Une méthode statistique pour détecter des ruptures multiples dans un arbre.

Solène Thépaut[1] & Guillem Rigaill[2,3]

[1]*Laboratoire de Mathématiques d'Orsay, Université Paris-Sud, CNRS, Université Paris-Saclay, 91405 Orsay, France*
solene.thepaut@u-psud.fr
[2]*Laboratoire de Mathématiques et Modélisation d'Évry, Université d'Évry Val d'Essonne*
[3]*Institute of Plant Sciences Paris Saclay IPS2, CNRS, INRA, Université Paris-Sud, Université d'Évry, Université Paris-Saclay, Gif sur Yvette, France*
guillem.rigaill@inra.fr

Résumé. Nous considérons le problème de détection de ruptures multiples dans la moyenne des nœuds d'un arbre. Ce problème est motivé par des applications en écologie où des mesures de diversité sont faites en n points d'un réseau de rivières. Ce réseau de rivières est représenté par un arbre. L'objectif est d'identifier des sous-arbres où les fluctuations d'abondance d'une espèce sont synchrones.

Nous proposons d'inférer la position des ruptures et le signal par minimisation d'un risque empirique pénalisé. Nous dérivons une pénalité adaptée au problème et contrôlant le risque au travers d'une inégalité oracle non-asymptotique. Nous proposons deux algorithmes de programmation dynamique élagués pour retrouver la segmentation optimisant ce critère. Nous montrons empiriquement que leur complexité est en moyenne de $O(n^2)$ ou moins avec n le nombre de nœuds de l'arbre. Nous avons testé le comportement de notre approche sur des simulations et sur notre jeu de données écologique.

Mots-clés. Détection de ruptures. Arbre. Programmation dynamique élaguée. Sélection de modèle.

Abstract. We study the problem of detecting multiple changes in the mean of the nodes of a tree. This problem is motivated by application in ecology where diversity measurements are made at n points of a river system. The river system is represented as a tree. The goal is to detect sub-trees where the diversity is abnormally high or low.

We propose to infer the signal and the position of the changes by minimizing a penalized empirical risk. We propose a penalty satisfying a non-asymptotic oracle inequality. We propose two pruned dynamic programming algorithms to solve this problem. We empirically show that their complexity is on average $O(n^2)$ or less with n the number of nodes of the tree. We tested our approach on simulations and used it on our ecological dataset.

Keywords. Multiple changepoint models. Tree. Dynamic programming. Model selection

1 Définition et modèle

1.1 Arbres et segmentations

Nous considérons un arbre $\mathcal{T}$ avec n nœuds $X_1, ...X_n$. Une segmentation en D morceaux de $\mathcal{T}$ est une partition de ces nœuds en D sous-arbres. Nous appelons $\mathcal{M}_{\mathcal{T}}^D$ l'ensemble de ces segmentations. Il y en a $\binom{n-1}{d-1}$. Nous appelons $\mathcal{M}_{\mathcal{T}} = \bigcup_D \mathcal{M}_{\mathcal{T}}^D$ l'ensemble de toutes les segmentations. Il y en a 2^{n-1}.

1.2 Modèle

Dans le cas des données simulées, étant donnée une segmentation m de $\mathcal{T}$ en D segments, notés m_d, nous considérons le modèle:

$$\forall\ i\ \in\ m_d \quad X_i \sim \mathcal{N}(\mu_{m_d}, \sigma^2)\ i.i.d$$

En pratique les sous-arbres et le nombre de sous-arbres n'est pas connu. Il faut donc l'inférer. C'est un problème difficile car il y a beaucoup de segmentations à envisager.

2 Inférence

2.1 Minimisation d'un risque empirique pénalisé

Nous proposons de faire l'inférence par minimisation d'un risque empirique pénalisé.

$$\min_{D, m \in \mathcal{M}_{\mathcal{T}}^D} \left(pen(D) + \sum_{d=1}^{D} \sum_{i \in m_d} (X_i - \bar{X}_{m_d})^2 \right),$$

où $\bar{X}_{m_d}$ est la moyenne empirique du segment m_d.

Nous justifions théoriquement une pénalité de la forme $pen(D) = D\sigma^2(c_1 log(n/D) + c_2)$, où c_1 et c_2 sont des constantes, au travers d'une inégalité oracle non-asymptotique. La preuve est inspirée de celle de Lebarbier (2005) dans le cas d'une chaîne.

2.2 Résolution algorithmique

D'un point de vue algorithmique un problème similaire a été étudié par Maravalle et al. (1997). Ils montrent notamment que la difficulté du problème dépend fortement de la forme de la fonction objective. Certaines formes sont pathologiques et donnent des problèmes NP-complets. Toutefois, ils n'étudient pas directement la fonction objectif qui nous intéressent : $\sum_{i \in m_d} (X_i - \bar{X}_{m_d})^2$. Par ailleurs, ils ne proposent pas de critère pour choisir D. Ils supposent D connu.

Nous résolvons le problème (2.1) en résolvant de manière itérative des problèmes plus simples de la forme suivante:

$$F(\lambda) = \min_{D, m \in \mathcal{M}_T^D} \left(\lambda D + \sum_{d=1}^{D} \sum_{i \in m_d} (X_i - \bar{X}_{m_d})^2 \right).$$

La pénalité du problème (2.2) est linéaire en le nombre de sous-arbres et non pas concave comme dans (2.1). Nous proposons deux algorithmes exacts de programmation dynamique élaguée pour résoudre (2.2). Ces algorithmes sont inspirés d'algorithmes de programmation dynamique élagués proposés pour des chaînes (Killick et al. (2012); Rigaill (2015); Maidstone et al. (2017)). Il faut noter que les principes de ces algorithmes ne peuvent s'appliquer naïvement dans le cas d'un arbre. En effet dans une chaîne le nombre de sous-arbre est en $O(n^2)$. Ce n'est pas le cas dans un arbre de manière général. Par exemple, dans le cas d'arbres binaires équilibrés le nombre de sous-arbres croit comme la suite suivante : $u_0 = 1$, $u_{n+1} = (u_n + 1)^2$. Cela pose un certain nombre de problèmes combinatoires que nous résolvons. Nous montrons empiriquement, c'est à dire par des simulations, que nos deux algorithmes ont une complexité en temps en moyenne de $O(n^2)$ ou moins. Cela suggère un élagage efficace relativement au grand nombre de sous-arbres à considérer. Cela suggère également que la forme de notre fonction objectif n'est pas pathologique au sens de Maravalle et al. (1997).

Pour optimiser (2.1) nous itérons l'utilisation de nos algorithmes pour résoudre (2.2) pour des valeurs particulières de λ. Par concavité de la pénalité, nous démontrons qu'il existe un λ tel que le minimiseur de (2.1) est le minimiseur de (2.2). Par ailleurs, toujours par concavité nous montrons que chacune de nos itérations permet d'améliorer le risque pénalisée (2.1). Il faut noter que le problème (2.1) peut avoir des minimums locaux. Notre procédure itérative ne garantie pas la découverte du minimum global. Initialiser plusieurs fois notre itération permet en partie de répondre à ce problème. Tout cela justifie heuristiquement notre procédure itérative.

3 Remerciements

Nous remercions Christophe Giraud et Nicolas Verzelen pour de fructueuses discussions.

References

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.

Lebarbier, É. (2005). Detecting multiple change-points in the mean of gaussian process by model selection. *Signal processing*, *85*(4), 717–736.

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

Maravalle, M., Simeone, B., & Naldini, R. (1997). Clustering on trees. *Computational Statistics & Data Analysis*, *24*(2), 217–234.

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.