alpha
31 beta_new = beta
32 lambda_new = lambda
33 repeat {
34   ## E Step
35   # Calculate each proba
36   for (l in 1:k) {
37     zeta[l,] = p[[l]] * f[[l]](data)
38   }
39   # Normalize proba
40   zeta = t(t(zeta) / rowSums(t(zeta)))
41   ## M step
42   # Lambda
43   lambda_new = sum(zeta[1,]) / sum(zeta[1,] * data)
44   # Alpha
45   c = log(sum(zeta[2,] * data) / sum(zeta[2,])) - sum(zeta[2,] * log(data))
46     / sum(zeta[2,])
47   alpha_new = 0.5 / c
48   alpha_temp = 0
49   repeat {
50     alpha_temp = 1 / (1 / alpha_new + (log(alpha_new) - digamma(alpha_new)
51       - c) / (alpha_new^2 * (1 / alpha_new - trigamma(alpha_new))))
52     if (abs(alpha_temp - alpha_new) < epsilon$alpha) {
53       break
54     } else {
55       alpha_new = alpha_temp
56     }
57   }
58   # Beta
59   beta_new = alpha_new * sum(zeta[2,]) / sum(zeta[2,] * data)
60   # P
61   for (l in 1:k) {
62     p_new[[l]] = mean(zeta[l,])
63   }
64   ## Evaluation
65   if (normVec(c(alpha, beta, lambda, p[[1]], p[[2]]) - c(alpha_new, beta_
66     new, lambda_new, p_new[[1]], p_new[[2]])) < epsilon$theta) {
67     break
68   } else {
69     alpha = alpha_new
70     beta = beta_new
71     lambda = lambda_new
72     p = p_new
73   }
74 }
75 # Final value update
76 alpha = alpha_new

```

```

75 beta = beta_new
76 lambda = lambda_new
77 p = p_new
78
79 # Illustration
80 f_theta = function(x) {
81   p[[1]] * f[[1]](x) + p[[2]] * f[[2]](x)
82 }
83 h_theta = hist(
84   data,
85   breaks = 40,
86   probability = TRUE,
87   main = "Mixage de la loi Exponentielle and Gamma",
88   xlab = "Dates de panne (mille d'heures)",
89   ylab = "Densité"
90 )
91 curve(
92   f_theta(x),
93   add = TRUE,
94   col = "violet",
95   from = min(h_theta$mids),
96   to = max(h_theta$mids)
97 )
98 # Kolmogorov-Smirnov test
99 F_theta = function(x) {
100   p[[1]] * pexp(x, rate = lambda) + p[[2]] * pgamma(x, shape = alpha, rate
      = beta)
101 }
102 test = ks.test(data, F_theta, exact = TRUE)

```

Références

- [1] Minka, Thomas P. (2002). "Estimating a Gamma distribution"
<https://tminka.github.io/papers/minka-gamma.pdf>