

Algorithme EM en fiabilité des logiciels

James Ledoux

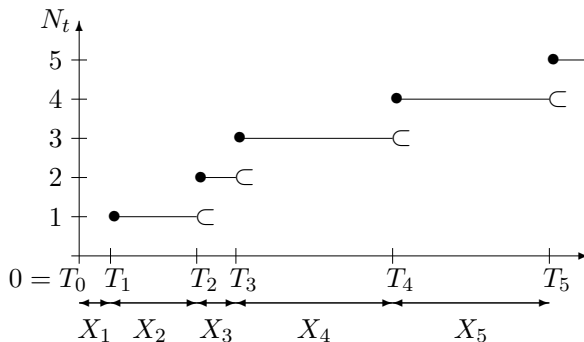
INSA de Rennes & IRMAR Rennes I

- 1 Processus de défaillance en fiabilité du logiciel
 - Modèle auto-excité
 - Modèles DIDE et conditionnellement DIDE
- 2 HMC discrète pour la fiabilité des logiciels
 - Modèles « boîte-noire »
 - Modèles « boîte-blanche »
- 3 Algorithme EM pour les HMC
 - Algorithme EM
 - Trois implémentations possible en temps discret
 - HMC en temps continu

Séquence des dates de détection d'une défaillance : $\{T_n\}_{n \geq 1}$

Après chaque défaillance :

- ▶ correction ou non
- ▶ puis relance avec durées de corrections considérées comme négligeable ou pas pris en compte



Processus ponctuel

Une suite de variables aléatoires positives $\{T_n\}_{n \geq 1}$, toutes définies sur $(\Omega, \mathcal{A}, \mathcal{P})$, telle que

- ① $T_1 > 0$,
- ② $T_n < T_{n+1}$ si $T_n < +\infty$ et $T_{n+1} = T_n$ si $T_n = +\infty$.

La suite $\{X_n\}_{n \geq 1}$ des durées d'inter-défaillances

$$X_{n+1} := \begin{cases} T_{n+1} - T_n & \text{si } T_n < +\infty \\ 0 & \text{si } T_n = +\infty. \end{cases}$$

Le processus de comptage des défaillances $\{N_t\}_{t \geq 0}$

$$N_0 := 0 \quad \text{et } t > 0 \quad N_t = \text{card}\{n \geq 1 \mid T_n \leq t\} := \sum_{n=1}^{+\infty} \mathbf{1}_{\{T_n \leq t\}}.$$

Spécifier la loi du processus $\{T_n\}_{n \geq 1}$, c'est spécifier ses lois de dimension finie : $(T_1, \dots, T_n)$ pour $n \geq 1$

- ☛ **Construction « récursive » de la séquence $\{T_n\}_{n \geq 1}$: spécifier la loi de T_n à partir du passé, ici $(T_1, \dots, T_{n-1})$**

$$f_{T_1, \dots, T_n}(t_1, \dots, t_n) = f_{T_1}(t_1) \prod_{k=2}^n f_{T_k | \mathbf{T}_1^{k-1} = \mathbf{t}_1^{k-1}}(t_k)$$

$$T_k = T_{k-1} + X_k : \text{pour } t \geq t_{k-1},$$

$$\mathbb{P}\{T_k \leq t \mid \mathbf{T}_1^{k-1} = \mathbf{t}_1^{k-1}\} = \mathbb{P}\{X_k \leq t - t_{k-1} \mid \mathbf{T}_1^{k-1} = \mathbf{t}_1^{k-1}\}$$

Une construction possible $\equiv$ une famille de taux de hasard des lois de X_k conditionnellement aux $\mathbf{T}_1^{k-1}$ pour $k \geq 1$:

$$r_{X_k | \mathbf{T}_1^{k-1} = \mathbf{t}_1^{k-1}}(x) = \frac{f_{X_k | \mathbf{T}_1^{k-1} = \mathbf{t}_1^{k-1}}(x)}{\mathbb{P}\{X_k \geq x \mid \mathbf{T}_1^{k-1} = \mathbf{t}_1^{k-1}\}} \quad x \geq 0$$

On parle alors d'un modèle de défaillance auto-excité

Intensité stochastique d'un modèle auto-excité

$\lambda_0 = r_{X_1}(0)$ et pour tout $t > 0$, par

$$\begin{aligned} \lambda_t &= \sum_{n \geq 0} r_{X_{n+1} | \mathbf{T}_1^n}(t - T_n) \mathbf{1}_{\{T_n < t \leq T_{n+1}\}} \\ &= \sum_{n \geq 0} r_{X_{n+1} | \mathbf{T}_1^n}(t - T_n) \mathbf{1}_{\{N_{t-} = n\}}. \end{aligned}$$

☛ $\lambda_t(N_{t-}, T_{N_{t-}}, \dots, T_1)$ ou $\lambda_t(\mathcal{H}_{t-}^N)$. avec $\mathcal{H}_{t-}^N := \sigma(N_s, s < t)$

Modèle DIDE

$$\forall k \geq 1, r_{X_k | T_1^{k-1} = t_1^{k-1}}(x) = \lambda_k$$

$$\lambda_t = \lambda_{N_{t-}+1}$$

alors

- ❶ $\{X_n\}_{n \geq 1}$ sont indépendants et $X_n \sim \mathcal{E}(\lambda_n)$
- ❷ la séquence $\{\lambda_n\}_{n \geq 1}$ donne un moyen de gérer la **qualité de la correction**
 - ▶ $\lambda_{n+1} < \lambda_n$: croissance de fiabilité
 - ▶ $\lambda_{n+1} = \lambda_n$: stabilité
 - ▶ $\lambda_{n+1} > \lambda_n$: dégradation
- ❸ $R_t(\tau) = \mathbb{P}\{N_{t+\tau} - N_t \mid \mathcal{H}_t^N\} = \exp(-\lambda_{N_t+1} \tau)$ et
 $\text{MTTF}_t = \mathbb{E}[T_{N_t+1} - t \mid \mathcal{H}_t^N] = \frac{1}{\lambda_{N_t+1}}$

Exemples : Modèle de Jelinski-Moranda, modèle géométrique de Moranda

Modèles conditionnellement DIDE

$$\mathcal{H}_t = \mathcal{H}_t^N \longrightarrow \mathcal{H}_t = \mathcal{H}_t^N \vee \mathcal{F}_0$$

$$\lambda_t(\mathcal{H}_t) \equiv h(N_{t-}, \mathcal{F}_0)$$

- 1 Analyse Bayésienne des modèles de Jelinski-Moranda, de Moranda :

$$\mathcal{F}_0 = \{N, \phi\} \quad \mathcal{F}_0 = \{N, \rho\}$$

- 2 Exemple fondamental : la suite $\{\lambda_n\}_{n \geq 1}$ est une suite de v.a.

$$\mathcal{F}_0 = \{\Lambda_n\}_{n \geq 1}$$

- 1 Littlewood [Lit81], Littlewood et Verrall [LV73]
- 2 $\{\Lambda_n\}_{n \geq 1}$ est une chaîne de Markov : **Modèle Markovien caché**

$(X_n, \Lambda_n)_{n \geq 1}$ chaîne de Markov cachée

- ❶ $\{\Lambda_n\}_{n \geq 1}$ est une chaîne de Markov homogène ;
- ❷ les durées d'inter-défaillance $\{X_n\}_{n \geq 1}$ sont indépendantes conditionnellement à $\{\Lambda_n\}_{n \geq 1}$;
- ❸ $\mathcal{L}(X_k \mid \Lambda_n, n \geq 1)$ ne dépend que Λ_k pour $k \geq 1$

- ▶ $\{X_n\}_{n \geq 1}$ le processus observé, $\{\Lambda_n\}_{n \geq 1}$ le processus état.
- ▶ Une conséquence de $\{\Lambda_n\}_{n \geq 1}$ Markov (homogène) :

$$\mathbb{E}[\Lambda_{n+1} \mid \Lambda_1^n] = \mathbb{E}[\Lambda_{n+1} \mid \Lambda_n] = f(\Lambda_n).$$

$f \equiv$ efficacité moyenne de la correction à la $n^{\text{ème}}$ défaillance.

- ▶ $f(x) < x$: la correction apporte une croissance de fiabilité ;
- ▶ $f(x) > x$: la correction entraîne une décroissance de fiabilité ;
- ▶ $f(x) = x$: on a stabilité.

Justification théorique [Sol88, GS92].

Dans un profil opérationnel Poissonnien homogène avec correction immédiate, il existe une chaîne de Markov $\{\Lambda_n\}_{n \geq 1}$, tel que, conditionnellement à $\{\Lambda_n = \lambda_n\}_{n \geq 1}$, les durées inter-défaillances X_n sont indépendantes et de lois exponentielles $\mathcal{E}(\lambda_n)$ respectivement.

- ▶ Modèle de Basu et Ebrahimi [BE03] : $f(x) = x$
- ▶ Modèle proportionnel lognormale [GLS94] : $f(x) = x^c$
- ▶ Modèle de correction imparfaite [Gau99] :
$$f(x) = (1 - \alpha - \beta)x + \mu\beta$$
- ▶ Modèle de Durand et Gaudoin [DG05] : $\{\Lambda_n\}_{n \geq 1}$ est une chaîne de Markov à espace d'état fini :
$$f(x) = \sum_{\lambda \in E} \lambda P(x, \lambda)$$

Modèle de type Cheung [Che80]

Un logiciel avec un nombre fini de modules $\{e_1, \dots, e_m\}$:

- ▶ Un graphe de contrôle du logiciel est supposé être une chaîne de Markov $\{Z_n\}_{n \geq 0}$: $Z_n :=$ module exécuté à la date t
- ▶ Modèle de défaillance : quand le module e_i est actif :
 - ▶ une défaillance apparaît avec probabilité p_i ;
 - ▶ quand une défaillance survient dans e_i , le contrôle est redirigé vers le module e_j avec probabilité $\alpha(i, j)$
- ▶ Le SRM est la combinaison des deux modèles précédents.
- ▶ Le délai pour recouvrer un état opérationnel est négligé ;

X_n le module actif à n dans ce modèle et

N_n le nombre de défaillances à n et $Y_n := N_n - N_{n-1}$

Alors $\{(Y_n, X_n)\}_{n \geq 0}$ est une « chaîne de Markov cachée »

C'est à dire, $\{(Y_n, X_n)\}_{n \geq 0}$ est une chaîne de Markov homogène d'espace d'état $\{0, 1\} \times \{e_1, \dots, e_m\}$ telle que

$$\begin{aligned} & \mathbb{P}\{ (Y_n, X_n) = (k, e_j) \mid (Y_{n-1}, X_{n-1}) = (l, e_i) \} \\ &= \mathbb{P}\{ (Y_n, X_n) = (k, e_j) \mid X_{n-1} = e_i \} \\ &=: D_k(i, j) \end{aligned}$$

où

$$D_0(i, j) = (1 - p_i)P(i, j) \quad \text{and} \quad D_1(i, j) = p_i \alpha(i, j)$$

- La chaîne $\{X_n\}_{n \geq 0}$ admet $P = D_0 + D_1$ comme matrice de transition

Considérons une HMC $\{(Y_n, X_n)\}_{n \geq 0}$:

- ▶ $L(\theta; Y)$: la *vraisemblance des données observées* pour un paramètre θ sachant les observations $Y_0, \dots, Y_n$
Très souvent, $L(\theta; Y)$ est fortement non-linéaire en θ et est difficile à maximiser
- ▶ Soit $L^c(\theta; Y, X)$ la *vraisemblance des données complètes* pour le paramètre, incluant également les données manquantes.
Cette fonction est souvent explicite et facile à manipuler comme fonction de θ .

Pour θ une estimation du paramètre, on définit

$$(\text{étape-E}) \quad Q(\theta^* \mid \theta) = \mathbb{E}_\theta [\log L^c(\theta^*; Y, X) \mid Y_0, \dots, Y_n],$$

Alors

$$(\text{étape-M}) \quad \hat{\theta} = \arg \max_{\theta^*} Q(\theta^* \mid \theta)$$

fait **croître la vraisemblance des données observées**

$$L(\hat{\theta}; Y) \geq L(\theta; Y)$$

Cette procédure est itérée.

Formules pour le modèle de type Cheung

$$\theta := \{D_0(i, j), D_1(i, j), \quad i, j = 1, \dots, n\}$$

satisfaisant, pour $i = 1, \dots, m$, $\sum_{k=0}^1 \sum_{j=1}^m D_k(i, j) = 1$.

- 1 Initialisation : une estimation initiale θ_0
- 2 Posons $\theta := \theta_m$

$$\widehat{D}_k(i, j) := \frac{\widehat{\mathcal{L}}_n^{k, ij}}{\widehat{\mathcal{O}}_n^{(i)}}, \quad i, j = 1, \dots, m$$

$$(2) \quad \widehat{\mathcal{L}}_n^{k, ij} := \mathbb{E}_\theta \left[\sum_{l=1}^n 1_{\{X_l = e_j, Y_l = k, X_{l-1} = e_i\}} \mid \mathbb{F}_n^Y \right] \quad k = 0, 1$$

$$(3) \quad \widehat{\mathcal{O}}_n^{(i)} := \mathbb{E}_\theta \left[\sum_{l=1}^n 1_{\{X_{l-1} = e_i\}} \mid \mathbb{F}_n^Y \right].$$

- 3 Retourner en 2 jà satisfaction d'un critère d'arrêt.

Les formules (2,3) peuvent être considérées comme

- 1 soit des fonctionnelles des espérances conditionnelles des statistiques de base

$$1_{\{X_l=e_i\}}, \quad 1_{\{X_{l-1}=e_i, X_l=e_j, Y_l=k\}}$$

☛ **Lissage** : $\mathbb{P}\{X_l = e_i \mid \mathbb{F}_n^Y\}, l = 0, \dots, n-1.$

- ▶ Approche avant-arrière ou de Baum-Welch
- ▶ Lissage récursif

- 2 soit des espérances conditionnelles des fonctionnelles additives de chaînes de Markov

$$\mathcal{L}_n^{k,ij} := \sum_{l=1}^n 1_{\{X_{l-1}=e_i, X_l=e_j, Y_l=k\}}, \quad \mathcal{O}_n^{(i)} := \sum_{l=0}^{n-1} 1_{\{X_l=e_i\}}$$

☛ **Filtrage récursif**

$$\mathbb{P}\{X_l = e_i \mid \mathbb{F}_n^Y\} = \frac{\hat{X}_l(i)\beta_{l|n}(i)}{\sum_{i=1}^n \hat{X}_l(i)\beta_{l|n}(i)}.$$

où les vecteurs $\beta_{l|n}$ et $\hat{X}_l$ sont calculés à partir des deux passes récursives bien connues :

Forme Avant-Arrière d'EM [Rab89]

Filtrage avant. $c_0 := 1$ and $\hat{X}_0 := x_0$
for $l = 1, \dots, n$: $c_l := \hat{X}_{l-1} D_{Y_l} \mathbf{1}^\top$
 $\hat{X}_l := \hat{X}_{l-1} D_{Y_l} / c_l$

Lissage arrière. $\beta_{n|n} := \mathbf{1}/c_n$
for $l = n - 1, \dots, 0$: $\beta_{l|n} := \beta_{l+1|n} D_{Y_l}^\top / c_l$

Pour chaque itération :

le coût de calcul est linéaire en le nombre d'observations n , quadratique en le nombre d'états cachés et

le coût de stockage est linéaire en le nombre n d'observations.

Lissage récursif pour l'état [CMR05]

Notations identique à celles de l'algorithme avant-arrière.

On a pour $l < n + 1$,

$$\mathbb{P}\{X_l = e_i \mid \mathbb{F}_{n+1}^Y\} = \sum_{j=1}^m \gamma_{l,n+1|n+1}(i, j)$$

où $\gamma_{l,n+1|n+1}(i, j) := \mathbb{P}\{X_l = e_i, X_{n+1} = e_j \mid \mathbb{F}_{n+1}^Y\}$.

La matrice $\gamma_{l,n+1|n+1}$ satisfait l'équation récursive suivante

$$\gamma_{l,n+1|n+1} = \gamma_{l,n|n} D_{Y_{n+1}} / c_{n+1}.$$

avec $\gamma_{n,n|n} := \text{diag}(\hat{X}_n)$ for $n \geq 0$.

Filtrage récursif [Led05]

x_0 est distribution de X_0

$e_i, i = 1, \dots, m$ est le $i^{\text{ème}}$ vecteur de la base canonique de $\mathbb{R}^m$.

① $\widehat{\mathcal{O}^{(i)}X_0} = x_0(i)e_i$ et pour $n \in \mathbb{N}$

$$\widehat{\mathcal{O}^{(i)}X_{n+1}} = \frac{\widehat{\mathcal{O}^{(i)}X_n} D_{Y_{n+1}} + \widehat{X}_n(i) e_i D_{Y_{n+1}}}{\widehat{X}_n D_{Y_{n+1}} \mathbf{1}^\top}.$$

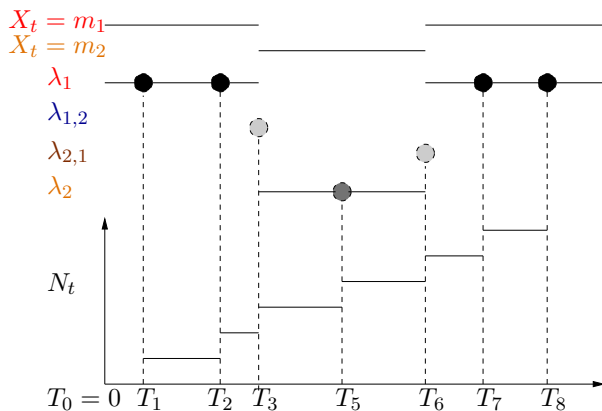
② $\widehat{\mathcal{L}^{k,ij}X_0} = 0$ et pour $n \in \mathbb{N}$

$$\widehat{\mathcal{L}^{k,ij}X_{n+1}} = \frac{\widehat{\mathcal{L}^{k,ij}X_n} D_{Y_{n+1}}}{\widehat{X}_n D_{Y_{n+1}} \mathbf{1}^\top} + \frac{D_k(i, j) \widehat{X}_n(i)}{\widehat{X}_n D_k \mathbf{1}^\top} \mathbf{1}_{\{Y_{n+1}=k\}} e_j.$$

Notons que $\widehat{\mathcal{O}^{(i)}}_n = \widehat{\mathcal{O}^{(i)}X_n} \mathbf{1}^\top$ et $\widehat{\mathcal{L}^{k,ij}}_n = \widehat{\mathcal{L}^{k,ij}X_n} \mathbf{1}^\top$.

À chaque exécution : le coût de calcul est linéaire en le nombre d'observations et d'ordre 4 en le nombre de paramètres

Le coût de stockage ne dépend pas du nombre d'observations



Littlewood [Lit75] $\begin{cases} D_0(i, j) = Q(i, j)(1 - \lambda_{i,j}) \\ D_0(i, i) = -\sum_{j \neq i} Q(i, j) - \lambda_i \end{cases} \quad \begin{cases} D_1(i, j) = Q(i, j)\lambda_{i,j} \\ D_1(i, i) = \lambda_i \end{cases}$

MMPP : $D_0 = Q - \text{diag}(\lambda_i) \quad D_1 = \text{diag}(\lambda_i)$

Modèle de Littlewood

Un processus de Markov bivarié $\{(N_t, X_t)\}_{t \geq 0}$ où

- ▶ $\{N_t\}_{t \geq 0}$ est le processus de comptage des défaillances et
- ▶ $\{X_t\}_{t \geq 0}$ est un modèle Markovien d'échanges de contrôle entre les modules du logiciel.

$\{N_t\}_{t \geq 0}$ est un PAI conditionnellement à $\{X_t\}_{t \geq 0}$.

Le modèle est paramétré par le vecteur

$$\theta = \{D_k(i, j), \quad k = 0, 1 \quad i, j = 1, \dots, n\}$$

$$\mathcal{L}(\theta; t_1, \dots, t_k)$$

$$= \alpha \exp(D_0 t_1) D_1 \exp(D_0(t_2 - t_1)) \cdots \exp(D_0(t_k - t_{k-1})) D_1 \mathbf{1}^\top.$$

avec $\alpha = (\mathbb{P}\{X_0 = e_i\})_{i=1, \dots, n}$.

Algorithme EM

Formules pour (re-)estimer θ à partir de $\mathbb{F}_t^N := \sigma(N_s, s \leq t)$:

- 1 Initialisation : estimation a priori de θ_0
- 2 Poser $\theta := \theta_m$

$$\hat{D}_1(i, j) = \frac{\mathbb{E}_\theta \left[\sum_{0 < s \leq t} 1_{\{X_{s-}=e_i, X_s=e_j\}} \Delta N_s \mid \mathbb{F}_t^N \right]}{\mathbb{E}_\theta \left[\int_0^t 1_{\{X_s=e_i\}} ds \mid \mathbb{F}_t^N \right]}$$

$$\hat{D}_0(i, j) = \frac{\mathbb{E}_\theta \left[\sum_{0 < s \leq t} 1_{\{X_{s-}=e_i, X_s=e_j\}} (1 - \Delta N_s) \mid \mathbb{F}_t^N \right]}{\mathbb{E}_\theta \left[\int_0^t 1_{\{X_s=e_i\}} ds \mid \mathbb{F}_t^N \right]}$$

$i \neq j$

- 3 Retourner en 2 jusqu'à satisfaction d'un critère d'arrêt

Approche Forward-backward : ok [Ryd96, KLL03, RED06]

$$\widehat{\mathcal{O}}^{(i)}_t = \int_0^t \mathbb{P}\{X_s = e_i \mid \mathcal{H}_t^N\} ds$$

$$\widehat{\mathcal{L}}_t^{0,ij} = \int_0^t \mathbb{P}\{\Delta N_s = 0, X_{s-} = e_i, X_s = e_j \mid \mathcal{H}_t^N\} ds$$

$$\widehat{\mathcal{L}}_t^{1,ij} = \int_0^t \mathbb{P}\{\Delta N_s = 1, X_{s-} = e_i, X_t = e_j \mid \mathcal{H}_t^N\} ds$$

en rappelant que $t = t_k$.

$f_0(x) = \exp(D_0x)$ et $f_1(x) = \exp(D_0x)D_1$
pour $l = 1, \dots, k$ $\Delta t_l := t_l - t_{l-1}$ avec $t_0 := 0$.

L'étape E prend la forme :

Récurrence Avant. $\alpha_0 := \alpha$ et pour $l = 1, \dots, k$ $\alpha_l = \alpha_{l-1}f_1(\Delta t_l)$, $c_l := \alpha_l \mathbf{1}^\top$

Récurrence Arrière. $\beta_{k+1}^\top = \mathbf{1}^\top$ et pour $l = k, \dots, 1$ $\beta_l^\top = f_1(\Delta t_l)\beta_{l+1}^\top$

pour $i, j = 1, \dots, n$:

$$L_0^{0,ij} = 0, L_0^{1,ij} = 0$$

pour $l = 1, \dots, k$:

$$L_l^{0,ij} = L_{l-1}^{0,ij} + \alpha_{l-1} \int_{t_{l-1}}^{t_l} f_0(t - t_{l-1}) \mathbf{e}_i^\top D_0(i, j) \mathbf{e}_j f_1(t_l - t) dt \beta_{l+1}^\top$$

$$L_l^{1,ij} = L_{l-1}^{1,ij} + \alpha_{l-1} f_0(\Delta t_l) \mathbf{e}_i^\top D_1(i, j) \mathbf{e}_j \beta_{l+1}^\top$$

$$O_l^{(i)} = O_{l-1}^{(i)} + \alpha_{l-1} \int_{t_{l-1}}^{t_l} f_0(t - t_{l-1}) \mathbf{e}_i^\top \mathbf{e}_j f_1(t_l - t) dt \beta_{l+1}^\top.$$

L'étape M est : $D_1(i, j) = \frac{L_k^{0,ij}}{O_k^{(i)}}$; $i \neq j$, $D_0(i, j) = \frac{L_k^{0,ij}}{O_k^{(i)}}$ et
 $D_0(i, i) = \sum_{j \neq i} D_0(i, j) + \sum_j D_1(i, j)$.

$$D_0 := \begin{pmatrix} -228 & 1 & 1 \\ 3 & -248 & 2 \\ 1 & 2 & -288 \end{pmatrix} \quad D_1 := \begin{pmatrix} 203 & 12 & 11 \\ 30 & 180 & 33 \\ 43 & 75 & 167 \end{pmatrix}$$

Modèle simulé sur 200000 arrivées

$$\hat{D}_0 = \begin{pmatrix} -227.69 & 1.02 & 1.2 \\ 3.24 & -243.71 & 1.9 \\ .97 & 1.95 & -286.34 \end{pmatrix} \quad \hat{D}_1 = \begin{pmatrix} 202.16 & 11.88 & 11.43 \\ 28.13 & 174.28 & 36.16 \\ 41.52 & 69.42 & 172.48 \end{pmatrix}$$

Modèle estimé avec 100 itérations

Filtrage récursif : ok [Led07]

À partir d'EDS, l'étape **E** :

$f_0(x) := \exp(D_0 x)$ et $f_1(x) := D_1 \exp(D_0 x)$; pour $l = 1, \dots, K$, $\Delta t_l := t_l - t_{l-1}$
avec $t_0 := 0$.

pour $l = 1, \dots, K$

$$\begin{aligned}\sigma(X_{t_l}) &= \sigma(X_{t_{l-1}}) f_1(\Delta t_l) \\ \sigma(\mathcal{L}_{t_l}^{0,ij} X_{t_l}) &= \sigma(\mathcal{L}_{t_{l-1}}^{0,ij} X_{t_{l-1}}) f_1(\Delta t_l) \\ &\quad + \sigma(X_{t_{l-1}}) \int_{t_{l-1}}^{t_l} f_0(s - t_{l-1}) e_i^\top D_0(i, j) e_j f_1(t_l - s) ds \\ \sigma(\mathcal{L}_{t_l}^{1,ij} X_{t_l}) &= \sigma(\mathcal{L}_{t_{l-1}}^{1,ij} X_{t_{l-1}}) f_1(\Delta t_l) + \sigma(X_{t_{l-1}}) f_0(\Delta t_l) e_i^\top D_1(i, j) e_j \\ \sigma(\mathcal{O}_{t_l}^{(i)} X_{t_l}) &= \sigma(\mathcal{O}_{t_{l-1}}^{(i)} X_{t_{l-1}}) f_1(\Delta t_l) + \sigma(X_{t_{l-1}}) \int_{t_{l-1}}^{t_l} f_0(s - t_{l-1}) e_i^\top e_i f_1(t_l - s) ds\end{aligned}$$

Le facteur $\exp(\Delta t_l)$ est omis dans les équations ci-dessus car les estimations à une date donnée de D_0, D_1 ne requiert que la connaissance des filtres qu'à une constante multiplicative près.

Lissage récursif :

Ok pour MMPPs [EM05] mais c'est un pb ouvert pour des HMC générales.



S. Basu and N. Ebrahimi.

Bayesian software reliability models based on martingale processes.

Technmetrics, 45 :150–158, 2003.



R. C. Cheung.

A user-oriented software reliability model.

IEEE Trans. Software Eng., 6 :118–125, 1980.



O. Cappé, E. Moulines, and T. Rydén.

Inference in Hidden Markov Models.

Springer, 2005.



J.-B. Durand and O. Gaudoin.

Software reliability modelling and prediction with hidden Markov chains.

Statist. Modelling, 5 :75–93, 2005.



R. J. Elliott and W. P. Malcolm.

General smoothing formulas for Markov-modulated poisson observations.

IEEE Trans. Automat. Control, 50(8) :1123–1134, 2005.



O. Gaudoin.

Software reliability models with two debugging rates.

Int. J. Reliab. Quality Safety, 6 :31–42, 1999.



O. Gaudoin, C. Lavergne, and J.-L. Soler.

A generalized geometric de-eutrophication software-reliability model.

IEEE Trans. Reliab., 43 :536–541, 1994.



O. Gaudoin and J.-L. Soler.

Modèles pour l'étude de la fiabilité des systèmes présentant des fautes de conception. application à l'évolution de la fiabilité des logiciels.

Revue de Statistique Appliquée, 40 :91–98, 1992.



A. Klemm, C. Lindemann, and M. Lohmann.

Modeling IP traffic using the Batch Markovian Arrival Process.

Performance Evaluation, 54 :149–173, 2003.



J. Ledoux.

Recursive filters for partially observable finite Markov chains.

J. Appl. Probab., 42(3) :687–697, 2005.



J. Ledoux.

Filtering and the EM-algorithm for the Markovian Arrival Process.

Comm. Statist. Theory Methods, 2007.

To appear.



B. Littlewood.

A reliability model for systems with Markov structure.

J. Roy. Statist. Soc. Ser. C, 24 :172–177, 1975.



B. Littlewood.

Stochastic reliability-growth : a model for fault-removal in computer programs and hardware designs.

IEEE Trans. Reliab., 30 :313–320, 1981.



B. Littlewood and J. L. Verrall.

A bayesian reliability growth model for computer software.

J. Roy. Statist. Soc., Serie C, pages 332–346, 1973.



L. R. Rabiner.

A tutorial on hidden Markov models and selected applications speech recognition.

Proceedings of the IEEE, 77 :257–286, 1989.



W. J. J. Roberts, Y. Ephraim, and E. Dieguez.

On Rydén's EM algorithm for estimating MMPPs.

IEEE Signal Process. Letters, 13 :373–376, 2006.



T. Rydén.

An EM-algorithm for estimation in Markov-modulated Poisson process.

Computational Statistics and Data Analysis, 21 :431–447, 1996.



J. L. Soler.

Modélisation des processus de risque, de défaillance et de correction des systèmes présentant des fautes de conception. application à la fiabilité des logiciels.

In *6ème Colloque National de Fiabilité et Maintainabilité*, $\lambda\mu 6$, pages 647–650, Strasbourg, 1988.