01 Chaos computation of K

Author

February 2022

Présentation de l'algorithme

Je ne présente pas la théorie derrière le test mais simplement l'écriture du programme. Voici la version de l'algorithme tel que présentée dans cette article et celui là. ϕ représente la donnée en entrée et N_t le nombre de valeurs dont nous disposons pour *phi*. N_c est le nombre de valeur différente de c qui sont utilisées.

Algorithm 1: 01 choas test original algorithm

Result: K_c

Read the file to initialize $\phi = \begin{pmatrix} \phi_1 \\ \vdots \\ \phi_{n_t} \end{pmatrix}$ et N_t

Initialize N_c

for $m \in [\![1, N_c]\!]$ **do**

  Choose a value for $c \in]0; \pi[$

  for $c \in [\![1, N_t]\!]$ **do**

    $p_n(c) \leftarrow \sum_{k=1}^{N_t} \phi_i \cos(kc)$

    $q_n(c) \leftarrow \sum_{k=1}^{N_t} \phi_i \cos(kc)$

    $M_n(c) \leftarrow \sum_{k=1}^{N_t} (p_{n+k}(c) - p_k(c))^2 + (q_{n+k}(c) - q_k(c))^2$

  end

  $M \leftarrow (M_1, ..., M_{N_t})$

  $t \leftarrow (t_1, ..., t_{N_t})$

  evaluate K_m

end

$K \leftarrow (K_1, ..., K_{N_c})$

$K_c \leftarrow \text{Median}(K)$

return K_c

Ceci est purement théorique puisqu'en pratique le remplissage des différentes valeurs de $p_n(c)$ et $q_n(c)$ se fait en utilisant les valeurs précédemment calculées. La particularité ici est qu'en réalité nous ne pouvons pas calculer $M_n(c)$ jusqu'à N_t. Nous nous limiterons à N_0 comme mentionnée dans les articles cités plus haut. L'algorithme implémenté s'écrit comme 3 en annexe.

Approximation de K

Maintenant si nous nous tenons à l'article à l'origine de ce test nous avons :

$$K = \lim_{n \to \infty} \frac{log M_n}{log\, n}$$

Or ne pouvant calculer la valeur de K de manières analytiques, deux approximation nous sont proposées. Les deux méthodes sont :

- méthode de corrélation
- méthode de régression

Correlation

Pour cette méthode nous introduisons :

$$D_n = M_n - (E\phi)^2 \frac{1 - \cos(cn)}{1 - \cos(c)} \text{ où } E\phi = \frac{1}{N_t} \sum_{k=1}^{N_t} \phi_k$$

Pour information la terme correctif se note :

$$V_{osc}(c, n) = (E\phi)^2 \frac{1 - \cos(cn)}{1 - \cos(c)}$$

Maintenant nous voulons calculer :

$$K = \lim_{n\to\infty} \frac{logD_n}{log\, n}$$

La limite est identique mais les propriétés de convergence sont plus intéressantes. L'approximation consiste simplement en :

$$K = \frac{Cov(D,t)}{\sqrt{Var(D)Var(t)}} \in [-1;1]$$

avec pour rappel $Cov(x,y) = \frac{1}{n}\sum_{k=1}^{n}(x - E(x))(y - E(y))$ et $Cov(x,x) = Var(x)$. Nous avons donc confondu K avec un coefficient de corrélation et sa valeur est donc comprise entre -1 et 1.

Régression

Ici nous continuons avec la définition de D_n mais nous introduisons aussi $\tilde{D}_n(c) = D_n(c) + a\min|D_n(c)|$ avec $a > 1$. Maintenant il suffit de faire un régression linéaire entre $\log\tilde{D}$ et $\log n$. Cependant il est indiqué que cette régression doit se faire par la minimisation de l'erreur selon la norme L^1 soit $||\phi - at - b1||_1 = \sum_{i=1}^{n}|\phi_i - at_i - b|$.

Pour la suite nous introduisons un abus de notation, 1 désignera le nombre mais aussi un vecteur dont toutes les composantes sont égals à 1. Contrairement à l'implémentation d'une méthode des moindres carrées ici il n'y a pas de solution analytique. Notre choix se porte sur la méthode Iteratively Reweighted Least Squares (IRLS). Ce qui se traduit par la formulation suivante :

$$\arg\min_{\beta} ||A\beta - \tilde{D}||_1 \quad \text{avec } A = \begin{pmatrix} t_1 & 1 \\ \vdots & \vdots \\ t_{N_t} & 1 \end{pmatrix},\ \tilde{D} = \begin{pmatrix} \tilde{D}_1 \\ \vdots \\ \tilde{D}_{N_t} \end{pmatrix} \text{ et } \beta = \begin{pmatrix} a \\ b \end{pmatrix}$$

La résolution se fait de manière itérative en voyant le problème comme un Weighted least squares problem. Une simple visite de wikipedia sur la méthode permet de clarifier la méthode utiliser. Pour notre cas nous avons

$$W = \text{diag}(w_1, ..., w_{N_t}) \text{ avec } \forall i \in [\![1, N_t]\!]\ w_i = \frac{1}{|\tilde{D}_i - at_i - b|}$$

$$\beta^{k+1} = \left(A^T W^k A\right)^{-1} A^T W^k \tilde{D}$$

Pour faciliter l'implémentation nous pouvons directement faire les calculs matricielles pour extraire les formules. Notre motivation est la simplicité des calcules qui vont aussi rendre le code plus facile à écrire mais à lire (d'où ce document) :

$$\begin{aligned}
\left[A^T W^k A\right]^{-1} &= \left[A^T \begin{pmatrix} w_1 t_1 & w_1 \\ \vdots & \vdots \\ w_{N_t} t_{N_t} & w_{N_t} \end{pmatrix}\right]^{-1} \\
&= \left[\begin{pmatrix} t_1 & t_2 & ... & t_{N_t} \\ 1 & 1 & ... & 1 \end{pmatrix} \begin{pmatrix} w_1 t_1 & w_1 \\ \vdots & \vdots \\ w_{N_t} t_{N_t} & w_{N_t} \end{pmatrix}\right]^{-1} \\
&= \left[\begin{pmatrix} \sum_{k=1}^{N_t} w_i t_i^2 & \sum_{k=1}^{N_t} w_i t_i \\ \sum_{k=1}^{N_t} w_i t_i & \sum_{k=1}^{N_t} w_i \end{pmatrix}\right]^{-1}
\end{aligned}$$

Finalement nous avons :

$$\left[A^T W^k A\right]^{-1} = \frac{1}{\sum_{k=1}^{N_t} w_i \sum_{k=1}^{N_t} w_i t_i^2 - \sum_{k=1}^{N_t} w_i t_i{}^2} \begin{pmatrix} \sum_{k=1}^{N_t} w_i & -\sum_{k=1}^{N_t} w_i t_i \\ -\sum_{k=1}^{N_t} w_i t_i & \sum_{k=1}^{N_t} w_i t_i^2 \end{pmatrix}$$

Nous pouvons facilement obtenir le résultat suivant :

$$A^T W^k \tilde{D} = \begin{pmatrix} \sum_{k=1}^{N_t} w_i \tilde{D}_i t_i \\ \sum_{k=1}^{N_t} w_i \tilde{D}_i \end{pmatrix}$$

Ce qui nous permet d'aboutir à la régression selon l'algorithme suivant avec pour notation S_{xy} indiquant la somme des $x_i y_i$:

Algorithm 2: regression

Result: K_c

init $l\tilde{D} = \begin{pmatrix} \log \tilde{D}_1 \\ \vdots \\ \log \tilde{D}_{N_t} \end{pmatrix}$ et $lt = \begin{pmatrix} \log t_1 \\ \vdots \\ \log t_{N_t} \end{pmatrix}$

Initialize a,b,a',b'

while $\sqrt{(a-a')^2 + (b-b')^2} > Threshold$ **do**

a'=a

b'=b

S_w=0

$S_{w\tilde{D}} = 0$

$S_{w\tilde{D}t} = 0$

$S_{wt} = 0$

$S_{wt^2} = 0$

for $c \in [\![1, N_0]\!]$ **do**

$S_w := S_w + w_i$

$S_{w\tilde{D}} := S_{w\tilde{D}} + w_i \tilde{D}_i$

$S_{w\tilde{D}t} := S_{w\tilde{D}t} + w_i \tilde{D}_i t_i$

$S_{wt} := S_{wt} + w_i t_i$

$S_{wt^2} := S_{wt^2} + w_i t_i^2$

end

$a = \frac{1}{S_{wt^2} S_w - S_w^2} \left(S_w S_{w\tilde{D}t} - S_{wt} S_{w\tilde{D}}\right)$

$b = \frac{1}{S_{wt^2} S_w - S_w^2} \left(S_{wt^2} S_{w\tilde{D}} - S_{wt} S_{w\tilde{D}t}\right)$

end

return a

Annexe

Algorithm 3: 01 choas test original algorithm

Result: K_c

Read the file to initialize $\phi = \begin{pmatrix} \phi_1 \\ \vdots \\ \phi_{n_t} \end{pmatrix}$ et N_t

Initialize N_c

for $m \in [\![1, N_c]\!]$ **do**

  Choose a value for $c \in]0; \pi[$

  $p_1(c) \leftarrow \phi_i \cos(c)$

  $q_1(c) \leftarrow \phi_i \sin(c)$

  for $c \in [\![2, N_t]\!]$ **do**

    $p_n(c) \leftarrow p_{n-1}(c) + \phi_i \cos(nc)$

    $q_n(c) \leftarrow q_{n-1}(c) + \phi_i \sin(nc)$

  end

  for $c \in [\![1, N_0]\!]$ **do**

    $M_n(c) \leftarrow \sum_{k=1}^{N_t} (p_{n+k}(c) - p_k(c))^2 + (q_{n+k}(c) - q_k(c))^2$

  end

  $M \leftarrow (M_1, ..., M_{N_0})$

  $t \leftarrow (t_1, ..., t_{N_0})$

  evaluate K_m

end

$K \leftarrow (K_1, ..., K_{N_c})$

$K_c \leftarrow \text{Median}(K)$

return K_c