Simulation et Monte Carlo : Sélection de variables

Théophile Froment, Emma Sarfati, Paul Vialard

ENSAE Paris

24 Avril 2020

Plan

- Introduction
- Algorithme de type *cross-entropy*
- Algorithme de type recuit simulé (basé sur un noyau de type Metropolis)
- Algorithme de type recuit simulé (basé sur un noyau de type Gibbs)
- Conclusion

Introduction

- Sélection de variables :
  - Choisir parmi p prédicteurs ceux réellement nécessaires au problème.
  - Problème difficile lorsque p est grand
- Trois approches différentes pour sélectionner le "meilleur" modèle :
  - *Cross-entropy*
  - Recuit simulé (Metropolis)
  - Recuit simulé (Gibbs)

Méthode *cross-entropy*

- **Idée de base** : on imagine notre set de variables $(X_i)_{i=1\cdots p}$ comme une matrice de taille $n \times p$, distribuées selon une loi $\mathcal{B}(\theta_i)$.
- p : nombre de colonnes de notre base prédictive et n : nombre de lignes de notre dataframe.

$$\begin{pmatrix} \mathcal{B}(\theta_1) & \mathcal{B}(\theta_2) & \mathcal{B}(\theta_3) & \cdots & \mathcal{B}(\theta_p) \\ \mathcal{B}(\theta_1) & \mathcal{B}(\theta_2) & \mathcal{B}(\theta_3) & \cdots & \mathcal{B}(\theta_p) \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \mathcal{B}(\theta_1) & \mathcal{B}(\theta_2) & \mathcal{B}(\theta_3) & \cdots & \mathcal{B}(\theta_p) \end{pmatrix}$$

- **Résultat** : à la fin de l'algorithme, on obtient des paramètres $\theta + /-$ proches de 1 ou 0.
- **Choix** : on sélectionne la variable X_i si θ_i proche de 1.

Cross-entropy : l'algorithme

- **Initialisation** : on créé une matrice de Bernoulli de taille $n \times p$ avec des paramètres $\frac{1}{2}$.

- **Boucle** :
  - n régressions linéaires, calculs de critère.
  - Sélection meilleurs critères.
  - Pour chaque colonne, moyenne sur les lignes ayant donné les "meilleurs" AIC/BIC.
  - Update des paramètres, itération de la boucle.
  - Critère d'arrêt : la distance entre les deux derniers vecteurs de paramètres est négligeable.

- **Output** : le vecteur final des paramètres. On sélectionne la variable si θ proche de 1.

Cross-entropy : convergence avec sélection de 10 critères

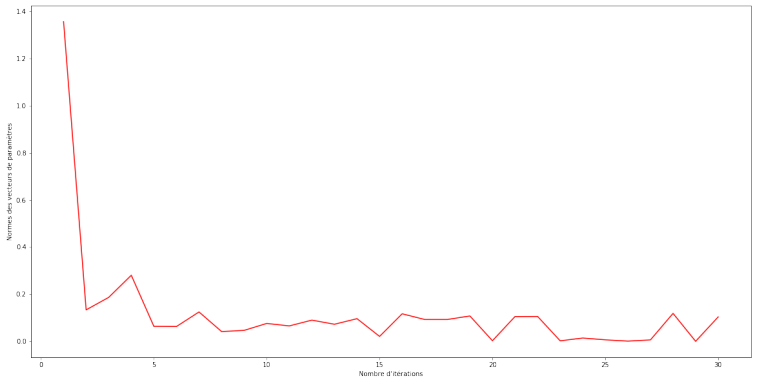


Figure: Convergence de l'algorithme CE basé sur l'AIC

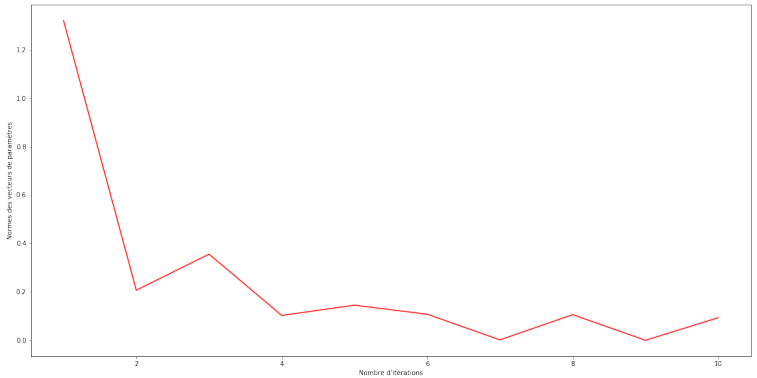


Figure: Convergence de l'algorithme CE basé sur le BIC

Cross-entropy : limites de l'approche

- Sélection finale subjective : on a rarement que des 0 ou des 1.
- Mesure de l'erreur difficilement réalisable.
- Convergence de l'algorithme lente, dépend du nombre de critères.
- Plus rapide avec le BIC que l'AIC : moins d'itérations pour atteindre une distance négligeable de vecteurs de paramètres.

Recuit simulé (Metropolis) : l'algorithme

- On associe la liste des features à une liste de 0 et de 1
- Inputs : k_{max}, T_0, E(), e_{max}
- Probabilité d'acceptation : $e^{-\delta E/T}$
- Variation de la température : $T_{i+1} = \lambda T_i$ avec $\lambda = 0.99$

Recuit simulé (Metropolis) : Résultats 1/4

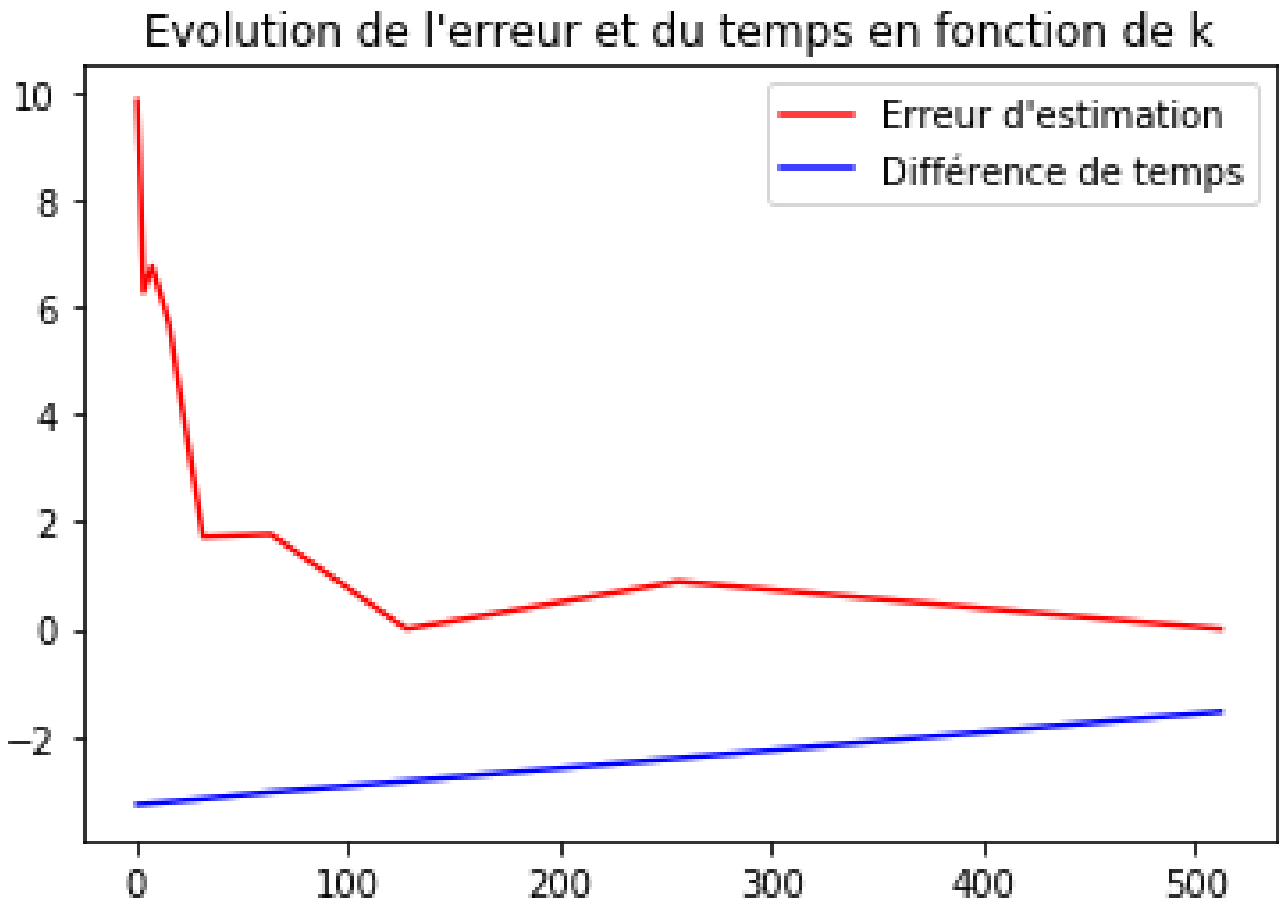


Figure: Evolution du temps et de l'erreur d'estimation en fonction du paramètre k

Recuit simulé (Metropolis) : Résultats 2/4

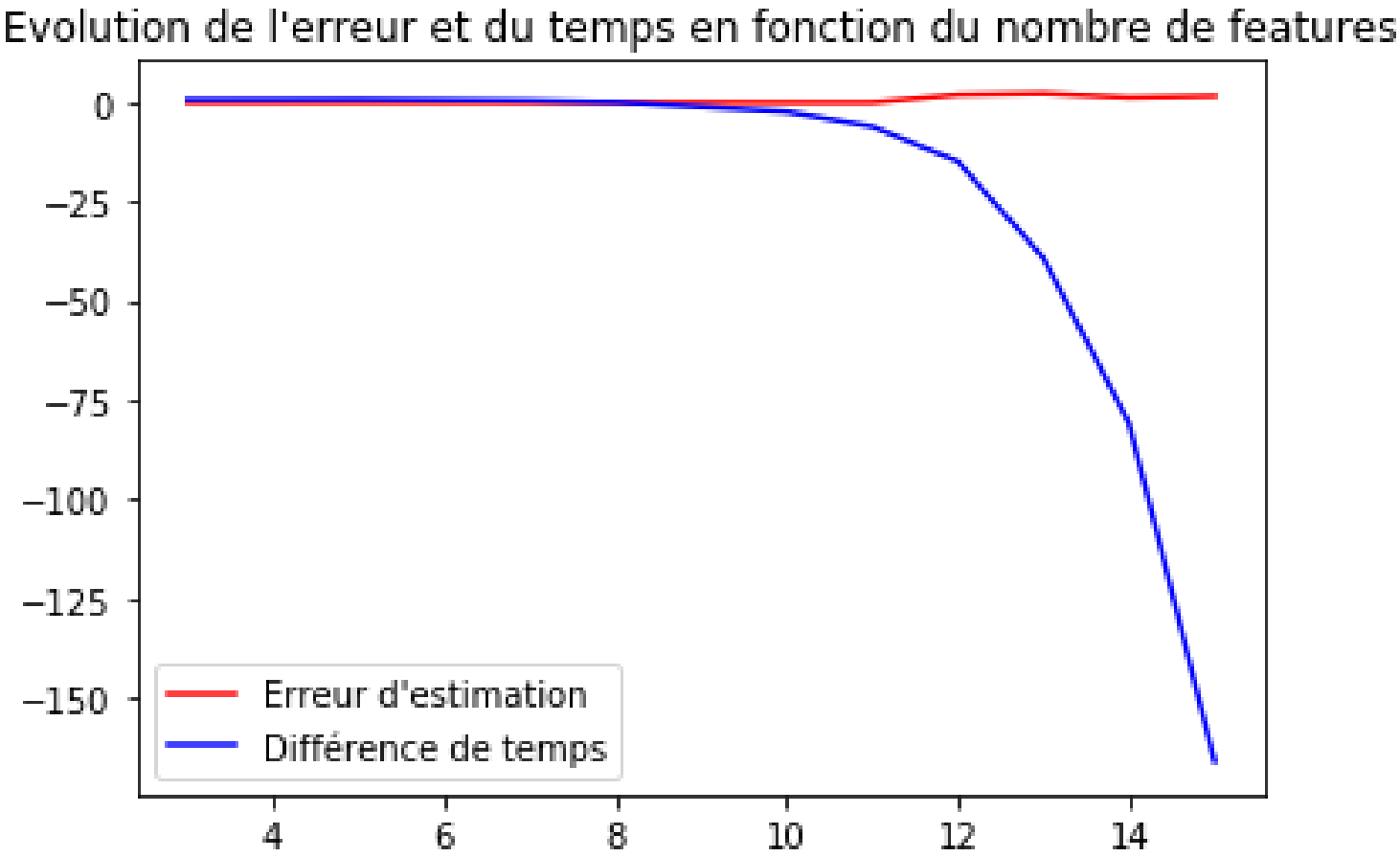


Figure: Evolution du temps et de l'erreur d'estimation en fonction du nombre de features

Recuit simulé (Metropolis) : Résultats 3/4

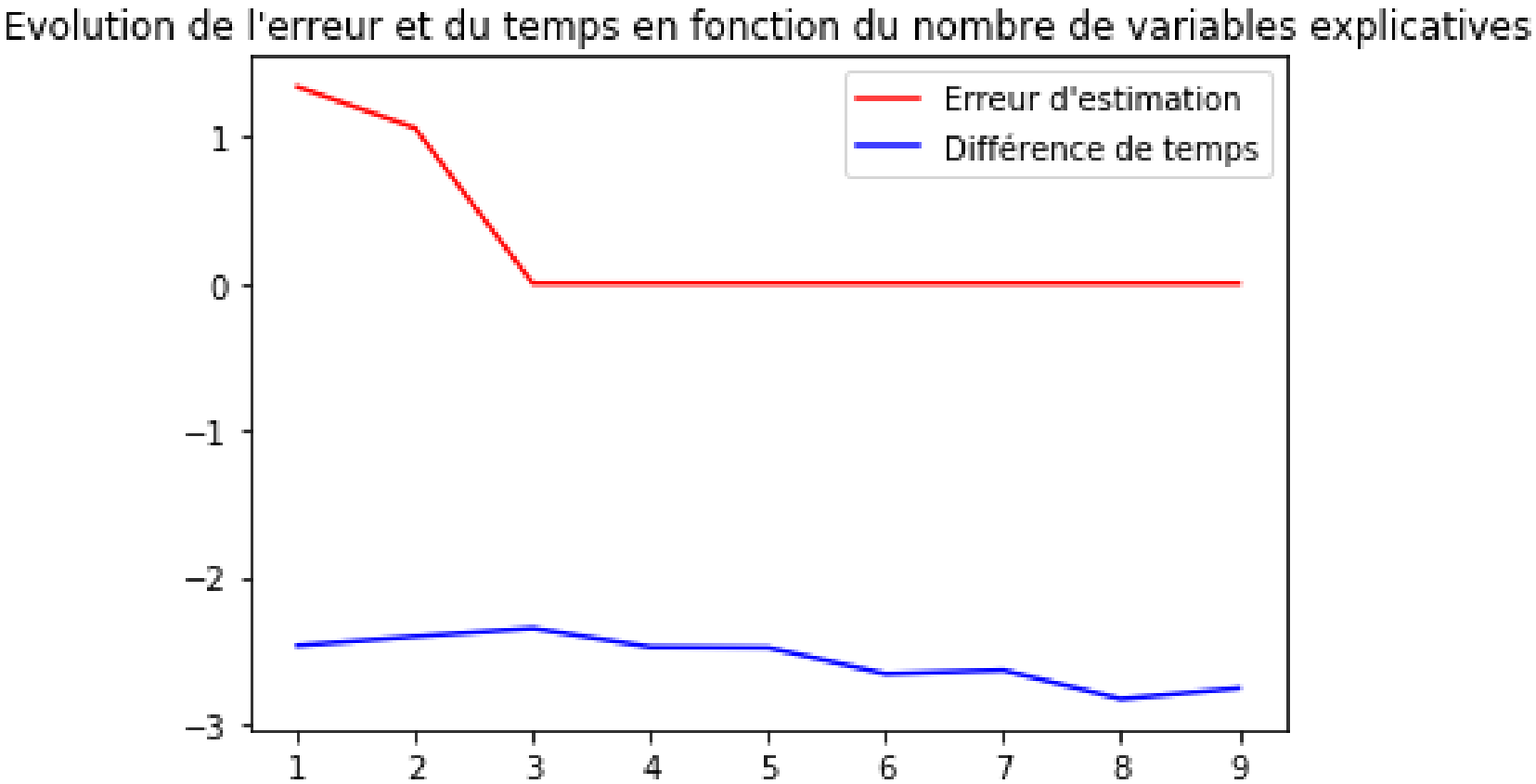


Figure: Evolution du temps et de l'erreur d'estimation en fonction du nombre de variables explicatives

Recuit simulé (Metropolis) : Résultats 4/4

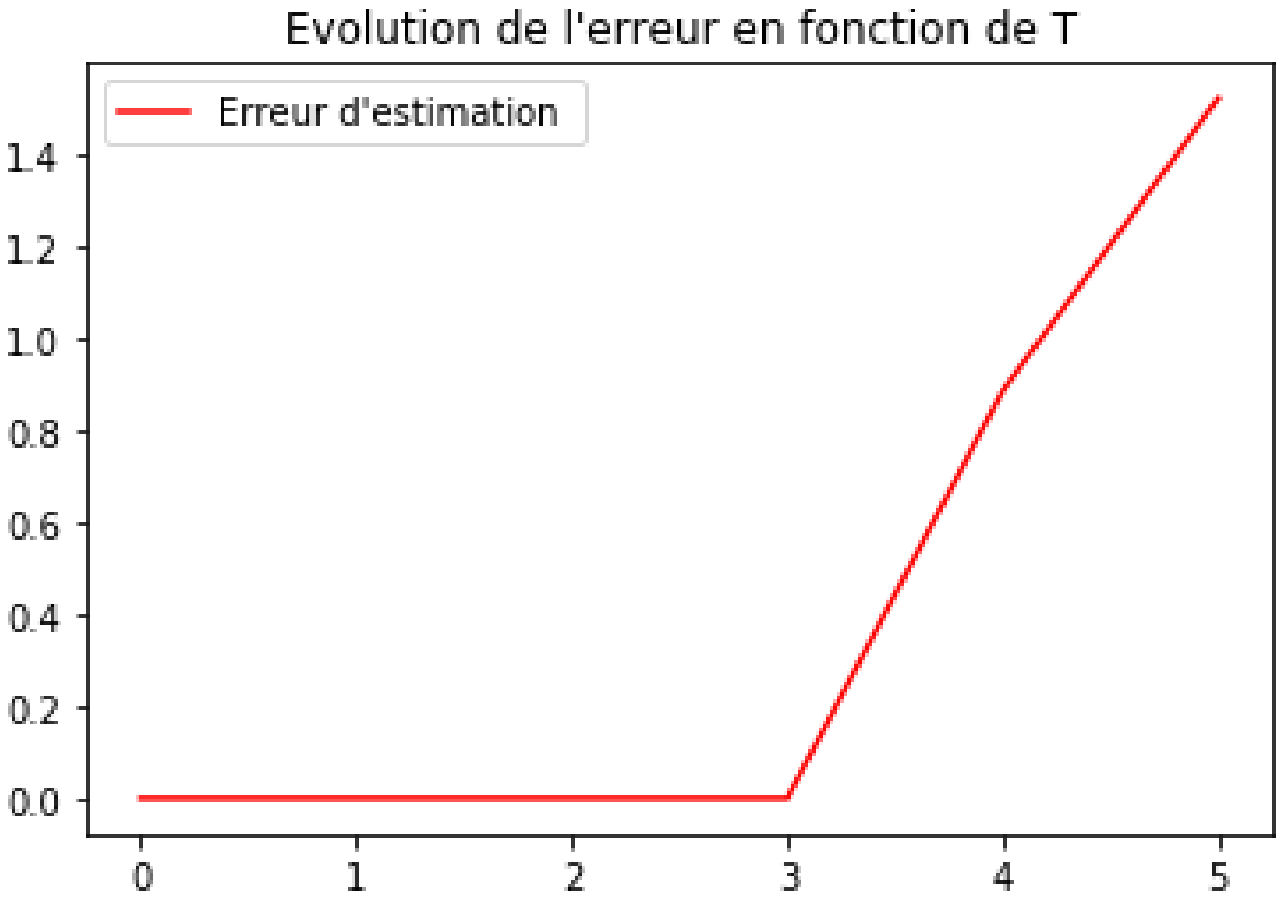


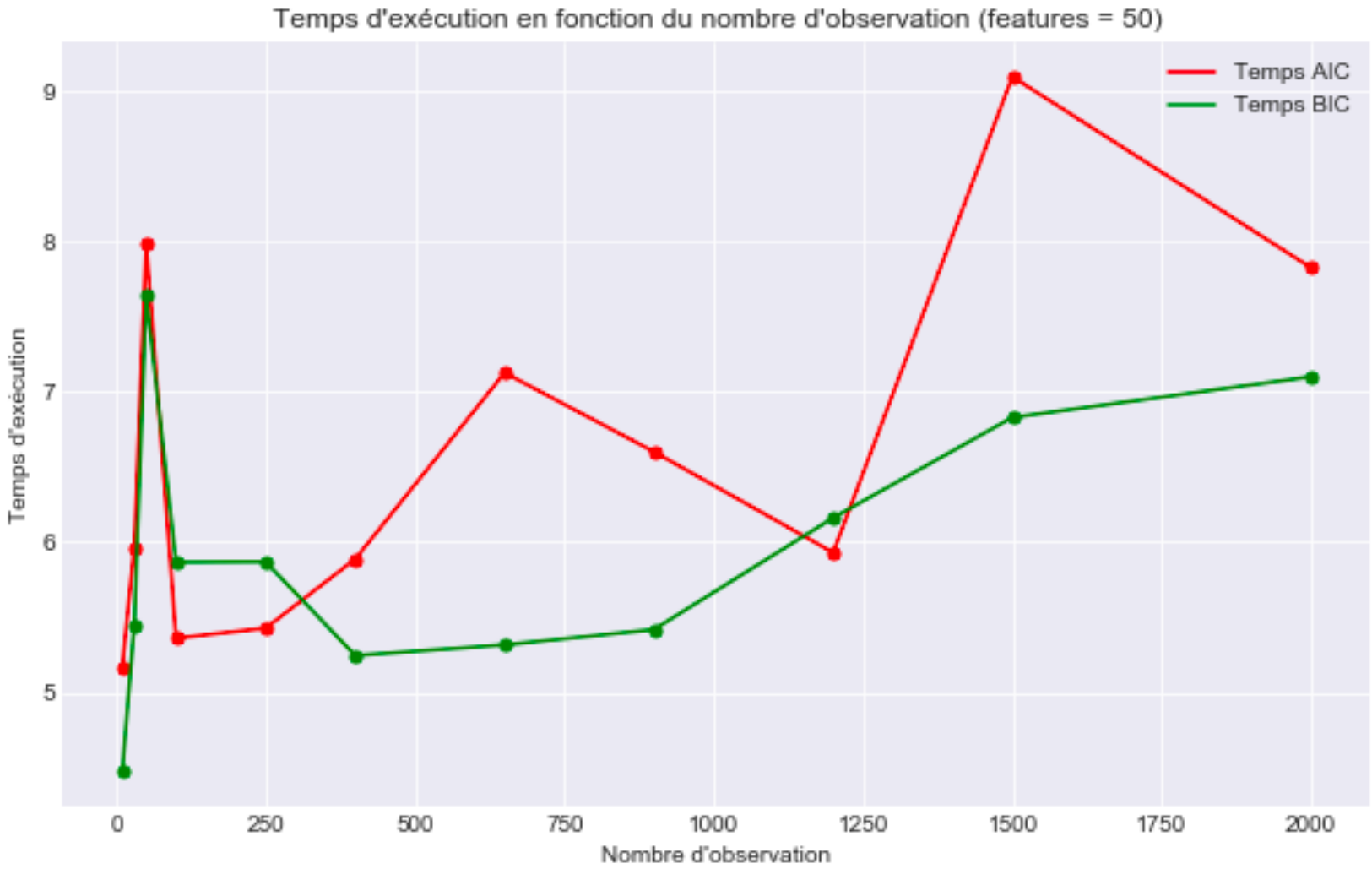
Figure: Evolution de l'erreur d'estimation en fonction de la température (échelle logarithmique)

Recuit simulé (Gibbs)

- Initialisation avec un vecteur ne contenant que des 1.
- On modifie la première composante en 0 et on teste si le nouveau vecteur diminiuer la valeur de l'AIC ou BIC.
- Si oui, on garde ce nouveau vecteur, on modifie la composante suivante et on réitère. Si non, on remet la valeur en 1, on modifie la compostante suivante et on réitère.
- Une fois le nouveau vecteur calculé, on effectue l'algortihme du recuit simulé.
- Ces opérations sont effectués jusqu'à ce que la température minimale soit atteinte.

Recuit simulé (Gibbs) : Résultats (1/2)

Recuit simulé (Gibbs) : Résultats (2/2)



Conclusion

- Les 3 méthodes sont plus efficaces que l'approche classique qui consiste à calculer le critère pour toutes les combinaisons possibles de régresseurs.
- La méthode de *cross-entropy* est plus longue que les deux autres.
- Les deux méthodes utilisant un algorithme de type recuit simulé sont équivalentes.