HAL
archives-ouvertes.fr

Algorithme rapide pour la détection optimale des ruptures

Guillem Rigaill

▶ **To cite this version:**

Guillem Rigaill. Algorithme rapide pour la détection optimale des ruptures. 42èmes Journées de Statistique, 2010, Marseille, France, France. inria-00494813

HAL Id: inria-00494813
https://hal.inria.fr/inria-00494813

Submitted on 24 Jun 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Algorithme rapide pour la détection optimale des ruptures

Guillem Rigaill[1,2]

1. UMR AgroParisTech / INRA MIA 518, 16 rue Claude Bernard, F - 75 231 Paris
2. Institut Curie, Département de Transfert, Laboratoire de Signalisation, Quadrilatère historique Porte 13, 1 rue Claude Vellefaux, Hôpital Saint-Louis, F - 75 010 Paris

Résumé

Dans les modèles de détection de ruptures, les données sont modélisées par un processus aléatoire dont les paramètres sont soumis à des changements brusques en des instants inconnus, appelés instants de ruptures. La recherche exhaustive des positions des ruptures de norme quadratique minimale se fait par un algorithme de programmation dynamique. L'intérêt de cet algorithme est qu'il permet d'obtenir la solution optimale en réduisant la complexité algorithmique de $O(n^K)$ à $O(Kn^2)$, où $K-1$ est le nombre de ruptures fixé et n la taille du signal. Même si le temps de calcul est réduit, cet algorithme ne peut être utilisé sur des signaux de grandes tailles. Nous proposons ici un nouvel algorithme de programmation dynamique permettant d'obtenir la solution optimale en un temps de calcul très nettement réduit. Notamment il permet d'analyser un signal d'un million de points en quelques minutes. Plus précisement, nous démontrons qu'au pire des cas sa complexité en temps et en espace sont respectivement de $O(Kn^2)$ et de $O(Kn)$ et nous montrons que son temps de calcul est empiriquement de l'ordre de $O(Kn\log(n))$. Par ailleurs, Nous comparons cet algorithme à l'algorithme de programmation dynamique classique. L'algorithme proposé est au pire des cas équivalent et a une complexité empirique bien plus faible .

In multiple change-points detection models, it is assumed that the observed data is a realization of an independent random process affected by $K-1$ abrupt changes, called change-points, at some unknown positions. Dynamic programming allow to retrieve the $K-1$ change-points with the smallest loss. This algorithm reduce the complexity from $O(n^K)$ to $O(Kn^2)$ where n is the number of observations. However the quadratic complexity in n restrict the use of such algorithm to small or intermediate value of n. We propose a new dynamic programming algorithm that recovers the $K-1$ change-points with the smallest Euclidean loss within a drastically reduce amount of time. Our algorithm can process a sequence of one million points in a matter of minutes. More precisely, we demonstrate that at worst the complexity is in $O(Kn^2)$ time and $O(Kn)$ space and we show that the empirical time complexity of our algorithm is approximately $O(Kn\log(n))$. Its complexity is compared with the classical dynamic programming algorithm. Overall, our algorithm is at worst equivalent and has a much better empirical run-time.

Mots-clés :
Grande dimension ; Détection de ruptures pour de grands signaux ; Programmation dynamique

La détection de ruptures est un problème rencontré dans de nombreux domaines, de l'analyse de signal acoustique [1] à la biologie moléculaire [2]. L'objectif est de découper un signal en K morceaux contigus et homogènes, de longueurs ou de durées variables. Ces segments sont délimités par des instants appelés ruptures. La détection de ces ruptures est ici réalisée "off-line", i.e. sur les données complètes, par opposition à la détection séquentielle ("on-line"). Un algorithme de programmation dynamique permet de trouver la solution optimale en un temps de calcul en $O(Kn^2)$ ([3], [4], [5]). Cette complexité quadratique limite l'utilisation de cet algorithme pour des signaux de grandes tailles. Pour résoudre ce problème, de nombreuses stratégies ont été proposées. Notamment dans [5] une légère modification du problème d'optimisation permet une implémentation efficace de l'algorithme LAR [6]. Une autre stratégie consiste à effectuer une présélection des ruptures, en utilisant par exemple l'algorithme CART [7]. Toutefois, ces procédures ne permettent pas d'obtenir la solution optimale. Elles sont donc des formes de compromis entre un temps de calcul raisonnable et une perte de précision de la solution.

L'algorithme que nous proposons ne fait pas ce type de compromis : il trouve systématiquement la meilleure solution et dans un temps raisonnable, même pour des signaux de très grandes tailles. Plus précisément, nous proposons un nouvel algorithme de programmation dynamique qui retrouve la meilleure solution avec une complexité théorique dans le pire des cas équivalente à celle de la programmation dynamique classique ($O(Kn^2)$) et une complexité empirique de l'ordre de $O(Kn\log(n))$ pour la norme Euclidienne. Cet algorithme permet d'analyser un signal d'un million de points en quelques secondes ou quelques minutes.

Ici on se place dans le cadre de la détection de ruptures dans la moyenne d'un signal gaussien. On observe des réalisations d'un processus aléatoire indépendant $Y = \{Y_t\}_{t=1,\dots,n}$. Ce processus est affecté par $K-1$ changements ou ruptures. Ces ruptures délimitent une partition m de $\{1,\dots,n\}$ en K segments. Pour un nombre de ruptures fixé, l'estimation de leurs positions se fait classiquement en minimisant la norme quadratique définie par :

$$min_{\{m \in \mathcal{M}_K\}}\{\sum_{r \in m}\sum_{t \in r}(Y_t - \bar{Y}_r)^2\}$$

où $\bar{Y}_r$ est la moyenne des observations sur le segment r et $\mathcal{M}_K$ est l'ensemble des segmentations possibles en K segments. On définit le coût c_r d'un segment r :

$$c_r = \sum_{t \in r}(Y_t - \bar{Y}_r)^2$$

et on note $\mathcal{M}_{K,t}$ l'ensemble des segmentations en K segments du signal jusqu'à la position t. La quantité que l'on cherche à minimiser est

$$C_{K,t} = min_{\{m \in \mathcal{M}_{K,t}\}}\{ \sum_{r \in m} c_r\}, \text{pour } t = n$$

L'algorithme de programmation dynamique classique ([3], [4], [5]) repose sur la formule de mise à jour suivante :

$$\forall\ t\ \in\ [\![K, n]\!] \qquad C_{K,t} = \min_{K-1 \leq i < t}\{\ C_{K-1,i} + c_{[\![i+1,t]\!]}\ \},$$

L'algorithme que nous proposons se fonde sur l'étude de la fonction $H_{K,t}(\mu)$:

$$H_{K,t}(\mu) = min_{\{m \in \mathcal{M}_{K,t}\}}\{ \sum_{r \in m, r \neq r_{m,K}} c_r + \sum_{t \in r_{m,K}} (Y_t - \mu)^2\ \},$$

où μ est dans $\mathbb{R}$ et $r_{m,K}$ est le dernier segment de la segmentation m de $\mathcal{M}_K$. On a $C_{K,t} = min_{\{\mu \in \mathbb{R}\}}\{H_{K,t}(\mu)\}$, donc connaître $H_{K,t}$ suffit pour avoir $C_{K,t}$. Les propriétés intrinsèques de $H_{K,t}$ permettent d'élaguer efficacement l'espace des segmentations candidates et ainsi retrouver la meilleure solution rapidement. Plus précisément, on utilise la formule de mise à jour suivante :

$$H_{K,t+1}(\mu) = min\{H_{K,t}(\mu),\ C_{K-1,t'}\ \} + (Y_{t+1} - \mu)^2,$$

qui permet de passer de la solution au point t à celle au point $t + 1$.

En utilisant un ordinateur 1.8 GHz, l'algorithme proposé trouve la meilleure segmentation en 50 segments d'un signal d'un million de points en 3 minutes contre 3 jours avec la programmation dynamique classique. Le temps de calcul empirique sur des données simulées et des données réelles est de l'ordre de $O(Kn\log(n))$.

Par ailleurs nous montrons que l'algorithme proposé s'étend à toute optimisation de la forme :

$$min_{\{m \in \mathcal{M}_K\}}\{ \sum_{r \in m} \quad min_{\{\mu \in \mathbb{R}\}}\{\textstyle\sum_{i \in r} \gamma(Y_i, \mu)\} \quad \},$$

où les observations Y_i sont dans un ensemble quelconque A et la fonction $\gamma : A \times \mathbb{R} \to \mathbb{R}$ est une fonction convexe de μ. Pour tous ces problèmes nous démontrons formellement que la complexité de l'algorithme est dans le pire des cas de $O(Kn^2)$ en temps et de $O(Kn)$ en espace.

Bibliographie

[1] Gillet 0. and Essid S. and Richard G. (2007), On the correlation of automatic audio and visual segmentation of music videos, IEEE Transactions on Circuits and Systems for Video Technology.
[2] Picard F., Robin S., Lavielle M., Vaisse C., Daudin J.J. (2005), A statistical approach for array CGH data analysis, BMC Bioinformatics, 11, 6-27.
[3] Guthery S.B. (1973), Partition Régression, Journal ofthe American Statistical. Association 69, 945-947.
[4] Bai J. and Perron P. (1998), Estimating and Testing Linear Models with Multiple Structural Changes, Econometrica. 66, 47-78.
[5] Harchaoui, Z. and Lévy-Leduc, C. (2007), Catching Change-points with Lasso, In NIPS, Vancouver, dec 2007.
[6] Efron B. Hastie T. and Tibshirani R. (2004) Least angle regression. Annals of Statistics, 32 :407-499
[7] Gey S. and Lebarbier E. (2008) Using CART to detect multiple change points in the mean for large samples. Research Report n°12 SSB group
[8] Breiman L. and Friedman J. and Olshen R. and Stone C. (1984) Classification and regression trees. Monterey, Calif., U.S.A. : Wadsworth, Inc.