

# TP 2 : Optimisation Bayésienne et Modèles Bayésiens à Noyau

ABCI Fella  
01/04/2025

## Partie 1 : Optimisation Bayésienne

### Fondements théoriques

#### Q1. Le principe de l'optimisation bayésienne

L'optimisation bayésienne c'est une méthode pour trouver le meilleur résultat possible (ex : un maximum ou un minimum) d'une fonction qu'on ne connaît pas exactement et qui est coûteuse à évaluer.

Par exemple : on veut tester la meilleure combinaison température + humidité pour maximiser un rendement agricole. Chaque test prend du temps (ou de l'argent), donc on veut faire le moins d'essais possible tout en trouvant la meilleure combinaison.

Elle fonctionne en 3 étapes :

- Elle modélise la fonction inconnue (le rendement en fonction des paramètres) avec un modèle probabiliste souvent un processus gaussien
- Elle choisit le prochain point à tester à l'aide d'une fonction d'acquisition, qui équilibre exploration (essayer de nouveaux endroits) et exploitation (creuser là où ça marche déjà bien).
- Elle met à jour le modèle avec le nouveau résultat et recommence

#### Q2. Définir et expliquer les processus gaussiens

Un processus gaussien est un outil statistique très puissant qu'on peut voir comme une extension de la loi normale gaussienne mais pour des fonctions entières

- Une variable gaussienne = un nombre avec incertitude
- Un processus gaussien = une fonction avec une incertitude à chaque point
- > Pourquoi c'est utile ?

Quand on connaît pas une fonction le GP permet de :

- Faire une prédiction à n'importe quel point
- Donner une incertitude sur cette prédiction

Et dans l'optimisation bayésienne on l'utilise comme modèle de la fonction objective (ce qu'on veut max ou min) parce que :

- il fonctionne bien même avec un peu de données
- il donne une estimation de ce qu'on sait déjà et ce qu'on ne sait pas

#### Q3. Décrire les principales fonctions d'acquisition

**Une fonction d'acquisition** est comme un GPS intelligent qui aide l'optimisation bayésienne à choisir le prochain point à tester elle regarde :

- Ce qu'on sait déjà (le modèle)
- L'incertitude
- Ce qu'on pourrait gagner à tester un nouveau point

Objectif : Trouver un bon compromis entre

- Exploitation : aller là où le modèle pense que c'est déjà bon
- Exploration : essayer de nouveaux endroits où on a peu de données

**Les fonctions d'acquisition:**

##### 1. Expected Improvement (EI) :

- Elle regarde l'espérance du gain par rapport au meilleur point connu.
- Si un point a une forte chance d'être meilleur que le meilleur actuel, il est testé.
- Bon équilibre entre exploration et exploitation.

## 2. Upper Confidence Bound (UCB) :

- Elle prend :  
Prédiction +  $k \times$  Incertitude  
(k est un paramètre qu'on peut régler)
- Si l'incertitude est grande, elle teste même si la prédiction n'est pas top.
- Très exploratoire, surtout quand k est grand.

## 3. Probability of Improvement (PI) :

- Elle mesure la probabilité que le point soit meilleur que le meilleur actuel.
- Moins utilisée car parfois trop timide (exploite trop).

## Exercice 2 : Implémentation et applications

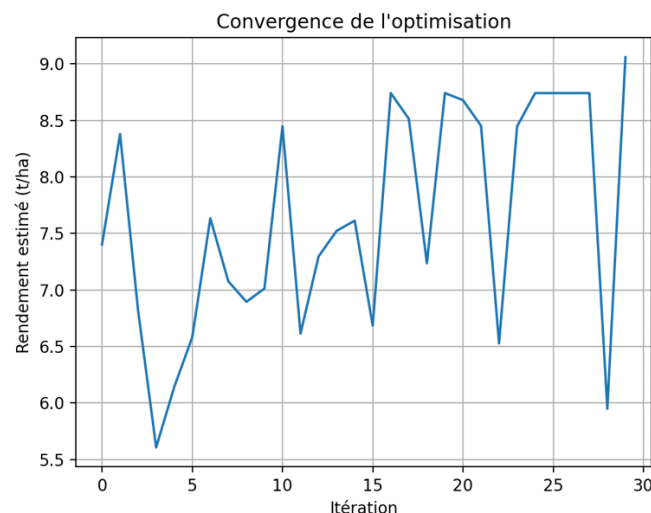
Q4. Resultat obtenu :

**Meilleure humidité : 89.57788776715802**

**Meilleure température : 30.949603422090313**

**Rendement max estimé : 9.058108768899977**

- ➔ J'ai eu un warning qui indique que l'algo est tombé plusieurs fois sur le même point optimal pendant ses essais ce qui est normal et montre qu'il a bien trouvé un point stable et prometteur



- ➔ La courbe montre l'évolution du rendement estimé au fil des 30 itérations de l'optimisation bayésienne
- On remarque que le rendement fluctue fortement car l'algo explore différentes zones de l'espace humidité/température
  - Puis les rendements estimés augmentent progressivement vers des valeurs élevées (entre 8 et 9)
  - vers la fin la courbe est autour de 9 ce qui indique que l'optimisation a probablement trouvé une zone optimale

**Q5.** On a testé trois méthodes pour optimiser les hyperparamètres d'un modèle de Random Forest utilisé pour prédire le rendement agricole :

Résultat obtenu :

Grid Search →  $MSE = 0.331$

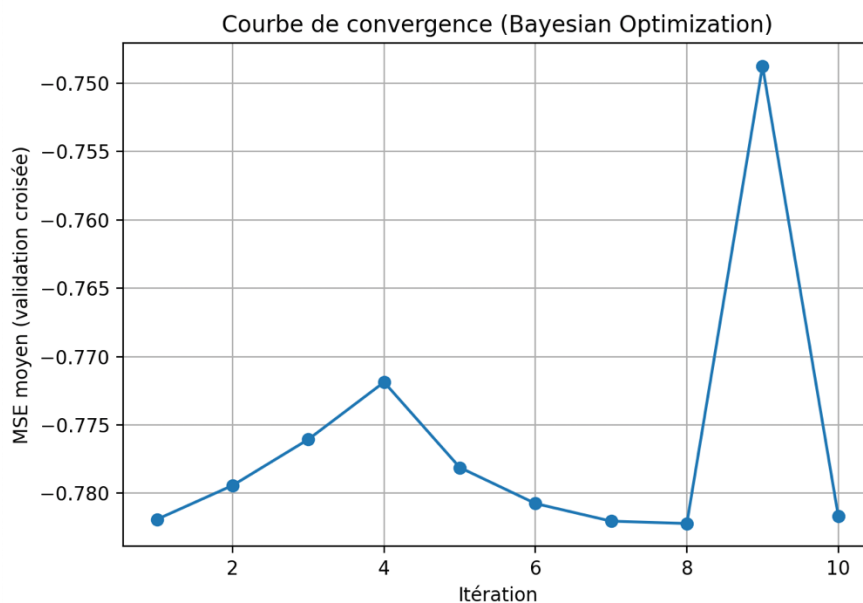
Random Search →  $MSE = 0.330$

Bayesian Opt. →  $MSE = 0.328$

->L'optimisation bayésienne obtient le meilleur résultat car elle choisit les points à tester de manière intelligente en s'appuyant sur le modèle probabiliste

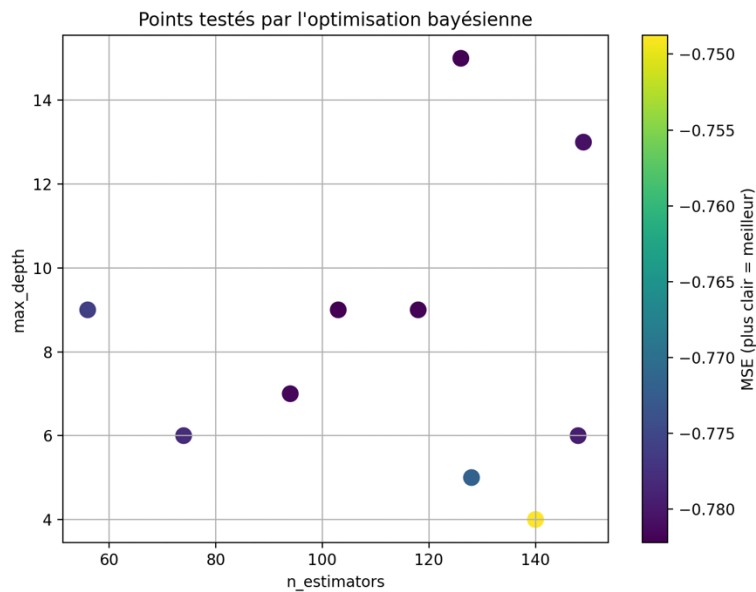
Contrairement à Grid Search qui teste systématiquement ou Random search qui teste au hasard, la méthode bayésienne réduit le nombre d'essais nécessaires tout en améliorant la performance du modèle

## **Q6. Visualisation du processus d'optimisation**



->La courbe montre l'évolution du MSE moyen au fil des itérations de l'optimisation bayésienne.

- On observe une baisse globale du MSE ce qui indique que l'algorithme améliore progressivement la qualité du modèle
  - Quelques sauts ou pics de MSE apparaissent (comme à l'itération 9) ce qui correspond à des phases d'exploration où l'algorithme teste des zones moins connues de l'espace
- ➔ Globalement le modèle converge vers une zone stable et performante.



->le graphique montre les combinaisons de `n_estimators` et `max_depth` testées par l'optimisation bayésienne.

- Les couleurs indiquent la qualité du modèle ->plus clair = meilleur
- On voit que l'algorithme teste d'abord un peu partout puis se concentre progressivement sur une région avec de bonnes performances (autour de `n_estimators`  $\approx$  140 et `max_depth`  $\approx$  4).
- ça illustre bien le compromis exploration / exploitation : d'abord découvrir l'espace puis exploiter ce qui fonctionne bien

## Q7. Analyse des avantages et limites de l'optimisation bayésienne face aux méthodes classiques

Avantages :

- Elle est intelligente, elle apprend de chaque essai pour mieux choisir le suivant
- Elle est efficace quand on a peu d'essais possibles (temps ou cout limité)
- elle gere bien les compromis explorations/exploitation : tester de nouveaux trucs vs ameliorer ce qui marche déjà

Limites :

- Elle est un peu plus complexe à comprendre et mettre en œuvre (besoin de packages comme `skpot` ou `GPyOpt`)
- Elle peut etre lente a chaque etape car elle fait beaucoup de calculs pour choisir le prochain point
- Elle est moins adaptée aux problemes avec beaucoup de dimensions (bcp d'hyperparametres a optimiser)

## Partie 2 : Modèles Bayésiens à Noyau

## Fondements théoriques

### Q8. Expliquer le concept d'inférence bayésienne

L'inférence bayésienne est une méthode pour mettre à jour ce qu'on pense (nos croyances) quand on reçoit de nouvelles informations.

Exemple :

On pense qu'un champ va produire 8 tonnes de maïs par hectare.

Mais après quelques tests on voit qu'en fait les résultats sont plutôt autour de 6 t/ha. on va mettre à jour notre croyance initiale en fonction des nouvelles données.

L'idée repose sur la **formule de Bayes** :

$$P(\text{modèle}|\text{données}) = \frac{P(\text{données}) \cdot P(\text{modèle})}{P(\text{données})}$$

- $P(\text{modèle})$  : ta croyance a priori (avant d'avoir les données)
- $P(\text{données} | \text{modèle})$  : la vraisemblance des données avec ton modèle
- $P(\text{modèle} | \text{données})$  : la croyance mise à jour (distribution a posteriori)

### Q9. Décrire la théorie des méthodes à noyau et leur lien avec les processus gaussiens

Une méthode à noyau permet de travailler sur des données complexes (pas linéaires) sans les transformer explicitement.

#### Comment ça fonctionne ?

Un noyau (ou kernel) est une fonction de similarité.

Exemples :

- Linéaire : mesure la corrélation simple.
- RBF (Radial Basis Function) : mesure si deux points sont proches ou éloignés.
- Polynomial : mesure des similarités complexes.

Lien avec les processus gaussiens :

Les processus gaussiens (GP) utilisent les noyaux pour :

- Définir la covariance entre les points,
- Et donc modéliser à quel point deux entrées sont liées.

Le choix du noyau dans un GP influence donc :

- La forme des prédictions,
- Et l'incertitude autour de ces prédictions.

### 10. Qu'est-ce qu'une distribution a priori et une distribution a posteriori ?

-Une distribution a priori (ou *prior*) représente ce qu'on croit ou suppose avant d'avoir vu les données.

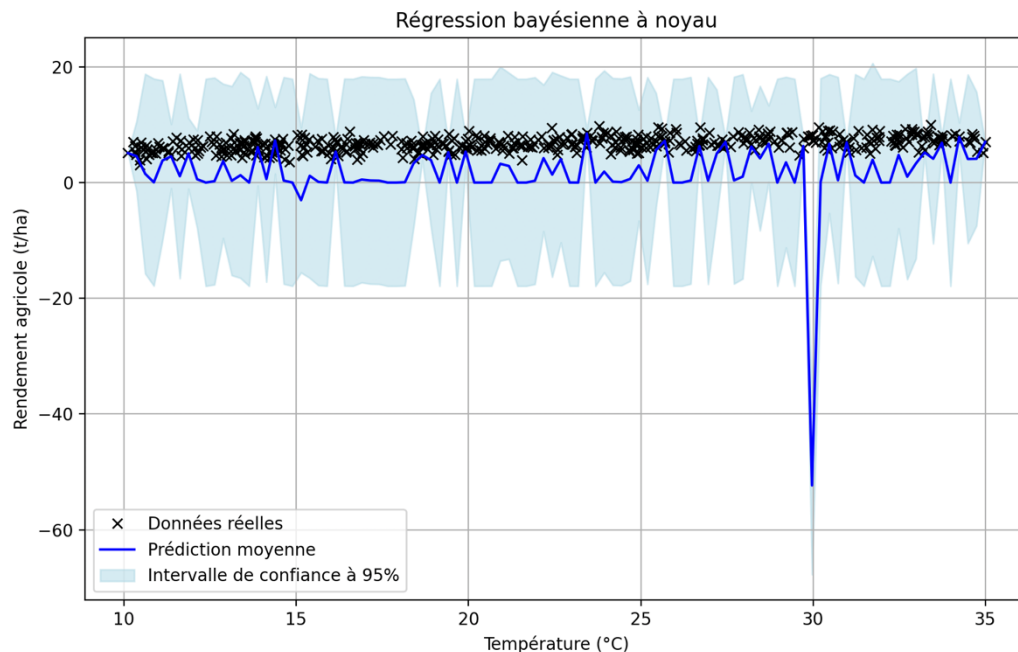
-Une distribution a posteriori (ou *posterior*) représente ce qu'on croit après avoir vu les données.

Elles sont liées par la formule de Bayes :

$$\text{Posterior} = \frac{\text{Vraisemblance} \cdot \text{A priori}}{\text{Evidence}}$$

### Q11. Implémentation d'une régression bayésienne à noyau

On utilise un modèle de régression bayésienne (comme un Gaussian Process Regressor)  
Prédire le rendement agricole en fonction de certaines variables



On remarque que :

Le modèle colle plutôt bien aux données globalement.

L'incertitude devient plus grande dans certaines zones, notamment autour de 30°C le modèle est moins sûr là.

Il y a un pic négatif très bas vers 30°C (c'est un artefact possible du modèle à corriger avec un autre noyau ou nettoyage des données mais ça reste un bon exemple de ce que peut faire le GP).

### Q12. Classification bayésienne à noyau pour prédire le type de sol

Résultat obtenu :

Précision (Bayesian GP) : 0.34

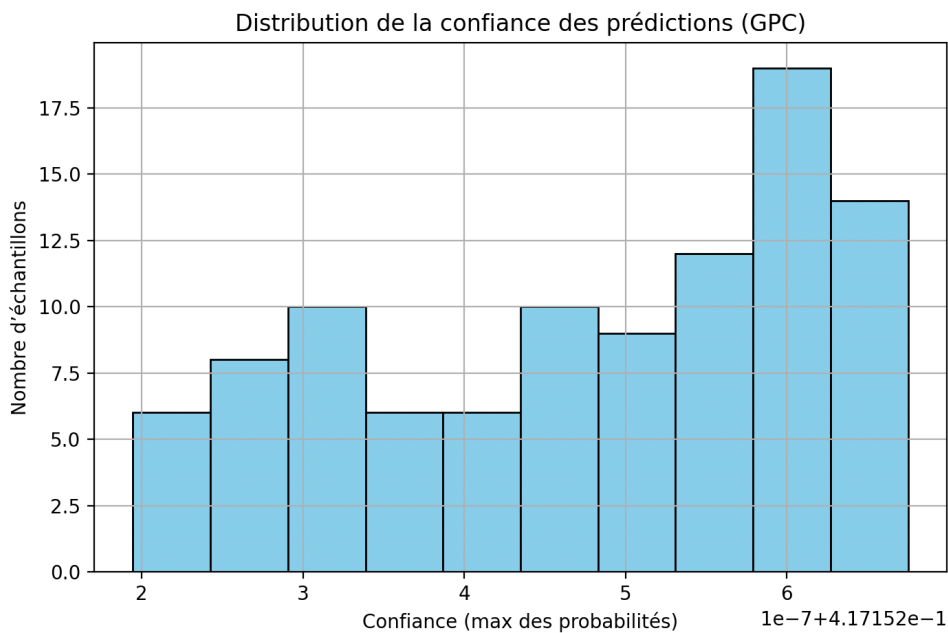
Précision (SVM classique) : 0.31

->les précisions sont faibles mais ça montre quand même que le modèle bayésien est légèrement plus précis

### Q13. Analyse de l'incertitude dans les prédictions

le modèle GaussianProcessClassifier ne donne pas juste une classe (argileux, sableux, limoneux),

il donne aussi les probabilités pour chaque classe → ce qui permet de mesurer l'incertitude.



On voit que la majorité des prédictions ont une confiance entre 0.4 et 0.6, ce qui confirme que le modèle est assez incertain (il ne donne pas souvent une classe avec une probabilité très élevée).

Ça peut indiquer que le modèle n'est pas sûr de ses choix pour une bonne partie des données. Cette incertitude peut provenir d'un manque d'informations discriminantes

#### Q14. Test différents noyaux (linéaire, RBF, polynomial)

Résultat obtenu

Noyau : RBF → Précision = 0.34

Noyau : Linéaire (DotProduct) → Précision = 0.29

Noyau : Polynomial (approx) → Précision = 0.35

Le noyau linéaire a donné de faibles performances ça suggère que les relations entre les variables climatiques et le type de sol ne sont pas linéaires.

Le noyau RBF a permis de modéliser des relations non linéaires avec une meilleure précision

Le noyau polynomial (degré 2) a légèrement surpassé le RBF ça indique qu'un compromis entre rigidité et complexité peut être optimal.

#### Q15.

Le choix du noyau est super important. Ça influence comment le modèle réagit aux données.

Et l'a priori c'est comme un pressentiment qu'on ajuste avec l'expérience

Le bon combo c'est un noyau pas trop simple ni trop complexe avec un modèle ouvert à apprendre mais pas naïf.

Le noyau linéaire → n'a pas bien marché (précision 29%)

Le noyau **RBF** plus souple qui regarde si les points sont proches → un peu mieux (précision 34%)

Le noyau **polynomial** qui comprend des relations plus tordues → légèrement meilleur (précision 35%)

-> Dans notre cas c'est le noyau polynomial qui a le mieux capté la réalité car