

# Sommaire.

- 1 Processus Gaussien
- 2 L'algorithme UCB
- 3 Algorithme Gaussian Process Upper Confidence Bound - GP-UCB-PE
  - Formalisme

# Définition

On définit un processus gaussien par :

$$f \sim \mathbf{GP}(m, k),$$

$$\text{avec } m : \mathcal{X} \rightarrow \mathbb{R} \\ x \mapsto E[f(x)]$$

$$\text{et } k : \mathcal{X} * \mathcal{X} \rightarrow \mathbb{R}^+ \\ (x, x') \mapsto E[(f(x) - m(x))(f(x') - m(x')))].$$

- Pour tout sous-ensemble fini de valeurs (abscisses de l'intervalle d'étude discrétisé), on forme une variable aléatoire gaussienne multivariée, de vecteur moyenne  $\mu$  et de matrice de covariance  $K_y$  donnés par la moyenne  $m$  et le noyau  $k$  du processus gaussien.

Ainsi, pour tout nombre fini  $(n + 1)$  de points et avec  $(x_0, \dots, x_n) \in \chi$ , on a :

$$(f(x_0), f(x_1), \dots, f(x_n)) \sim \mathcal{N}(\mu, K_y),$$

$$\text{avec } \forall i \in [0; n], \mu[x_i] = m(x_i)$$

$$\text{et } \forall (i, j) \in [0; n]^2, K_y[x_i, x_j] = k(x_i, x_j).$$

- Méthode d'estimation par noyau, où le noyau est une fonction symétrique définie semi-positive.
- $f(x_i)$  est une collection de variables gaussiennes
- $\mathcal{N}(\mu, K_y)$  est la distribution gaussienne multivariée de vecteur moyenne  $\mu$  et de matrice de covariance  $K_y$  définie par le noyau  $k(x_i, x_j)$  avec  $(i, j) \in [0; n]^2$ .
- Base de tests ( $\sim$  base d'apprentissage) :  $X^*$  points d'abscisses  $(x_0^*, x_1^*, \dots, x_n^*) \in \chi_*$   
 $\chi_*$  : famille de valeurs du domaine de  $(n+1)$  éléments.

# Choix des noyaux

Les noyaux les plus couramment utilisés pour la modélisation par processus gaussien sont :

- **Le noyau polynomial** de degré  $\alpha$  entier naturel :

$$k(x_1, x_2) = (x_1^T x_2 + c)^\alpha \text{ avec } c \geq 0$$

un paramètre libre (lorsque  $c=0$ , on parle de noyau polynomial homogène).

- **Le noyau Gaussien** (aussi appelé RBF : radial basis function) de grandeur  $l > 0$  :

$$k(x_1, x_2) = \exp\left(-\frac{\|x_1, x_2\|^2}{2l^2}\right)$$

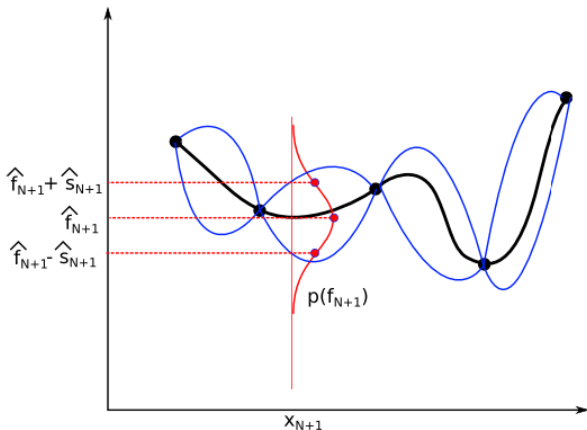
- **Les noyaux Matérn** de grandeur  $l > 0$  et de paramètre  $\nu$  :

$$k(x_1, x_2) = \frac{2^{1-\nu}}{\Gamma(\nu)} * \left(\frac{\sqrt{2\nu} * \|x_1, x_2\|}{l}\right)^\nu * K\left(\frac{\sqrt{2\nu} * \|x_1, x_2\|}{l}\right)$$

avec  $K$  la fonction Bessel modifiée de second genre et d'ordre  $\nu$ .

3 paramètres principaux  $\nu = 1/2$ ,  $\nu = 3/2$  et  $\nu = 5/2$ .

# Illustration



Métamodèle de type processus Gaussien : à partir des observations (points noirs), la densité de probabilité (rouge) fournit en tout point une valeur moyenne (noir) et une variance (bleu).

# Sommaire.

- 1 Processus Gaussien
- 2 L'algorithme UCB**
- 3 Algorithme Gaussian Process Upper Confidence Bound - GP-UCB-PE
  - Formalisme

# Formalisme

- Chaque point  $x_i^*$ ,  $i \in [0, n]$ , se voit attribuer, à chaque itération  $j$ , une valeur estimée  $\widehat{\mu}_j[x_i^*]$  et un  $\widehat{\sigma}_j^2[x_i^*] \sim$  la confiance que l'on prête à cette valeur estimée (sa variance).
- On construit la base d'entraînement  $X' : (x'_0, x'_1, \dots, x'_m)$ , qui sont les  $(m + 1)$  abscisses des points calculés à l'itération  $j$ , classés dans l'ordre dans lequel ils ont été calculés.
- À partir des  $(n + 1)$  points discrétisés de l'intervalle d'étude, à chaque itération  $j$ , on crée le vecteur  $\widehat{\mu}_j$  de dimension  $(n + 1, 1)$ , où  $\widehat{\mu}_j[i]$  est la valeur estimée de l'ordonnée de  $x_i^*$ .
- Itération  $j+1$  : on calcule le  $\widehat{\mu}_{j+1}$  et  $\widehat{\sigma}_{j+1}^2$

$$\widehat{\mu}_{j+1} = \widehat{\mu}_j(X_*) + K_{*j}^t * K_{y_j}^{-1} * (Y_j - \widehat{\mu}_j(X'))$$

$$\widehat{\sigma}_{j+1}^2 = K_{**} - \text{diag}(K_{*j}^t * K_{y_j}^{-1} * K_{*j})$$

$\widehat{\mu}_j(X_*)$   $(n + 1, 1)$  vecteur tel que  $\widehat{\mu}_j(X_*) = \bar{\mu}_j * I(n + 1)$

$\widehat{\mu}_j(X')$   $(m + 1, 1)$  vecteur tel que  $\widehat{\mu}_j(X') = \bar{\mu}_j * I(m + 1)$

avec  $I(n)$  le vecteur identité et  $\bar{\mu}_j = \frac{1}{m+1} \sum_{i=0}^m f(x'_i)$ .

$\widehat{\mu}_j(X')$  et  $\widehat{\mu}_j(X_*)$  sont des prédicteurs de la moyenne des réalisations du processus gaussien, à l'itération  $j$ .

- $\mathbf{k}(\mathbf{x}_1, \mathbf{x}_2)$  noyau gaussien avec  $l$  un paramètre fixé arbitrairement.
- $\mathbf{K}_*(m+1, n+1)$  la matrice de covariance des points d'entraînement avec les points tests :

$$K_* = \kappa(X', X_*) = \begin{pmatrix} \kappa(x'_0, x_0^*) & \kappa(x'_0, x_1^*) & \dots & \kappa(x'_0, x_n^*) \\ \kappa(x'_1, x_0^*) & \kappa(x'_1, x_1^*) & \dots & \kappa(x'_1, x_n^*) \\ \vdots & \vdots & \ddots & \vdots \\ \kappa(x'_m, x_0^*) & \kappa(x'_m, x_1^*) & \dots & \kappa(x'_m, x_n^*) \end{pmatrix} \quad \text{et } K_*^t \text{ sa transposée.}$$

- $\mathbf{K}_y(m+1, m+1)$  la matrice de covariance entre les points tests :

$$K_y = \kappa(X', X') = \begin{pmatrix} \kappa(x'_0, x'_0) + \epsilon & \kappa(x'_0, x'_1) & \dots & \kappa(x'_0, x'_m) \\ \kappa(x'_1, x'_0) & \kappa(x'_1, x'_1) + \epsilon & \dots & \kappa(x'_1, x'_m) \\ \vdots & \vdots & \ddots & \vdots \\ \kappa(x'_m, x'_0) & \kappa(x'_m, x'_1) & \dots & \kappa(x'_m, x'_m) + \epsilon \end{pmatrix} \quad \text{et } K_y^{-1} \text{ son inverse.}$$

Avec  $\epsilon$  : le "bruit" venant du fait que, d'une part il existe des incertitudes de mesure, et d'autres part, qu'on ne peut pas ajuster la courbe en ayant exactement  $y = \mu$  (avec  $y$  la valeur exacte et  $\mu$  sa valeur prise sur la courbe).



- $\mathbf{K}_{**}$  ( $n + 1, 1$ ) avec  $K_{**} = \kappa(X_*, X_*) = \begin{pmatrix} \kappa(x_0^*, x_0^*) \\ \kappa(x_1^*, x_1^*) \\ \vdots \\ \kappa(x_n^*, x_n^*) \end{pmatrix}$ .

- $Y$  ( $m + 1, 1$ ) : ensemble des ordonnées des points calculés  $Y = \left( f(x'_0), f(x'_1), \dots, f(x'_m) \right)^t$

Quand on vient de calculer l'ordonnée  $y$  d'un point d'abscisse  $x$  de la base d'apprentissage (point qui rentre alors dans la base d'entraînement), l'algorithme attribuera une valeur estimée  $\mu[i] \approx y$ , ce qui se traduit sur le graphique par le passage de la courbe estimée au plus proche du point  $(x_i, y_{x_i})$ .

Ici  $y(x_i) \approx \mu[i]$  car on ne peut pas toujours avoir, même assez rarement, l'égalité parfaite  $y(x_i) = \mu[i]$  (à cause des incertitudes numériques notamment).

Au même titre que le  $\mu$ , on utilise le  $\sigma^2$  pour déterminer le nouveau point à calculer, il faut donc calculer ce  $\sigma^2$  à chaque itération pour tous les points.

Comme  $\mu$ ,  $\sigma^2$  est un vecteur de dimension  $(n+1, 1)$  où  $\sigma^2[i]$  est la variance du point d'abscisse  $x_i$ , avec  $i \in [0; n]$ .

$K_*^t * K_y^{-1} * K_*$  est de dimension  $(n+1, n+1)$  et  $\text{diag}()$  forme le vecteur  $v$ , composé des éléments diagonaux de la matrice, tel que :  $\forall i \in [0; n], v[i] = (K_*^t * K_y^{-1} * K_*)[i][i]$ .

On a donc bien  $\sigma^2$  un vecteur de dimension  $(n+1, 1)$ . Le précédent calcul aboutit bien toujours à un  $\sigma^2$  compris entre 0 et 1.

# Exploration

Initialisation : choix arbitraire  $x'_0 = x_i^*$   $i \in [0; n]$ , (nombre arbitraire de points dans  $(x_0^*, x_1^*, \dots, x_n^*)$ ).  
calcul de son ordonnée  $y_0 = f(x_i^*)$ .

Remarque : Dans notre exemple, l'ordonnée est obtenue simplement par application de la fonction  $f$ , mais dans la partie pratique cette ordonnée est obtenue par résolution de systèmes d'équations (car on ne connaît pas la forme explicite de la fonction  $f$  : c'est justement ce que l'on veut approximer).

On procède de même pour les points suivants  $x'_1, x'_2, \dots, x'_m$  tel que  $0 \leq m \leq n$ , le nombre de points calculés au cours de l'algorithme.

Pour chaque valeur de l'intervalle discrétisé  $x_0^*, x_1^*, \dots, x_n^*$ , on a pour une valeur d'ordonnée estimée  $\hat{\mu}$  et une variance (incertitude)  $\hat{\sigma}^2$ , elle aussi estimée.

On recherche sur notre fonction précédemment estimée (et potentiellement imprécise) l'argument maximisant :

$$\underbrace{\hat{\mu}_j[i]}_{\text{gain moyen}} + \underbrace{\hat{\sigma}_j[i]\sqrt{\beta_j}}_{\text{bonus d'exploration}} .$$

Le prochain point exploré sera donc le point d'abscisse  $\arg \max_{\{x_i\}} (\hat{\mu}_j + \hat{\sigma}_j \sqrt{\beta_j})$ .

Dans notre cas, on parcourt l'axe des abscisses avec  $\hat{\mu}_j[i]$  la valeur estimée de son ordonnée et  $\hat{\sigma}_j[i]\sqrt{\beta}$  correspondant à la borne supérieure de l'intervalle de confiance en ce point (le  $i$  se rapporte à chacun des points de notre échantillon).  $\hat{\sigma}_j[i]$  est l'écart type en chaque point  $i$  de

On a alors  $\widehat{f}_j^+ = \widehat{\mu}_j + \widehat{\sigma}_j \sqrt{\beta_j} = \begin{pmatrix} \widehat{\mu}_j[0] \\ \widehat{\mu}_j[1] \\ \vdots \\ \widehat{\mu}_j[n] \end{pmatrix} + \sqrt{\beta_j} \begin{pmatrix} \widehat{\sigma}_j[0] \\ \widehat{\sigma}_j[1] \\ \vdots \\ \widehat{\sigma}_j[n] \end{pmatrix}$ , o  $j$  est le numéro de l'itération.

On classe ensuite les indices  $i \in [0 ; n]$  de ce vecteur par ordre de  $\widehat{\mu}_j[i] + \widehat{\sigma}_j[i] \sqrt{\beta_j}$  décroissant. Une fois cet ordonnancement fait, on regarde le premier indice (celui de plus grand  $\widehat{\mu}_j + \widehat{\sigma}_j \sqrt{\beta_j}$ ). S'il est non-évalué, ce point d'abscisse  $i$  est choisi pour entrer dans la base d'entrainement.

Le  $\widehat{\sigma}^2$  entrant en compte, l'abscisse d'un point déjà évalué a, de toute façon, peu de chance de maximiser  $\widehat{\mu}_j[i] + \widehat{\sigma}_j[i] \sqrt{\beta_j}$  avec  $i \in [0 ; n]$ . On peut prendre, comme dans notre modèle :

$$\beta_j = 2 * \ln\left(\frac{(n + 1) * (j\pi)^2}{6 * 0,9}\right)$$

, avec  $n+1$  le nombre de points dans l'intervalle discrétisé et  $j$  le numéro de l'itération.

# Critère d'arrêt

Dans l'algorithme, le critères d'arrêt est l'évaluation du changement apporté par la nouvelle itération par rapport à la précédente. On utilise le principe de classement des points : on classe les points à chaque itération, selon la valeur qu'on leur attribue (qu'on la connaisse car explorée ou qu'on l'estime), on relativise alors le nombre de permutations effectuées lors de cette itération au nombre maximal de permutations possibles. On compare ainsi le nombre de permutations effectuées dans le classement des  $\mu[i]$  par rapport à l'itération précédente avec le nombre total de permutations possibles.

On définit ainsi  $\pi_j$  le vecteur des classements des  $\mu_j[i]$ ,  $i \in [0; n]$ . Ainsi, on a :

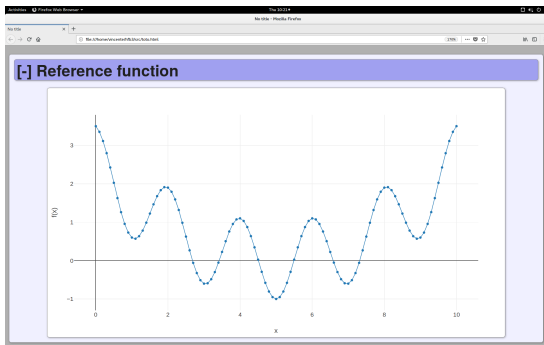
$$d(\pi_{j+1}, \pi_j) = \sum_{i=0}^n \left( \frac{\pi_{j+1} - \pi_j}{\pi_{j+1}} \right) \quad (1)$$

Le nombre total de permutations possibles étant de  $2^{n+1}$ , on compare  $\frac{d(\pi_{j+1}, \pi_j)}{2^{n+1}}$  à la valeur que nous nous sommes fixer comme critères d'arrêt. On stoppe donc l'algorithme quand le nombre  $\frac{d(\pi_{j+1}, \pi_j)}{2^{n+1}}$  devient plus petit que ce paramètre préalablement fixé : on considère qu'une itération n'apporte plus assez d'amélioration à l'approximation.

# Exemple

$$f(x) = \cos(\pi * x) + \frac{(x - 5)^2}{10}$$

Soit  $(x_0^*, x_1^*, \dots, x_n^*) = (0, 0.1, 0.2, \dots, 10)$   $n = 101$  points de la base d'apprentissage.



# Initialisation

• On choisit arbitrairement 2 points d'abscisse  $x'_0 = 2.5$  et  $x'_1 = 7.5$ , on a alors :  $y_0 = f(x'_0) = 0.625$  et  $y_1 = f(x'_1) = 0.625$ . Ces points entrent donc dans la base d'entraînement, tel que  $X' = \{2.5, 7.5\}$ .

•  $\widehat{\mu}_1 = \mu_0(X_*) + K_{*0}^t * K_{y_0}^{-1} * (Y_0 - \mu_0(X'))$

•  $\bar{\mu}_0 = \frac{1}{2} \sum_{i=0}^1 f(x'_i)$

•  $\mu_0(X_*) (101, 1)$  tel que  $\mu_0(X_*) = \bar{\mu}_0 * I(101, 1) = 0.625 * I(101, 1)$ .

•  $\mu_0(X') (2, 1)$  tel que  $\mu_0(X') = \begin{pmatrix} 1 \\ 1 \end{pmatrix} * \bar{\mu}_0 = \begin{pmatrix} 0.625 \\ 0.625 \end{pmatrix}$ .

• En utilisant  $\kappa(x_1, x_2)$  gaussien ( $l = 0.5$ )

$$K_{*0} (2, 101) = \begin{pmatrix} \kappa(2.5, 0) & \kappa(2.5, 0.1) & \dots & \kappa(2.5, 10) \\ \kappa(7.5, 0) & \kappa(7.5, 0.1) & \dots & \kappa(7.5, 10) \end{pmatrix}$$

$$\Rightarrow K_{*0}^t (101, 2) = \begin{pmatrix} 0.000003727 & 0 \\ 0.00000993 & 0 \\ \cdot & \cdot \\ \cdot & \cdot \\ 0 & 0 \end{pmatrix} \dots$$

## Initialisation (suite)

$$\bullet Y_0 - \mu_0(X') = \begin{pmatrix} 0.625 - 0.625 \\ 0.625 - 0.625 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

$$\bullet \epsilon = 10^{-3} K_{y_0} = \begin{pmatrix} \kappa(2.5, 2.5) + 0.001 & \\ & \kappa(7.5, 2.5) \end{pmatrix} = \begin{pmatrix} \kappa(2.5, 7.5) & \\ \kappa(7.5, 7.5) + 0.001 & \end{pmatrix} = \begin{pmatrix} 0 & \\ 0 & 0.999001 \end{pmatrix}, \quad Y_0 = \begin{pmatrix} 0.625 \\ 0.625 \end{pmatrix}.$$

$$\begin{pmatrix} 1.001 & 0 \\ 0 & 1.001 \end{pmatrix} \text{ et } K_{y_0}^{-1} = \begin{pmatrix} 0.999001 & \\ & 0 \end{pmatrix}$$

$$\bullet \widehat{\mu}_1 = \mu_0(X_*) + K_{*0}' * K_{y_0}^{-1} * (Y_0 - \mu_0(X'))$$

$$\widehat{\mu}_1 = \begin{pmatrix} 0.625 \\ 0.625 \\ \vdots \\ \vdots \\ 0.625 \end{pmatrix} + \begin{pmatrix} 0.000003727 & 0 \\ 0.00000993 & 0 \\ \vdots & \vdots \\ \vdots & \vdots \\ 0 & 0 \end{pmatrix} * \begin{pmatrix} 0.999001 & 0 \\ 0 & 0.999001 \end{pmatrix} * \begin{pmatrix} 0 \\ 0 \end{pmatrix} \widehat{\mu}_1 = \begin{pmatrix} 0.625 \\ 0.625 \\ \vdots \\ \vdots \\ 0.625 \end{pmatrix}.$$

On a ainsi une valeur estimée pour chaque point de l'intervalle. On peut alors tracer la courbe estimée :

$$((x_0^*, \widehat{\mu}_1[0]), (x_1^*, \widehat{\mu}_1[1]), \dots, (x_{100}^*, \widehat{\mu}_1[100]))$$

# Initialisation (fin)

- $\hat{\sigma}_1^2 = K_{**} - \text{diag}(K_{*0}^t * K_{y0}^{-1} * K_{*0})$

$$K_{**} = \begin{pmatrix} 1 \\ 1 \\ \cdot \\ \cdot \\ 1 \end{pmatrix} = I^*.$$

$$\text{Ainsi, } \hat{\sigma}_1^2 = I^* - \begin{pmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ 0.999001 \\ \cdot \\ \cdot \\ 0.999001 \\ \cdot \\ \cdot \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ \cdot \\ \cdot \\ 0.000999 \\ \cdot \\ \cdot \\ 0.000999 \\ \cdot \\ \cdot \\ 1 \end{pmatrix}.$$

Si on calcule les variances des points calculés, on a :

$$\hat{\sigma}_1^2[2.5] = 1 - 1 * 0.999001 * 1 = 0.000999$$

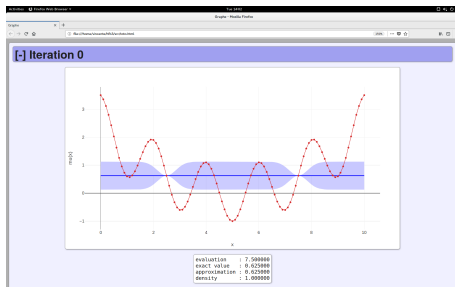
$$\text{et } \hat{\sigma}_1^2[7.5] = 1 - 1 * 0.999001 * 1 = 0.000999.$$

On observe donc bien que la variance en ces points n'est pas pour autant nulle. On comprend également que plus un point est proche d'un (ou plusieurs) point(s) calculé(s), plus sa variance est faible.



# Exemple

$$f(x) = \cos(\pi * x) + \frac{(x - 5)^2}{10}$$



# Iteration 1

- Choix du point d'entraînement.

À partir du  $\widehat{\mu}_1$  et du  $\widehat{\sigma}_1^2$ , on cherche :  $\forall i \in [0; 101]$ ,

$$\arg \max_{\{x_i\}} (\widehat{\mu}_1(x_i) + \widehat{\sigma}_1(x_i) \sqrt{\beta_1})$$

Calcul pour chaque point de l'intervalle  $f_1^+ = \widehat{\mu}_1 + \widehat{\sigma}_1 \sqrt{\beta_1}$  avec  $\beta_1 = 2 * \ln(\frac{101 * (\pi)^2}{6 * 0,9}) = 10.436$ .

$$f_1^+ = \begin{pmatrix} \widehat{\mu}_1[0] \\ \widehat{\mu}_1[1] \\ \vdots \\ \widehat{\mu}_1[100] \end{pmatrix} + \sqrt{\beta_1} \begin{pmatrix} \widehat{\sigma}_1[0] \\ \widehat{\sigma}_1[1] \\ \vdots \\ \widehat{\sigma}_1[100] \end{pmatrix} \text{ donc, } f_1^+ = \begin{pmatrix} 0.625 \\ 0.625 \\ \vdots \\ 0.625 \end{pmatrix} + \sqrt{10.43636267} \begin{pmatrix} \sqrt{1} \\ \sqrt{1} \\ \vdots \\ \sqrt{0.000999} \\ \vdots \\ \sqrt{0.000999} \\ \vdots \\ \sqrt{1} \end{pmatrix}.$$

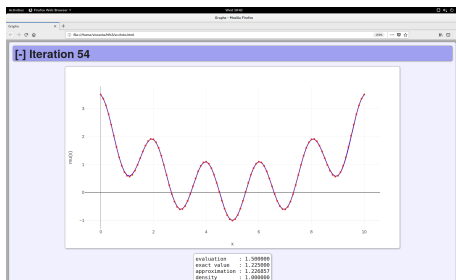
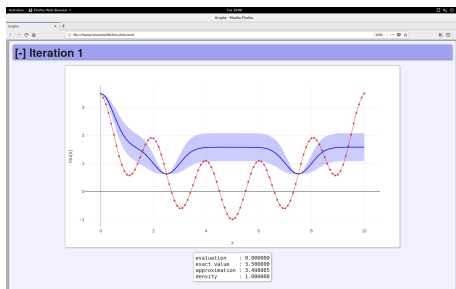
## Iteration 1

$$\widehat{f}_1^+ = \begin{pmatrix} 3.85553597 \\ 3.85553597 \\ \cdot \\ \cdot \\ 0.727107425 \\ \cdot \\ \cdot \\ 0.727107425 \\ \cdot \\ \cdot \\ 3.85553597 \end{pmatrix} \cdot$$

On réalise un classement des abscisses  $i$  en fonction de la valeur décroissante de  $\widehat{\mu}_1[i] + \widehat{\sigma}_1[i]\sqrt{\beta_1}$ . Étant au début de l'algorithme, de nombreux points sont très incertains. Ces points incertains, comme les points d'abscisse 0 ou 10 par exemple, sont ceux éloignés des points calculés et ont le  $\widehat{\sigma}_1[i]$  maximal, à savoir 1. On a ainsi de nombreux points ayant la même valeur de  $\widehat{\mu}_1[i] + \widehat{\sigma}_1[i]\sqrt{\beta_1}$  (en plus de l'approximation éventuelle dans les calculs). On pourrait prendre ensuite au hasard parmi les abscisses maximisant  $\widehat{\mu}_i + \widehat{\sigma}_i\sqrt{\beta_j}$ , mais on choisit de classer les points ex-æquo par valeur d'abscisse croissante. Ici le point d'abscisse  $i = 0$  ayant la plus grande valeur  $\widehat{\mu}_1[i] + \widehat{\sigma}_1[i]\sqrt{\beta_1}$  ex-æquo, c'est celui-ci qui est alors choisi.

Le point d'abscisse 0 ayant été choisi, on calcule donc l'ordonnée de ce troisième point :  $f(x'_3) = f(0) = 3.5$ .

## Exemple



# Exemple

# Sommaire.

- 1 Processus Gaussien
- 2 L'algorithme UCB
- 3 Algorithme Gaussian Process Upper Confidence Bound - GP-UCB-PE
  - Formalisme

# Algorithme GP-UCB-PE

• Supposons qu'un ensemble de  $N$  observations  $F_N = \{f_1, f_2, \dots, f_N\}$ , ait été réalisé aux points  $X_N = \{x_1, x_2, \dots, x_N\}$ . On fait l'hypothèse que  $F_N$  correspond à la réalisation d'un processus Gaussien multi-varié à moyenne nulle (Krigage simple), dont la densité de probabilité s'écrit :

$$p(F_N|X_N) = \frac{\exp\left(-\frac{1}{2} F_N^T C_N^{-1} F_N\right)}{\sqrt{(2\pi)^N \det(C_N)}} \quad (2)$$

où  $C_N$  représente la matrice de covariance. Ses coefficients  $C_{mn}$ , qui décrivent la covariance des valeurs  $f_m$  et  $f_n$  correspondant aux points  $x_m$  et  $x_n$ , sont exprimées sous la forme d'une fonction de corrélation  $C_{mn} = c(x_m, x_n; \Theta)$  où les paramètres  $\Theta$  servent à calibrer le modèle. Sous ces hypothèses, la densité de probabilité d'obtenir une valeur additionnelle  $f_{N+1}$  en un point  $x_{N+1}$  s'écrit de même :

$$p(F_{N+1}|X_{N+1}) = \frac{\exp\left(-\frac{1}{2} F_{N+1}^t C_{N+1}^{-1} F_{N+1}\right)}{\sqrt{(2\pi)^{N+1} \det(C_{N+1})}}.$$

En introduisant la règle des probabilités conditionnelles :

$$p(f_{N+1}|F_N, X_{N+1}) = \frac{p(F_{N+1}|X_{N+1})}{p(F_N|X_N)}$$

# Algorithme GP-UCB-PE

A partir des deux expressions précédentes, on obtient la densité de probabilité de la valeur  $f_{N+1}$ , connaissant les valeurs  $F_N$ . La densité est gaussienne de moyenne  $\hat{f}_{N+1}$  et variance  $\hat{\sigma}_{f_{N+1}}^2$  :

$$p(f_{N+1}|F_N, X_{N+1}) = \frac{1}{\sqrt{2\pi} \hat{\sigma}_{f_{N+1}}} \exp\left[-\frac{(f_{N+1} - \hat{f}_{N+1})^2}{2\hat{\sigma}_{f_{N+1}}^2}\right]$$

avec

$$\hat{f}_{N+1} = k^t C_N^{-1} F_N \quad \hat{\sigma}_{f_{N+1}}^2 = \kappa - k^t C_N^{-1} k,$$

$$k = [c(x_1, x_{N+1}), c(x_2, x_{N+1}), \dots, c(x_N, x_{N+1})]^t$$

et

$$\kappa = c(x_{N+1}, x_{N+1}).$$

- L'algorithme GP-UCB-PE permet de minimiser une fonction en supposant qu'elle suive un PG. Problème d'optimisation :

$$\max_{x \in \mathcal{X}} f(x),$$

quelque soit la fonction  $f : \mathcal{X} \rightarrow \mathbb{R}$  and  $\mathcal{X} \subset \mathbb{R}^n$ .



# Régression sans bruit

Soit  $\mathcal{X}$  la base d'apprentissage  $\{\mathcal{X}, y = f\}$  constituée de  $n$  points localisés par  $x_1, \dots, x_n \in \mathcal{X}$ . On suppose que l'on évalue la fonction après  $T$  itérations au cours desquelles sont estimés  $m$  points localisés auxquels on veut connaître la fonction régressée

# Régression avec bruit



On se pose ici la question de trouver les valeurs  $f_T$  ( $f_T$  est le vecteur rassemblant les valeurs que l'on cherche) de la fonction  $f$  pour certaines entrées  $x_T$ , alors qu'on en observe des échantillons bruités  $y_i = f(x_i) + \epsilon_i$  ou encore  $y = f + \epsilon$ . On supposera  $\epsilon \sim \mathcal{N}(0, \sigma^2)$ . On a alors :

$$\begin{bmatrix} f \\ f_T \end{bmatrix} \approx \mathcal{N} \left( 0; \begin{bmatrix} K(\mathcal{X}, \mathcal{X}) + \sigma^2 \mathbf{I} & K(\mathcal{X}, \mathcal{X}_T) \\ K(\mathcal{X}_T, \mathcal{X}) & K(\mathcal{X}_T, \mathcal{X}_T) \end{bmatrix} \right)$$

et que

$$f_T | \mathcal{X}, \mathcal{X}_T, y \sim \mathcal{N} \left( K(\mathcal{X}_T, \mathcal{X}_T) [K(\mathcal{X}, \mathcal{X}) + \sigma^2 \mathbf{I}]^{-1} \mathbf{y} \right)$$

$$K(\mathcal{X}_T, \mathcal{X}_T) - K(\mathcal{X}_T, \mathcal{X}) [K(\mathcal{X}, \mathcal{X}) + \sigma^2 \mathbf{I}]^{-1} K(\mathcal{X}, \mathcal{X}_T)$$