gfpop : Un package R pour la détection de ruptures contrainte par un graphe

Vincent Runge[1] & Toby Hocking[2] & Guillem Rigaill[1,3]

[1]*Laboratoire de Mathématiques et Modélisation d'Évry, Université d'Évry Val d'Essonne*
vincent.runge@univ-evry.fr
[2]*Northern Arizona University. School of Informatics, Computing and Cyber Systems*
toby.hocking@nau.edu
[3]*Institute of Plant Sciences Paris-Saclay, INRA*
guillem.rigaill@inra.fr

Résumé. T. D. Hocking et al. (2017) ont proposé un algorithme pour l'inférence de modèles de détection de ruptures avec des contraintes entre les paramètres des segments successifs. L'algorithme est exact au sens où il optimise un risque pénalisé. Leur implémentation traite le cas particulier d'une contrainte haut-bas pour une perte Poisson. Nous avons développé un package `R` implémentant l'algorithme de manière générique traitant plusieurs pertes (avec leur équivalent robuste) et de nombreuses contraintes décrites par un graphe. Nous illustrerons d'abord les performances de notre algorithme dans le cas de la régression isotonique. Nous mettons en évidence l'intérêt d'utiliser des pertes robustes – récemment suggéré par Bach (2018) et Fearnhead and Rigaill (2018) – et celui de pénaliser le nombre de ruptures quand il y a effectivement des changements abrupts dans la moyenne du signal. Nous présentons enfin plus généralement les potentialités du package avec des graphes de contraintes plus exotiques.

Mots-clés. Détection de ruptures contraint par un graphe. Programmation dynamique élaguée. Inférence robuste. Régression isotonique.

Abstract. T. D. Hocking et al. (2017) proposed an algorithm for model inference of changepoint models with constraints between successive segment parameters. The algorithm is exact as it optimizes a penalised risk. There implementation deals with the particular case of an up-down constraint with a Poisson loss. We developed an `R` package implementing the algorithm in a generic way with several loss functions (and their robust equivalent) and many constraint structures encoded into a graph. We first illustrate performances of our algorithm in the case of the isotonic regression. We illustrate the benefit of using robust losses – recently suggested by Bach (2018) and Fearnhead and Rigaill (2018) – and also of penalizing the number of changes when there are steps in the signal. We finally present the potentialities of our package with more exotic constraints graphs.

Keywords. Graph-constrained changepoint detection. Pruned dynamic programming. Robust inference. Isotonic regression.

1 Introduction

Le problème de la détection de ruptures multiples consiste à retrouver des changements abrupts dans un signal. Dans le cas d'un bruit Gaussien le modèle est $Y_t \sim \mathcal{N}(\mu_t, \sigma^2)$, où $t \in \{1, ..., n\}$ et $t \mapsto \mu_t$ est constant par morceaux. Nous cherchons alors à inférer la position des ruptures, c'est-à-dire des t tels que $\mu_t \neq \mu_{t+1}$ ainsi que le nombre de ces ruptures à partir des données observées $(y_t)_{t=1,...,n}$.

Une manière classique de procéder est d'optimiser un risque quadratique en fixant un nombre de ruptures. Il est aussi possible de pénaliser le nombre de rupture (même si ce n'est pas exactement équivalent) en minimisant en $\mu = (\mu_1, ..., \mu_t)^T \in \mathbb{R}^n$ la quantité

$$\sum_{t=1}^{n}(y_t - \mu_t)^2 + \lambda \sum_{t=1}^{n-1} I_{\mu_t \neq \mu_{t+1}} ,$$

où I est la fonction indicatrice. Dans les deux cas des algorithmes de programmation dynamique rapides et exacts répondent à notre question (Killick et al. (2012); Rigaill (2015); Maidstone et al. (2017)).

Dans de nombreuses applications, il est souhaitable de contraindre les paramètres des segments successifs (T. Hocking et al. (2015); Maidstone et al. (2017); Jewell et al. (2018); Baranowski and Fryzlewicz (2014)). Le cas le plus simple et le plus étudié est sans doute celui de la régression isotonique (Barlow (1972)). Dans ce cas l'objectif est de minimiser en μ la quantité:

$$\sum_{t=1}^{n}(y_t - \mu_t)^2 , \text{ sous-constrainte } \mu_t \leq \mu_{t+1} , t = 1, ..., n-1 .$$

Il faut bien noter que l'estimateur obtenu est constant par morceaux, ce qui fait le lien avec le problème de détection de ruptures.

De manière plus générale on peut vouloir imposer des motifs plus complexes, comme l'unimodalité (Stout (2008)) ou l'alternance de sauts vers le haut puis vers le bas (T. Hocking et al. (2015)) pour détecter des pics. Il existe des algorithmes très efficaces pour le cas isotonique et unimodal (Best and Chakravarti (1990); Stout (2008)) au moins si le nombre de ruptures n'est pas pénalisé ou contraint. Pour des motifs de contraintes plus complexes comme le cas haut-bas, T. Hocking et al. (2015) ont proposé un algorithme exact. Cet algorithme est une généralisation des algorithmes de programmation dynamique fonctionnelle de Rigaill (2015) et Maidstone et al. (2017). L'algorithme permet de pénaliser ou contraindre le nombre de ruptures. L'algorithme permet aussi de considérer des pertes robustes notamment la perte biweight. Dans le cas de la détection de ruptures sans contrainte cette perte à de bonnes propriétés statistiques et est souvent meilleure qu'une perte l_1 ou Huber (Fearnhead and Rigaill (2018)). Les travaux de Bach (2018) dans le cadre de la régression isotonique montre aussi l'intérêt de ce genre de perte.

L'algorithme de notre package permet de considérer une grande variété de contraintes. Ces contraintes sont modélisées sous la forme d'un graphe. À chaque temps t un certain nombre d'états est possible, ces états sont les nœuds du graphe. Les transitions entre les états des temps t et $t+1$ sont représentées par les arêtes du graphe. À chaque arête est associé une contrainte (par exemple $\mu_t \leq \mu_{t+1}$) et une pénalité (éventuellement nulle).

Dans la suite nous présentons d'abord formellement ces graphes de contraintes puis le problème d'optimisation résolu par le package. Nous illustrons les potentialités de notre package par des simulations de régression isotonique avec données corrompues. D'autres exemples (sur données de Chip-Seq notamment) seront présentés lors de la présentation.

2 Graphes des contraintes

2.1 Définition

Le graphe $\mathcal{G}_n$ des contraintes est un graphe dirigé acyclique composé:

(1) de nœuds indicés par le temps $t \in \{1, ..., n\}$ et un état $s \in \{1, ...S\}$;

(2) Tous les nœuds ont un temps t dans $\{1, ..., n\}$ à l'exception du nœuds de départ $v_0 = (0, \emptyset)$ et du nœuds d'arrivée $v_{n+1} = (n+1, \emptyset)$, avec $\emptyset$ désignant un état indéfini;

(3) Les arêtes sont des transitions avec des nœuds consécutifs "en temps" du type $v = (t, s)$ et $v' = (t+1, s')$ ce qui donne des arêtes de type $e = (t, s, s')$ pour $t \in \{0, ..., n\}$;

(4) Chaque nœud e est associé à une fonction indicatrice $I_e : \mathbb{R} \times \mathbb{R} \mapsto \{0, 1\}$ contraignant les moyennes successives μ_t et μ_{t+1} (par exemple $I_e(\mu_t, \mu_{t+1}) = I_{\mu_t \leq \mu_{t+1}}$) et la pénalité $\beta_e \geq 0$.

Avec notre package nous nous plaçons dans le cas simple où les transitions ne dépendent pas du temps.

2.2 Quelques exemples

a) Contrainte isotonique. Le graphe est sur la Figure 1 a). Il y a un seul état 0.
Arêtes : une arête de 0 à 0 non pénalisée et contrainte à $\mu_t = \mu_{t+1}$ et une arête de 0 à 0 avec une contrainte $\mu_t \leq \mu_{t+1}$ ($I_e(\mu_t, \mu_{t+1}) = I_{\mu_t \leq \mu_{t+1}}$) pénalisée par $\beta \geq 0$.

b) Contrainte haut-bas. Le graphe est sur la Figure 1 b). Il y a deux états 0 (bas) et 1 (haut).
Arêtes : deux arêtes non pénalisées et contraintes à $\mu_t = \mu_{t+1}$ de 0 vers 0 et de 1 vers 1; deux arêtes pénalisées avec contrainte $\mu_t \leq \mu_{t+1}$ de 0 vers 1 et contrainte $\mu_t \geq \mu_{t+1}$ de 1 vers 0.

c) Contrainte haut-isotonique décroissante. Son graphe est présenté sur la Figure 1 c).

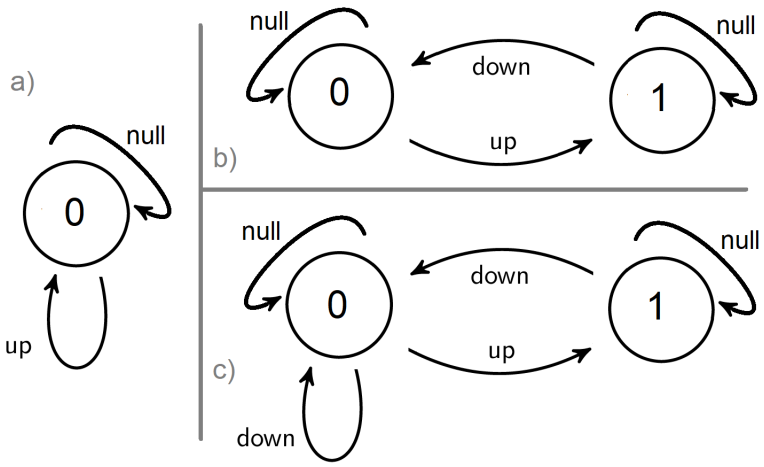

Figure 1: graphes a) isotonique. b) alternance haut-bas. c) alternance haut isotonique décroissante.

2.3 Chemin et validation des contraintes

Un chemin $p \in \mathcal{G}_n$ du graphe est une collection de $n+2$ nœuds $(v_0, ..., v_{n+1})$ avec $v_0 = (0, \emptyset)$ et $v_{n+1} = (n+1, \emptyset)$ et $v_t = (t, s_t)$ pour $t \in \{1, ..., n\}$ et $s_t \in \{1, ...S\}$. De plus, le chemin est fait de $n+1$ arêtes nommées $e_0, ..., e_n$ avec $\beta_{e_n} = 0$. Un vecteur $\mu \in \mathbb{R}^n$ valide le chemin p si pour tout $t \in \{1, ..., n-1\}$, nous avons $I_{e_t}(\mu_t, \mu_{t+1}) = 1$ (vraie). On écrira $p(\mu)$ pour dire que le vecteur μ vérifie le chemin p.

3 Formalisation du problème

Nous pouvons maintenant présenter la formalisation du problème de segmentation sous-contrainte. L'objectif est de minimiser la quantité suivante :

$$Q_n = \min_{\substack{p \in \mathcal{G}_n \\ \mu | p(\mu)}} \left\{ \sum_{t=1}^{n} (\gamma(y_t, \mu) + \beta_{e_t}) \right\} .$$

T. D. Hocking et al. (2017) ont montré qu'il est possible d'optimiser cette quantité par des techniques de programmation dynamique fonctionnelle. L'idée est de considérer Q_n conditionnellement à la valeur θ et à l'état s du dernier point :

$$Q_n^s(\theta) = \min_{\substack{p \in \mathcal{G}_n | v_n = (n,s) \\ \mu | s.c \ p(\mu) \\ \mu_n = \theta}} \left\{ \sum_{t=1}^{n} (\gamma(y_t, \mu) + \beta_{e_t}) \right\} .$$

Il est alors possible de définir et exploiter une formule de mise à jour permettant de calculer tous les $Q_n^s(\theta)$ à partir de tous les $Q_{n-1}^s(\theta)$.

4 Illustration avec données isotoniques corrompues

Nous illustrons ci-dessous l'intérêt d'utiliser une perte bornée (biweight) dans un cadre de la régression isotonique. Les simulations de Bach (2018) montrent l'intérêt de ce type de perte par rapport à la ℓ_1 et ℓ_2 en présence de données corrompues. Un exemple de résultat est présenté dans la Figure 2. La perte biweight (rouge) est nettement plus proche du signal (noir). Nous avons confirmé ce résultat sur un grand nombre de simulations.

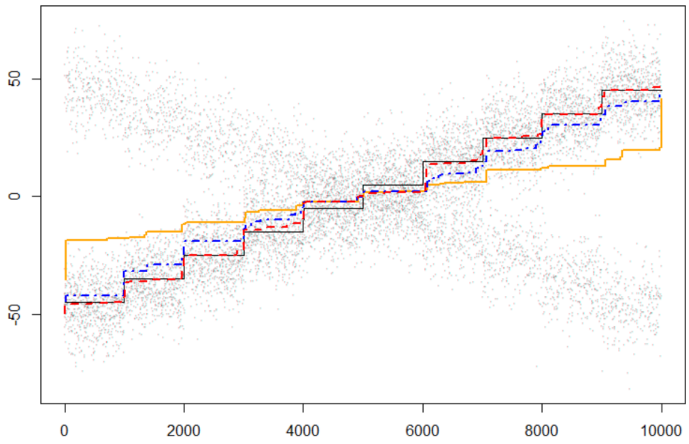

Figure 2: Régression isotonique avec 30% de données corrompues. Il y a $n = 10^5$ avec une variance de 10, le signal est la ligne noire. 30 % des points sont corrompus c'est-à-dire multiplié par -1 comme dans les simulations de Bach (2018). La ligne orange est le résultat d'isoreg avec la perte ℓ_2, la bleue pointillée le résultat avec la perte ℓ_1 reg et la rouge pointillée la perte biweight pénalisée (on obtient dans ce cas $D = 20$ segments).

5 Le package `gfpop`, utilisation et temps de calcul

Nous avons implémenté l'algorithme pour une structure de graphe générique en Rcpp. Le code est rendu accessible par le package `gfpop` disponible sur github[1]. Un exemple de code `R` avec notre package est donné ci-dessous pour un graphe de contraintes haut-bas.

```
n <- 1000
myData <- dataGenerator(n, c(0.1,0.3,0.5,0.8,1), c(1,2,1,3,1), 1)
myGraph <- graph(penalty = 2*log(n), type = "updown")
gfpop(vectData = myData, mygraph = myGraph, type = "gauss")
```

De nombreux graphes de contraintes sont possibles. Plusieurs pertes sont disponibles. Le temps de calcul dépend fortement des contraintes et de la perte utilisée. Par exemple avec

[1] `https://github.com/vrunge/gfpop`

une perte biweight et $\beta = 2\log(n)$ on a les temps d'exécution moyen sur 10 simulations :

	$K = 3$ (biweight)			$K = +\infty$ (ℓ_2)		
n	non contraint	isotonique	haut-bas	non contraint	isotonique	haut-bas
10^4	0.027	4.27	0.15	0.019	0.056	0.11
10^5	0.33	·	1.96	0.18	0.52	1.15
10^6	6.43	·	37.1	1.88	5.63	12.9

References

Bach, F. (2018). Efficient algorithms for non-convex isotonic regression through submodular optimization. In *Advances in neural information processing systems* (pp. 1–10).

Baranowski, R., & Fryzlewicz, P. (2014). wbs: Wild binary segmentation for multiple change-point detection, 2014. *R package version*, *1*.

Barlow, R. E. (1972). Statistical inference under order restrictions; the theory and application of isotonic regression.

Best, M. J., & Chakravarti, N. (1990). Active set algorithms for isotonic regression; a unifying framework. *Mathematical Programming*, *47*(1-3), 425–439.

Fearnhead, P., & Rigaill, G. (2018). Changepoint detection in the presence of outliers. *Journal of the American Statistical Association*, 1–15.

Hocking, T., Rigaill, G., & Bourque, G. (2015, 07–09 Jul). Peakseg: constrained optimal segmentation and supervised penalty learning for peak detection in count data. In F. Bach & D. Blei (Eds.), *Proceedings of the 32nd international conference on machine learning* (Vol. 37, pp. 324–332). Lille, France: PMLR. Retrieved from `http://proceedings.mlr.press/v37/hocking15.html`

Hocking, T. D., Rigaill, G., Fearnhead, P., & Bourque, G. (2017). A log-linear time algorithm for constrained changepoint detection. *arXiv preprint arXiv:1703.03352*.

Jewell, S., Hocking, T. D., Fearnhead, P., & Witten, D. (2018). Fast nonconvex deconvolution of calcium imaging data. *arXiv preprint arXiv:1802.07380*.

Killick, R., Fearnhead, P., & Eckley, I. A. (2012). Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, *107*(500), 1590–1598.

Maidstone, R., Hocking, T., Rigaill, G., & Fearnhead, P. (2017). On optimal multiple changepoint algorithms for large data. *Statistics and computing*, *27*(2), 519–533.

Rigaill, G. (2015). A pruned dynamic programming algorithm to recover the best segmentations with 1 to k_max change-points. *Journal de la Société Française de Statistique*, *156*(4), 180–205.

Stout, Q. F. (2008). Unimodal regression via prefix isotonic regression. *Computational Statistics & Data Analysis*, *53*(2), 289–297.