

Prévision à partir des minimums locaux d'une fonction objectif

A. Dermoune, D. Ounaissi, Y. Slaoui

Université de Lille, L'école Supérieure des agricultures d'Angers, Université de Poitiers

$$f(T, \theta) = \frac{\sum_{t=1}^T c(m(t, \theta), y(t))}{T},$$

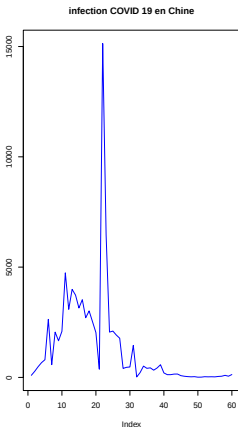
où $y(t)$ désigne l'observation à la date t et $m(t, \theta)$ est le modèle qui dépend du paramètre θ , c mesure la perte entre l'observation $y(t)$ et le modèle $m(t, \theta)$. L'estimation de θ nécessite la minimisation de la fonction $\theta \in \Theta \rightarrow f(T, \theta)$. Ayant trouvé ce minimum $\hat{\theta}$ on estime les valeurs $y(t)$ par $m(t, \hat{\theta})$ dans le futur $t > T$.

- Le minimum global est-il toujours le bon choix?
- Dans le cas de plusieurs minimums locaux, comment peut-on les utiliser dans la prévision?
- Chaque algorithme de minimisation nécessite une initialisation et un critère d'arrêt. Comment peut-on choisir les initialisations pour explorer tous les minimums?

Nous allons illustrer ces questions à travers les données du COVID 19 en Chine.

Exemple de données

Le nombre quotidien $I(t)$ des infections par le COVID 19 en Chine durant la première vague.



Valeurs observées du pic à la fin de la première vague

Le pic a lieu à la date $I_{pic} = 22$. Son amplitude est égal à $a_{pic} = 15136$.

Modélisation par une fonction sigmoïde

$$G(t, a, l, s) = a \exp\left(-\frac{(l - t)^2}{s^2}\right), t > 0,$$

- a est l'amplitude (le pic de la pandémie).
- l le jour du pic de la pandémie.
- s l'étendue de la pandémie.
- Les paramètres du modèle sont (a, l, s) .

$$f(T, a, l, s) = \frac{\sum_{t=1}^T |G(t, a, l, s) - l(t)|}{T}.$$

$$\arg \min \{f(T, a, l, s) : a, l, s\}.$$

Interprétation probabiliste: La loi de Laplace symétrique sur les erreurs

Si on suppose que

$$I(t) = G(t, a, l, s) + e(t),$$

où les erreurs ($e(t)$) sont i.i.d. avec la densité commune

$$\frac{1}{2\lambda} \exp\left(-\frac{|e|}{\lambda}\right).$$

Dans ce cas la vraisemblance du modèle est

$$\frac{1}{(2\lambda)^T} \exp\left(-\frac{1}{\lambda} \sum_{t=1}^T |G(t, a, l, s) - I(t)|\right).$$

Elle dépend des paramètres $((a, l, s), \lambda)$. L'estimateur du maximum de vraisemblance

$$\begin{aligned}(\hat{a}, \hat{l}, \hat{s}) &= \arg \min \{f(T, a, l, s) : a, l, s\}, \\ \hat{\lambda} &= \min \{f(T, a, l, s) : a, l, s\}.\end{aligned}$$

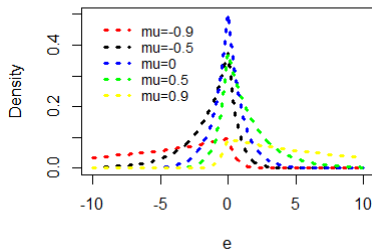
La loi de Laplace "déséquilibrée" sur les erreurs: Dermoune, Ounaissi, Rahmania (2015)

$$\frac{(1 - \mu^2)}{2\lambda} \exp\left(-\frac{|e|}{\lambda}(1 - \operatorname{sgn}(e)\mu)\right)$$

avec $\lambda > 0$, $\mu \in (-1, 1)$, dans ce cas la fonction objective devient

$$\sum_{t=1}^T |G(t, a, l, s) - I(t)|(1 - \operatorname{sgn}(G(t, a, l, s) - I(t))\mu).$$

La loi de Laplace "déséquilibrée"



Intervalle de confiance: Erreurs de Laplace

L'intervalle de confiance asymétrique $[-q, q]$ de la loi de Laplace de paramètre λ de niveau 0.95 est donné par l'équation

$$\int_{-q}^q \frac{1}{2\lambda} \exp\left(-\frac{|e|}{\lambda}\right) de = 0.95.$$

On obtient $q = -\lambda \ln(0.05) = 2.995732\lambda$.

On en déduit l'intervalle de confiance

$$\hat{a} \exp\left(-\frac{(t - \hat{l})^2}{(\hat{s})^2}\right) \pm 2.995732\hat{\lambda},$$

de $I(t)$ de niveau de confiance 0.95 avec $t > T$.

Les algorithmes d'optimisation sont groupés en ceux qui utilisent les dérivées de la fonction objectif (algorithmes d'ordre 1,2,...) et ceux qui n'utilisent pas les dérivées (algorithmes d'ordre 0) Tous dépendent de la condition initiale et d'un critère d'arrêt.

- Les algorithmes classiques utilisent les dérivées premières ou secondes (la descente du gradient, Newton, ...).
- Les algorithmes d'optimisation directs (e.g. algorithme du simplexe), algorithmes stochastiques (e.g. recuit simulé) n'utilisent pas les dérivées.

La minimisation en utilisant l'algorithme du simplexe de Nelder-Mead (1965)

La méthode de minimisation de Nelder-Mead est basée sur la comparaison des valeurs de la fonction dans les $p + 1$ sommets $s_1, \dots, s_{p+1}$ d'un simplexe général. Le simplexe se modifie à travers les opérations de réflexion, d'expansion et de contraction jusqu'à la réduction à un point.

- 1) Le premier sommet s_1 du simplexe initial (point initial) est fourni par l'utilisateur, $s_i = s_1 + b_i$ avec (b_i) désigne la base canonique de $\mathbb{R}^p$. On obtient le simplexe $(s_1, \dots, s_{p+1})$ dans $\mathbb{R}^p$,
- 2) Calcul des valeurs $(f(s_i))$ puis on les ré-indexe de sorte que

$$f(s_1) \leq \dots \leq f(s_{p+1}),$$

- 3) Calcul du centre de gravité $s_0 = \frac{\sum_{i=1}^p s_i}{p}$ de tous les points sauf s_{p+1} ,

La minimisation en utilisant l'algorithme du simplexe de Nelder-Mead (1965)

- 4) Calcul de la réflexion $s_r = s_0 + (s_0 - s_{p+1})$ du point s_{p+1} par rapport au centre s_0 ,
- 5) Si $f(s_r) < f(s_p)$, on calcul $s_e = s_0 + 2(s_0 - s_{p+1})$ (étirement du simplexe): si $f(s_e) < f(s_r)$ on remplace s_{p+1} par s_e , sinon on remplace s_{p+1} par s_r puis retour à l'étape 2).
- 6) Si $f(s_p) < f(s_r)$, calcul de $s_c = s_{p+1} + \frac{(s_0 - s_{p+1})}{2}$ (contraction du simplexe). Si $f(s_c) < f(s_p)$, remplacement de s_{p+1} par s_c et retour à l'étape 2).
- 7) Si $f(s_c) \geq f(s_p)$, similitude de rapport 1/2 et de centre s_1 : remplacement de s_i par $s_1 + \frac{s_i - s_1}{2}$ et retour à 2).

L'algorithme du simplexe de Nelder-Mead (1965)

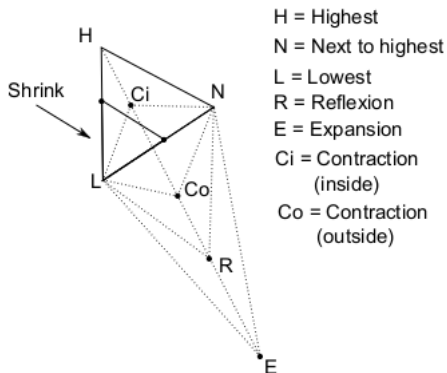


Figure 4.3: Nelder-Mead simplex steps

La longueur orienté du simplexe $S = S(s_1, \dots, s_{p+1})$ est définie par

$$\max(\|s_k - s_{p+1}\| : k = 1, \dots, p) = \sigma(S).$$

Critères d'arrêt:

1) avec une tolérance Tol : $\sigma(S) \leq Tol$,

2) avec les valeurs de f : $\frac{\sum_{i=1}^{p+1} |f(s_i) - \bar{f}|^2}{p+1} \leq \varepsilon$,

.....

La fonction `optim()`

L'algorithme du simplexe de Nelder-Mead est implémentée sous R avec la fonction `optim(init,f)`. Il est bien connu qu'en général la fonction `optim` ne produit pas un vrai minimum. Mais si on l'itère plusieurs fois alors on augmente la chance de la convergence vers un minimum.

Dans notre cas on propose l'itération de la fonction optim jusqu'à la stabilisation:

$$init = optim(init).$$

Réduction de la minimisation à deux variables: médiane avec poids

On utilise la médiane avec poids comme suit:

$$\sum_{t=1}^T |a \exp(-\frac{(t-l)^2}{s^2}) - I(t)| =$$
$$\sum_{t=1}^T \exp(-\frac{(t-l)^2}{s^2}) |a - \exp(\frac{(t-l)^2}{s^2}) I(t)|,$$

pour l, s fixé le minimum comme fonction du paramètre a est égal à la médiane $a(T, l, s)$ de la suite $(\exp(\frac{(t-l)^2}{s^2}) I(t), t = 1, \dots, T)$ avec les poids $(\exp(-\frac{(t-l)^2}{s^2}), t = 1, \dots, T)$.

Let us consider a sequence $(x(t), w(t))$ of real numbers with positive weights $w(t) > 0$ and $t = 1, \dots, n$. The minimizer of the function $a \rightarrow \sum_{t=1}^n w(t)|a - x(t)|$ (called the weighted median) is given as follows. We calculate the permutation $p(1), \dots, p(n)$ which rearranges the sequence $(x(t) : t = 1, \dots, n)$ into ascending order. We form the sequence $(w(p(t)) : t = 1, \dots, n)$, then we find the largest integer t^* which satisfies

$$\sum_{t=1}^{t^*} w(p(t)) \leq \frac{\sum_{t=1}^n w(t)}{2}.$$

If

$$\sum_{t=1}^{t^*} w(p(t)) < \frac{\sum_{t=1}^n w(t)}{2},$$

then weighted median equals $x(p(t^* + 1))$.

If $\sum_{t=1}^{t^*} w(p(t)) = \frac{\sum_{t=1}^n w(t)}{2}$, then the weighted median equals the interval $[x(p(t^*)), x(p(t^* + 1))]$.

Grâce à la fonction `optim` on montre que la surface

$$(l, s) \rightarrow f(T, a(T, l, s), l, s)$$

a un seul minimum.

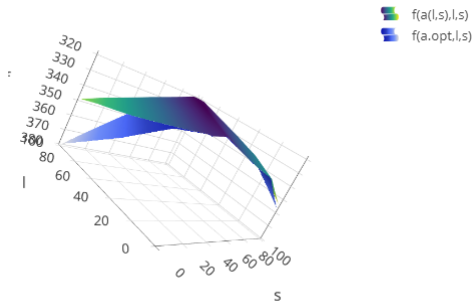
Exploration des minimums à la date T

Pour chaque a fixé la surface

$$(l, s) \rightarrow f(T, a, l, s)$$

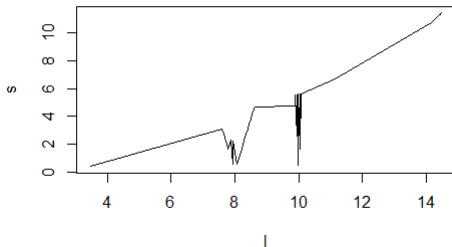
est au dessus de la surface

$$(a, l, s) \rightarrow f(T, a(T, l, s), l, s).$$



Exploration des minimums à la date T

Les deux surfaces s'intersectent le long de la courbe $a = a(T, l, s)$.



Nous allons montrer grâce à la fonction `optim` que la fonction objectif

$$(a, l, s) \rightarrow f(T, a, l, s)$$

a plusieurs minimums.

Initialisation à la date T par tâtonnement cohérent

Le cas $T = 10$, initialisations $l = 11$, $s = 1 : 28$

	$l,1$	$l,2$	$l,3$	$l,4$
[1,]	2585.578	1.372342e+01	-8.062748e+00	324.1095
[2,]	1293.375	5.010983e+08	6.877449e+08	713.3000
[3,]	2089.035	1.002303e+01	5.644370e+00	319.5720
[4,]	2089.035	1.002303e+01	5.644370e+00	319.5720
[5,]	2187.822	1.140679e+01	6.543475e+00	321.8125
[6,]	2089.035	1.002303e+01	5.644371e+00	319.5720
[7,]	2089.035	1.002303e+01	5.644372e+00	319.5720
[8,]	2089.035	1.002303e+01	5.644372e+00	319.5720
[9,]	2089.035	1.002303e+01	5.644371e+00	319.5720
[10,]	2089.035	1.002303e+01	5.644369e+00	319.5720
[11,]	2089.035	1.002303e+01	5.644370e+00	319.5720
[12,]	2089.035	1.002303e+01	5.644370e+00	319.5720
[13,]	2089.035	1.002303e+01	5.644370e+00	319.5720
[14,]	2089.035	1.002303e+01	5.644373e+00	319.5720
[15,]	1148.930	2.159124e+07	3.330589e+07	713.2999
[16,]	1270.095	1.551944e+08	2.286770e+08	713.3000
[17,]	1185.149	4.191666e+08	6.380754e+08	713.3000
[18,]	1144.730	1.885265e+08	2.924252e+08	713.3000
[19,]	1149.235	7.579279e+08	1.175040e+09	713.3000
[20,]	1292.189	5.279987e+08	7.537467e+08	713.3000
[21,]	1373.573	1.707965e+08	2.281007e+08	713.3000
[22,]	1334.005	5.442447e+08	7.260206e+08	713.3000
[23,]	1199.166	4.425709e+08	6.673183e+08	713.3000
[24,]	1215.301	5.369460e+07	8.053014e+07	713.3000
[25,]	1133.424	8.396538e+07	1.319578e+08	713.3000
[26,]	1070.928	2.946326e+08	4.975136e+08	713.3000
[27,]	1362.187	4.414539e+07	6.025008e+07	713.3000
[28,]	2089.035	1.002303e+01	5.644371e+00	319.5720

Le cas $T = 10$, initialisations $l = 11$, $s = 29 : 40$

```
[29,] 2089.035 1.002303e+01 5.644370e+00 319.5720
[30,] 2089.035 1.002303e+01 5.644371e+00 319.5720
[31,] 2089.035 1.002303e+01 5.644370e+00 319.5720
[32,] 2089.035 1.002303e+01 5.644370e+00 319.5720
[33,] 2089.035 1.002303e+01 5.644370e+00 319.5720
[34,] 2089.035 1.002303e+01 5.644370e+00 319.5720
[35,] 2089.035 1.002303e+01 5.644369e+00 319.5720
[36,] 2089.035 1.002303e+01 5.644369e+00 319.5720
[37,] 2089.035 1.002303e+01 5.644370e+00 319.5720
[38,] 2089.035 1.002303e+01 5.644370e+00 319.5720
[39,] 2089.035 1.002303e+01 5.644371e+00 319.5720
[40,] 2131.875 1.091563e+01 6.423846e+00 321.1121
```

Le cas $T = 10$, initialisations $l = 11$, $s = 84 : 100$

```
[83,] 2089.035 1.002303e+01 5.644371e+00 319.5720
[84,] 2721.319 1.434972e+01 8.458729e+00 324.6848
[85,] 2912.913 1.515657e+01 8.943052e+00 325.3661
[86,] 2089.035 1.002303e+01 5.644370e+00 319.5720
[87,] 2089.035 1.002303e+01 5.644370e+00 319.5720
[88,] 2089.035 1.002303e+01 5.644371e+00 319.5720
[89,] 2089.035 1.002303e+01 5.644370e+00 319.5720
[90,] 2089.035 1.002303e+01 5.644370e+00 319.5720
[91,] 2089.035 1.002303e+01 5.644370e+00 319.5720
[92,] 2089.035 1.002303e+01 5.644371e+00 319.5720
[93,] 2089.035 1.002303e+01 5.644370e+00 319.5720
[94,] 2089.035 1.002303e+01 5.644370e+00 319.5720
[95,] 2089.035 1.002303e+01 5.644370e+00 319.5720
[96,] 2089.035 1.002303e+01 5.644371e+00 319.5720
[97,] 2089.035 1.002303e+01 5.644372e+00 319.5720
[98,] 2840.306 1.486001e+01 8.768146e+00 325.1237
[99,] 2089.035 1.002303e+01 5.644370e+00 319.5720
[100,] 2089.035 1.002303e+01 5.644370e+00 319.5720
```

Le cas $T = 10$, initialisations $l = 11 : 30$, $s = 1$

	[,1]	[,2]	[,3]	[,4]
[1,]	2585.578	13.72342	-8.062748	324.1095
[2,]	2089.035	10.02303	5.644370	319.5720
[3,]	2089.035	10.02303	5.644370	319.5720
[4,]	2089.035	10.02303	5.644371	319.5720
[5,]	2089.035	10.02303	5.644371	319.5720
[6,]	2089.035	10.02303	5.644371	319.5720
[7,]	2555.976	13.57933	7.968869	323.9718
[8,]	2089.035	10.02303	5.644370	319.5720
[9,]	2713.546	14.31522	8.437402	324.6541
[10,]	2588.139	13.73574	8.070729	324.1211
[11,]	2089.035	10.02303	5.644371	319.5720
[12,]	2089.035	10.02303	5.644370	319.5720
[13,]	2089.035	10.02303	5.644370	319.5720
[14,]	2089.035	10.02303	5.644370	319.5720
[15,]	2089.035	10.02303	-5.644371	319.5720
[16,]	2089.035	10.02313	5.644438	319.5723
[17,]	2089.035	10.02303	5.644371	319.5720
[18,]	2089.035	10.02303	5.644372	319.5720
[19,]	2721.199	14.34919	8.458403	324.6843

Le cas $T = 10$, initialisations $l = 31 : 58$, $s = 1$

[19,]	2721.199	14.34919	8.458403	324.6843
[20,]	2303.890	12.19099	7.001948	322.6003
[21,]	2089.035	10.02303	5.644370	319.5720
[22,]	2590.206	13.74567	8.077154	324.1305
[23,]	2089.035	10.02303	5.644370	319.5720
[24,]	2089.035	10.02303	5.644370	319.5720
[25,]	2585.358	13.72235	8.062061	324.1084
[26,]	2089.035	10.02303	5.644371	319.5720
[27,]	2089.035	10.02303	5.644370	319.5720
[28,]	2884.627	15.04227	8.876049	325.2738
[29,]	2089.035	10.02303	5.644370	319.5720
[30,]	2089.035	10.02303	5.644371	319.5720
[31,]	2599.713	13.79118	8.106527	324.1735
[32,]	2089.035	10.02303	5.644370	319.5720
[33,]	2852.023	14.90858	8.797031	325.1640
[34,]	2089.035	10.02303	5.644370	319.5720
[35,]	2716.916	14.33020	8.446668	324.6675
[36,]	2727.814	14.37843	8.476437	324.7102
[37,]	2089.035	10.02303	5.644370	319.5720
[38,]	2884.126	15.04023	8.874851	325.2721

Initialisation fonction de $(l(1), \dots, l(T-1))$

- Le minimum global de $f(1, a, l, s)$ est atteint sur la surface

$$a = \exp\left(\frac{(1-l)^2}{s^2}\right)l(1).$$

En partant d'un point a_1, l_1, s_1 de cette surface on obtient un minimum a_2, l_2, s_2 de $f(2, a, l, s)$. Puis on minimise $(a, l, s) \rightarrow f(3, a, l, s)$ en partant de a_2, l_2, s_2 . On itère cet algorithme jusqu'à $(a, l, s) \rightarrow f(T, a, l, s)$.

Matrice des restarts

```
printoptimizer<-function(T,init,K){  
  y<-matrix(0,K,4)  
  for (k in 1:K)  
    {y[k,]<-c(optimizer(T,init,k),f(T,optimizer(T,init,k)))}  
  return(y)  
}
```

Algorithme 1

```
optimiter1T<-function(T,init,k,K){  
  y<-matrix(0,T,3)  
  y[1,]<-init  
  for (j in 1:(T-2)){  
    y[(j+1),]<-optimiter((j+1),y[j,],k)  
  }  
  return(c(optimiter(T,y[(T-1),],K),f(T,optimiter(T,y[(T-1),],K))))  
}
```

```
T=10  
a=i[1]  
l=1  
s=1  
init=c(a,l,s)  
K=20  
optimiter1T(T,init,k,K)  
alg1_sortie<-matrix(0,5,4)  
for (k in 1:5){  
  alg1_sortie[k,]=optimiter1T(T,init,k,K)  
}  
alg1_sortie|
```

Initialisation à partir de $T = 1$ et variation des nombre de restarts

```
> alg1_sortie
      [,1]      [,2]      [,3]      [,4]
[1,] 2089.035 10.02303 -5.644371 319.5720
[2,] 2089.035 10.02303 -5.644371 319.5720
[3,] 2125.639 10.83868 -6.360411 320.9765
[4,] 14526.281 31.63087 15.532969 336.7485
[5,] 14526.281 31.63087 15.532969 336.7485
> |
```

Initialisation à partir de $T = 1$ et variation des nombre de restarts

```
~ optimisation_2  
, , 1
```

	[,1]	[,2]	[,3]	[,4]
[1,]	2089.035	10.02303	-5.644371	319.5720
[2,]	2089.035	10.02303	-5.644371	319.5720
[3,]	2125.639	10.83868	-6.360411	320.9765
[4,]	14526.281	31.63087	15.532969	336.7485
[5,]	14526.281	31.63087	15.532969	336.7485

```
, , 2
```

	[,1]	[,2]	[,3]	[,4]
[1,]	2089.035	10.02303	5.644371	319.5720
[2,]	2089.035	10.02303	-5.644371	319.5720
[3,]	2125.639	10.83868	-6.360411	320.9765
[4,]	14526.281	31.63087	15.532969	336.7485
[5,]	14526.281	31.63087	15.532969	336.7485

Initialisation à partir de $T = 1$ et variation des nombre de restarts

, , 3

	[,1]	[,2]	[,3]	[,4]
[1,]	2106.903	10.56633	-6.130627	320.4943
[2,]	2089.035	10.02303	-5.644370	319.5720
[3,]	2089.035	10.02303	-5.644371	319.5720
[4,]	2089.035	10.02303	5.644371	319.5720
[5,]	2089.035	10.02303	5.644371	319.5720

, , 4

	[,1]	[,2]	[,3]	[,4]
[1,]	2089.035	10.02303	-5.644371	319.5720
[2,]	2217.306	11.63024	-6.677325	322.0614
[3,]	2249.870	11.85460	-6.809067	322.2904
[4,]	2089.035	10.02303	-5.644372	319.5720
[5,]	2089.035	10.02303	-5.644370	319.5720

Initialisation à partir de $T = 1$ et variation des nombre de restarts

, , 5

	[,1]	[,2]	[,3]	[,4]
[1,]	2089.035	10.02303	5.644371	319.5720
[2,]	2089.035	10.02303	-5.644370	319.5720
[3,]	2125.639	10.83868	-6.360411	320.9765
[4,]	14526.281	31.63087	15.532969	336.7485
[5,]	14526.281	31.63087	15.532969	336.7485

, , 6

	[,1]	[,2]	[,3]	[,4]
[1,]	2089.035	10.02303	-5.644371	319.5720
[2,]	2089.035	10.02303	-5.644371	319.5720
[3,]	2125.639	10.83868	-6.360411	320.9765
[4,]	14526.281	31.63087	15.532969	336.7485
[5,]	14526.281	31.63087	15.532969	336.7485

Initialisation à partir de $T = 1$ et variation des nombre de restarts

, , 7

	[,1]	[,2]	[,3]	[,4]
[1,]	2267.257	11.96708	-6.874162	322.3982
[2,]	2182.027	11.35989	-6.515039	321.7574
[3,]	2089.035	10.02303	-5.644370	319.5720
[4,]	17693.418	33.55918	-16.117884	337.3699
[5,]	29460.636	38.52439	-17.534408	338.6248

, , 8

	[,1]	[,2]	[,3]	[,4]
[1,]	2089.035	10.02303	-5.644370	319.5720
[2,]	2089.035	10.02303	-5.644370	319.5720
[3,]	2089.035	10.02303	5.644370	319.5720
[4,]	2089.035	10.02303	5.644371	319.5720
[5,]	13524.189	30.93065	15.315042	336.4985

Variation du nombre de restarts

, , 9

	[,1]	[,2]	[,3]	[,4]
[1,]	2089.035	10.02303	5.644370	319.5720
[2,]	2089.035	10.02303	-5.644370	319.5720
[3,]	2125.639	10.83868	-6.360411	320.9765
[4,]	14526.281	31.63087	15.532969	336.7485
[5,]	14526.281	31.63087	15.532969	336.7485

, , 10

	[,1]	[,2]	[,3]	[,4]
[1,]	2089.035	10.02303	-5.644371	319.5720
[2,]	2089.035	10.02303	-5.644370	319.5720
[3,]	2125.639	10.83868	-6.360411	320.9765
[4,]	14526.281	31.63087	15.532969	336.7485
[5,]	14526.281	31.63087	15.532969	336.7485

Sortie pour $T = 10$ avec $a_1 = l(1)$, $l_1 = 1$, $s_1 = 1 : 100$

```
> L10C(11)[1:]  
      [,1]      [,2]      [,3]      [,4]  
[1,] 3181.602 16.17395 90.60544 326.1289  
[2,] 13826.498 31.14730 236.63059 336.5774  
[3,] 2227.795 11.75031 47.62022 322.2710  
[4,] 2210.562 11.58110 44.19744 322.0085  
[5,] 3054.362 15.72552 86.02698 325.9717  
[6,] 3185.468 16.18778 90.74996 326.1385  
[7,] 2296.994 12.15005 48.70304 322.5645  
[8,] 2119.134 10.75259 39.54759 320.8243  
[9,] 2627.843 13.92417 67.10492 324.2981  
[10,] 2305.333 12.20039 49.10325 322.6166  
> |
```

Sortie pour $T = 10$ avec 40 itérations de optim

```
> Liste(n)[[2]]  
      [,1]      [,2]      [,3]      [,4]  
[1,] 2089.035 10.02303 31.85892 319.5720  
[2,] 13826.456 31.14727 236.63022 336.5774  
[3,] 2089.035 10.02303 31.85891 319.5720  
[4,] 2089.035 10.02303 31.85891 319.5720  
[5,] 2263.910 11.94577 47.08537 322.3782  
[6,] 2089.035 10.02303 31.85892 319.5720  
[7,] 2089.035 10.02303 31.85891 319.5720  
[8,] 2089.035 10.02303 31.85894 319.5720  
[9,] 2627.839 13.92415 67.10472 324.2980  
[10,] 2304.457 12.19433 49.05376 322.6032  
> |
```

Sortie pour $T = 10$ avec $a_1 = l(1)$, $l_1 = 1$, $s_1 = 1 : 100$

```
> L10C(10)/[[1]]  
      [,1]      [,2]      [,3]      [,4]  
[1,] 3181.602 16.17395 90.60544 326.1289  
[2,] 13826.498 31.14730 236.63059 336.5774  
[3,] 2227.795 11.75031 47.62022 322.2710  
[4,] 2210.562 11.58110 44.19744 322.0085  
[5,] 3054.362 15.72552 86.02698 325.9717  
[6,] 3185.468 16.18778 90.74996 326.1385  
[7,] 2296.994 12.15005 48.70304 322.5645  
[8,] 2119.134 10.75259 39.54759 320.8243  
[9,] 2627.843 13.92417 67.10492 324.2981  
[10,] 2305.333 12.20039 49.10325 322.6166  
> |
```

Sortie pour $T = 10$ avec 40 itérations de optim

```
> Liste(n)[[2]]  
      [,1]      [,2]      [,3]      [,4]  
[1,] 2089.035 10.02303 31.85892 319.5720  
[2,] 13826.456 31.14727 236.63022 336.5774  
[3,] 2089.035 10.02303 31.85891 319.5720  
[4,] 2089.035 10.02303 31.85891 319.5720  
[5,] 2263.910 11.94577 47.08537 322.3782  
[6,] 2089.035 10.02303 31.85892 319.5720  
[7,] 2089.035 10.02303 31.85891 319.5720  
[8,] 2089.035 10.02303 31.85894 319.5720  
[9,] 2627.839 13.92415 67.10472 324.2980  
[10,] 2304.457 12.19433 49.05376 322.6032  
> |
```

Algo1: liste des minimums

a.opt	l.opt	s.opt	min.opt
2088.911	10.11929	-5.712177	319.5446
2128.204	10.87978	6.370550	320.8094
2145.336	11.07451	6.528464	321.1384
2153.056	11.15474	6.592431	321.2725
2158.027	11.20450	6.631791	321.3551
2224.886	11.69203	-6.714658	321.8885
2225.025	11.69300	-6.715230	321.8895
2249.680	11.86072	-6.813429	322.0586
2253.529	11.88597	-6.828094	322.0831
2256.011	11.90214	-6.837462	322.0987
2266.977	11.97247	-6.878080	322.1655
2271.453	12.00070	-6.894315	322.1918
2278.152	12.04244	-6.918256	322.2303
2278.642	12.04547	-6.919990	322.2330
2280.267	12.05551	-6.925729	322.2422
2283.831	12.07739	-6.938233	322.2620
2304.857	12.20350	-7.009853	322.3734
2305.080	12.20481	-7.010591	322.3745
2328.151	12.34677	7.112471	322.5107
2385.756	12.68629	-7.357375	322.8470
2934.332	15.24451	8.990657	325.1856
2961.585	15.35206	9.052868	325.2699
3129.861	15.98855	-9.412633	325.7453
3692.467	17.84610	-10.391605	326.9218
3694.603	17.85251	-10.394823	326.9253
9898.549	27.80715	-14.274613	335.3236
10299.681	28.19739	-14.404736	335.5083
10876.948	28.73248	-14.581277	335.7510
11255.129	29.06751	14.690733	335.8971
13910.732	31.13899	-15.350165	336.7118
17746.211	33.51140	-16.072194	337.4910
18057.835	33.68070	-16.122482	337.5414
36877.378	40.60046	18.058501	339.1720

n=10 # i : les observations
l0=1 # k0= 1:K0 ; K = rest
j0=1
s0=100
K0=5
K=40

Algorithme 2: idée générale

On veut explorer tous les minimums d'une fonction $f(\theta)$ en utilisant un algorithme de minimisation.

1) On part d'une initialisation θ_0 puis on met la main sur un minimum θ_1 .

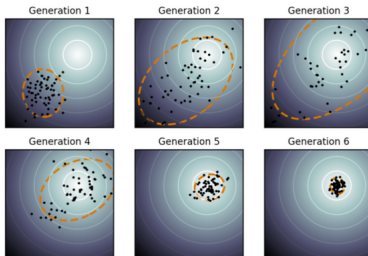
2) On tire un échantillon (θ_1^i) selon une loi de probabilité paramétrée par θ_1 (dans notre cas c'est la loi gaussienne tronquée). On produit une liste de minimum (θ_2^i) à partir des initialisations (θ_1^i).

3) On recommence les étapes 1) et 2) jusqu'à la stabilisation de la liste des minimums.

Algorithme 2: les stratégies d'évolution

```
Initialize distribution parameters  $\theta^{(0)}$ 
For generation  $g = 0, 1, 2, \dots$ 
  Sample  $\lambda$  independent points from distribution  $P(x|\theta^{(g)}) \rightarrow x_1, \dots, x_\lambda$ 
  Evaluate the sample  $x_1, \dots, x_\lambda$  on  $f$ 
  Update parameters  $\theta^{(g+1)} = F_\theta(\theta^{(g)}, (x_1, f(x_1)), \dots, (x_\lambda, f(x_\lambda)))$ 
  break, if termination criterion met
```

Figure 2: Randomized black box search. $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is the objective function



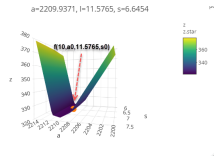
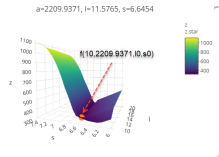
Liste produite par l'algorithme 2

a.opt	l.opt	s.opt	min.opt
2088,91087	10,1192944	5,7121776	319,544595
2088,91277	10,1194326	5,71230428	319,544817
2127,52101	10,8713995	6,36366801	320,795195
2128,71464	10,8860026	6,37565862	320,820005
2129,49682	10,8954801	6,38342734	320,836098
2131,09727	10,9146565	6,39912089	320,86864
2135,4166	10,9650642	6,44018936	320,954021
2136,5972	10,9785253	6,45111157	320,976775
2140,92468	11,0268223	6,49015025	321,058247
2141,14332	11,0292212	6,49208309	321,062287
2142,66799	11,0458461	6,50546282	321,090264
2144,10215	11,0613203	6,51789113	321,116261
2145,14657	11,0724954	6,52685257	321,135019
2146,92575	11,0913539	6,5419455	321,166632
2153,51741	11,1594166	6,5961342	321,280277
2155,12921	11,17566	6,60900296	321,307289

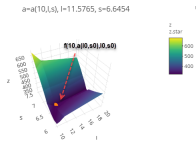
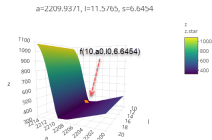
Le plus grand minimum

$$a = 39997 \quad l = 41.39 \quad s = 18.27 \quad 339.32.$$

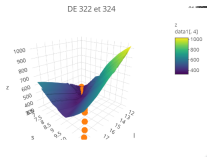
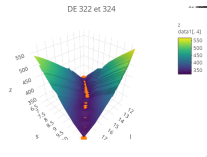
Visualisations graphiques



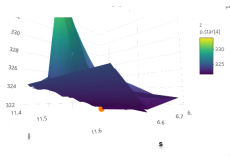
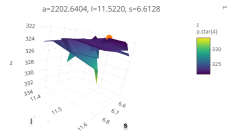
Visualisations graphiques



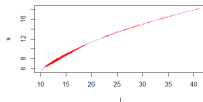
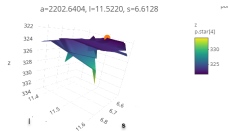
Visualisations graphiques



Visualisations graphiques



Visualisations graphiques



lm(formula = log(z) ~ log(l))

Residuals:

Min	1Q	Median	3Q	Max
-0.014254	-0.031100	0.003227	0.026236	0.060380

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.058478	0.004416	1.32	0.085
log(l)	0.790029	0.005400	564.41	<2e-16 ***

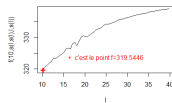
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.03104 on 2369 degrees of freedom
Multiple R-squared: 0.9926,
F-statistic: 3.186e+05 on 1 and 2369 DF, p-value: < 2.2e-16

Adjusted R-squared: 0.9926

lm(a)

Interpolation des minimums



Retour à la prévision à partir de $T = 10$ en utilisant notre liste des minimums

Intervalle de confiance

$$\hat{a} \exp\left(-\frac{(t - \hat{l})^2}{(\hat{s})^2}\right) \pm 2.995732\hat{\lambda}$$

avec $t > T$

Retour à la prévision à partir de $T = 10$ en utilisant notre liste des minimums

Retour à la prévision à partir de $T = 11$ en utilisant notre liste des minimums

Retour à la prévision à partir de $T = 12$ en utilisant notre liste des minimums

Retour à la prévision à partir de $T = 13$ en utilisant notre liste des minimums

Retour à la prévision à partir de $T = 14$ en utilisant notre liste des minimums

Retour à la prévision à partir de $T = 15$ en utilisant notre liste des minimums

Retour à la prévision à partir de $T = 16$ en utilisant notre liste des minimums

Retour à la prévision à partir de $T = 17$ en utilisant notre liste des minimums

Retour à la prévision à partir de $T = 18$ en utilisant notre liste des minimums

Retour à la prévision à partir de $T = 19$ en utilisant notre liste des minimums

Retour à la prévision à partir de $T = 20$ en utilisant notre liste des minimums

Retour à la prévision à partir de $T = 21$ en utilisant notre liste des minimums

- Le minimum global est-il toujours le bon choix? Non.
- Dans le cas de plusieurs minimums locaux, comment peut-on les utiliser dans la prévision? Intervalles de confiances.
- Chaque algorithme de minimisation nécessite une initialisation et un critère d'arrêt. Comment peut-on choisir les initialisations pour explorer tous les minimums? Tâtonnement cohérent, stratégies d'évolutions, ...
- Quand l'algorithme s'arrête comment peut-on être sûr qu'il s'agit d'un vrai minimum? Utiliser d'autres algorithmes, tirage aléatoire.
- Une bonne fonction objective doit avoir plusieurs minimums au début de la période, et uniquement un seul minimum à la fin.
- Dans le cas où certains minimums sont proches, comment décider de leur égalité? Pas encore clair pour nous.