Aplicații ale algoritmului EM și nu numai...

Ciobanu Sebastian

Coordonator: Conf. dr. Liviu Ciortuz

Cuprins

- **Teorie**
  - Algoritmul EM
  - Procese gaussiene
- **Aplicații**
  - Image segmentation/compression
  - Missing data
  - Image deblurring
  - Regression. Gene expression

0.1. Algoritmul EM

Algoritmul EM (1)

- Este, de fapt, o schemă algoritmică
- E = Expectation
- M = Maximization
- La ce este folosit?
  - Maximizarea unei funcții
- Care funcție?
  - Funcția de log-verosimiliate a datelor (observabile)
- Când îl folosim?
  - Când avem de-a face cu date latente

Funcția de verosimilitate

- f(h) = P(D|h)
  - D = date (observabile)
  - h = ipoteză/model
    - De ex.: parametrii unei distribuții probabiliste:
      - D = {1,2,3}
      - h_0 = {mu = 0, sigma = 1}; distribuția normală
      - $f(h_0) = p(D|h_0)$ = p(1,2,3|mu = 0, sigma = 1)

        = p(1|mu = 0, sigma = 1) p(2|mu = 0, sigma = 1) p(3|mu = 0, sigma = 1)

        = ...

        = 5.78987e-05

Funcția de log-verosimilitate

- $f(h) = \ln P(D|h)$
  - D = date (observabile)
  - h = ipoteză/model
- De ce?
  - P(D|h) va fi un produs de numere din [0,1]
  - Avem de-a face cu derivate
  - Computerul folosește aproximări

Date - exemplu

- Complete

Notație: Y = (X, Z)

- Observabile
- Neobservabile

Preț	Tip produs
2.1	1
3	2
4	3
5	1
3.1	2
2	3
7	3

Latente

Notație: X

Notație: Z

… log-verosimilitatea datelor observabile…

Cum maximizăm

… log-verosimilitatea datelor observabile… ?

Ex:

$$\sum_{i=1}^{n} \ln\left(\sum_{j=1}^{k} p_{X|Z,h}(x_i|j,h)\, p_{Z|h}(j|h)\right)$$

1. Metode numerice standard (ex.: metoda gradientului)
2. **Algoritmul EM**

Algoritmul EM (2)

- Inițializare: ? => [W,] h
- While(...)
  - W = pasE(X,h)
  - h = pasM(X,W)
  - (implicit crește P(X|h_var))

Algoritmul EM (2)

- Inițializare: ? => [W,] h
- While(...)
  - W = E[g(Z)|X,h]
  - h = ln P(X,Z|h_var)
  - (implicit crește P(X|h_var))

Algoritmul EM (2)

- Inițializare: ? => [W,] h
- While(...)
  - W = E[g(Z)|X,h]
  - h = $E_{P(g(Z)|X,h)}$ [ln P(X,Z|h_var)]
  - (implicit crește P(X|h_var))

Algoritmul EM (2)

- Inițializare: ? => [W,] h
- While(...)
  - $W = E[g(Z)|X,h]$
  - $h = \text{argmax}_{h\_var}\ E_{P(g(Z)|X,h)}\ [\ln P(X,Z|h\_var)]$
  - (implicit crește $P(X|h\_var)$)

Algoritmul EM (2)

- Inițializare: ? => [W,] h
- While(...)
  - W = E[g(Z)|X,h]
  - $h = \text{argmax}_{h\_var} \overbrace{E_{P(g(Z)|X,h)}\,[\ln P(X,Z|h\_var)]}^{\overset{\mathbf{Q(h\_var|h)}}{\text{not.}}}$
  - (implicit crește P(X|h_var))

0.2. Procese gaussiene

Proces stocastic (aleator)

- Colecție indexată de variabile aleatoare

$f : \mathcal{X} \rightarrow \mathcal{F}(\Omega, \mathbb{R})$ - proces aleator
$\mathcal{X}$ - mulțime de indecși
$\mathcal{F}(\Omega, \mathbb{R})$ - mulțime de variabile aleatoare

Proces stocastic (aleator)

- Induce o distribuție de probabilitate peste:
  - Funcții: dacă $\mathcal{X}$ este finită
  - Aproximări de funcții: dacă $\mathcal{X}$ este infinită
    - Aproximăm $\mathcal{X}$ printr-un număr finit de indecși
- Deci, avem:

$$\mathcal{X} = \{x_1, \ldots, x_n\} \text{ sau } \mathcal{X} \approx \{x_1, \ldots, x_n\}$$

Proces stocastic (aleator)

$$f \overset{\text{not.}}{=} \begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_n) \end{bmatrix} \qquad v \overset{\text{not.}}{=} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

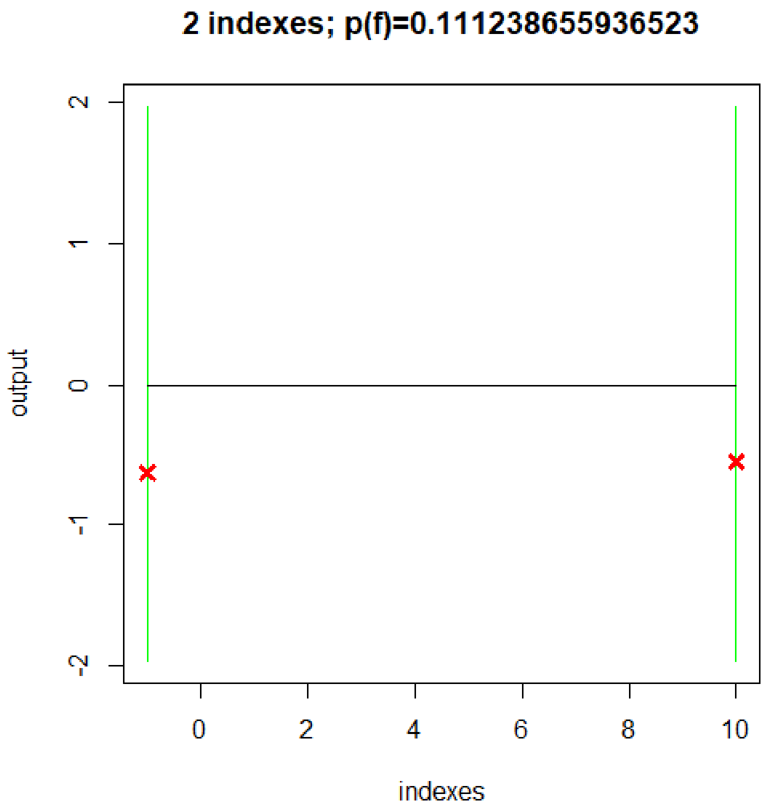
$$P(f = v) = P(f(x_1) = v_1, \ldots, f(x_n) = v_n)$$
$$p_f(v) = p_{f(x_1),\ldots,f(x_n)}(v_1, \ldots, v_n)$$

Proces gaussian

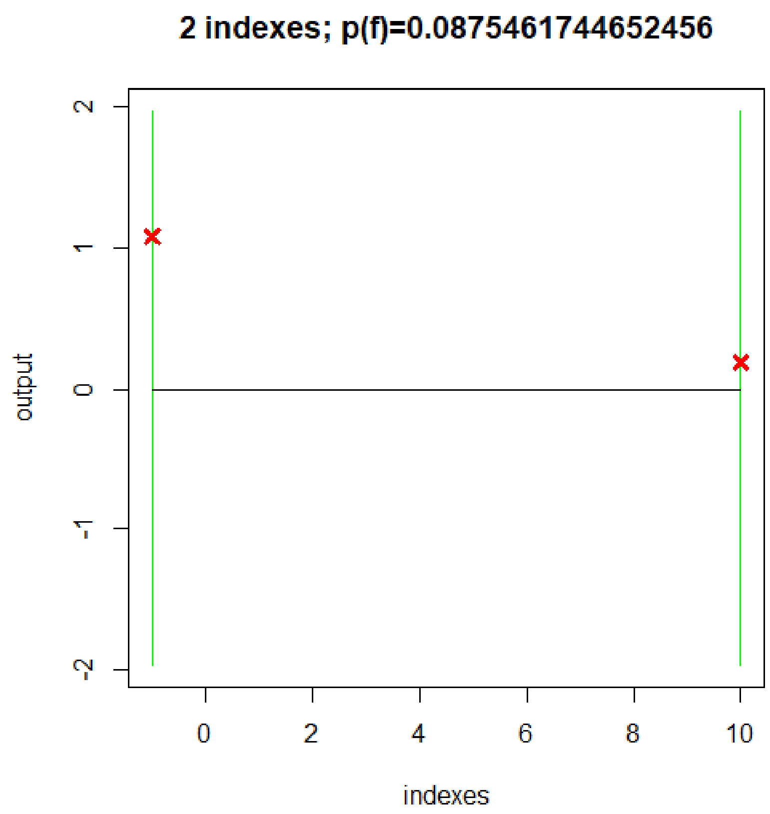
$\forall x_1, \ldots, x_n \in \mathcal{X} : (f(x_1), \ldots, f(x_n))$ urmează o distribuție normală

=> vom putea calcula P(f = x)

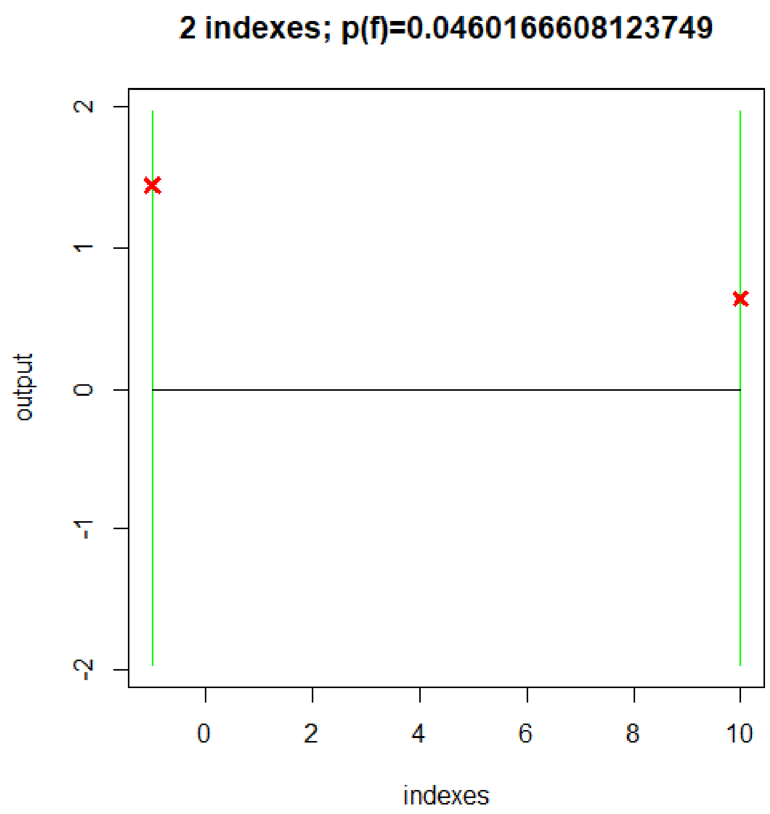
Proces gaussian



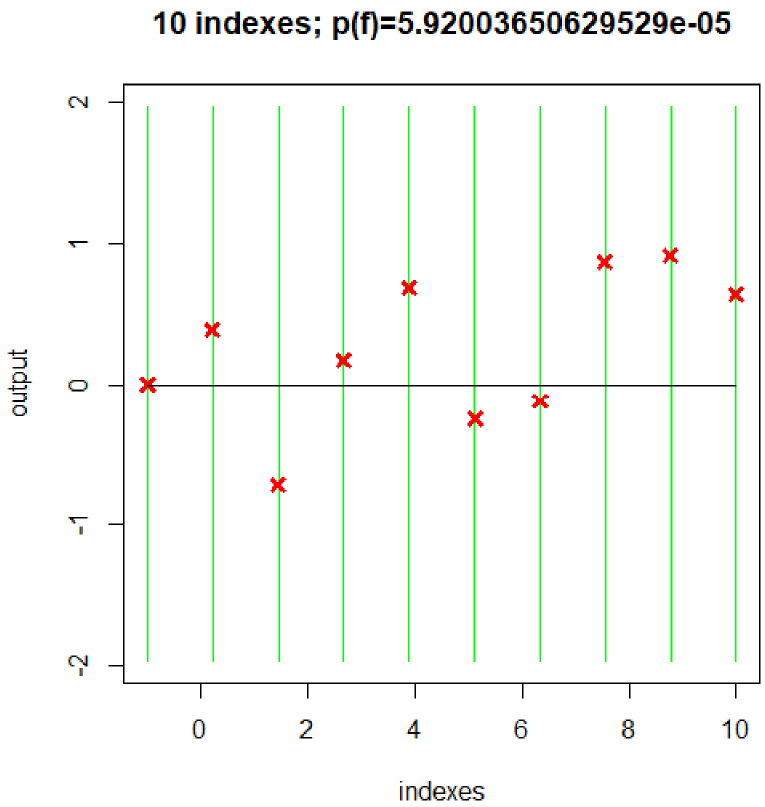
Proces gaussian



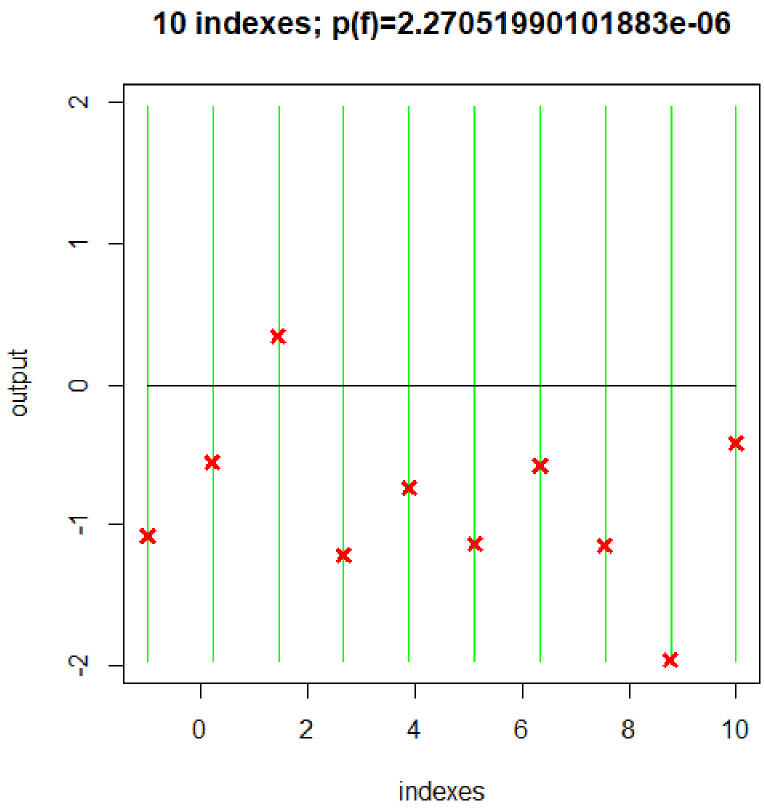
Proces gaussian



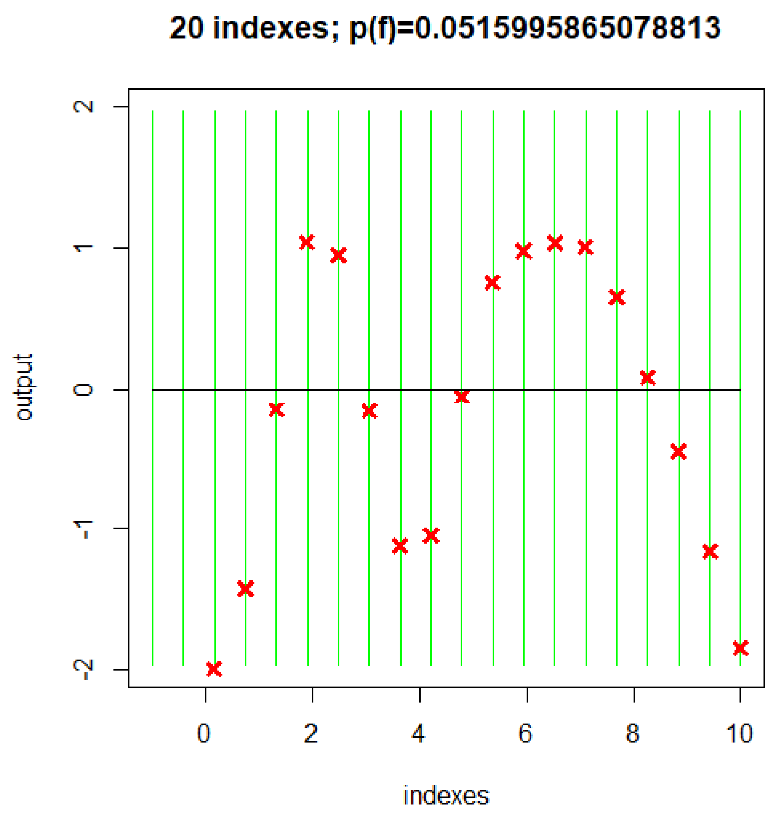
Proces gaussian



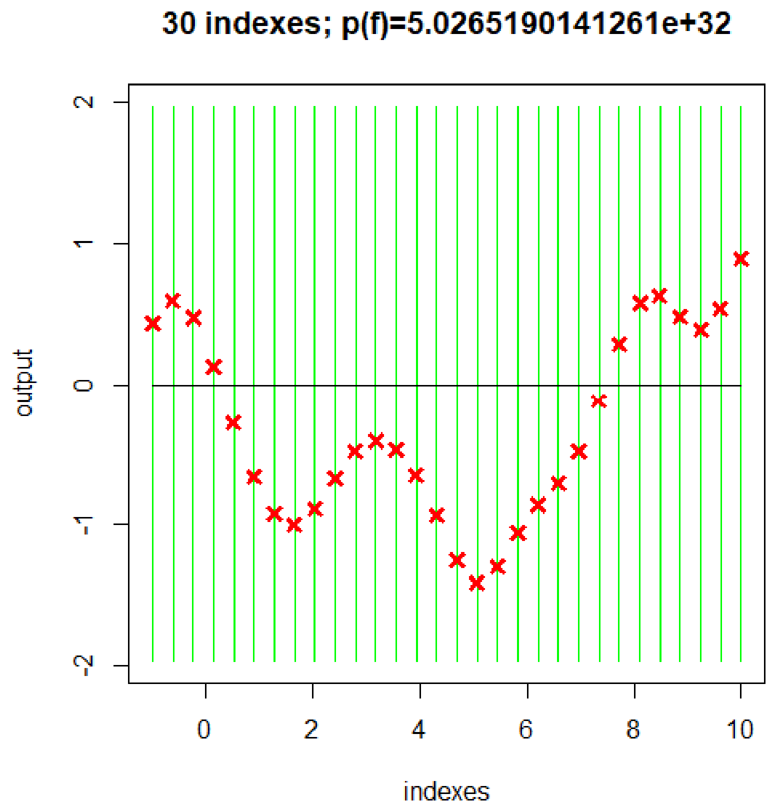
Proces gaussian



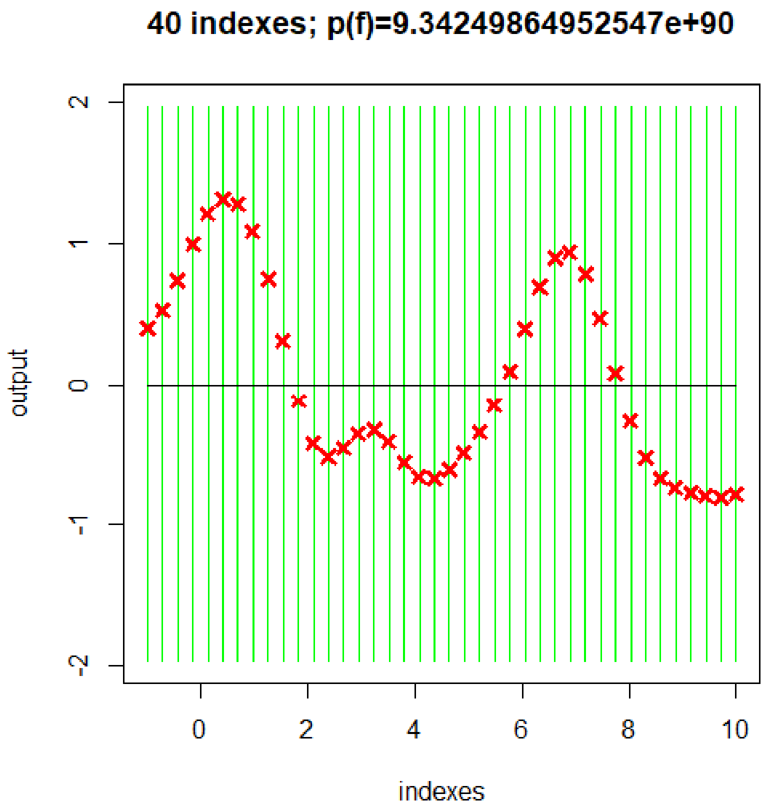
Proces gaussian



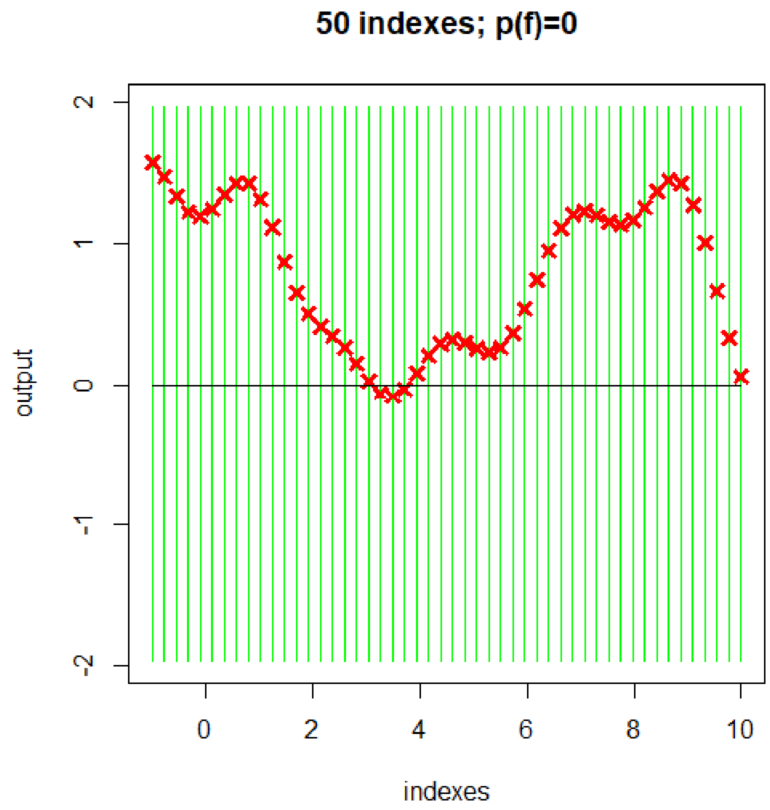
Proces gaussian



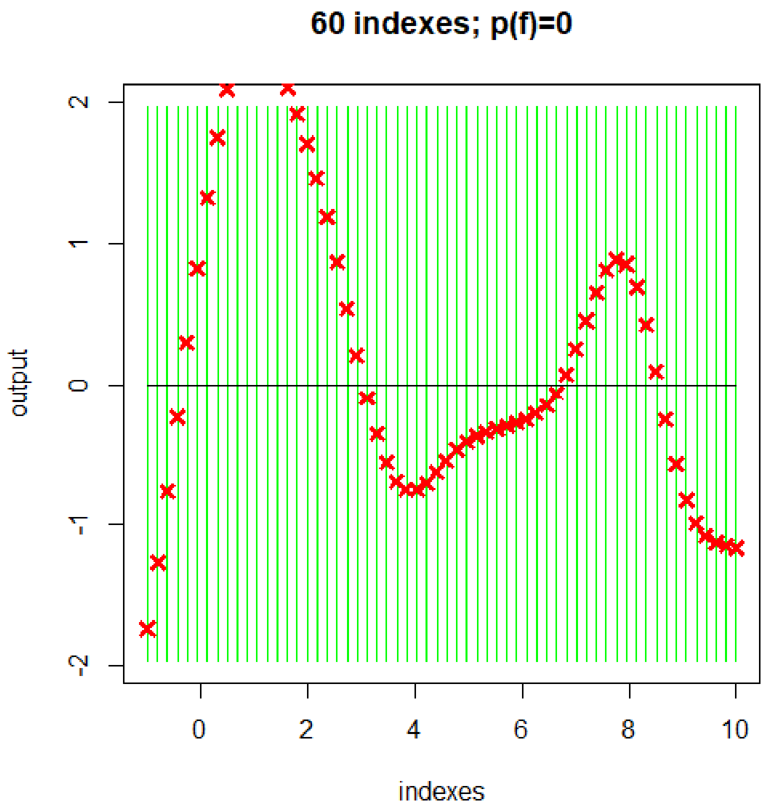
Proces gaussian



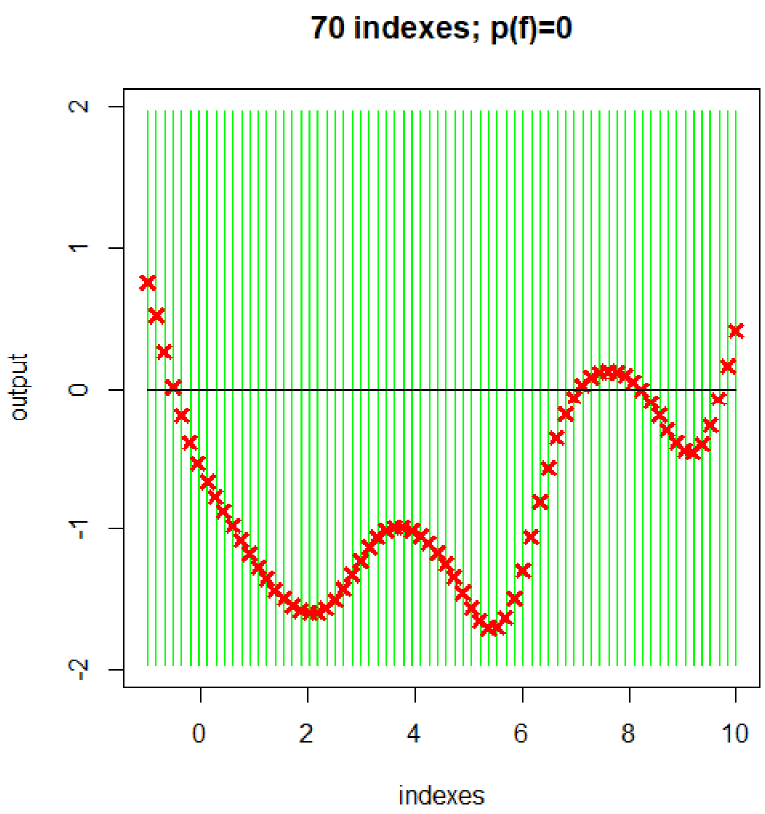
Proces gaussian



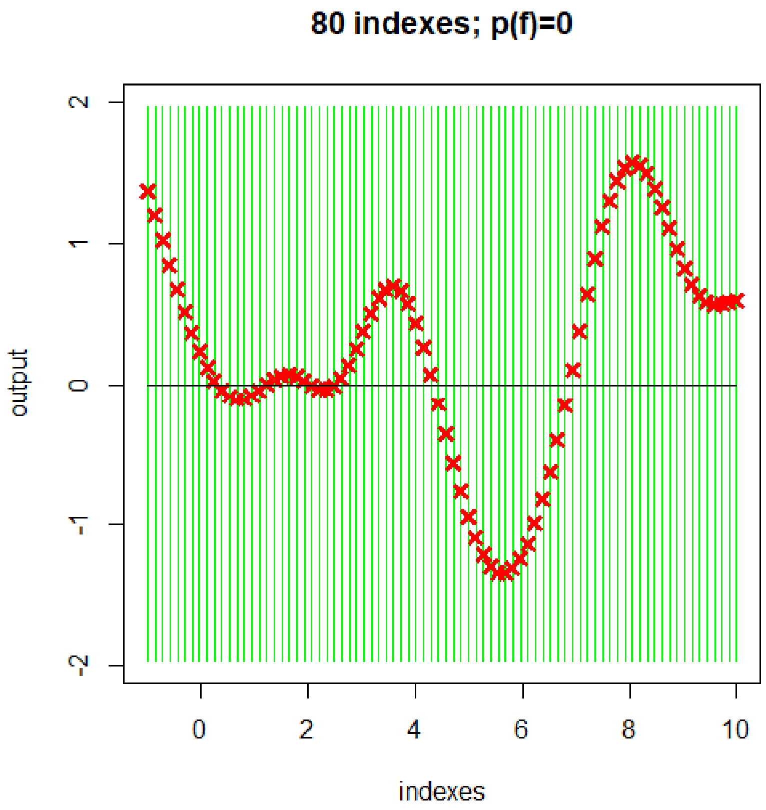
Proces gaussian



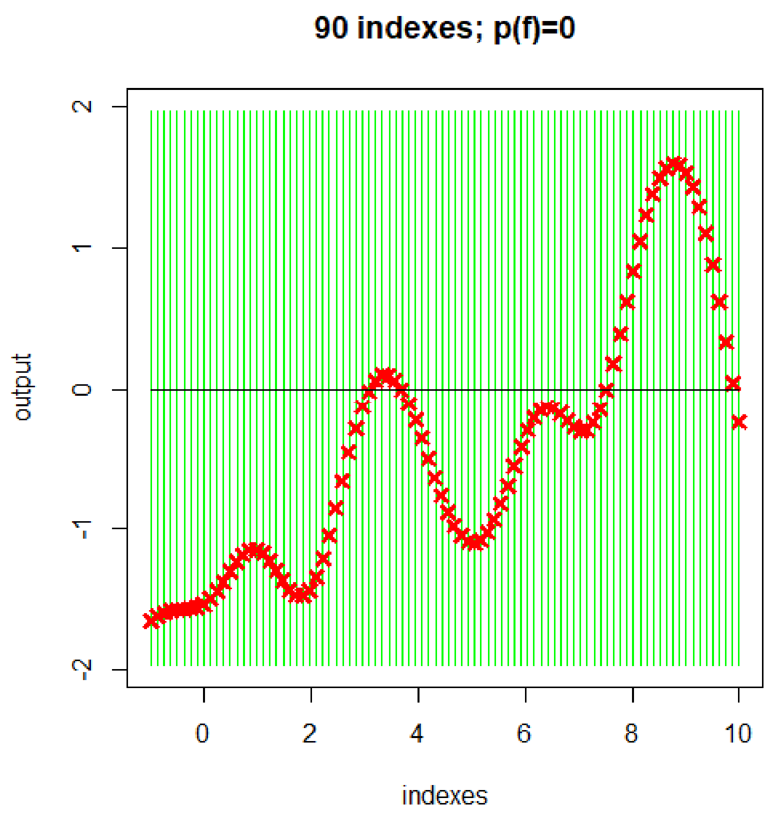
Proces gaussian



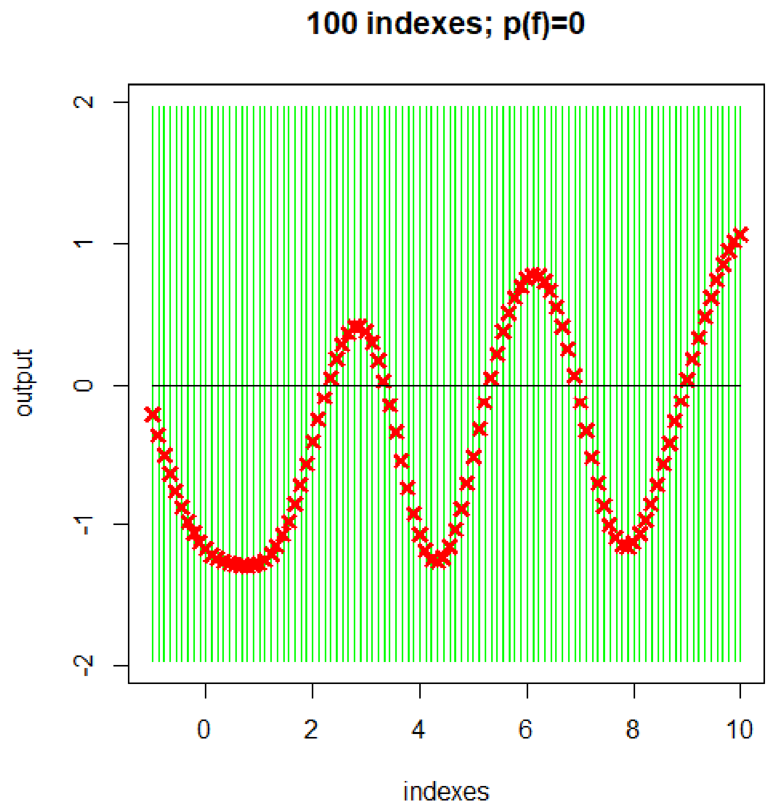
Proces gaussian



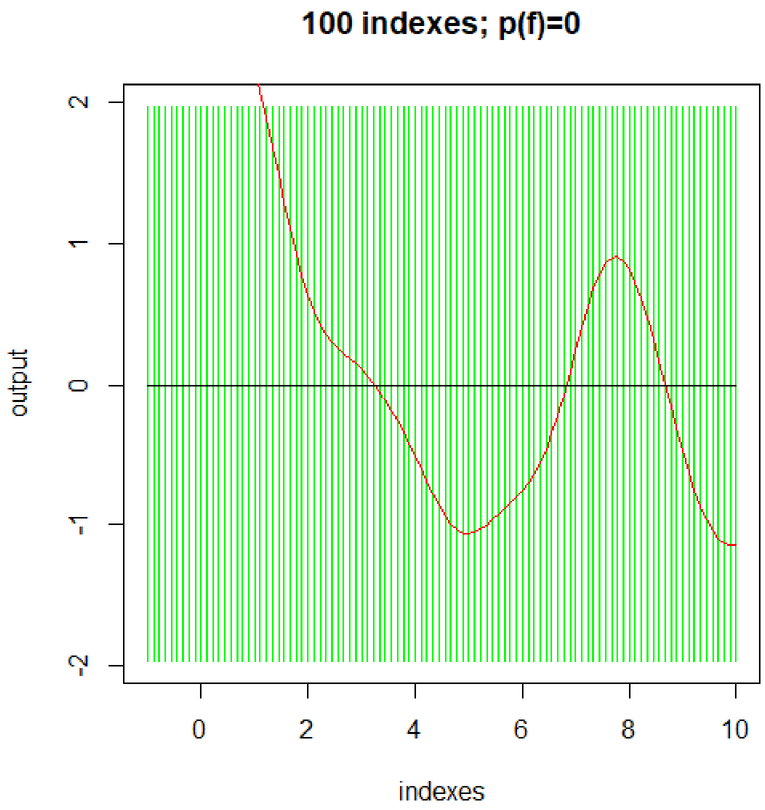
Proces gaussian



Proces gaussian



Proces gaussian



Proces gaussian

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

$$m(x) = E[f(x)]$$
$$k(x, x') = E[(f(x) - m(x))(f(x') - m(x'))]$$

$$x_1, \ldots, x_m \in \mathcal{X}$$

$$\begin{bmatrix} f(x_1) \\ \vdots \\ f(x_m) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(x_1) \\ \vdots \\ m(x_m) \end{bmatrix}, \begin{bmatrix} k(x_1, x_1) & \cdots & k(x_1, x_m) \\ \vdots & \ddots & \vdots \\ k(x_m, x_1) & \cdots & k(x_m, x_m) \end{bmatrix} \right)$$

- m – funcție
- k – funcție nucleu (kernel)

Regresia prin procese gaussiene

$$y^{(i)} = f(x^{(i)}) + \varepsilon^{(i)}, \qquad i = 1, \ldots, m$$

$$f(\cdot) \sim \mathcal{GP}(0, k(\cdot, \cdot))$$

Regresia prin procese gaussiene

Antrenament

– În același timp cu testarea!!!

Regresia prin procese gaussiene

Testare

$$\begin{bmatrix} \vec{f} \\ \vec{f}_* \end{bmatrix} \Bigg| X, X_* \sim \mathcal{N}\left(\vec{0}, \begin{bmatrix} K(X,X) & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right)$$

$$\begin{bmatrix} \vec{\varepsilon} \\ \vec{\varepsilon}_* \end{bmatrix} \sim \mathcal{N}\left(\vec{0}, \begin{bmatrix} \sigma^2 I & \vec{0} \\ \vec{0}^T & \sigma^2 I \end{bmatrix}\right)$$

Regresia prin procese gaussiene

Testare

$$\begin{bmatrix} \vec{y} \\ \vec{y}_* \end{bmatrix} \Bigg| X, X_* = \begin{bmatrix} \vec{f} \\ \vec{f}_* \end{bmatrix} + \begin{bmatrix} \vec{\varepsilon} \\ \vec{\varepsilon}_* \end{bmatrix} \sim \mathcal{N}\left(\vec{0}, \begin{bmatrix} K(X,X) + \sigma^2 I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) + \sigma^2 I \end{bmatrix}\right)$$

$$\vec{y_*} \mid \vec{y}, X, X_* \sim \mathcal{N}(\mu^*, \Sigma^*)$$

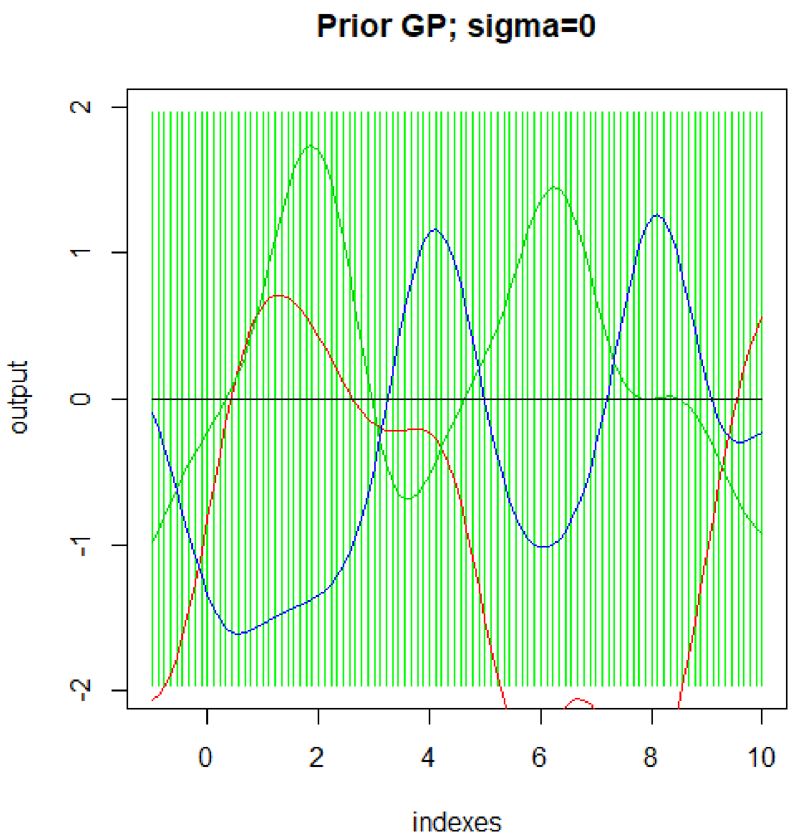
$$\mu^* = K(X_*, X)\left(K(X,X) + \sigma^2 I\right)^{-1} \vec{y}$$

$$\Sigma^* = K(X_*, X_*) + \sigma^2 I - K(X_*, X)\left(K(X,X) + \sigma^2 I\right)^{-1} K(X, X_*)$$

Regresia prin procese gaussiene

Vizualizare = **aplicarea definiției pentru multe puncte (indecși)**

Înainte de regresie/antrenament/testare



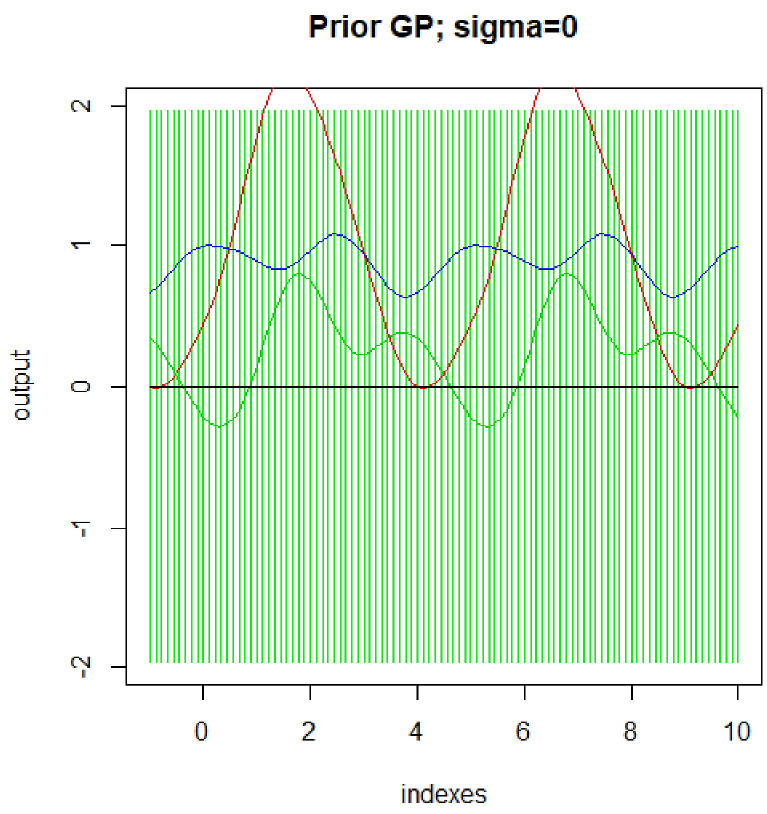
$$f(\cdot) \sim \mathcal{GP}(0, k(\cdot,\cdot))$$

$$k(x,y) = 1^2 \cdot e^{-\frac{\|x-y\|^2}{2\cdot 1^2}}$$

Regresia prin procese gaussiene

Vizualizare **= aplicarea definiției pentru multe puncte (indecși)**

Înainte de regresie/antrenament/testare



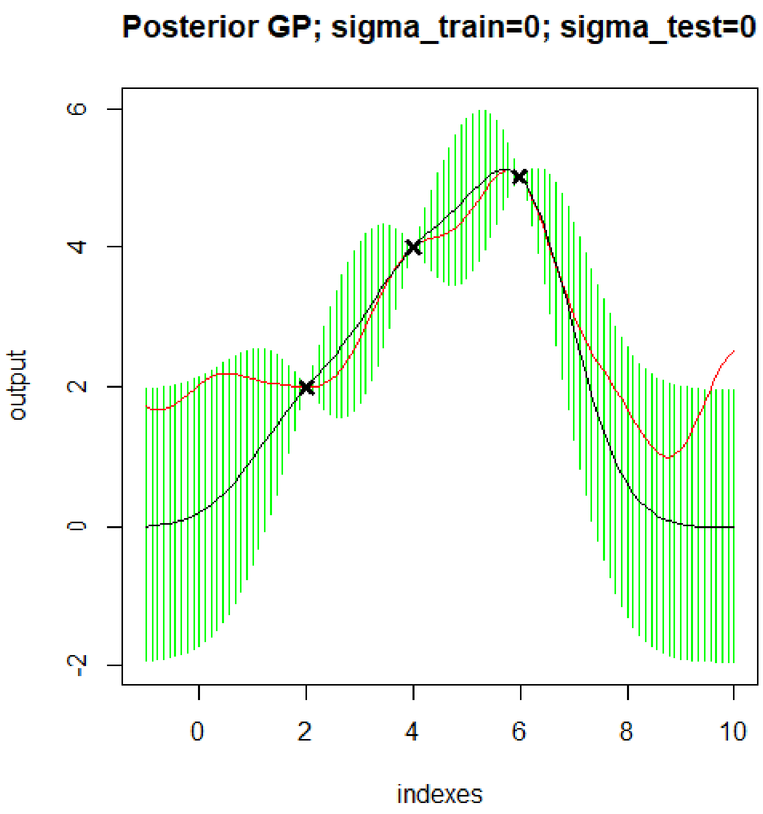
$$f(\cdot) \sim \mathcal{GP}(0, k(\cdot, \cdot))$$

$$k(x,y) = 1^2 \cdot e^{-\frac{2}{2^2}\sin^2(\pi\frac{(x-y)}{5})}$$

Regresia prin procese gaussiene

Vizualizare = testare pe multe puncte (indecși)

La regresie/antrenament/testare



$$\vec{y_*} \mid \vec{y}, X, X_* \sim \mathcal{N}(\mu^*, \Sigma^*)$$

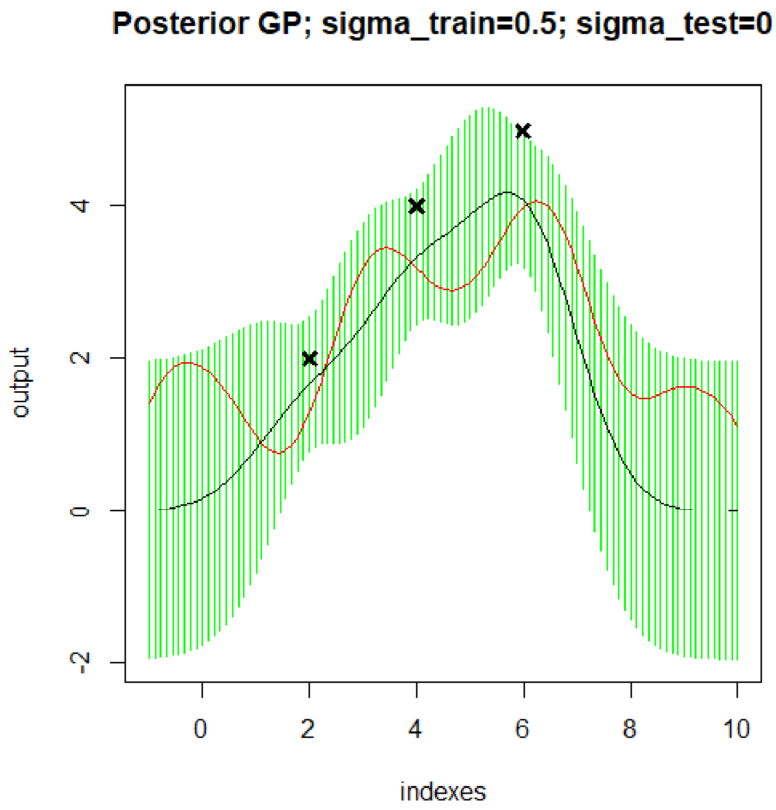
$$\mu^* = K(X_*, X)\left(K(X,X) + \sigma^2 I\right)^{-1}\vec{y}$$
$$\Sigma^* = K(X_*, X_*) + \sigma^2 I - K(X_*, X)\left(K(X,X) + \sigma^2 I\right)^{-1} K(X, X_*)$$

$$k(x,y) = 1^2 \cdot e^{-\frac{\|x-y\|^2}{2\cdot 1^2}}$$

Regresia prin procese gaussiene

Vizualizare = testare pe multe puncte (indecși)

La regresie/antrenament/testare



$$\vec{y_*} \mid \vec{y}, X, X_* \sim \mathcal{N}(\mu^*, \Sigma^*)$$

$$\mu^* = K(X_*, X)\left(K(X,X) + \sigma^2 I\right)^{-1} \vec{y}$$
$$\Sigma^* = K(X_*, X_*) + \sigma^2 I - K(X_*, X)\left(K(X,X) + \sigma^2 I\right)^{-1} K(X, X_*)$$

$$k(x,y) = 1^2 \cdot e^{-\frac{\|x-y\|^2}{2\cdot 1^2}}$$

++Rezultate de echivalență dpvd al outputului

Remember: kernelized bayesian linear regression

$$
\begin{aligned}
&y^{(i)} = \phi(x^{(i)}) \cdot w + \epsilon^{(i)} \\
&\epsilon^{(i)} \sim \mathcal{N}(0, \sigma^2) \\
&w \sim \mathcal{N}(0, \Sigma_p) \\
&A\alpha = f
\end{aligned}
$$

- Dacă **output =** $p(y_* | X_*, x, y)$

Dacă $k_{GP}(x, z) = \psi(x) \cdot \psi(z)$, unde $\psi(x) = \Sigma_p^{1/2} \phi(x)$:
GP regression $\equiv$ kernelized (fully) bayesian linear regression.

++Rezultate de echivalență dpvd al outputului

Remember: kernelized bayesian linear regression

$$
\begin{aligned}
&y^{(i)} = \phi(x^{(i)}) \cdot w + \epsilon^{(i)} \\
&\epsilon^{(i)} \sim \mathcal{N}(0, \sigma^2) \\
&w \sim \mathcal{N}(0, \Sigma_p) \\
&A\alpha = f
\end{aligned}
$$

- Dacă **output** = predicția

Dacă $\Sigma_p = \tau^2 I$ (cazul ridge regression):

$$
\begin{aligned}
\text{GP regression} &\equiv \text{kernelized (fully) bayesian linear regression} \\
&\equiv \text{kernelized \{classical=semi-bayesian=with MAP\}} \\
&\text{ridge regression}
\end{aligned}
$$

++Formula pentru media GP posterior

- **Regresia liniară/polinomială**
  - **Start**
    - interpolare polinominală: prin n puncte distincte trece un unic polinom de grad n-1
    - Notă: doar 1D
  - **Cum prevenim overfittingul?**
    - pentru n puncte vom considera un polinom de

    grad < n-1
    - interpolare în sensul celor mai mici pătrate

++Formula pentru media GP posterior

- **Regresia liniară/polinomială**

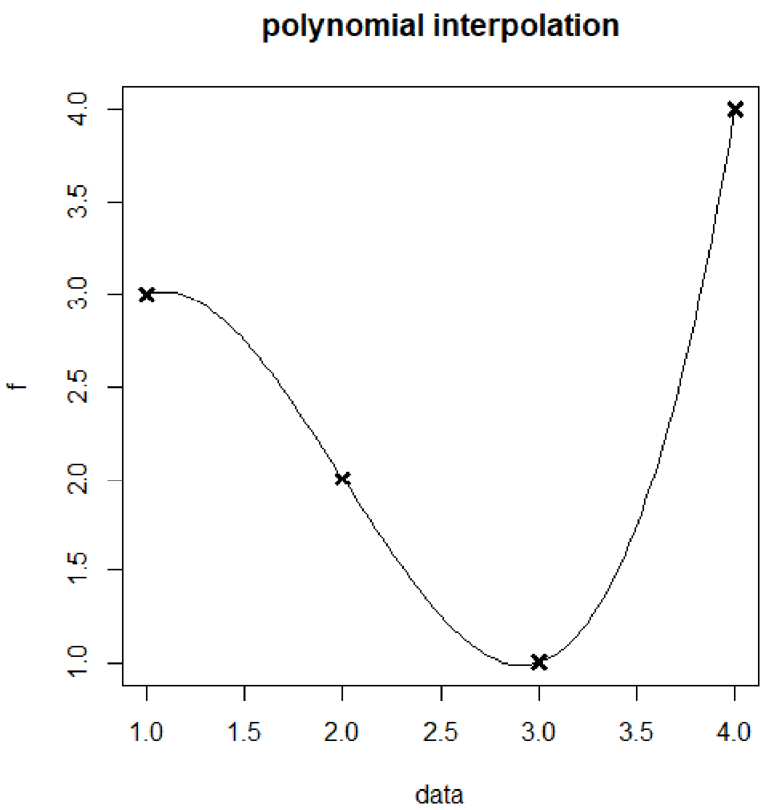
$$\phi(x) = \begin{bmatrix} 1 & x^1 & \ldots & x^{n-1} \end{bmatrix}$$

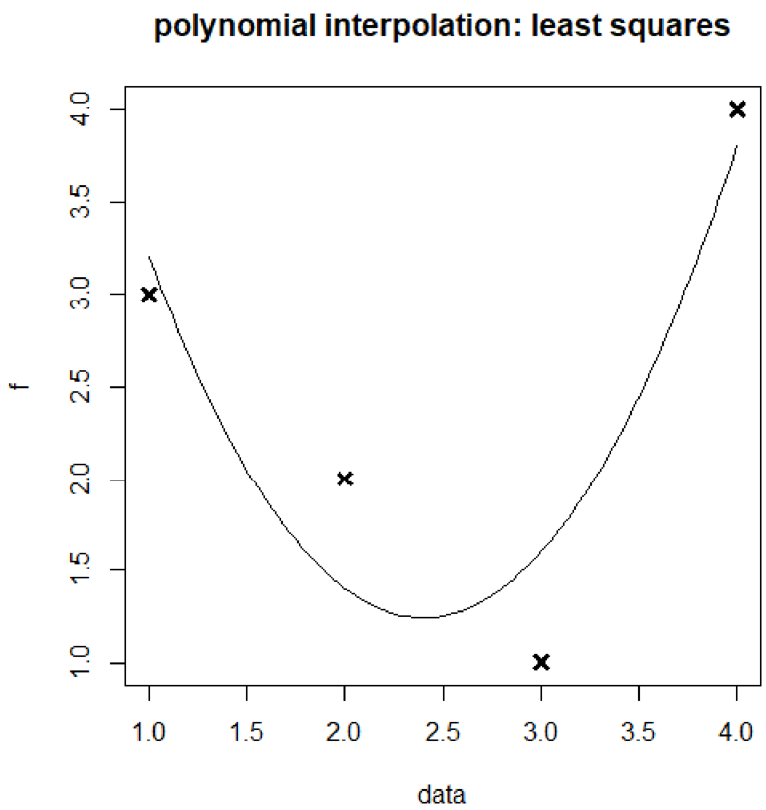
$$A\alpha = f$$

$$A = \begin{bmatrix} 1 & x_1^1 & \ldots & x_1^{n-1} \\ 1 & x_2^1 & \ldots & x_2^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n^1 & \ldots & x_n^{n-1} \end{bmatrix}$$

++Formula pentru media GP posterior

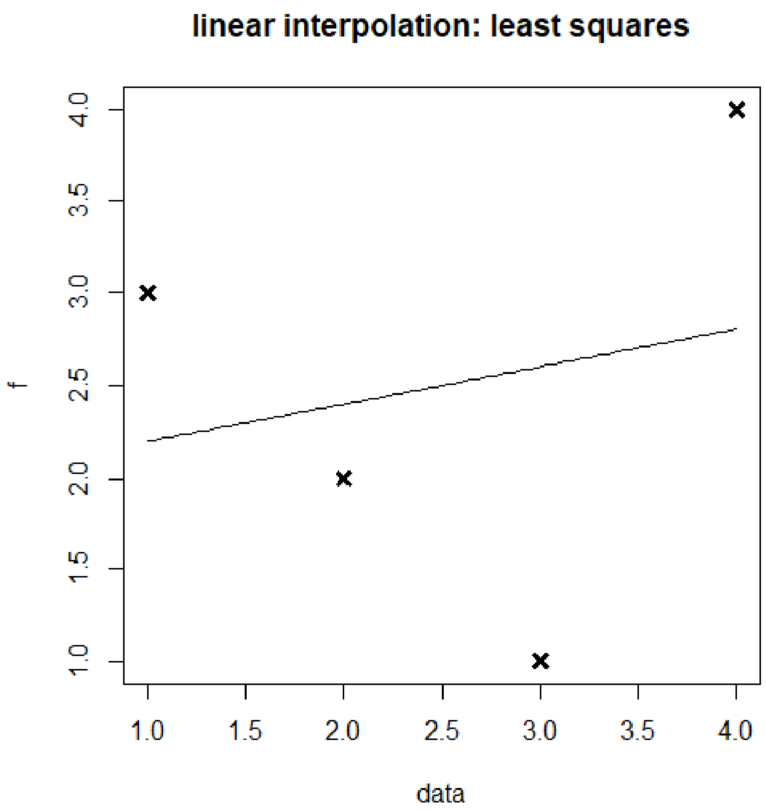
- **Regresia liniară/polinomială**





++Formula pentru media GP posterior

- **Regresia liniară/polinomială**



++Formula pentru media GP posterior

- **Regresia prin procese gaussiene**
  - **Start**
    - interpolare prin funcții nucleu
    - Notă: 1D sau mmD
  - **Cum prevenim overfittingul?**
    - Adăugăm zgomot pe diagonala principală a matricii A

++Formula pentru media GP posterior

- **Regresia prin procese gaussiene**

$$\phi(x) = \begin{bmatrix} k(x, x_1) & k(x, x_2) & \dots & k(x, x_n) \end{bmatrix}$$

$$A\alpha = f$$

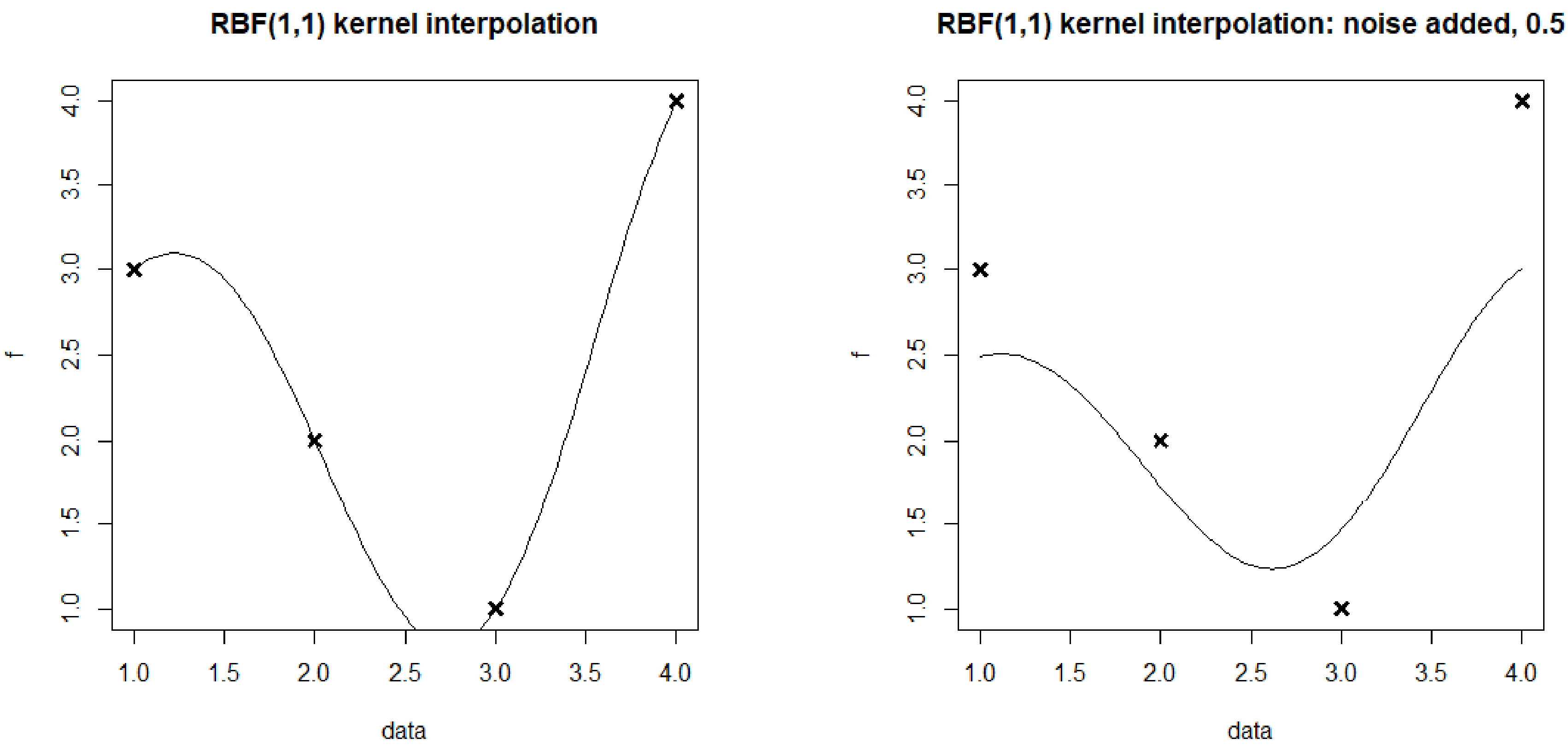
$$A = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \dots & k(x_1, x_n) \\ k(x_2, x_1) & k(x_2, x_2) & \dots & k(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n, x_1) & k(x_n, x_2) & \dots & k(x_n, x_n) \end{bmatrix}$$

Testare (media):

$$A_*\alpha = A_* A^{-1} f \stackrel{\text{not.}}{=} K(X_*, X)(K(X, X) + \sigma^2 I)^{-1} y$$

++Formula pentru media GP posterior

- **Regresia prin procese gaussiene**



Surse de inspirație:

Stanford, 2012 spring, Andrew Ng, HW9

sau OpenClassroom Stanford, Machine Learning, Andrew Ng

CMU, 2008 fall, Eric Xing, HW4, pr. 2.2

1. Image segmentation, compression

RGB

K-means vs EM/GMM

$$Z_i | h \sim \text{Categorical}(h.\pi)$$
$$X_i | Z_i = j, h \sim \text{NumeDistribuție}(h_j)$$

Date observabile și latente

Red	Green	Blue	Generat de gaussiana...
123	144	255	1
102	154	144	2
10	1	102	1
…	…	…	…

Algoritmul EM particular

EM pentru mixtura de distribuții gaussiene multivariate

Pasul E:

$$\gamma_{ij}^{(t+1)} \stackrel{\text{not.}}{=} \frac{\pi_j^{(t)} \mathcal{N}(x_i; \mu_j^{(t)}, \Sigma_j^{(t)})}{\sum_{l=1}^{k} \pi_l^{(t)} \mathcal{N}(x_i; \mu_l^{(t)}, \Sigma_l^{(t)})}$$

Pasul M:

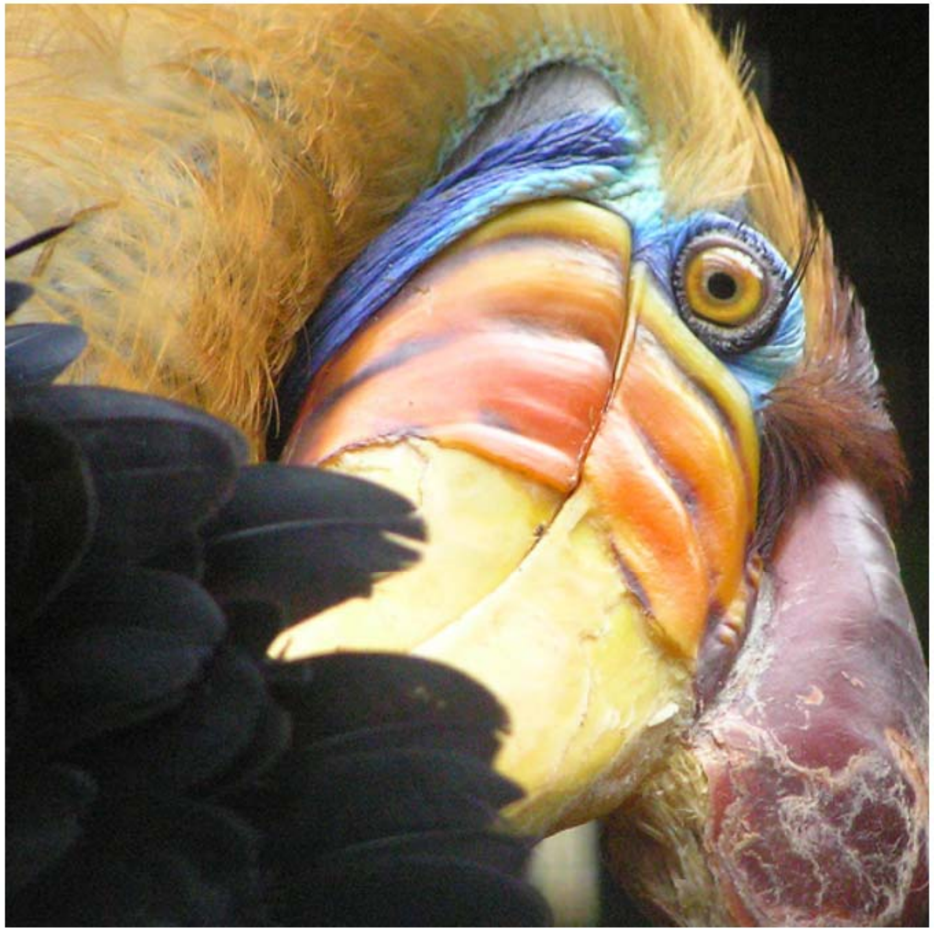
$$\pi_j^{(t+1)} = \frac{\sum_{i=1}^{n} \gamma_{ij}^{(t+1)}}{n}$$

$$\mu_j^{(t+1)} = \frac{\sum_{i=1}^{n} \gamma_{ij}^{(t+1)} x_i}{\sum_{i=1}^{n} \gamma_{ij}^{(t+1)}}$$

$$\Sigma_j^{(t+1)} = \frac{\sum_{i=1}^{n} \gamma_{ij}^{(t+1)} (x_i - \mu_j)(x_i - \mu_j)^T}{\sum_{i=1}^{n} \gamma_{ij}^{(t+1)}}$$

Rezultate

Originalele



Rezultate: k-means != EM/GMM

k=2, first run, k-means

k=2, first run, EM/GMM

Rezultate: k-means != EM/GMM

k=4, first run, k-means

k=4, first run, EM/GMM

Rezultate: k-means != EM/GMM

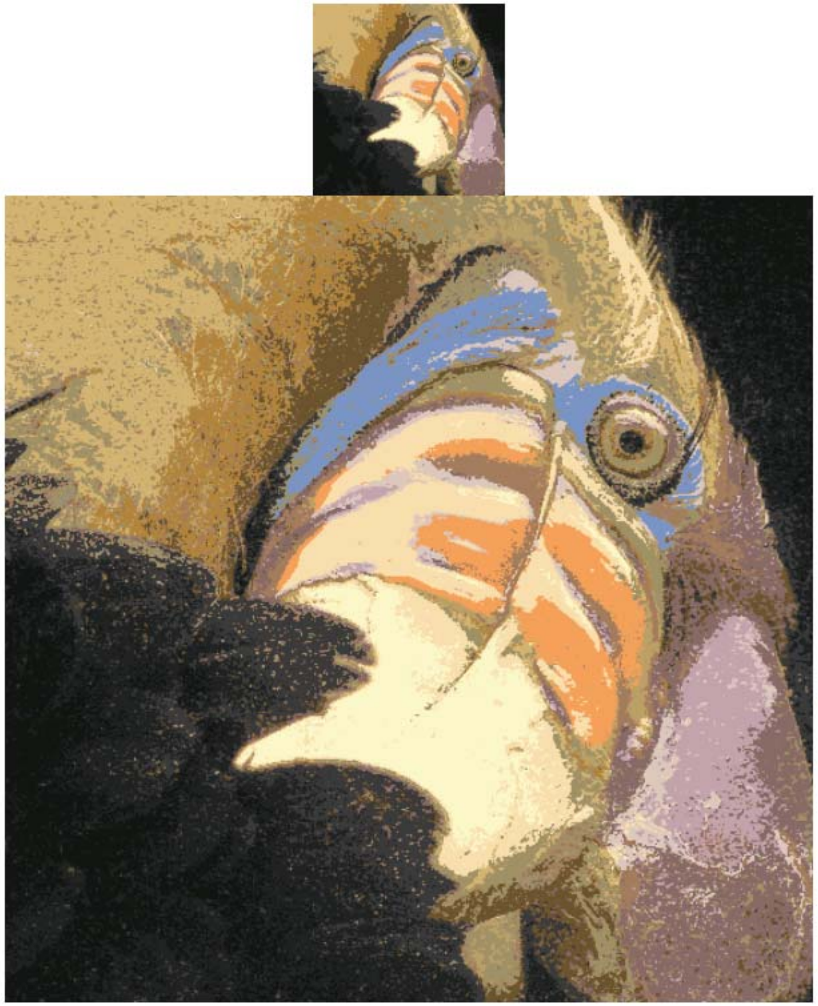
k=8, first run, k-means

k=8, first run, EM/GMM

Rezultate: k-means != EM/GMM

k=16, first run, k-means

k=16, first run, EM/GMM



Rezultate: k-means == EM/GMM

k=2, second run, k-means

k=2, second run, EM/GMM, sigma = 6

k=2, second run, EM/GMM, sigma = 50

Rezultate: k-means == EM/GMM

k=4, second run, k-means

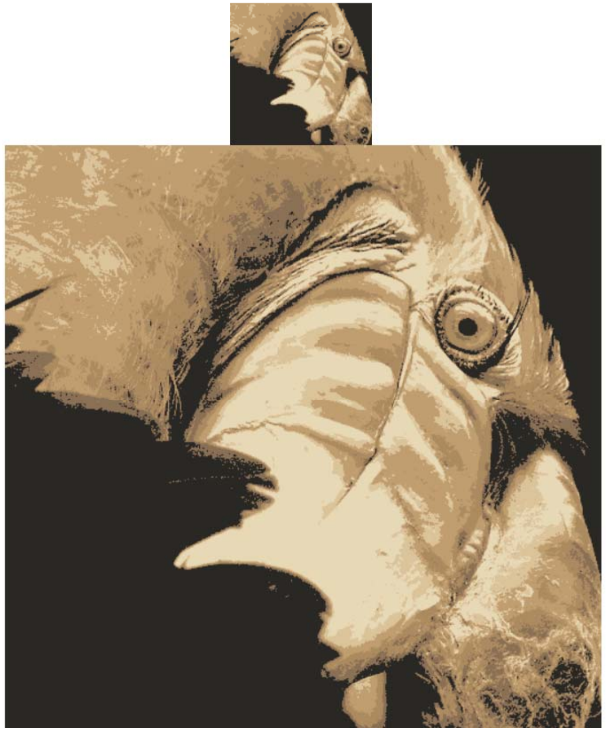
k=4, second run, EM/GMM, sigma = 6

k=4, second run, EM/GMM, sigma = 50

Rezultate: k-means == EM/GMM

k=8, second run, k-means

k=8, second run, EM/GMM, sigma = 50



k=8, second run, EM/GMM, sigma = 6

Rezultate: k-means == EM/GMM

k=16, second run, k-means

k=16, second run, EM/GMM, sigma = 6

k=16, second run, EM/GMM, sigma = 50

Pe lângă rezultate,
Ce am mai aflat nou?

- Ce să faci când imaginea este mare
- *Testarea* EM pe un nou set de date

Sursa de inspirație:

Gamma Mixture Classifier for Plaque. Detection in Intravascular Ultrasonic Images.
Gonzalo Vegas-sánchez-Ferrero

1. Image segmentation, compression

Gray

EM: GMM, GammaMM, RayleighMM, NakagamiMM

$$Z_i | h \sim \text{Categorical}(h.\pi)$$
$$X_i | Z_i = j, h \sim \text{NumeDistribuție}(h_j)$$

Date observabile și latente

Gray	Generat de distribuția...
255	1
144	2
102	1
...	...

Algoritmul EM particular (1)

EM pentru mixtura de distribuții gaussiene univariate

Pasul E:

$$\gamma_{ij}^{(t+1)} \stackrel{\text{not.}}{=} \frac{\pi_j^{(t)} \mathcal{N}(x_i; \mu_j^{(t)}, (\sigma_j^2)^{(t)})}{\sum_{l=1}^{k} \pi_l^{(t)} \mathcal{N}(x_i; \mu_l^{(t)}, (\sigma_l^2)^{(t)})}$$

Pasul M:

$$\pi_j^{(t+1)} = \frac{\sum_{i=1}^{n} \gamma_{ij}^{(t+1)}}{n}$$

$$\mu_j^{(t+1)} = \frac{\sum_{i=1}^{n} \gamma_{ij}^{(t+1)} x_i}{\sum_{i=1}^{n} \gamma_{ij}^{(t+1)}}$$

$$(\sigma_j^2)^{(t+1)} = \frac{\sum_{i=1}^{n} \gamma_{ij}^{(t+1)} (x_i - \mu_j^{(t+1)})^2}{\sum_{i=1}^{n} \gamma_{ij}^{(t+1)}}$$

Algoritmul EM particular (2)

EM pentru mixtura de distribuții Gamma (univariate)
Pasul E:

$$\gamma_{ij}^{(t+1)} \stackrel{\text{not.}}{=} \frac{\pi_j^{(t)} \mathrm{Gamma}(x_i; r_j^{(t)}, \beta_j^{(t)})}{\sum_{l=1}^{k} \pi_l^{(t)} \mathrm{Gamma}(x_i; r_l^{(t)}, \beta_l^{(t)})}$$

Pasul M:

$$\pi_j^{(t+1)} = \frac{\sum_{i=1}^{n} \gamma_{ij}^{(t+1)}}{n}$$

$$\boxed{\mathrm{Gamma}(x; r, \beta) = \frac{1}{\beta^r \Gamma(r)} x^{r-1} e^{-\frac{x}{\beta}}}$$

$$\ln(r_j^{(t+1)}) - \frac{\Gamma'(r_j^{(t+1)})}{\Gamma(r_j^{(t+1)})} = \ln\left(\frac{\sum_{i=1}^{n} \gamma_{ij}^{(t+1)} x_i}{\sum_{i=1}^{n} \gamma_{ij}^{(t+1)}}\right) - \frac{\sum_{i=1}^{n} \gamma_{ij}^{(t+1)} \ln(x_i)}{\sum_{i=1}^{n} \gamma_{ij}^{(t+1)}}$$

$f(x) = \ln(x) - \frac{\Gamma'(x)}{\Gamma(x)} - K,\ K > 0$
$f(\frac{1}{2K}) f(\frac{1}{K}) < 0$
Pentru a afla x cu $f(x) = 0$ vom folosi metoda bisecției pe $[\frac{1}{2K}, \frac{1}{K}]$.

$$\beta_j^{(t+1)} = \frac{\sum_{i=1}^{n} \gamma_{ij}^{(t+1)} x_i}{r_j^{(t+1)} \sum_{i=1}^{n} \gamma_{ij}^{(t+1)}}$$

Algoritmul EM particular (3)

EM pentru mixtura de distribuții Rayleigh (univariate)

Pasul E:

$$\gamma_{ij}^{(t+1)} \stackrel{\text{not.}}{=} \frac{\pi_j^{(t)} \text{Rayleigh}(x_i; (\sigma_j^2)^{(t)})}{\sum_{l=1}^{k} \pi_l^{(t)} \text{Rayleigh}(x_i; (\sigma_l^2)^{(t)})}$$

Pasul M:

$$\pi_j^{(t+1)} = \frac{\sum_{i=1}^{n} \gamma_{ij}^{(t+1)}}{n}$$

$$(\sigma_j^2)^{(t+1)} = \frac{\sum_{i=1}^{n} \gamma_{ij}^{(t+1)} x_i^2}{2 \sum_{i=1}^{n} \gamma_{ij}^{(t+1)}}$$

Algoritmul EM particular (4)

EM pentru mixtura de distribuții Nakagami (univariate)

Pasul E:

$$\gamma_{ij}^{(t+1)} \stackrel{\text{not.}}{=} \frac{\pi_j^{(t)} \text{Nakagami}(x_i; \Omega_j^{(t)}, m_j^{(t)})}{\sum_{l=1}^{k} \pi_l^{(t)} \text{Nakagami}(x_i; \Omega_l^{(t)}, m_l^{(t)})}$$

Pasul M:

$$\pi_j^{(t+1)} = \frac{\sum_{i=1}^{n} \gamma_{ij}^{(t+1)}}{n}$$

$$\Omega_j^{(t+1)} = \frac{\sum_{i=1}^{n} \gamma_{ij}^{(t+1)} x_i^2}{\sum_{i=1}^{n} \gamma_{ij}^{(t+1)}}$$

$$\ln(m_j^{(t+1)}) - \frac{\Gamma'(m_j^{(t+1)})}{\Gamma(m_j^{(t+1)})} = \ln\left(\frac{\sum_{i=1}^{n} \gamma_{ij}^{(t+1)} x_i^2}{\sum_{i=1}^{n} \gamma_{ij}^{(t+1)}}\right) - 2\frac{\sum_{i=1}^{n} \gamma_{ij}^{(t+1)} \ln(x_i)}{\sum_{i=1}^{n} \gamma_{ij}^{(t+1)}}$$

Rezultate

Originalele

Rezultate: k=2

Gaussian

Gamma

Rayleigh

Nakagami

Rezultate: k=4

Gaussian

Gamma

Rayleigh

Nakagami

Rezultate: k=8

Gaussian

Gamma

Rayleigh

Nakagami

Rezultate: k=16

Gaussian

Gamma

Rayleigh

Nakagami

Pe lângă rezultate,
Ce am mai aflat nou?

- Cum construiești pentru o distribuție (alta decât gaussiană) un prototip/centroid
- La pasul M, actualizările pot să nu fie doar formule (*closed-form*) – metoda bisecției

Sursa de inspirație:

University of Chicago, Michael Eichler, Statistics 24600, Spring 2004, Homework 3, pr. 2

2. Missing data

BivariateBernoulli

Date observabile și latente

X	Y
0	1
0	1
1	0
1	1
0	NA
1	0
1	NA
1	NA
1	1
0	1
1	NA
…	…

$$(X, Y) \sim \mathrm{BivariateBernoulli}(p_{00}, p_{01}, p_{10}, p_{11})$$

Algoritmul EM particular

EM pentru date lipsă (distribuţia Bernoulli bivariată)

Pasul E:

$$n_{00}^{(t)} = \#\{0\_\} \cdot \frac{p_{00}^{(t)}}{p_{00}^{(t)} + p_{01}^{(t)}} + \#\{00\}$$

$$n_{01}^{(t)} = \#\{0\_\} \cdot \frac{p_{01}^{(t)}}{p_{00}^{(t)} + p_{01}^{(t)}} + \#\{01\}$$

$$n_{10}^{(t)} = \#\{1\_\} \cdot \frac{p_{10}^{(t)}}{p_{10}^{(t)} + p_{11}^{(t)}} + \#\{10\}$$

$$n_{11}^{(t)} = \#\{1\_\} \cdot \frac{p_{11}^{(t)}}{p_{10}^{(t)} + p_{11}^{(t)}} + \#\{11\}$$

Pasul M:

$$p_{00}^{(t+1)} = \frac{n_{00}^{(t)}}{n}$$

$$p_{01}^{(t+1)} = \frac{n_{01}^{(t)}}{n}$$

$$p_{10}^{(t+1)} = \frac{n_{10}^{(t)}}{n}$$

$$p_{11}^{(t+1)} = \frac{n_{11}^{(t)}}{n}$$

Rezultate

#{X=0,Y=0} = 54

#{X=0,Y=1} = 104

#{X=1,Y=0} = 4

#{X=1,Y=1} = 142

#{X=0,Y=NA} = 53

#{X=1,Y=NA} = 62

Valorile inițiale sunt:

$p_{00}^{(0)} = 0.1776316, p_{01}^{(0)} = 0.3421053, p_{10}^{(0)} = 0.01315789, p_{11}^{(0)} = 0.4671053.$

După 150 de iterații valorile parametrilor devin:

$p_{00}^{(150)} = 0.1721096, p_{01}^{(150)} = 0.3314703, p_{10}^{(150)} = 0.01360055, p_{11}^{(150)} = 0.4828195.$

Pe lângă rezultate, Ce am mai aflat nou?

- Variabilele/Datele latente pot fi și altceva (nu doar clusterul de care aparține) => EM nu doar pentru clusterizare
- Cum afli ce trebuie să calculezi la pasul E

Sursa de inspirație:

The EM Algorithm and Extensions (2008), Geoffrey J. McLachlan, T. Krishnan

pages 54-58

3. Image deblurring

Algoritmul Richardson-Lucy

Date observabile și latente

Gray	Numărul de fotoni emiși în pixelul i și înregistrați de detectorul 1	Numărul de fotoni emiși în pixelul i și înregistrați de detectorul 2	...	Numărul de fotoni emiși în pixelul i și înregistrați de detectorul k
20 = 10 + 1 + ... + 1	10	1	...	1
120 = 20 + 30 + ... + 0	20	30	...	0
1 = 1 + 0 + ... + 0	1	0	...	0
0 = 0 + 0 + ... + 0	0	0	...	0
...	...	...	...	...

Intuiție: La un moment dat, detectoarele sunt plasate dens la o anumită poziție (pixel) și numără.

Algoritmul EM particular

$X = A\mu$ și zgomot
$x_i = \sum_{j=1}^{k} A_{ij}\mu_j$ și zgomot
X = imaginea cu blur ca vector
μ = imaginea originală (ideală!) ca vector

A = o matrice constantă
= point spread function (pentru un sistem optic, gradul de blurare a unui punct de lumină)
= generat de un convolution kernel (de smoothing)
= P(un foton e numărat de detectorul j|fotonul a fost emis în pixelul j)

$Z_{ij}|\mu \sim \text{Poisson}(A_{ij}, \mu_j)$
$X_i = \sum_{j=1}^{k} Z_{ij}$

$$\text{Poisson}(x;\lambda) = \frac{1}{e^{\lambda}} \cdot \frac{\lambda^x}{x!}$$

Algoritmul EM particular

EM = Richardson-Lucy

Pasul E:

$$\gamma_{ij}^{(t+1)} = x_i \frac{A_{ij}\mu_j^{(t)}}{\sum_{l=1}^{k} A_{il}\mu_l^{(t)}}$$

Pasul M:

$$\mu_j^{(t+1)} = \frac{\sum_{i=1}^{n} \gamma_{ij}^{(t+1)}}{\sum_{i=1}^{n} A_{ij}}$$

Varianta comprimată, dacă $\sum_{i=1}^{n} A_{ij} = 1$:

$$\mu^{(t+1)} = \mu^{(t)} .* (A^T(x./(A\mu^{(t)})))$$

Operațiile .* și ./ sunt operațiile de înmulțire și împărțire a vectorilor element cu element.

Rezultate

Original

Rezultate

Input: blurred

Rezultate: 10, 20, 30, 40, 50, 60 it.

Rezultate: 70, 80, 90 it.

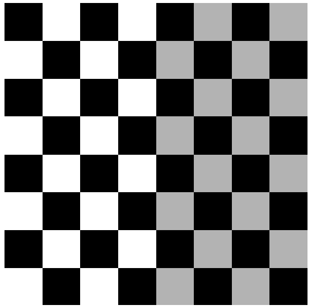
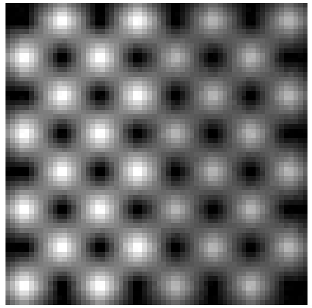
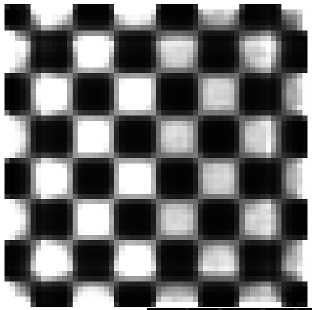
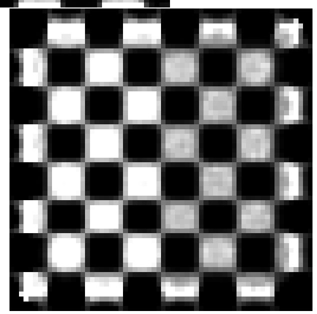
Rezultate

Input: blurred + noisy

Rezultate: 10, 20, 30, 40, 50, 60 it.

Rezultate: 70, 80, 90 it.

Rezultate

- Original 
- Input: blurred + noisy 
- Rezultatul meu 
- Rezultatul Matlab 

Pe lângă rezultate,
Ce am mai aflat nou?

- Variabilele/Datele latente pot fi și altceva (nu doar clusterul de care aparține) => EM nu doar pentru clusterizare
- Cum afli ce trebuie să calculezi la pasul E
- Convolution kernel => matricea A
- Eficiența foarte mare dată de implementarea vectorială (comprimată)

Sursa de inspirație:
MIT, 6867 ML, Fall 2006, Tommi Jaakkola, HW5, pr. 2

4. Regression. Gene expression

EM pentru mixturi de procese gaussiene

Date observabile și latente

Nivel de expresie la t_1	Nivel de expresie la t_2	…	Nivel de expresie la t_{30}	Generat de procesul gaussian…
-0.09900893	0.2818237	…	0.1033819	1
-0.00857957	0.1534053	…	0.08532405	2
-0.03709729	-0.00954268	…	0.01441636	1
…	…	…	…	…

t_1	t_2	…	t_{30}
0.0000000	0.2166616	…	6.2831853

Algoritmul EM particular

EM pentru mixtura de procese gaussiene

$$k(t,t') = \sigma_f^2 \exp\left(-\frac{(t-t')^2}{2\rho^2}\right) + \sigma_n^2 \delta(t,t')$$

$$K_j^{(t)} \overset{\text{not.}}{=} K(h_j^{(t)})$$

Pasul E:

$$\gamma_{ij}^{(t+1)} \overset{\text{not.}}{=} \frac{\pi_j^{(t)} \mathcal{N}(x_i; 0, K_j^{(t)}))}{\sum_{l=1}^{k} \pi_l^{(t)} \mathcal{N}(x_i; 0, K_l^{(t)}))}$$

Pasul M:

$$\pi_j^{(t+1)} = \frac{\sum_{i=1}^{n} \gamma_{ij}^{(t+1)}}{n}$$

Algoritmul EM particular

$$\theta_l \in \{(\sigma_f)_l, (\sigma_n)_l, \rho_l\}$$

$$\frac{\partial Q}{\partial \theta_l} = \frac{1}{2}\sum_{i=1}^{n}\left(\gamma_{il}^{(t+1)} x_i^T (K_l^{(t)})^{-1}\left(\frac{\partial K_l}{\theta_l}\right)^{(t)}(K_l^{(t)})^{-1}x_i\right) - \frac{\sum_{i=1}^{n}\gamma_{il}^{(t+1)}}{2}\mathrm{Tr}\left((K_l^{(t)})^{-1}\left(\frac{\partial K_l}{\theta_l}\right)^{(t)}\right)$$

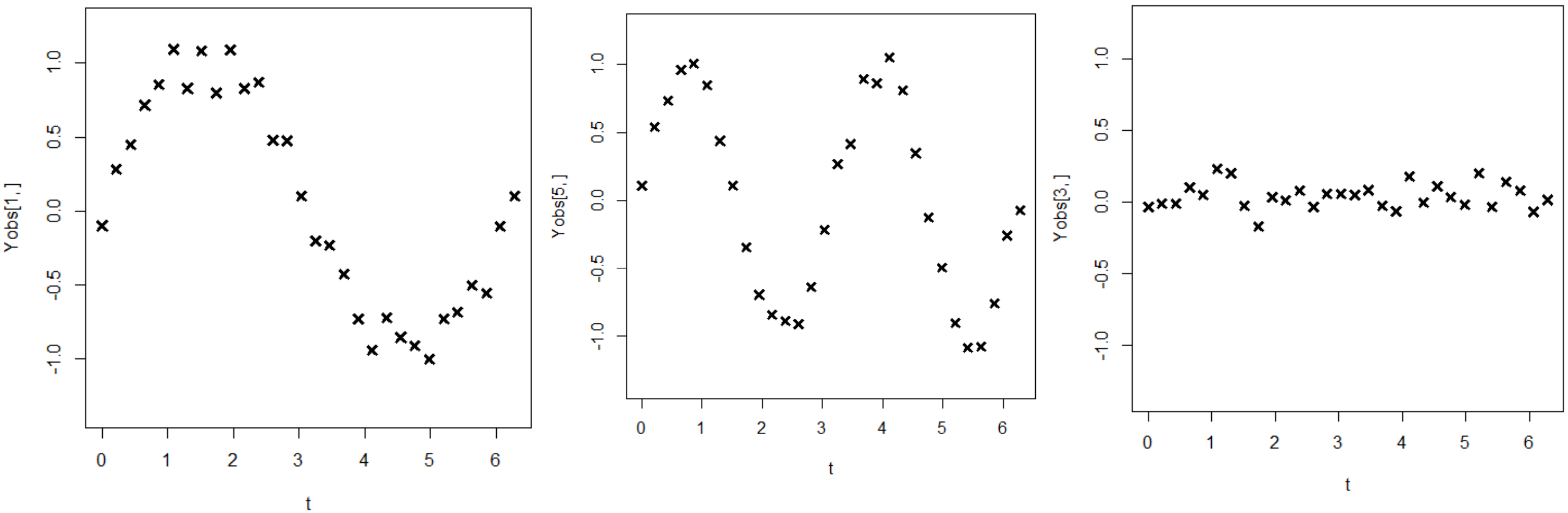
$$\frac{\partial k(x,y)}{\partial \sigma_f} = 2\sigma_f \exp\left(-\frac{(x-y)^2}{2\rho^2}\right)$$

$$\frac{\partial k(x,y)}{\partial \sigma_n} = 2\sigma_n \delta(x,y)$$

$$\frac{\partial k(x,y)}{\partial \rho} = \sigma_f^2 \exp\left(-\frac{(x-y)^2}{2\rho^2}\right)\frac{(x-y)^2}{\rho^3}$$

Pentru a afla $\theta_l^{(t+1)}$ vom folosi metoda gradientului (ascendent).

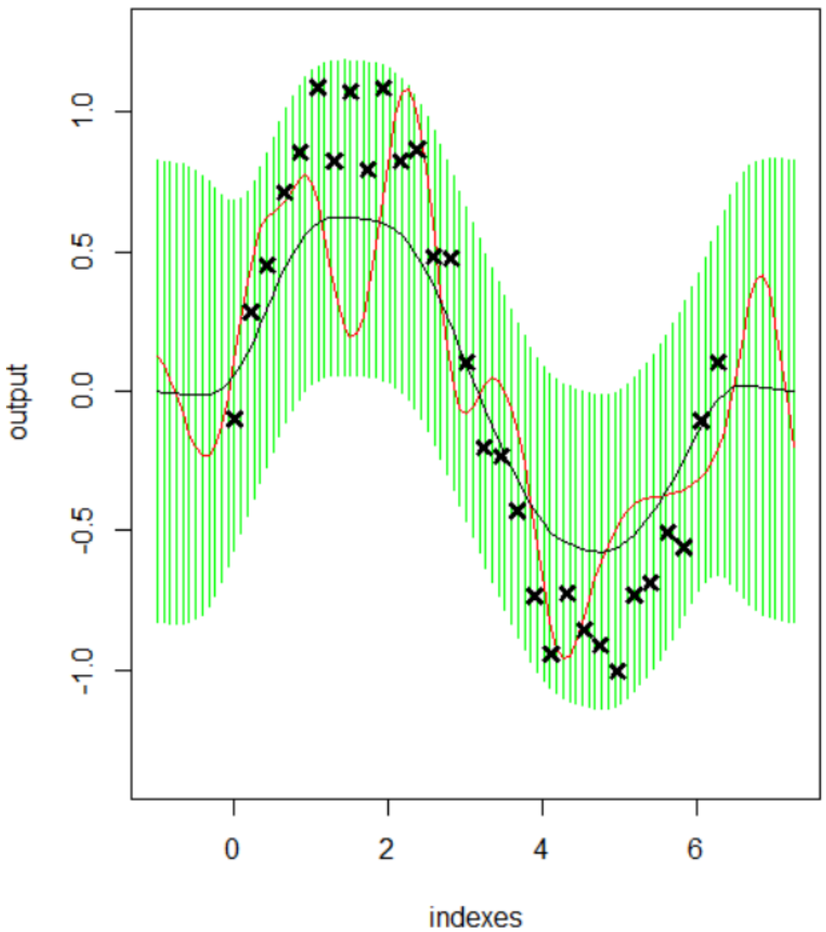
Rezultate



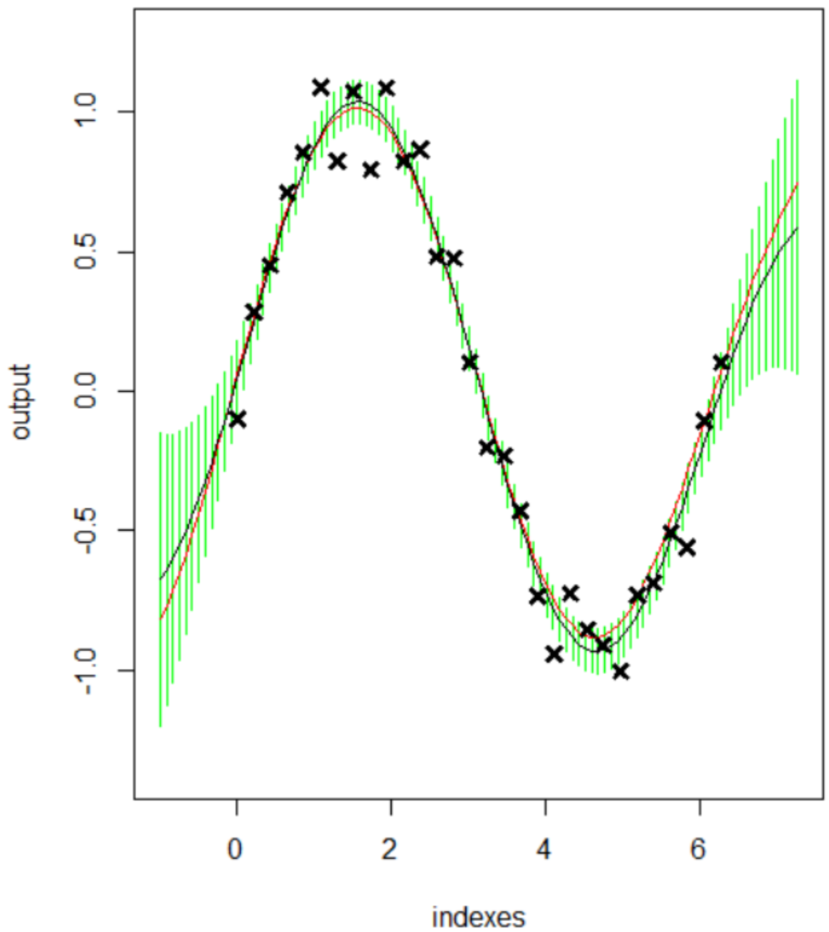
- => 3 tipuri de funcții

Rezultate

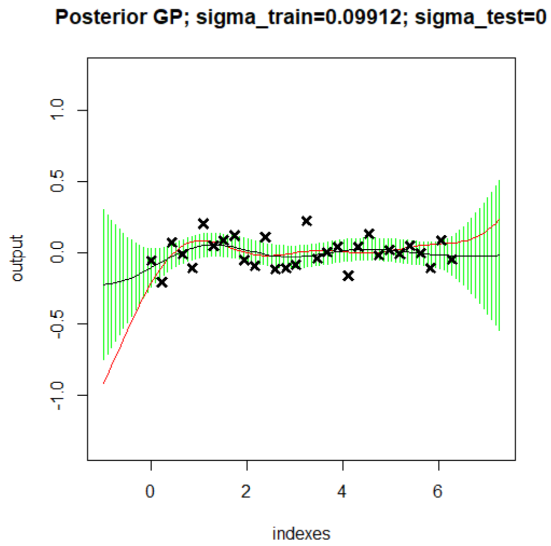
Random fit

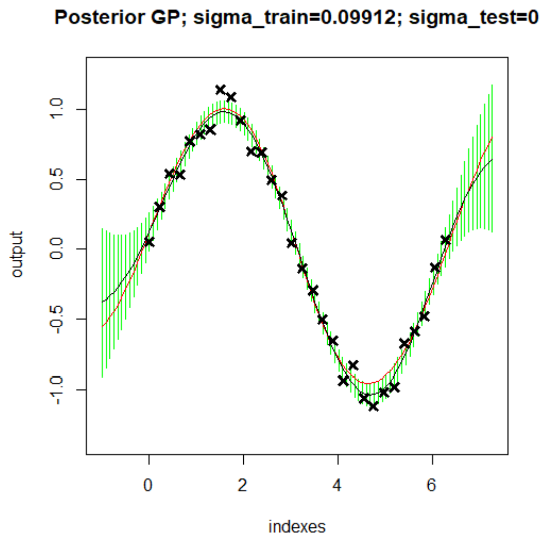


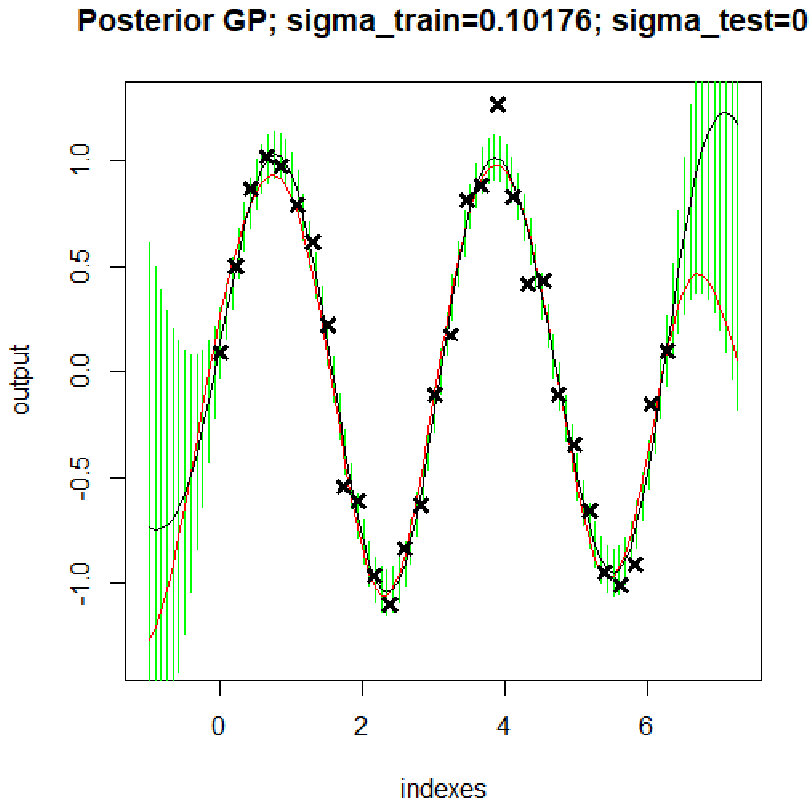
EM/GPMM fit



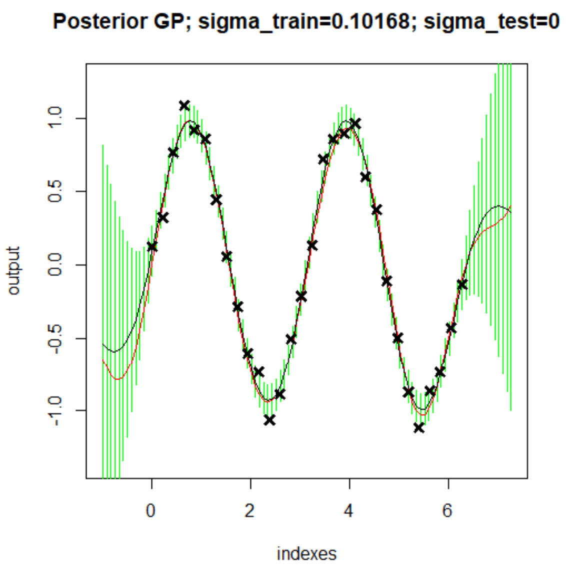
Rezultate: k=2

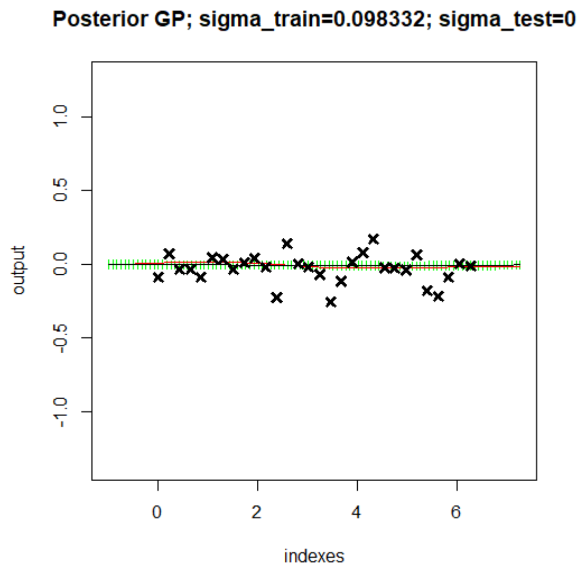


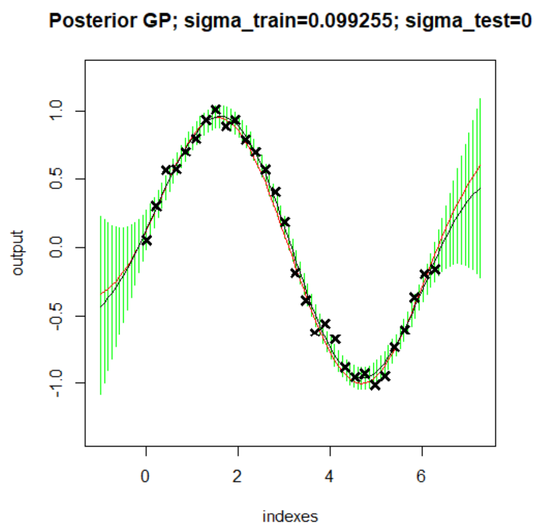




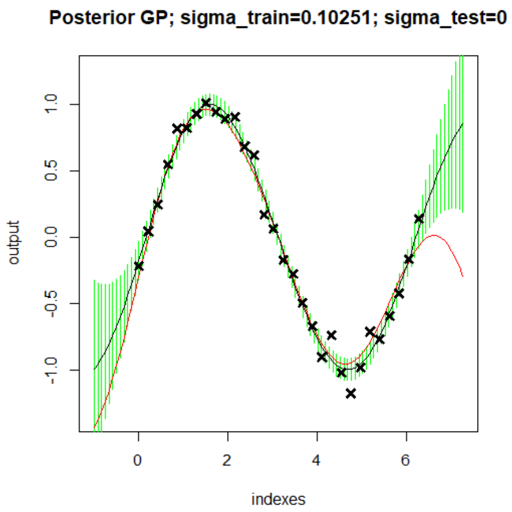
Rezultate: k=3

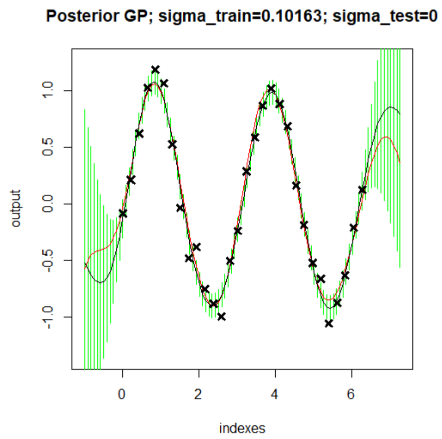


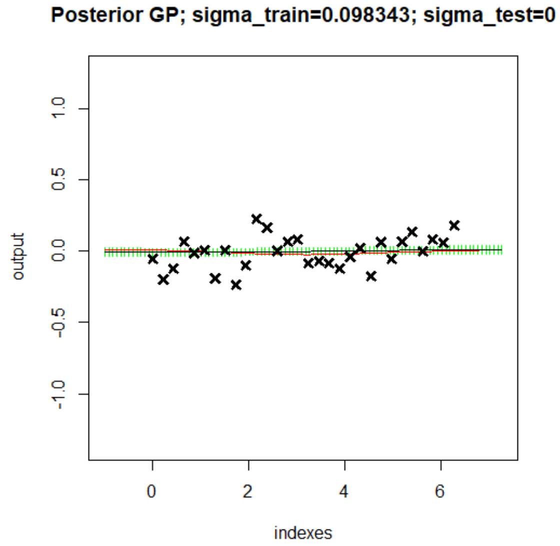


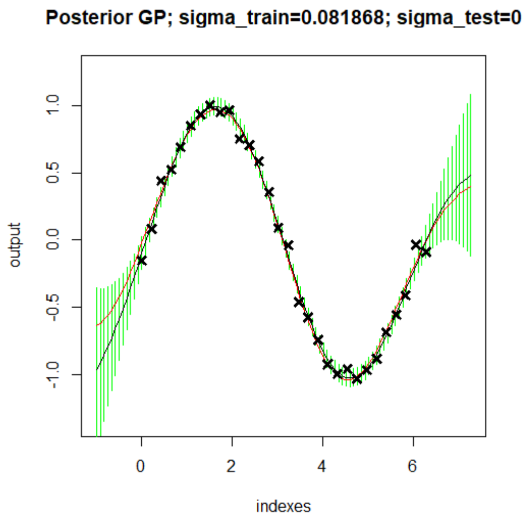


Rezultate: k=4



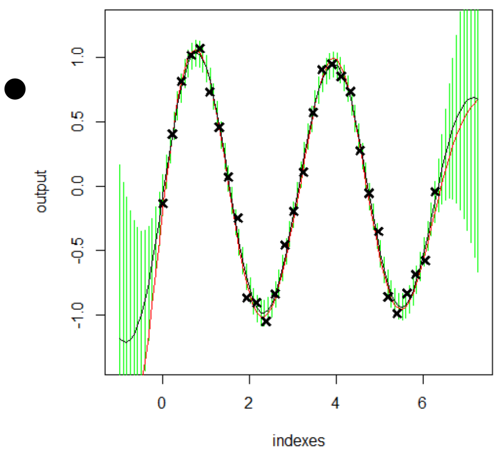




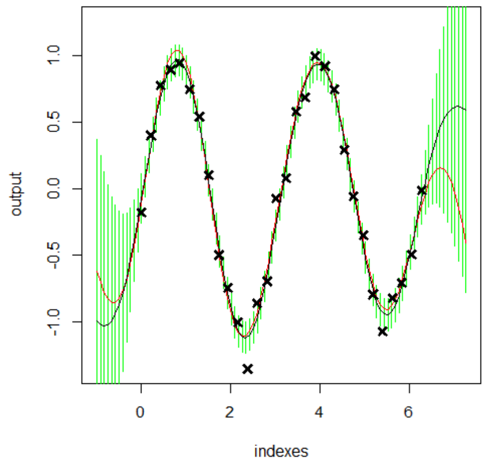


Rezultate: k=5

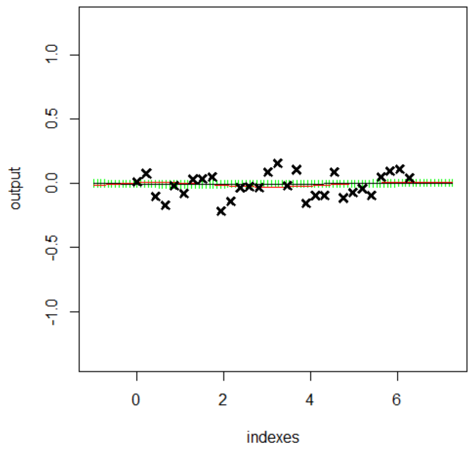
Posterior GP; sigma_train=0.09535; sigma_test=0



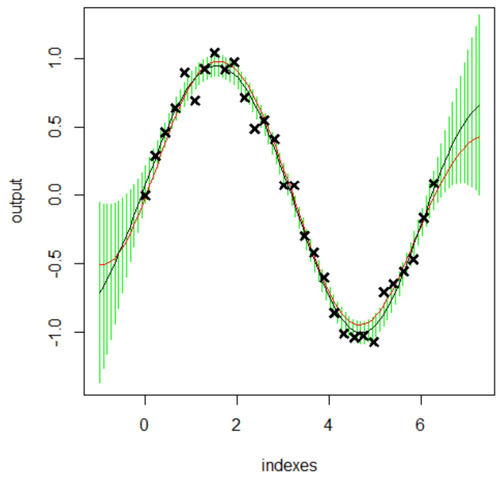
Posterior GP; sigma_train=0.1129; sigma_test=0



Posterior GP; sigma_train=0.098334; sigma_test=0



Posterior GP; sigma_train=0.099246; sigma_test=0



NULL cluster

Pe lângă rezultate,
Ce am mai aflat nou?

- Inițializarea contează
- Inițializarea W-ului
- Asemănări foarte mari între MLE pentru o distribuție (GP) și alg. EM pentru o mixtură de distribuții (GPs)
- Metoda gradientului poate fi îmbunătățită ca timp de rulare
- Variante ale metodei gradientului (k=4,5; momentum)
- La pasul M, actualizările pot să nu fie doar formule (*closed-form*) – metoda gradientului