

Codificación Dispersa Adaptada a la Estimación de Modelos Sustitutos

(Adapted Sparse Coding for on-line Estimation
of Surrogate Models for Expensive Functions)

M.Sc. Cindy Calderón Arce
Dr.-Ing. Pablo Alvarado-Moya

Escuela de Matemática
Junio, 2018

- 1 Preliminares
- 2 Sparse encoded surrogate model (SESM)
- 3 Resultados

Algunos conceptos...

- Señales dispersas
- Codewords o átomos
- Diccionarios
- Diccionarios sobrecompletos

Algunos conceptos...

- Señales dispersas
- Codewords o átomos
- Diccionarios
- Diccionarios sobrecompletos

Ejemplo

Representar números del 0 al 9:

Algunos conceptos...

- Señales dispersas
- Codewords o átomos
- Diccionarios
- Diccionarios sobrecompletos

Ejemplo

Representar números del 0 al 9:

- Codewords: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Algunos conceptos...

- Señales dispersas
- Codewords o átomos
- Diccionarios
- Diccionarios sobrecompletos

Ejemplo

Representar números del 0 al 9:

- Codewords: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- Diccionarios:
(6, 3, 9, 1, 4, 5, 7, 8, 0, 2), (6, 3, 9, 1, 4, 5, 7, 8), (6, 3, 9, 1, 4, 5, 7, 8, 0, 2, 6)

Algunos conceptos...

- Señales dispersas
- Codewords o átomos
- Diccionarios
- Diccionarios sobrecompletos

Ejemplo

Representar números del 0 al 9:

- Codewords: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- Diccionarios:
(6, 3, 9, 1, 4, 5, 7, 8, 0, 2), (6, 3, 9, 1, 4, 5, 7, 8), (6, 3, 9, 1, 4, 5, 7, 8, 0, 2, 6)
- Diccionario sobrecompleto: $D = (6, 0, 3, 9, 1, 4, 5, 7, 1, 8, 0, 2, 6)$

Algunos conceptos...

- Señales dispersas
- Codewords o átomos
- Diccionarios
- Diccionarios sobrecompletos

Ejemplo

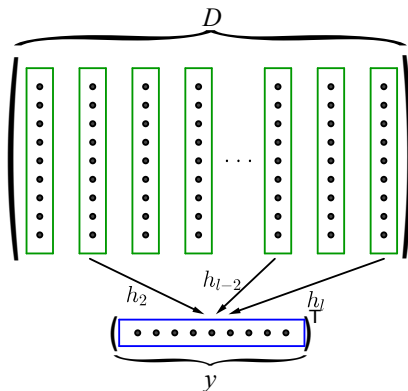
Representar números del 0 al 9:

- Codewords: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- Diccionarios:
(6, 3, 9, 1, 4, 5, 7, 8, 0, 2), (6, 3, 9, 1, 4, 5, 7, 8), (6, 3, 9, 1, 4, 5, 7, 8, 0, 2, 6)
- Diccionario sobrecompleto: $D = (6, 0, 3, 9, 1, 4, 5, 7, 1, 8, 0, 2, 6)$
- Representación dispersa: $3 = D \cdot (0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0)^T$

Sparse Coding (SC)

Utiliza un diccionario sobrecompleto D para establecer una representación dispersa h de una señal y

$$y = Dh$$



Sparse Coding (SC)

Dado un conjunto de entrenamiento

$$Y = \{y^{(1)}, y^{(2)}, \dots, y^{(T)}\}$$

con $y^{(t)} \in \mathbb{R}^m$, cada $y^{(t)}$ representa un conjunto de muestras. El objetivo es encontrar un diccionario $D \in \mathbb{R}^{m \times l}$ y un vector disperso $h^{(t)}$ tal que

$$Dh^{(t)} = y^{(t)}$$

Sparse Coding (SC)

Dado un conjunto de entrenamiento

$$Y = \{y^{(1)}, y^{(2)}, \dots, y^{(T)}\}$$

con $y^{(t)} \in \mathbb{R}^m$, cada $y^{(t)}$ representa un conjunto de muestras. El objetivo es encontrar un diccionario $D \in \mathbb{R}^{m \times l}$ y un vector disperso $h^{(t)}$ tal que

$$Dh^{(t)} = y^{(t)}$$

Es posible encontrar D y $h^{(t)}$ por medio del siguiente problema de optimización convexa

$$\min_{D, h^{(t)}} \sum_{t=1}^T \|y^{(t)} - Dh^{(t)}\|_2^2 + \lambda \|h^{(t)}\|_1$$

$$\min_{D, h^{(t)}} \sum_{t=1}^T \|y^{(t)} - Dh^{(t)}\|_2^2 + \lambda \|h^{(t)}\|_1$$

El problema general de SC se resuelve de manera iterativa utilizando algoritmos de descenso de gradiente, fija D para encontrar $h^{(t)}$ por medio de Iterative Shrinkage and Thresholding Algorithm (ISTA) y luego fija $h^{(t)}$ para encontrar D utilizando un problema de mínimos cuadrados y un algoritmo de Dictionary Learning (DL).

ISTA

$$h^{(t)} = \Upsilon_{\lambda\alpha} \left(h^{(t)} - \alpha D^T \left(y^{(t)} - Dh^{(t)} \right) \right)$$

$$\Upsilon_{\beta}(v) = \text{sign}(v) \max\{|v| - \beta, 0\}$$

ISTA

$$h^{(t)} = \Upsilon_{\lambda\alpha} \left(h^{(t)} - \alpha D^T \left(y^{(t)} - Dh^{(t)} \right) \right)$$

$$\Upsilon_{\beta}(v) = \text{sign}(v) \max\{|v| - \beta, 0\}$$

Conv. $\frac{1}{\alpha}$ mayor que el mayor valor propio de $D^T D$

ISTA

$$h^{(t)} = \Upsilon_{\lambda\alpha} \left(h^{(t)} - \alpha D^T \left(y^{(t)} - Dh^{(t)} \right) \right)$$

$$\Upsilon_{\beta}(v) = \text{sign}(v) \max\{|v| - \beta, 0\}$$

Conv. $\frac{1}{\alpha}$ mayor que el mayor valor propio de $D^T D$

DL

$$D = D - \alpha (Dh - y) h$$

ISTA

$$h^{(t)} = \Upsilon_{\lambda\alpha} \left(h^{(t)} - \alpha D^T \left(y^{(t)} - Dh^{(t)} \right) \right)$$

$$\Upsilon_{\beta}(v) = \text{sign}(v) \max\{|v| - \beta, 0\}$$

Conv. $\frac{1}{\alpha}$ mayor que el mayor valor propio de $D^T D$

DL

$$D = D - \alpha (Dh - y) h$$

Conv. $\|D_{:,j}\|_2 \leq 1 (\forall j)$

Surrogate models

Dado un conjunto de parámetros $x \in \mathbb{R}^n$, es posible medir su desempeño por medio de una función de aptitud (fitness) $f(x) = y$. Suponiendo que f es una función computacionalmente costosa, se crea un modelo $\psi(x)$ de la forma

$$f(x) \approx \tilde{f}(x) = \psi(x)$$

por medio de un conjunto de datos observados $X = \{x^{(r)}\}_{r=1}^m$, a partir de los cuales se obtiene el conjunto de valores objetivo $Y = \{y^{(r)}\}_{r=1}^m = \{f(x^{(r)})\}_{r=1}^m$

Problema original

$$g(D) = \min_{D,h} \{ \|Y - Dh\|_2^2 + \lambda \|h\|_1 \}$$

Problema original

$$g(D) = \min_{D,h} \{ \|Y - Dh\|_2^2 + \lambda \|h\|_1 \}$$

Versión "online"

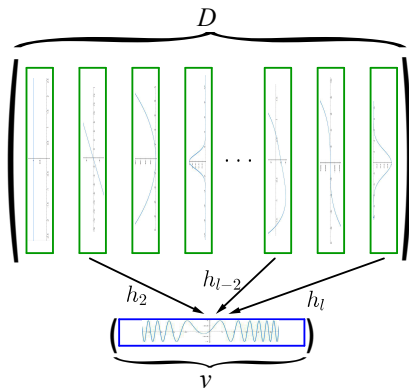
$$\tilde{g}_p(D_p) = \min_{D_p, h_p} \{ \|Y_p - D_p h_p\|_2^2 + \lambda \|h_p\|_1 \}$$

Es posible obtener D_p , de manera eficiente, utilizando D_{p-1} como parámetro inicial. Tal que $X_p = X_{p-1} \cup X_{new}$.

DL con "codewords" paramétricos

Utiliza un diccionario sobrecompleto $D(x)$, de funciones base, para establecer una representación dispersa h de un conjunto de muestras y

$$y = D(x) \cdot h$$



DL con "codewords" paramétricos

Diccionario paramétrico

$$D(x) = (\phi_1(x), \phi_2(x), \phi_3(x), \dots, \phi_l(x))$$

DL con "codewords" paramétricos

Diccionario paramétrico

$$D(x) = (\phi_1(x), \phi_2(x), \phi_3(x), \dots, \phi_l(x))$$

Diccionario evaluado

$$D = \begin{pmatrix} D(x^{(1)}) \\ D(x^{(2)}) \\ \vdots \\ D(x^{(m)}) \end{pmatrix} = \begin{pmatrix} \phi_1(x^{(1)}) & \phi_2(x^{(1)}) & \phi_3(x^{(1)}) & \dots & \phi_l(x^{(1)}) \\ \phi_1(x^{(2)}) & \phi_2(x^{(2)}) & \phi_3(x^{(2)}) & \dots & \phi_l(x^{(2)}) \\ \vdots & \vdots & \vdots & \dots & \vdots \\ \phi_1(x^{(m)}) & \phi_2(x^{(m)}) & \phi_3(x^{(m)}) & \dots & \phi_l(x^{(m)}) \end{pmatrix}$$

"Codewords" Gaussianos

Cada "codeword" es una función $\phi_j : \mathbb{R}^n \rightarrow \mathbb{R}$ de la forma

$$\phi_j(x) = \exp\left(-\frac{1}{2}(x - \mu^{(j)})^T (\Sigma^{(j)})^{-1} (x - \mu^{(j)})\right)$$

donde $\mu^{(j)} \in \mathbb{R}^n$ representa el vector de promedios y $\Sigma^{(j)} \in \mathbb{R}^{n \times n}$ la matriz de covarianza.

DL con "codewords" paramétricos

"Codewords" Gaussianos

Cada "codeword" es una función $\phi_j : \mathbb{R}^n \rightarrow \mathbb{R}$ de la forma

$$\phi_j(x) = \exp\left(-\frac{1}{2}(x - \mu^{(j)})^T (\Sigma^{(j)})^{-1} (x - \mu^{(j)})\right)$$

donde $\mu^{(j)} \in \mathbb{R}^n$ representa el vector de promedios y $\Sigma^{(j)} \in \mathbb{R}^{n \times n}$ la matriz de covarianza.

$$\mu^{(j)} = \left(\mu_1^{(j)}, \mu_2^{(j)}, \mu_3^{(j)}, \dots, \mu_n^{(j)}\right)^T$$

$$\left(\Sigma^{(j)}\right)^{-1} = \begin{pmatrix} \rho_{1,1}^{(j)} & \rho_{1,2}^{(j)} & \rho_{1,3}^{(j)} & \cdots & \rho_{1,n-1}^{(j)} & \rho_{1,n}^{(j)} \\ \rho_{1,2}^{(j)} & \rho_{2,2}^{(j)} & \rho_{2,3}^{(j)} & \cdots & \rho_{2,n-1}^{(j)} & \rho_{2,n}^{(j)} \\ & & & \ddots & & \\ \rho_{1,n-1}^{(j)} & \rho_{2,n-1}^{(j)} & \rho_{3,n-1}^{(j)} & \cdots & \rho_{n-1,n-1}^{(j)} & \rho_{n-1,n}^{(j)} \\ \rho_{1,n}^{(j)} & \rho_{2,n}^{(j)} & \rho_{3,n}^{(j)} & \cdots & \rho_{n-1,n}^{(j)} & \rho_{n,n}^{(j)} \end{pmatrix}$$

Surrogate model

Modelo sustituto

$$\psi(x) = \sum_{j=1}^l \phi_j(x) h_j = D(x) \cdot h$$

Surrogate model

Modelo sustituto

$$\psi(x) = \sum_{j=1}^l \phi_j(x) h_j = D(x) \cdot h$$

Arreglo de parámetros "code"

$$\begin{aligned} \theta_j &= \left(\rho_{1,1}^{(j)} \cdots \rho_{1,n}^{(j)} \rho_{2,2}^{(j)} \cdots \rho_{2,n}^{(j)} \rho_{3,3}^{(j)} \cdots \rho_{3,n}^{(j)} \cdots \rho_{n-1,n-1}^{(j)} \rho_{n-1,n}^{(j)} \rho_{n,n}^{(j)} \mu^{(j)} \right)^T \\ &= \left(\rho_{1,1}^{(j)} \cdots \rho_{1,n}^{(j)} \rho_{2,2}^{(j)} \cdots \rho_{2,n}^{(j)} \cdots \rho_{n-1,n-1}^{(j)} \rho_{n-1,n}^{(j)} \rho_{n,n}^{(j)} \mu_1^{(j)} \mu_2^{(j)} \cdots \mu_n^{(j)} \right)^T \end{aligned}$$

$$\Theta = (\theta_1, \theta_2, \theta_3, \cdots, \theta_{l-1}, \theta_l)$$

Error cuadrático medio

$$E(\Theta) = \frac{1}{2m} \sum_{r=1}^m \left(\psi(x^{(r)}) - y^{(r)} \right)^2$$

Surrogate model

Error cuadrático medio

$$E(\Theta) = \frac{1}{2m} \sum_{r=1}^m \left(\psi(x^{(r)}) - y^{(r)} \right)^2$$

Batch Gradient Descent update rule

$$\Theta_p = \Theta_p - \alpha \frac{\partial E(\Theta)}{\partial \Theta_p}$$

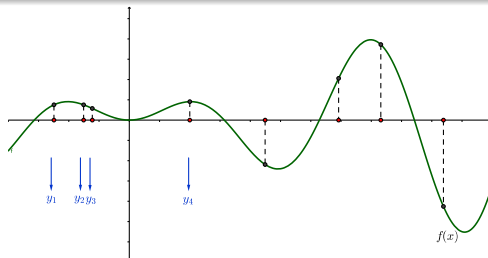
Surrogate model

Error cuadrático medio

$$E(\Theta) = \frac{1}{2m} \sum_{r=1}^m \left(\psi(x^{(r)}) - y^{(r)} \right)^2$$

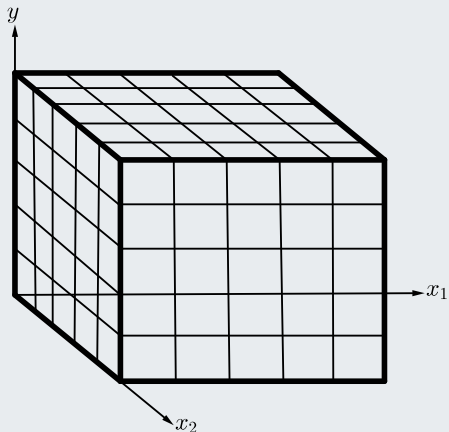
Batch Gradient Descent update rule

$$\Theta_p = \Theta_p - \alpha \frac{\partial E(\Theta)}{\partial \Theta_p}$$



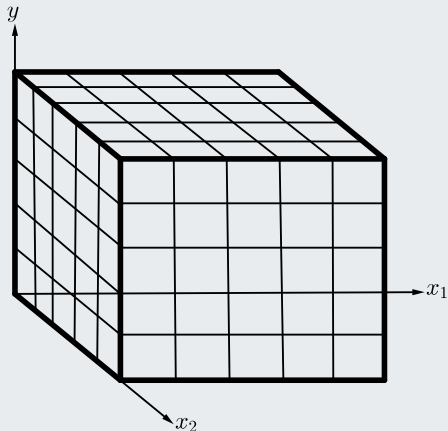
Surrogate model

Partición del espacio



Surrogate model

Partición del espacio

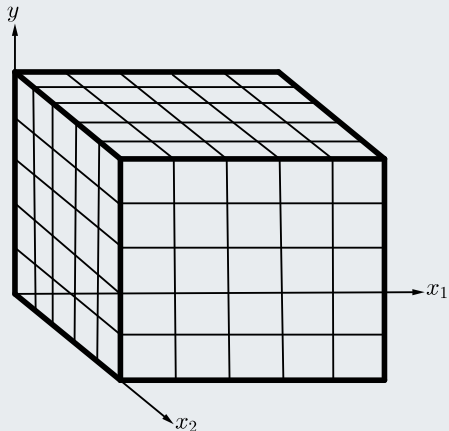


- Espacio de búsqueda:

$$[a, b]^n$$

Surrogate model

Partición del espacio



- Espacio de búsqueda:

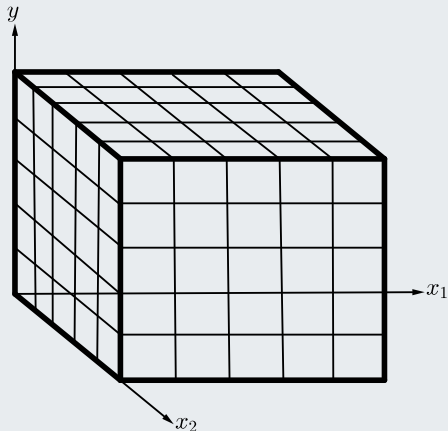
$$[a, b]^n$$

- Partición en sub-blocks de dimensión:

$$\left(\frac{b-a}{T}\right)^n$$

Surrogate model

Partición del espacio



- Espacio de búsqueda:

$$[a, b]^n$$

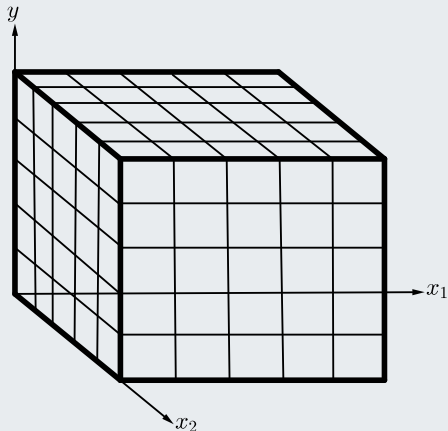
- Partición en sub-blocks de dimensión:

$$\left(\frac{b-a}{T}\right)^n$$

- T : cantidad de sub-blocks por dimensión

Surrogate model

Partición del espacio



- Espacio de búsqueda:

$$[a, b]^n$$

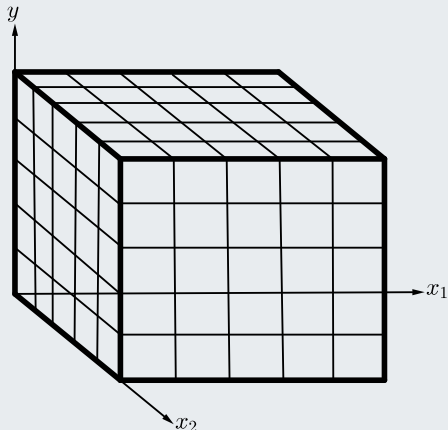
- Partición en sub-blocks de dimensión:

$$\left(\frac{b-a}{T}\right)^n$$

- T : cantidad de sub-blocks por dimensión
- T^n : total de sub-blocks

Surrogate model

Partición del espacio



- Espacio de búsqueda:

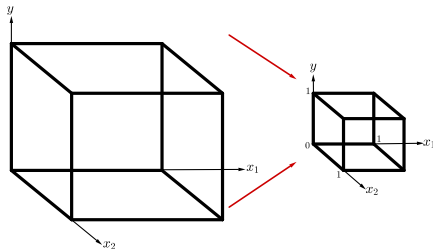
$$[a, b]^n$$

- Partición en sub-blocks de dimensión:

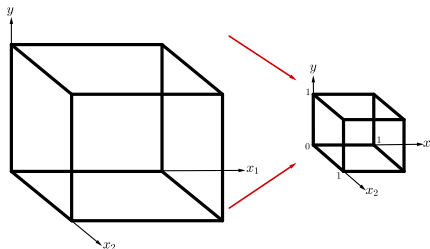
$$\left(\frac{b-a}{T}\right)^n$$

- T : cantidad de sub-blocks por dimensión
- T^n : total de sub-blocks
- **Un modelo sustituto por sub-block**

Surrogate model



Surrogate model



Normalización: $[0, 1]^n \rightarrow [0 \pm \varepsilon, 1 \pm \varepsilon]$

$$x_i^{(t,r)} \leftarrow \frac{x_i^{(t,r)} - a_i^{(t,r)}}{b_i^{(t,r)} - a_i^{(t,r)}}$$

$$y^{(t)} \leftarrow \xi_t \cdot y^{(t)}$$

Suavizamiento

$$a_{i_{ext}}^{(t)} = \min \left\{ \max \left\{ x_i^{(r)}; \forall r \text{ tal que } x_i^{(r)} < a_i^{(t)} \right\}, a_i^{(t)} \right\}$$

$$b_{i_{ext}}^{(t)} = \max \left\{ \min \left\{ x_i^{(r)}; \forall r \text{ tal que } x_i^{(r)} > b_i^{(t)} \right\}, b_i^{(t)} \right\}$$

Suavizamiento

$$a_{i_{ext}}^{(t)} = \min \left\{ \max \left\{ x_i^{(r)}; \forall r \text{ tal que } x_i^{(r)} < a_i^{(t)} \right\}, a_i^{(t)} \right\}$$

$$b_{i_{ext}}^{(t)} = \max \left\{ \min \left\{ x_i^{(r)}; \forall r \text{ tal que } x_i^{(r)} > b_i^{(t)} \right\}, b_i^{(t)} \right\}$$

$$t_{i_{ext}} = [a_{i_{ext}}^{(t)}, b_{i_{ext}}^{(t)}]$$

subintervalo extendido, correspondiente al subintervalo t_i de la dimensión i en el espacio de búsqueda.

Surrogate model

- $h^{(t_{ext})}$: vector disperso de la combinación lineal del modelo sustituto correspondiente al sub-block extendido t_{ext} .

Surrogate model

- $h^{(t_{ext})}$: vector disperso de la combinación lineal del modelo sustituto correspondiente al sub-block extendido t_{ext} .
- Modelo sustituto correspondiente al sub-block t :

$$\psi^{(t)}(x) = D(x) \cdot h^{(t_{ext})}$$

Surrogate model

- $h^{(t_{ext})}$: vector disperso de la combinación lineal del modelo sustituto correspondiente al sub-block extendido t_{ext} .
- Modelo sustituto correspondiente al sub-block t :

$$\psi^{(t)}(x) = D(x) \cdot h^{(t_{ext})}$$

- $I = \{1, 2, 3, \dots, T^n\}$: conjunto de índices correspondientes a los sub-blocks.

Surrogate model

- $h^{(t_{ext})}$: vector disperso de la combinación lineal del modelo sustituto correspondiente al sub-block extendido t_{ext} .
- Modelo sustituto correspondiente al sub-block t :

$$\psi^{(t)}(x) = D(x) \cdot h^{(t_{ext})}$$

- $I = \{1, 2, 3, \dots, T^n\}$: conjunto de índices correspondientes a los sub-blocks.
- I_{act} : conjunto de índices de los sub-blocs activos.

Surrogate model

- $h^{(t_{ext})}$: vector disperso de la combinación lineal del modelo sustituto correspondiente al sub-block extendido t_{ext} .
- Modelo sustituto correspondiente al sub-block t :

$$\psi^{(t)}(x) = D(x) \cdot h^{(t_{ext})}$$

- $I = \{1, 2, 3, \dots, T^n\}$: conjunto de índices correspondientes a los sub-blocks.
- I_{act} : conjunto de índices de los sub-blocs activos.
- $\mathcal{H} = (h^{(1_{ext})}, h^{(2_{ext})}, h^{(3_{ext})}, \dots)$, con $t_{ext} \in I_{act}$.

Surrogate model

- $h^{(t_{ext})}$: vector disperso de la combinación lineal del modelo sustituto correspondiente al sub-block extendido t_{ext} .
- Modelo sustituto correspondiente al sub-block t :

$$\psi^{(t)}(x) = D(x) \cdot h^{(t_{ext})}$$

- $I = \{1, 2, 3, \dots, T^n\}$: conjunto de índices correspondientes a los sub-blocks.
- I_{act} : conjunto de índices de los sub-blocs activos.
- $\mathcal{H} = (h^{(1_{ext})}, h^{(2_{ext})}, h^{(3_{ext})}, \dots)$, con $t_{ext} \in I_{act}$.
- $\mathcal{D} = (D^{(1_{ext})}, D^{(2_{ext})}, D^{(3_{ext})}, \dots)$, con $t_{ext} \in I_{act}$ y $D^{(t_{ext})} = D(X^{(t_{ext})})$.

Surrogate model

- $h^{(t_{ext})}$: vector disperso de la combinación lineal del modelo sustituto correspondiente al sub-block extendido t_{ext} .
- Modelo sustituto correspondiente al sub-block t :

$$\psi^{(t)}(x) = D(x) \cdot h^{(t_{ext})}$$

- $I = \{1, 2, 3, \dots, T^n\}$: conjunto de índices correspondientes a los sub-blocks.
- I_{act} : conjunto de índices de los sub-blocs activos.
- $\mathcal{H} = (h^{(1_{ext})}, h^{(2_{ext})}, h^{(3_{ext})}, \dots)$, con $t_{ext} \in I_{act}$.
- $\mathcal{D} = (D^{(1_{ext})}, D^{(2_{ext})}, D^{(3_{ext})}, \dots)$, con $t_{ext} \in I_{act}$ y $D^{(t_{ext})} = D(X^{(t_{ext})})$.
- Se toma un conjunto de muestras iniciales X_0 , con $\text{card}\{X_0\} = m$.

Surrogate model

- $h^{(t_{ext})}$: vector disperso de la combinación lineal del modelo sustituto correspondiente al sub-block extendido t_{ext} .
- Modelo sustituto correspondiente al sub-block t :

$$\psi^{(t)}(x) = D(x) \cdot h^{(t_{ext})}$$

- $I = \{1, 2, 3, \dots, T^n\}$: conjunto de índices correspondientes a los sub-blocks.
- I_{act} : conjunto de índices de los sub-blocs activos.
- $\mathcal{H} = (h^{(1_{ext})}, h^{(2_{ext})}, h^{(3_{ext})}, \dots)$, con $t_{ext} \in I_{act}$.
- $\mathcal{D} = (D^{(1_{ext})}, D^{(2_{ext})}, D^{(3_{ext})}, \dots)$, con $t_{ext} \in I_{act}$ y $D^{(t_{ext})} = D(X^{(t_{ext})})$.
- Se toma un conjunto de muestras iniciales X_0 , con $\text{card}\{X_0\} = m$.
- Se van agregando nuevas muestras X_{new} , en cada instante $p = 1, 2, 3, \dots, \eta$, por medio de un algoritmo de búsqueda lineal (BackTracking).

Algorithm 1: SESM-Sequential Mode**Input:** $X_0, \mathcal{H}_0, \Theta_0, \alpha, \lambda, \zeta, \eta, \omega, tol, iterMax$ **Output:** $\mathcal{H}, D(x)$

```

1  Calcular  $Y_0$  y  $I_{act}$ ;  $\Theta \leftarrow \Theta_0$ 
2  for  $p = 1 : \eta$  do
3      for  $t \in I_{act}$  do
4          Calcular y normalizar  $X_{p-1}^{(text)}$  y  $Y_{p-1}^{(text)}$ 
5           $h^{(t)} \leftarrow \mathcal{H}_0(:, t)$ 
6      end
7       $Perm \leftarrow [P_1(I_{act}), \dots, P_\omega(I_{act})]$ 
8      for  $cont = 1 : \omega \cdot T$  do
9           $t \leftarrow Perm(cont)$ ;  $iter \leftarrow 1$ ;  $E(\Theta) \leftarrow tol + 1$ 
10         while  $iter \leq iterMax \wedge E(\Theta) > tol$  do
11              $\Theta \leftarrow \text{D-GD}(X_{p-1}^{(text)}, Y_{p-1}^{(text)}, h^{(text)}, \Theta, iterD)$ 
12             Calcular  $D^{(text)}$ 
13              $h^{(t)} \leftarrow \text{h-ISTA}(Y_{p-1}^{(text)}, h^{(t)}, D^{(text)}, \alpha, \lambda, iterh)$ 
14             Calcular  $E(\Theta)$ ;  $iter \leftarrow iter + 1$ 
15         end
16     end
17      $X_{new} \leftarrow \zeta$  nuevas muestras (Backtracking line search)
18      $X_p \leftarrow X_{p-1} \cup X_{new}$ 
19     Actualizar  $Y_p$  y  $I_{act}$ 
20 end
21 Calcular  $D(x)$ 
22 for  $t \in I_{act}$  do
23      $\mathcal{H}(:, t) \leftarrow \frac{h^{(t)}}{\xi_{text}}$ 
24 end

```


Surrogate model

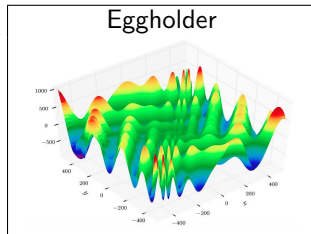
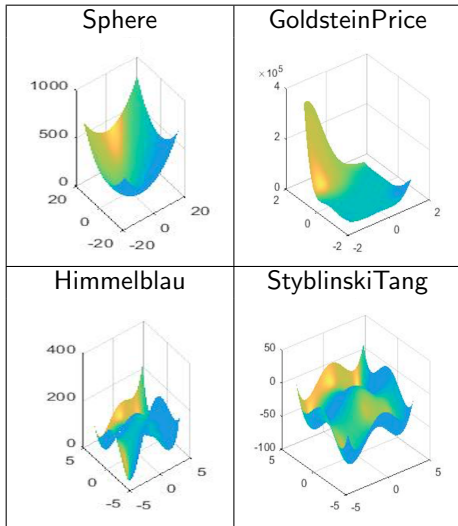
Algorithm 2: SESM-Batch Mode**Input:** $X_0, \mathcal{H}_0, \Theta_0, \alpha, \lambda, \zeta, \eta, tol, iterMax$ **Output:** $\mathcal{H}, D(x)$

```

1  Calcular  $Y_0$  y  $I_{act}$ ;  $\Theta \leftarrow \Theta_0$ 
2  for  $p = 1 : \eta$  do
3      for  $t \in I_{act}$  do
4          Calcular y normalizar  $X_{p-1}^{(t_{ext})}$  y  $Y_{p-1}^{(t_{ext})}$ 
5           $h^{(t)} \leftarrow \mathcal{H}_0(:, t)$ 
6      end
7       $iter \leftarrow 1$ ;  $E(\Theta) \leftarrow tol + 1$ 
8      Calcular  $X_{ext_{p-1}}$  y  $Y_{ext_{p-1}}$ 
9      while  $iter \leq iterMax \wedge E(\Theta) > tol$  do
10          $\Theta \leftarrow \text{D-GD}(X_{ext_{p-1}}, Y_{ext_{p-1}}, h^{(t_{ext})}, \Theta, iterD)$ 
11         Calcular  $\mathcal{D}$ 
12          $h^{(t)} \leftarrow \text{h-ISTA}(Y_{ext_{p-1}}, h^{(t)}, \mathcal{D}, \alpha, \lambda, iterh)$ 
13         Calcular  $E(\Theta)$ ;  $iter \leftarrow iter + 1$ 
14     end
15      $X_{new} \leftarrow \zeta$  nuevas muestras (Backtracking line search)
16      $X_p \leftarrow X_{p-1} \cup X_{new}$ 
17     Actualizar  $Y_p$  y  $I_{act}$ 
18 end
19 Calcular  $D(x)$ 
20 for  $t \in I_{act}$  do
21      $\mathcal{H}(:, t) \leftarrow \frac{h^{(t)}}{\xi_{t_{ext}}}$ 
22 end

```

Funciones de prueba



Análisis de error (iterD - iterh)

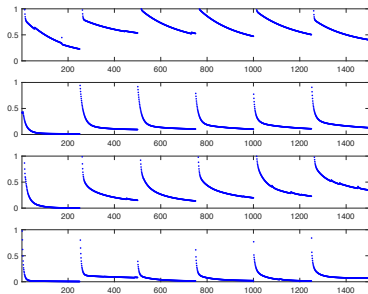
Configuración

- $n = 2$
- $m = 3$
- $\zeta = 3$
- $\eta = 3$
- $\omega = 3$
- $l = 250$
- $\alpha = 0,001$
- $\lambda = 0,01$
- $iterMax = 250$
- $iterD = \left\{ 1, \frac{0,15nl(n+3)}{2} \right\}$
- $iterh = \{1, 0,15l\}$

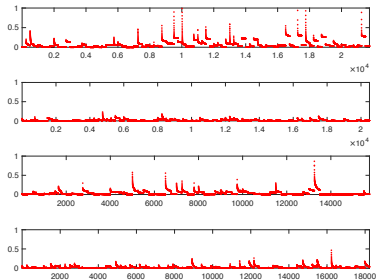
Análisis de error (iterD - iterh)

Sphere

Batch Mode



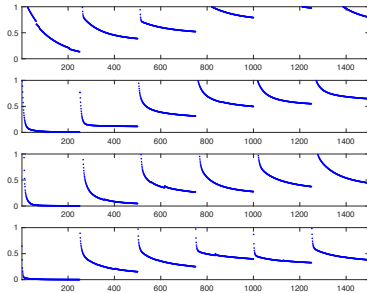
Sequential Mode



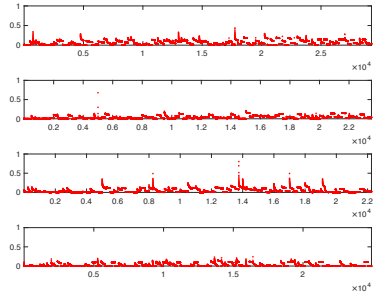
Análisis de error (iterD - iterh)

GoldsteinPrice

Batch Mode



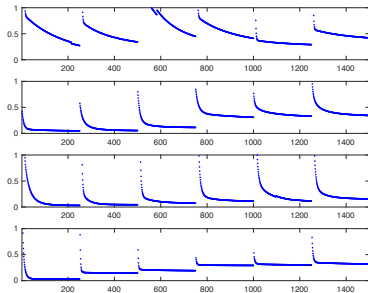
Sequential Mode



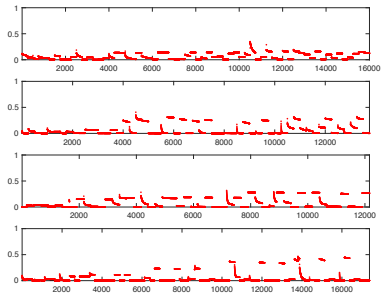
Análisis de error (iterD - iterh)

Himmelblau

Batch Mode



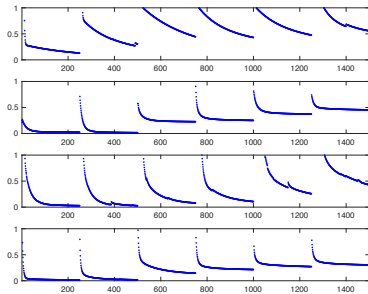
Sequential Mode



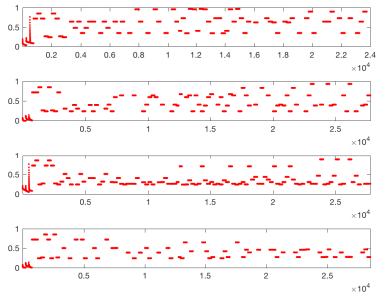
Análisis de error (iterD - iterh)

StyblinskiTang

Batch Mode



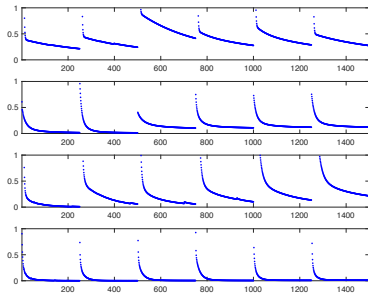
Sequential Mode



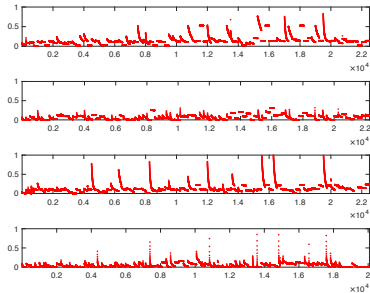
Análisis de error (iterD - iterh)

Eggholder

Batch Mode



Sequential Mode



Análisis de error (iterD - iterh)

- ① Una sola iteración por subrutina no le permite al método aprender lo suficiente de los datos observados.

Análisis de error (iterD - iterh)

- ① Una sola iteración por subrutina no le permite al método aprender lo suficiente de los datos observados.
- ② El aumento en la cantidad de iteraciones de las subrutinas ayudan a la eficiencia de los métodos.

Análisis de error (iterD - iterh)

- ① Una sola iteración por subrutina no le permite al método aprender lo suficiente de los datos observados.
- ② El aumento en la cantidad de iteraciones de las subrutinas ayudan a la eficiencia de los métodos.
- ③ Una cantidad considerable ($\geq 50\%$) de iteraciones, por subrutina, sobreajusta los parámetros a las muestras en el instante p . Obligándolo a empezar de 0 al ajustar con las nuevas muestras en el instante $p + 1$.

Análisis de error (iterD - iterh)

- 1 Una sola iteración por subrutina no le permite al método aprender lo suficiente de los datos observados.
- 2 El aumento en la cantidad de iteraciones de las subrutinas ayudan a la eficiencia de los métodos.
- 3 Una cantidad considerable ($\geq 50\%$) de iteraciones, por subrutina, sobreajusta los parámetros a las muestras en el instante p . Obligándolo a empezar de 0 al ajustar con las nuevas muestras en el instante $p + 1$.
- 4 El aumento en la cantidad de iteraciones por subrutina debe ser pequeño, comparado con la cantidad de parámetros que ajusta cada método.

Análisis de error (iterD - iterh)

- 1 Una sola iteración por subrutina no le permite al método aprender lo suficiente de los datos observados.
- 2 El aumento en la cantidad de iteraciones de las subrutinas ayudan a la eficiencia de los métodos.
- 3 Una cantidad considerable ($\geq 50\%$) de iteraciones, por subrutina, sobreajusta los parámetros a las muestras en el instante p . Obligándolo a empezar de 0 al ajustar con las nuevas muestras en el instante $p + 1$.
- 4 El aumento en la cantidad de iteraciones por subrutina debe ser pequeño, comparado con la cantidad de parámetros que ajusta cada método.
- 5 Parece que la combinación con 2-2 produce un estancamiento del método en general, se está probando una variación híbrida que cambia la cantidad de iteraciones a 1-2 o 2-1 (con cierta probabilidad) cuando se detecte algún estancamiento en la disminución del RMES.

Ejemplo de aproximación

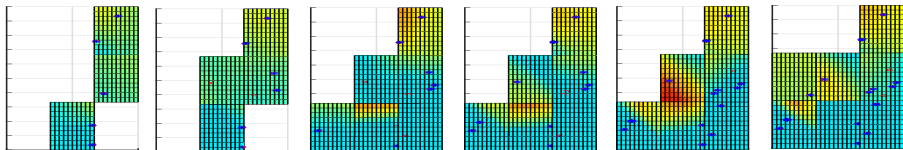
Himmelblau

- Batch Mode
- $n = 2$
- $m = 5$
- $\eta = 5$
- $\zeta = 3$
- $l = 250$
- $\alpha = 0,001$
- $\lambda = 0,01$
- $iterMax = 1000$
- $iterD = \frac{0,15nl(n+3)}{2}$
- $iterh = 0,15l$

Ejemplo de aproximación

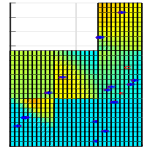
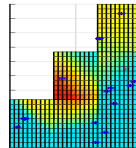
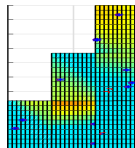
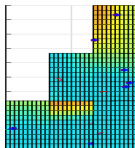
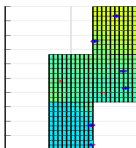
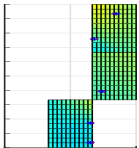
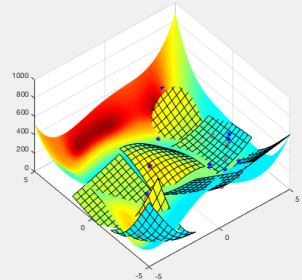
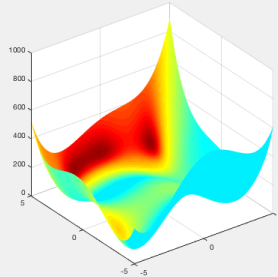
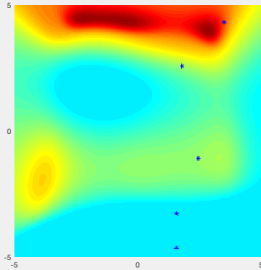
Himmelblau

- Batch Mode
- $n = 2$
- $m = 5$
- $\eta = 5$
- $\zeta = 3$
- $l = 250$
- $\alpha = 0,001$
- $\lambda = 0,01$
- $iterMax = 1000$
- $iterD = \frac{0,15nl(n+3)}{2}$
- $iterh = 0,15l$



Ejemplo de aproximación

Himmelblau



- Optimizador de parámetros:

$$Param = (l, T, iterD, iterh)$$

$$F = (Tiempo, RMSE, ErrorTotal, Ruido)$$

Siguiete presentación

- Optimizador de parámetros:

$$Param = (l, T, iterD, iterh)$$

$$F = (Tiempo, RMSE, ErrorTotal, Ruido)$$

- Integración del modelo sustituto propuesto con el optimizador de funciones costosas (NSGA-II, SPEA-2)

- Optimizador de parámetros:

$$Param = (l, T, iterD, iterh)$$

$$F = (Tiempo, RMSE, ErrorTotal, Ruido)$$

- Integración del modelo sustituto propuesto con el optimizador de funciones costosas (NSGA-II, SPEA-2)
- Aplicaciones

Consultas, sugerencias, comentarios,...?