

# Maximum de vraisemblance, inférence bayésienne & Metropolis-Hastings.

**Professeur :** Christophe ANCEY

**Assistants :** Clemente GOTELLI, Mehrdad KIANI

## 1 Ajustement de lois de valeurs extrêmes sur des données (fin du TD précédent)

Nous allons maintenant faire un ajustement de loi de valeurs extrêmes sur les données de températures et de précipitations à Davos. Il s'agit de déterminer le type de loi de valeurs extrêmes (Weibull, Gumbel ou Fréchet) qui décrit le mieux le comportement des maxima de chaque série. Dans cet exercice, nous utiliserons directement la fonction `gevfit` qui estime les paramètres de la loi généralisée des valeurs extrêmes par la méthode du maximum de vraisemblance vue dans l'exercice précédent (voir TD3). Nous procédons de la manière suivante :

- définissez des blocs (classiquement des années) ;
- calculez le maximum de chaque bloc (donc par exemple le maximum annuel) ;
- utilisez ces maxima pour caler la loi des valeurs extrêmes ;
- tracez cette loi et comparez avec la courbe empirique.

### 1. Précipitations

De quel type (Gumbel, Fréchet ou Weibull) est la distribution ?

### 2. Températures

Le fichier possède plusieurs colonnes de données, les trois dernières colonnes représentent la température à 7 h, 13 h et 21 h. Choisissez-en une et trouvez la loi de valeurs extrêmes qui convient le mieux pour décrire la distribution des maxima.

## Inférence bayésienne, présentation de la méthode

Lors du dernier TD nous avons vu comment trouver les paramètres  $\theta$  d'une loi pour décrire au mieux  $n$  observations  $d_i$ . Les méthodes employées étaient la méthode des moments qui ajuste les moments des données à ceux de la loi et celle du maximum de vraisemblance qui consiste à maximiser  $\text{Prob}(d_1, d_2, \dots, d_n | \theta)$ . Aujourd'hui nous allons voir une autre méthode basée sur le théorème de Bayes :

$$\text{Prob}(\theta | \mathbf{d}) = \frac{\text{Prob}(\mathbf{d} | \theta) \text{Prob}(\theta)}{\int \text{Prob}(\mathbf{d} | \theta) d\theta} \quad (1)$$

En d'autres termes, la probabilité que les paramètres soient  $\theta$  est proportionnelle à la vraisemblance  $\text{Prob}(\mathbf{d} | \theta)$  pondérée par  $\text{Prob}(\theta)$ .  $\text{Prob}(\theta)$  est la probabilité a priori d'avoir  $\theta$ , et est appelée le prior. Cette loi  $\text{Prob}(\theta)$  est déduite d'une connaissance experte ou d'un précédent calage, si ni l'une ni l'autre de ces informations n'est disponible on utilise une loi uniforme.

L'algorithme d'Hastings-Metropolis permet de trouver quels paramètres  $\theta$  maximisent l'équation 1, pour cela on se promène sur  $Q(\theta) = \text{Prob}(\theta | \mathbf{d})$  par sauts successifs. Si  $Q(\theta_{i+1}) \geq Q(\theta_i)$  on accepte toujours le saut car on se dirige vers des vraisemblances plus élevées, dans le cas contraire on accepte le saut avec une certaine probabilité. Peu à peu, saut après saut, nous allons donc approcher des maxima de  $Q(\theta)$  sans pour autant avoir une probabilité nulle de redescendre (ce qui permet de sortir d'un éventuel maxima secondaire par exemple). Les sauts sont contrôlés par la loi de probabilité instrumentale que l'on note ici  $q$  et que l'on va considérer, pour simplifier, comme symétrique (i.e.  $q(\theta_{i+1} | \theta_i) = q(\theta_i | \theta_{i+1}) = q(|\theta_{i+1} - \theta_i|)$ ). De manière plus explicite :

1. On part d'un état  $\theta_0$ .
2. On tire une valeur candidate selon la loi  $q(\theta_{i+1} | \theta_i)$ .
3. On définit un taux d'acceptation  $r$  de la manière suivante :

$$r = \min \left[ \frac{Q(\theta_{i+1})}{Q(\theta_i)}, 1 \right]$$

A cette étape, on voit que l'intégrale se simplifie dans le quotient. Notons aussi que cette expression est plus complexe lorsque la loi de probabilité instrumentale n'est pas symétrique.

4. On accepte les paramètres  $\theta_{i+1}$  (c'est-à-dire on fait le saut) avec une probabilité  $r$ .
5. On répète la procédure jusqu'à convergence.

## 2 Algorithme d'Hastings-Metropolis

1. Pour commencer, générez  $N = 20$  valeurs tirées aléatoirement selon une loi de poisson de paramètre  $\lambda = 2$ . Ces valeurs représentent les données sur lesquelles nous allons caler une loi de poisson et essayer de retrouver le paramètre  $\lambda = 2$  utilisé pour générer ces valeurs. [**poissrnd**]
2. Codez l'algorithme d'Hastings-Metropolis. Pour cela faites une boucle qui va de 1 jusqu'à **nPas**, le nombre de sauts autorisé (p. ex. 3000). A chaque itération  $i$ , calculez le nouveau candidat potentiel  $\theta_p$  ainsi que le paramètre d'acceptation  $r$  correspondant. Comme loi instrumentale  $q$ , on prendra une loi normale d'espérance  $\theta_i$  et d'écart-type 0,01. Considérant que nous n'avons pas de connaissances a priori, le prior sera pris uniforme dans un premier temps (par exemple selon une loi  $U(0,100)$ ). Pour éviter des erreurs numériques lors du calcul de  $r$ , utilisez la log-vraisemblance :

$$r = \frac{Q(\theta_p)}{Q(\theta_i)} = \exp [l(\theta_p) + \ln \text{Prob}(\theta_p) - l(\theta_i) - \ln \text{Prob}(\theta_i)]$$

où  $l(\theta)$  est la log-vraisemblance.

Une fois  $\theta_p$  déterminé et  $r$  calculé en conséquence, il faut effectuer un test : si  $r \geq 1$  on accepte dans tous les cas la nouvelle position (car on monte) et  $\theta_{i+1} = \theta_p$ , sinon on accepte  $\theta_{i+1}$  avec une probabilité  $r$  uniquement. Pour coder le taux d'acceptation  $r < 1$  on procède de la manière suivante : on tire une valeur aléatoire  $u$  d'une distribution uniforme (cela se fait facilement dans Matlab avec la commande **rand**) ; si  $r > u$  alors on accepte  $\theta_p$ , si  $r < u$  on reste sur la position  $\theta_i$ .

3. A chaque itération, enregistrez la position  $\theta_i$  dans un vecteur **D<sub>o</sub>**, lorsque les *nPas* sont effectués faites un graph de ce vecteur et calculez la moyenne de celui-ci (restreignez cette moyenne à la partie convergente du vecteur, par exemple les 1500 derniers points).
4. Comparez le résultat obtenu avec ce que l'on obtient en utilisant la commande Matlab **poissfit** sur le même set de donnée.
5. Introduisez à présent un prior de type gaussien centré en 2 et d'écart type 0,1. Calculez ensuite le vecteur **D** correspondant et superposez sur un graph **D** et **D<sub>o</sub>**, qu'observez-vous ?
6. Changez  $N$ , augmentez le à une valeur supérieur (par exemple 100) et refaites cette comparaison (prior uniforme vs prior centré en 2) ; augmentez encore  $N$  à 200, que remarquez-vous ?
7. Introduisez un prior de type gaussien centré en 0,5 et d'écart type 0,1. Que constatez-vous pour  $N = 20, 100, 200$  ? Faites la même chose pour un prior complètement erroné (par exemple centré en 5 et d'écart type 1), que constatez-vous pour  $N = 20, 100, 200$  ?