## Die Aufgabe

Betrachte $\Omega \subset \mathbb{R}^d$. Für ein Modell $f : \Omega \to \mathbb{R}$ konstruieren wir iterativ Approximationen $s_f^{(n)} : \Omega \to \mathbb{R}$, die $f$ an $n$ ausgewählte Daten $x_1, ..., x_n \in \Omega$ interpolieren.

Sei $k : \Omega \times \Omega \to \mathbb{R}$ ein positiv definiter Kern, sodass $f \in \mathcal{H}_k(\Omega)$. Hierbei bezeichnet $(\mathcal{H}_k(\Omega), \langle \cdot, \cdot \rangle)$ den Raum von Funktionen $f : \Omega \to \mathbb{R}^q$, zu dem $k$ der reproduzierende Kern ist, d.h. es gilt

1. $k(\cdot, x) : \Omega \to \mathbb{R} \in \mathcal{H}_k(\Omega) \, \forall x \in \Omega$
2. $\langle f, k(\cdot, x) \rangle = f(x) \, \forall x \in \Omega, f \in \mathcal{H}_k(\Omega)$.

Für iterativ gewählte $X_n := \{x_1, ..., x_n\} \subset \Omega$ konstruieren wir die Approximationen $s_f^{(n)}$ als die orthogonale Projektion in

$$V(X_n) := \text{span}\{k(\cdot, x_1), ..., k(\cdot, x_n)\} \subset \mathcal{H}_k(\Omega).$$

Dies ist die beste Approximation von $f$ bezüglich $\| \cdot \|_{\mathcal{H}_k(\Omega)}$. Schreibe $\langle \cdot, \cdot \rangle$ bzw. $\| \cdot \|$ für das Skalarprodukt bzw. die Norm in $\mathcal{H}_k(\Omega)$. Gesucht sind also die Koeffizienten $a_i$, $i = 1, ..., n$ sodass

$$s_f(x) = \sum_{i=1}^{n} a_i k(\cdot, x_i).$$

Diese finden wir mittels einer Orthonormalbasis $(v_1, ..., v_n)$ von $V(X_n)$. Denn dann gilt

$$s_f(x) = \Pi_{V(X_n)}(f) = \sum_{i=1}^{n} c_i v_i,$$

wobei die $c_i := \langle f, v_i \rangle$ die Koeffizienten der Linearkombination beschreiben und im $n$. Iterationsschritt können die bisher berechneten $n-1$ Basisfunktionen und Koeffizienten übernommen werden. Die Aufgabe des Algorithmus besteht also darin, im $n$. Iterationsschritt die $n$. Basisfunktion $v_n \in V(X_n)$ und den dazugehörigen Koeffizienten $c_n \in \mathbb{R}$ zu bestimmen.

Diese Überlegungen lassen sich leicht auf Funktionen $f : \Omega \to \mathbb{R}^q$ für $q > 1$ übertragen, in dem man den Algorithmus auf alle $q$ Komponenten von $f$ anwendet und den Raum

$$\mathcal{H}_k(\Omega)^q := \{f : \Omega \to \mathbb{R}^q | (f)_j \in \mathcal{H}_k(\Omega)\}$$

betrachtet.

## Das Vorgehen

Sie $\tilde{\Omega} \subset \Omega$ eine Diskretisierung von $\Omega$.

Mit $P_{V(X_n)}$ bezeichnen wir die Power Function, die den Approximationsfehler der schlechtesten Funktion $f \in \mathcal{H}_k(\Omega)$ an einer Stelle $x \in \Omega$ im $n$. Iterationsschritt angibt:

$$P_{V(X_n)}(x) := \sup_{f \in \mathcal{H}_k(\Omega)} \frac{|f(x) - s_f^{(n)}(x)|}{\|f\|}.$$

Beim P-Greedy Algorithmus wird im $n$. Iterationsschritt, der Datenpunkt $x_n$ ausgewählt, der die Power Function für den nächsten Iterationsschritt minimiert. Dieser liefert dann die $n$. Basisfunktion $k(\cdot, x_n)$ bzw. $v_n$.

**Lemma 1.** *Für alle $x \in \Omega$ gilt*

$$P_{V(X_n)}(x) = \|k(\cdot, x) - \Pi_{V(X_n)}(k(\cdot, x))\|.$$

*Beweis.* Sei $f \in \mathcal{H}_k(\Omega)$, o.B.d.A. sei $\|f\| \leq 1$. Außerdem sei $v_1, ..., v_n$ eine Orthonormalbasis von $V(X_n)$. Für $x \in \Omega$ und $v_x := k(\cdot, x)$ gilt dann

$$\begin{aligned}
f(x) - \Pi_{V(X_n)}(f)(x) &= \langle v_x, f \rangle - \langle v_x, \sum_{i=0}^{n} \langle f, v_i \rangle v_i \rangle \\
&= \langle v_x, f \rangle - \sum_{i=0}^{n} \langle f, v_i \rangle \langle v_x, v_i \rangle \\
&= \langle v_x, f \rangle - \langle \sum_{i=0}^{n} \langle v_x, v_i \rangle v_i, f \rangle \\
&= \langle v_x - \sum_{i=0}^{n} \langle v_x, v_i \rangle v_i, f \rangle \\
&\leq \|v_x - \sum_{i=0}^{n} \langle v_x, v_i \rangle v_i\| \|f\| \\
&\leq \|v_x - \Pi_{V(X_n)}(v_x)\|
\end{aligned}$$

Also muss für alle $x \in \Omega$

$$P_{V(x_n)}(x) = \sup_{f \in \mathcal{H}_k(\Omega), \|f\| \leq 1} |f(x) - \Pi_{V(X_n)}(f)(x)| \leq \|v_x - \Pi_{V(X_n)}(v_x)\|$$

Da für $f := \frac{v_x - \Pi_{V(X_n)}(v_x)}{\|v_x - \Pi_{V(X_n)}(v_x)\|} \in \mathcal{H}_k(\Omega)$

$$f(x) - \Pi_{V(X_n)}(f)(x) = \langle v_x - \Pi_{V(X_n)}(v_x), f \rangle \;= \|v_x - \Pi_{V(X_n)}(v_x)\|$$

gilt, folgt

$$P_{V(X_n)}(x) = \|v_x - \Pi_{V(X_n)}(v_x)\|.$$

□

Insbesondere gilt also

$$P_{V(X_0)}(x)^2 = \|k(\cdot, x)\|^2 = \langle k(\cdot, x), k(\cdot, x)\rangle = k(x, x),$$

und ist also für translationsinvariante Kerne konstant, da $k(x,x) = \phi(x-x) = \phi(0)\,\forall x \in \Omega$.

Außerdem folgt

$$\begin{aligned} P_{V(X_n)}(x)^2 &= \|k(\cdot, x) - \Pi_{V(X_n)}(k(\cdot, x))\|^2 \\ &= k(x, x) - \sum_{i=1}^{n} v_i(x)^2 \\ &= P_{V(X_0)}(x)^2 - \sum_{i=1}^{n} v_i(x)^2 \\ &= P_{V(X_{n-1})}(x)^2 - v_n(x)^2. \end{aligned}$$

Wir bezeichnen das Residuum der $n$. Approximation mit

$$\begin{aligned} r_n(x) &:= f(x) - \Pi_{V(X_n)}(f)(x) \\ &= f(x) - \sum_{i=1}^{n} c_i v_i \\ &= r_{n-1}(x) - c_n v_n. \end{aligned}$$

Bei der iterativen Approximation gehen wir nun wie folgt vor. Wir konstruieren eine Orthonormalbasis $(v_1, ..., v_n)$ der $V(X_n)$, die sich in jedem Iterationsschritt erweitern lässt. Dabei wählen wir in der $n$. Iteration das $x_n$ aus $\tilde{\Omega} \setminus X_{n-1}$ aus, das die Power function $P_{V(X_{n-1})}(x)$ maximiert und orthogonalisieren $k(\cdot, x_n)$ mithilfe des Gram-Schmidt-Verfahrens. Das bedeutet

$$\begin{aligned} x_1 &= \arg\max_{x \in \tilde{\Omega}} \sqrt{k(x, x)} \\ v_1 &= \frac{k(\cdot, x_1)}{\sqrt{k(x_1, x_1)}} \end{aligned}$$

und

$$\begin{aligned}
x_n &= \underset{x\in\tilde{\Omega}\setminus X_{n-1}}{\arg\max}\, P_{V(X_{n-1})}(x) \\
v_n' &= k(\cdot,x_n) - \sum_{i=1}^{n-1}\langle k(\cdot,x_n), v_i\rangle v_i \\
&= k(\cdot,x_n) - \sum_{i=1}^{n-1} v_i(x_n) v_i \\
v_n &= \frac{v_n'}{\|v_n'\|}.
\end{aligned}$$

**Lemma 2.** *Es gilt*

$$\| v_n' \| = P_{V(X_{n-1})}(x)$$

*Beweis.*

$$\begin{aligned}
\|v_n'\|^2 &= \langle k(\cdot,x_n) - \sum_{i=1}^{n-1} v_i(x_n)v_i, k(\cdot,x_n) - \sum_{i=1}^{n-1} v_i(x_n)v_i\rangle \\
&= k(x_n,x_n) - 2\sum_{i=1}^{n-1} v_i(x_n)^2 + \sum_{i=1}^{n-1}\sum_{j=1}^{n-1} v_i(x_n)v_j(x_n)\langle v_i, v_j\rangle \\
&= P_{V(X_0)}(x_n)^2 - \sum_{i=1}^{n-1} v_i(x_n)^2 \\
&= P_{V(X_{n-1})}(x_n)^2.
\end{aligned}$$

□

Also ist

$$v_n = \frac{k(\cdot,x_n) - \sum_{i=1}^{n-1} v_i(x_n)v_i}{P_{V(X_{n-1})}(x_n)}.$$

**Lemma 3.** *Für die Koeffizienten $c_n$ der Approximation folgt*

$$c_n = \langle f, v_n\rangle = \frac{r_{n-1}(x_n)}{P_{V(X_{n-1})}(x_n)}.$$

*Beweis.*

$$\begin{aligned}
c_n &= \langle f, v_n\rangle \\
&= \langle f, \frac{k(\cdot,x_n) - \sum_{i=1}^{n-1} v_i(x_n)v_i}{P_{V(X_{n-1})}(x_n)}\rangle \\
&= \frac{f(x_n) - \sum_{i=1}^{n-1} v_i(x_n)\langle f, v_i\rangle}{P_{V(X_{n-1})}(x_n)} \\
&= \frac{f(x_n) - \sum_{i=1}^{n-1} c_i v_i(x_n)}{P_{V(X_{n-1})}(x_n)} \\
&= \frac{r_{n-1}(x_n)}{P_{V(X_{n-1})}(x_n)}
\end{aligned}$$

□

# Implementierung

Grob ist die Implementierung wie folgt umgesetzt:

```
Initialisieren aller benötigten Größen;
for i = 1, ..., maxIter do
    Wähle argmax der Power Function aus;
    Berechne i. Basisfunktion v_i ausgewertet an Omega;
    Berechne i. Koeffizient c_i der Linearkombination bzgl. v_i;
    Aktualisiere die Residuen an Omega;
    Aktualisiere die Basiswechselmatrix;
    Aktualisierte die Power Function ausgewertet an Omega;
    if Maximum der Power Function < Toleranz then
        Speichere i als Anzahl benötigter Iterationen;
        break;
    end
end
Berechne die Approximation als ∑_i c_i v_i an Omega;
Berechne die Koeffizienten α_i für ∑_i α_i k(·, x_i) mittels der
  Basiswechselmatrix;
```

Sei $f : \Omega \to \mathbb{R}^q$ und $\tilde{\Omega}$ eine Diskretisierung von $\Omega$. Zur Implementierung notwendige Größen:

- $\tilde{\Omega} = \{x_1, ..., x_N\}$ Menge aller Punkte, aus denen wir $\tilde{N}$ viele auswählen
- $A = (k(x_i, x_j))_{i,j=1,...,N}$

- $f = \begin{pmatrix} f(x_1) \\ \vdots \\ f(x_N) \end{pmatrix} \in \mathbb{R}^{N\times q}$ die Auswertungen der zu approximierenden Funktion an $\tilde{\Omega}$

Dabei wird im $k$. Schritt gespeichert:

- $X_k := \{x_{j_1}, ..., x_{j_k}\} = X_{k-1} \cup \{x_{j_k}\},\ J_k := \{j_1, ..., j_k\} = J_{k-1} \cup \{j_k\}$ die Auswahl von $x_{j_k}$ aus $\tilde{\Omega} \setminus X_{k-1}$, sodass

$$x_{j_k} = \operatorname*{arg\,max}_{x \in \tilde{\Omega} \setminus X_{k-1}} P_{V(X_{k-1})}(x).$$

- Eine Matrix der zu berechnenden Basisfunktionen ausgewertet an den ausgewählten Punkten $x_{j_i} \in X_k$, d.h.

$$V = \begin{pmatrix} \vdots & & & & & \\ v_1(x_{j_3}) & v_2(x_{j_3}) & v_3(x_{j_3}) & 0 & \dots & 0 \\ \vdots & & & & & \\ v_1(x_{j_1}) & 0 & & & \dots & 0 \\ \vdots & & & & & \\ v_1(x_{j_2}) & v_2(x_{j_2}) & 0 & & \dots & 0 \\ \vdots & & & & & \end{pmatrix} \in \mathbb{R}^{N\times\tilde{N}}$$

wobei $V_{nm} = v_n(x_{j_m})$ für $x_{j_m} \in X_k$, $n \leq j_m$. Im $k$. Schritt wird also die $k$. Spalte gesetzt:

$$V_{ik} = v_k(x_i) \quad \forall i \in \{1, ..., N\} \setminus X_k$$

Es ist

$$\begin{aligned} v_k(x_i) &= \frac{k(x_i, x_{j_k}) - \sum_{j=1}^{k-1} v_j(x_{j_k}) v_j(x_i)}{P_{V(X_{k-1})}(x_{j_k})} \\ &= \frac{A_{i,j_k} - \sum_{j=1}^{k-1} v_j(x_{j_k}) v_j(x_i)}{P_{V(X_{k-1})}(x_{j_k})} \end{aligned}$$

mit

$$\sum_{j=1}^{k-1} v_j(x_{j_k}) v_j(x_i) = \begin{pmatrix} v_1(x_{j_k}) & \dots & v_{k-1}(x_{j_k}) \end{pmatrix} \cdot \begin{pmatrix} v_1(x_i) \\ \vdots \\ v_{k-1}(x_i) \end{pmatrix}.$$

$V$ wird mit $V = 0 \in \mathbb{R}^{N\times\tilde{N}}$ initialisiert.

- Die Koeffizienten der Linearkombination von $s_f^{(k)}$:

$$c = \begin{pmatrix} c_1 \\ \vdots \\ c_k \\ 0 \\ \vdots \\ 0 \end{pmatrix} \in \mathbb{R}^{\tilde{N} \times q}$$

  wobei $c_k = \langle f, v_k \rangle = \frac{r_{k-1}(x_{j_k})}{P_{V(X_{k-1})}(x_{j_k})}$. Es wird $c := 0 \in \mathbb{R}^{\tilde{N} \times q}$ gesetzt.

- Das Residuum im $k$. Schritt ausgewertet an $x_i \in \tilde{\Omega}$, d.h

$$R^{(k)} = \begin{pmatrix} r_k(x_1) \\ \vdots \\ r_k(x_N) \end{pmatrix} \in \mathbb{R}^{N \times q}$$

  wobei $r_k(x_{j_i}) = 0 \quad \forall i \in J_k$. Dabei wird wie folgt aktualisiert:

$$r_k(x_i) = r_{k-1}(x_i) - c_k v_k(x_i) \quad \forall i \in \{1, ..., N\} \setminus X_{k-1}.$$

  Es wird $R^{(0)} = f \in \mathbb{R}^{N \times q}$ initialisiert.

- Die Power Function ausgewertet an den Punkten $x_i \in \tilde{\Omega}$

$$P^{(k)} = \begin{pmatrix} P_{V(X_k)}(x_1) \\ \vdots \\ P_{V(X_k)(x_N)} \end{pmatrix},$$

  wobei $P_{V(X_k)}(x_i) = 0 \quad \forall i \in J_k$. Dabei wird wie folgt aktualisiert:

$$P_{V(X_k)}(x_i) = \sqrt{P_{V(X_{k-1})}(x_i)^2 - v_{j_k}(x_i)^2} \quad \forall i \in \{1, ..., N\} \setminus X_{k-1}.$$

  Es wird

$$P^{(0)} = \begin{pmatrix} P_{V(X_0)}(x_1) \\ \vdots \\ P_{V(X_0)(x_N)} \end{pmatrix} = \begin{pmatrix} k(x_1, x_1) \\ \vdots \\ k(x_1, x_1) \end{pmatrix}$$

  gesetzt.

- Die Basiswechselmatrix $T = (d_{ij})_{i,j=1,...,\tilde{N}}$ von $(v_1, ..., v_{\tilde{N}})$ zu $(k(\cdot, x_{j_1}), ..., k(\cdot, x_{j_{\tilde{N}}}))$, wobei

$$v_j = \sum_{i=0}^{\tilde{N}} d_{ij} k(\cdot, x_{j_i}) = \sum_{i=0}^{j} d_{ij} k(\cdot, x_{j_i}) \quad \forall j = 1, \dots, \tilde{N}.$$

Im $k$. Schritt wird also die $k$. Spalte gesetzt mit

$$
\begin{aligned}
d_{lk} &= -\frac{1}{P_{V(X_{k-1})}(x_{j_k})} \begin{pmatrix} d_{l1} & \dots & d_{l,k-1} \end{pmatrix} \cdot \begin{pmatrix} v_1(x_{j_k}) \\ \vdots \\ v_{k-1}(x_{j_k}) \end{pmatrix} \\
&= -\frac{1}{P_{V(X_{k-1})}(x_{j_k})} \sum_{i=1}^{k-1} v_i(x_{j_k}) d_{li} \quad \forall l < k
\end{aligned}
$$

und $d_{kk} = \frac{1}{P_{V(X_{k-1})}(x_{j_k})}$, $d_{lk} = 0$ falls $l > k$, da

$$
\begin{aligned}
v_k' &= k(\cdot, x_{j_k}) - \sum_{i=1}^{k-1} v_i(x_{j_k}) v_i \\
&= k(\cdot, x_{j_k}) - \sum_{i=1}^{k-1} v_i(x_{j_k}) \sum_{l=1}^{i} d_{li} k(\cdot, x_{j_l}) \\
&= k(\cdot, x_{j_k}) - \sum_{i=1}^{k-1} \sum_{l=1}^{k-1} v_i(x_{j_k}) d_{li} k(\cdot, x_{j_l}) \\
&= k(\cdot, x_{j_k}) - \sum_{l=1}^{k-1} \sum_{i=1}^{k-1} v_i(x_{j_k}) d_{li} k(\cdot, x_{j_l})
\end{aligned}
$$

und

$$
v_k = \frac{v_k'}{P_{V(X_{k-1})}(x_{j_k})}.
$$

Initialisiert wird $T$ mit

$$
T = 0 \in \mathbb{R}^{\tilde{N} \times \tilde{N}}.
$$