Procesarea semnalelor

Reprezentări rare. Antrenarea dicționarelor.

Paul Irofti

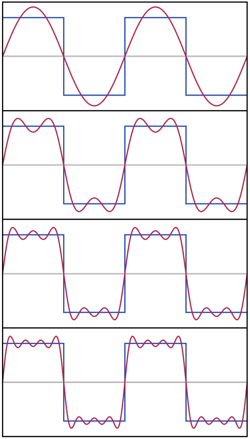
Universitatea din București

Facultatea de Matematică și Informatică

Departmentul de Informatică

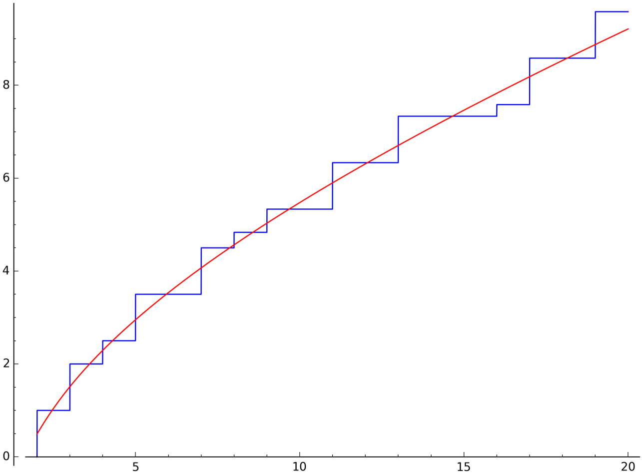
Email: paul.irofti@fmi.unibuc.ro

Aproximarea treptei cu sinusoide



https://en.wikipedia.org/wiki/Fourier_series

Aproximarea sinusoidei cu trepte



http://www.riemannhypothesis.info

Reprezentare rară – *sparse representation (SR)*

Dat un semnal $\boldsymbol{y} \in \mathbb{R}^m$, apar câteva întrebări

- care este o bază "bună" pentru a-l reprezenta?
- care este numărul minim de $s < m$ vectori necesari pentru reprezentare dintr-o bază dată $\boldsymbol{D}$?
- dar dacă pot alege eu baza cât este s?

Reprezentare rară – *sparse representation (SR)*

Dat un semnal $\boldsymbol{y} \in \mathbb{R}^m$, apar câteva întrebări

- care este o bază "bună" pentru a-l reprezenta?
- care este numărul minim de $s < m$ vectori necesari pentru reprezentare dintr-o bază dată $\boldsymbol{D}$?
- dar dacă pot alege eu baza cât este s?

Răspuns: Aleg baza ce conține pe $\boldsymbol{y}$ printre vectorii săi a.î. $s = 1$

Reprezentare rară – *sparse representation (SR)*

Dat un semnal $\boldsymbol{y} \in \mathbb{R}^m$, apar câteva întrebări

- care este o bază "bună" pentru a-l reprezenta?
- care este numărul minim de $s < m$ vectori necesari pentru reprezentare dintr-o bază dată $\boldsymbol{D}$?
- dar dacă pot alege eu baza cât este s?

Răspuns: Aleg baza ce conține pe $\boldsymbol{y}$ printre vectorii săi a.î. $s = 1$

Ce se întâmplă dacă vreau să reprezint o familie de $N > m$ semnale diferite?

Reprezentare rară – *sparse representation (SR)*

Dat un semnal $\boldsymbol{y} \in \mathbb{R}^m$, apar câteva întrebări

- care este o bază "bună" pentru a-l reprezenta?
- care este numărul minim de $s < m$ vectori necesari pentru reprezentare dintr-o bază dată $\boldsymbol{D}$?
- dar dacă pot alege eu baza cât este s?

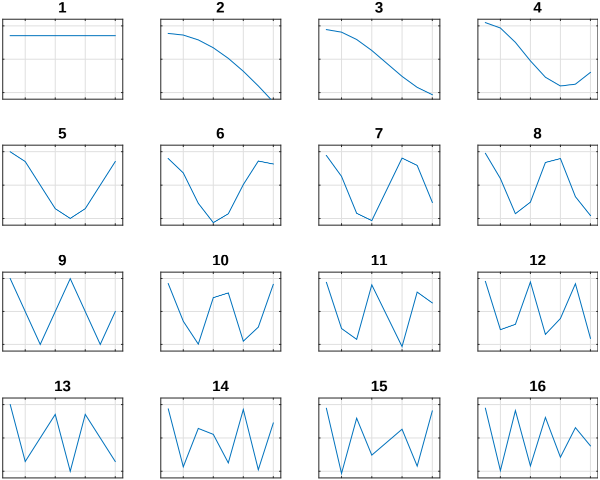
Răspuns: Aleg baza ce conține pe $\boldsymbol{y}$ printre vectorii săi a.î. $s = 1$

Ce se întâmplă dacă vreau să reprezint o familie de $N > m$ semnale diferite?

Răspuns: O soluție este să aleg o bază redundantă $\boldsymbol{D} \in \mathbb{R}^{m \times n}$, unde $m < n$. $\boldsymbol{D}$ se numește **dicționar** iar coloanele sale **atomi**.

Exemplu DCT cu $m = 8$ și $n = 16$

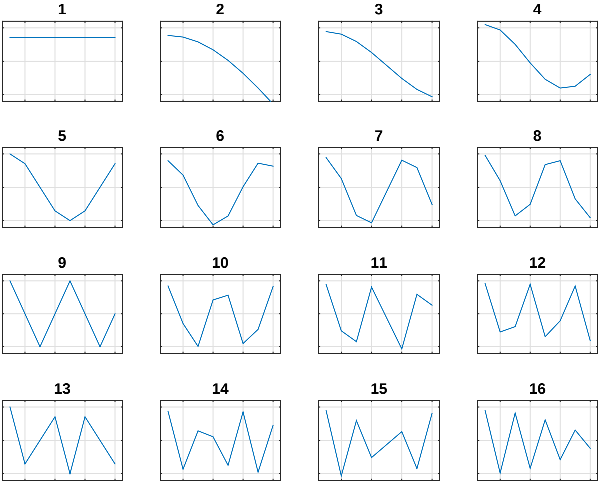
Coloanele 1 și 3 sunt baza canonică, iar 2 și 4 sunt redundante.



$$d_{ij} = \cos(\pi(i-1)(j-1)/n),\ i = 1:m,\ j = 1:n$$

Exemplu DCT cu $m = 8$ și $n = 16$

Cum se reprezintă $\boldsymbol{y} = 0.5\boldsymbol{d}_1 - 0.2\boldsymbol{d}_6$ canonic? Dar redundant?



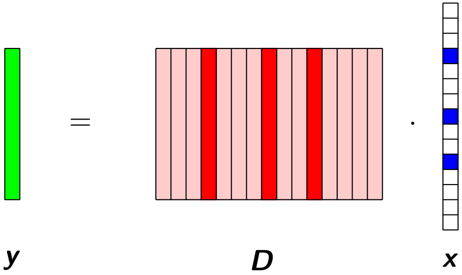
$$d_{ij} = \cos(\pi(i-1)(j-1)/n),\ i = 1:m,\ j = 1:n$$

Țel reprezentări rare

Scopul nostru devine astfel să reprezentăm un semnal $\boldsymbol{y}$ a.î.

$$\boldsymbol{y} = \boldsymbol{D}\boldsymbol{x} = \sum_{j=1}^{n} x_j \boldsymbol{d}_j = \sum_{j \in \mathcal{S}} x_j \boldsymbol{d}_j, \tag{1}$$

unde mulți x_j sunt zero, iar $\mathcal{S} = \{j \mid x_j \neq 0\}$ e suportul semnalului.



Definiție: $\boldsymbol{x}$ se numește reprezentarea rară a lui $\boldsymbol{y}$.

Posibilități de alegere

Fie $\boldsymbol{y} \in \mathbb{R}^m$ și $\boldsymbol{D} \in \mathbb{R}^{m \times n}$ date a.î.:

- știm că $\boldsymbol{y}$ există într-un subspațiu de dimensiune s
- acest subspațiu este mic în comparație cu $\mathbb{R}^m$
- deci reprezentarea rară $\boldsymbol{x}$ are avea un suport $|\mathcal{S}| = s$

Câte posibilități de alegere a unui set de s atomi din dicționar am?

Posibilități de alegere

Fie $\boldsymbol{y} \in \mathbb{R}^m$ și $\boldsymbol{D} \in \mathbb{R}^{m \times n}$ date a.î.:

- știm că $\boldsymbol{y}$ există într-un subspațiu de dimensiune s
- acest subspațiu este mic în comparație cu $\mathbb{R}^m$
- deci reprezentarea rară $\boldsymbol{x}$ are avea un suport $|\mathcal{S}| = s$

Câte posibilități de alegere a unui set de s atomi din dicționar am?

Răspuns: $\binom{n}{s} \approx n^s / s!$

Posibilități de alegere

Fie $\boldsymbol{y} \in \mathbb{R}^m$ și $\boldsymbol{D} \in \mathbb{R}^{m \times n}$ date a.î.:

- știm că $\boldsymbol{y}$ există într-un subspațiu de dimensiune s
- acest subspațiu este mic în comparație cu $\mathbb{R}^m$
- deci reprezentarea rară $\boldsymbol{x}$ are avea un suport $|\mathcal{S}| = s$

Câte posibilități de alegere a unui set de s atomi din dicționar am?

Răspuns: $\binom{n}{s} \approx n^s / s!$

Dar dintr-o bază?

Posibilități de alegere

Fie $\boldsymbol{y} \in \mathbb{R}^m$ și $\boldsymbol{D} \in \mathbb{R}^{m \times n}$ date a.î.:

- știm că $\boldsymbol{y}$ există într-un subspațiu de dimensiune s
- acest subspațiu este mic în comparație cu $\mathbb{R}^m$
- deci reprezentarea rară $\boldsymbol{x}$ are avea un suport $|\mathcal{S}| = s$

Câte posibilități de alegere a unui set de s atomi din dicționar am?

Răspuns: $\binom{n}{s} \approx n^s/s!$

Dar dintr-o bază?

Răspuns: $\binom{m}{s} \approx m^s/s!$

Posibilități de alegere

Fie $\boldsymbol{y} \in \mathbb{R}^m$ și $\boldsymbol{D} \in \mathbb{R}^{m \times n}$ date a.î.:

- știm că $\boldsymbol{y}$ există într-un subspațiu de dimensiune s
- acest subspațiu este mic în comparație cu $\mathbb{R}^m$
- deci reprezentarea rară $\boldsymbol{x}$ are avea un suport $|\mathcal{S}| = s$

Câte posibilități de alegere a unui set de s atomi din dicționar am?

Răspuns: $\binom{n}{s} \approx n^s/s!$

Dar dintr-o bază?

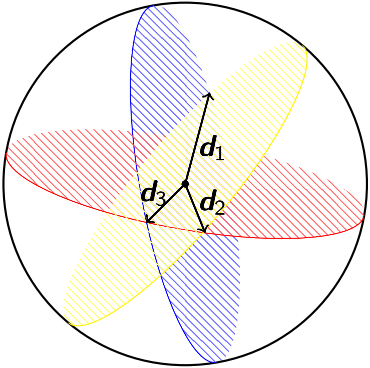
Răspuns: $\binom{m}{s} \approx m^s/s!$

Deci dicționarul este mai bogat cu $(n/m)^s$ posibili candidați. (Ce presupunere am făcut legată de atomii dicționarului?)

Exemplu: subspații $s = 2$, $m = n = 3$

Doar $\binom{3}{2} = 3$ subspații pot fi obținute.

- $\boldsymbol{d}_1$ și $\boldsymbol{d}_2$ generează subspațiul galben
- $\boldsymbol{d}_1$ și $\boldsymbol{d}_3$ generează subspațiul albastru
- $\boldsymbol{d}_2$ și $\boldsymbol{d}_3$ generează subspațiul roșu



Pentru $n = 6$ ar fi $\binom{6}{2} = 15$ subspații posibile!

Problema de optimizare

Formularea exactă

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \|\boldsymbol{x}\|_0 \\ \text{s.t.} \quad & \boldsymbol{y} = \boldsymbol{D}\boldsymbol{x} \end{aligned} \tag{2}$$

unde $\|.\|_0$ este pseudo-norma-zero care numără elemente nenule.

De ce este o $\|.\|_0$ o pseudo-normă? Ce proprietate nu îndeplinește?

Dacă ar exista o astfel de soluție, cât de ușor ar fi să o găsesc?

Problema de optimizare

O formulare mai relaxată a problemei precedente

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \|\boldsymbol{x}\|_0 \\ \text{s.t.} \quad & \|\boldsymbol{y} - \boldsymbol{D}\boldsymbol{x}\| \leq \varepsilon \end{aligned} \tag{3}$$

unde ε este o toleranță acceptată.

Putem căuta o soluție urmărind ca suportul să fie cât mai rar:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \|\boldsymbol{y} - \boldsymbol{D}\boldsymbol{x}\|^2 \\ \text{s.t.} \quad & \|\boldsymbol{x}\|_0 \leq s \end{aligned} \tag{4}$$

Aceste soluții se pretează foarte bine cazului în care semnalul măsurat este perturbat de un zgomot $\boldsymbol{v}$

$$\boldsymbol{y} = \boldsymbol{D}\boldsymbol{x} + \boldsymbol{v}. \tag{5}$$

care se pierde în urma aproximării

Algoritmi

Greedy

- construiesc suportul adăugând câte un atom la fiecare iterație
- rezolvă o sub-problemă restrânsă la pasul curent
- sunt foarte rapizi

Relaxare convexă

- înlocuiesc norma-0 cu norma-1
- problema se transformă într-o problemă de optimizare convexă
- norma-1 promovează soluțiile rare (dar nu la fel de rare ca norma-0)
- norma-1 poate fi mutată ca regularizare în obiectiv pentru a controla mai bine cât de rară va fi soluția

Orthogonal Matching Pursuit (OMP)

- cel mai popular algoritm (greedy)
- construiește iterativ suportul $\mathcal{S}$
- folosește reziduu curent

$$\boldsymbol{e} = \boldsymbol{y} - \sum_{j \in \mathcal{S}} x_j \boldsymbol{d}_j. \tag{6}$$

- pentru a alege atomul ce corelează cel mai mult cu el

$$|\boldsymbol{e}^T \boldsymbol{d}_k| = \max_{j \notin \mathcal{S}} |\boldsymbol{e}^T \boldsymbol{d}_j|. \tag{7}$$

- după care recalculează coeficienții din $\boldsymbol{x}$

$$\boldsymbol{x}_{\mathcal{S}} = (\boldsymbol{D}_{\mathcal{S}}^T \boldsymbol{D}_{\mathcal{S}})^{-1} \boldsymbol{D}_{\mathcal{S}}^T \boldsymbol{y}, \tag{8}$$

Algoritmul OMP

Algorithm 1: Orthogonal Matching Pursuit

Data: dictionary $\boldsymbol{D} \in \mathbb{R}^{m \times n}$
signal $\boldsymbol{y} \in \mathbb{R}^{m}$
sparsity level s
stopping error ε

Result: representation support $\mathcal{S}$, solution $\boldsymbol{x}$

```
Initialize S = ∅, e = y
while |S| < s and ||e|| > ε do
    Find new index: k = arg max_{j∉S} |e^T d_j|
    Build new support: S ← S ∪ {k}
    Compute new solution: x_S = (D_S^T D_S)^{-1} D_S^T y
    Compute new residual: e = y − D_S x_S
```

Initialize $\mathcal{S} = \emptyset$, $\boldsymbol{e} = \boldsymbol{y}$

while $|\mathcal{S}| < s \text{ and } \|\boldsymbol{e}\| > \varepsilon$ **do**

- Find new index: $k = \arg\max_{j \notin \mathcal{S}} |\boldsymbol{e}^T \boldsymbol{d}_j|$
- Build new support: $\mathcal{S} \leftarrow \mathcal{S} \cup \{k\}$
- Compute new solution: $\boldsymbol{x}_{\mathcal{S}} = (\boldsymbol{D}_{\mathcal{S}}^T \boldsymbol{D}_{\mathcal{S}})^{-1} \boldsymbol{D}_{\mathcal{S}}^T \boldsymbol{y}$
- Compute new residual: $\boldsymbol{e} = \boldsymbol{y} - \boldsymbol{D}_{\mathcal{S}} \boldsymbol{x}_{\mathcal{S}}$

Algoritmul OMP – complexitate

Algorithm 2: Orthogonal Matching Pursuit $O(ms(n+s^2))$

Data: dictionary $\boldsymbol{D} \in \mathbb{R}^{m \times n}$
signal $\boldsymbol{y} \in \mathbb{R}^m$
sparsity level s
stopping error ε

Result: representation support $\mathcal{S}$, solution $\boldsymbol{x}$

Initialize $\mathcal{S} = \emptyset$, $\boldsymbol{e} = \boldsymbol{y}$
while $|\mathcal{S}| < s \text{ and } \|\boldsymbol{e}\| > \varepsilon$ **do**
  Find new index: $k = \arg\max_{j \notin \mathcal{S}} |\boldsymbol{e}^T \boldsymbol{d}_j|$ $O(mn)$
  Build new support: $\mathcal{S} \leftarrow \mathcal{S} \cup \{k\}$
  Compute new solution: $\boldsymbol{x}_\mathcal{S} = (\boldsymbol{D}_\mathcal{S}^T \boldsymbol{D}_\mathcal{S})^{-1} \boldsymbol{D}_\mathcal{S}^T \boldsymbol{y}$ $O(ms^2)$
  Compute new residual: $\boldsymbol{e} = \boldsymbol{y} - \boldsymbol{D}_\mathcal{S} \boldsymbol{x}_\mathcal{S}$ $O(ms)$

Garanții

Un dicționar $\boldsymbol{D}$ satisface *restricted isometry property (RIP)* dacă pentru orice bloc de s coloane $\tilde{\boldsymbol{D}}$ constanta δ_s este cea mai mică a.î.

$$(1-\delta_s)\|\boldsymbol{x}\|^2 \leq \|\tilde{\boldsymbol{D}}\boldsymbol{x}\|^2 \leq (1+\delta_s)\|\boldsymbol{x}\|^2 \tag{9}$$

pentru orice vector $\boldsymbol{x}$ de dimensiune s.

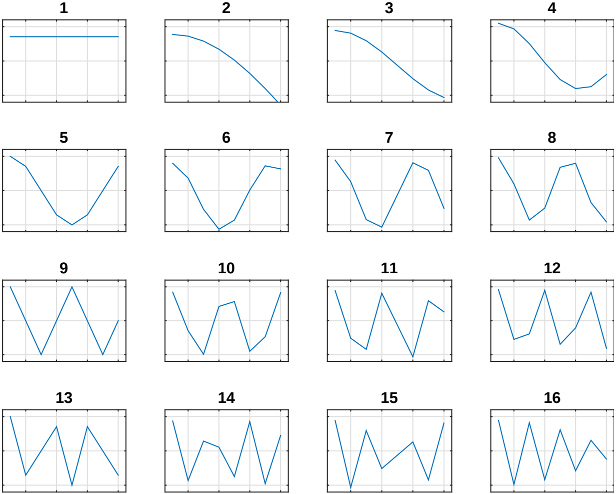
OMP găsește cea mai rară soluție dacă $\delta_{s+1} < \dfrac{\sqrt{4s+1}-1}{2s}$.

Alegerea dicționarului

- ortogonal: DFT, DCT, wavelet
- redundant: exemplul cu DCT de mai devreme
- aleator: cu proprietăți bune (ex. RIP)
- structurat: compunerea a mai multor transformări (DCT+wavelet)
- învățat: dicționarul este adaptat pentru un set de N semnale de antrenare dat (*dictionary learning (DL)*)

Exemplu învățare cu $m = 8$ și $n = 16$

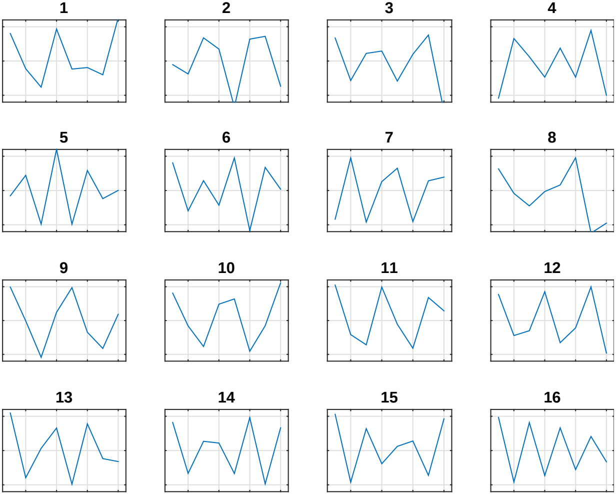
Inițializare cu DCT redundant



Semnale de antrenare $\eta_i = \cos(2\pi i\omega/m)$, $\omega \in [\frac{m}{4}, \frac{m}{2}]$

Exemplu învățare cu $m = 8$ și $n = 16$

După antrenare



Rezultat: reprezentări rare cu eroarea de două ori mai mică.

Problema de antrenare

Date semnalele de antrenare $\boldsymbol{Y} \in \mathbb{R}^{m \times N}$ și s, antrenarea dicționarului $\boldsymbol{D}$ presupune rezolvarea problemei de optimizare

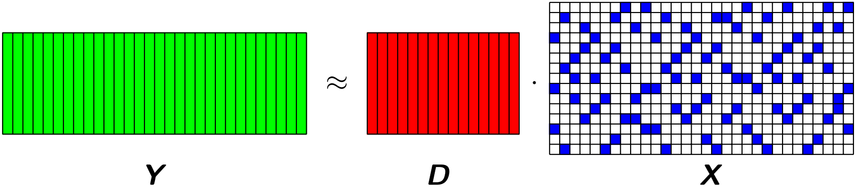
$$\begin{aligned} \min_{\boldsymbol{D},\boldsymbol{X}} \quad & \|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{X}\|_F^2 \\ \text{s.t.} \quad & \|\boldsymbol{x}_\ell\|_0 \leq s, \ \ell = 1 : N \\ & \|\boldsymbol{d}_j\| = 1, \ j = 1 : n \end{aligned} \tag{10}$$

unde variabilele sunt $\boldsymbol{D} \in \mathbb{R}^{m \times n}$ și $\boldsymbol{X} \in \mathbb{R}^{n \times N}$.

De ce e nevoie de normalizarea atomilor?

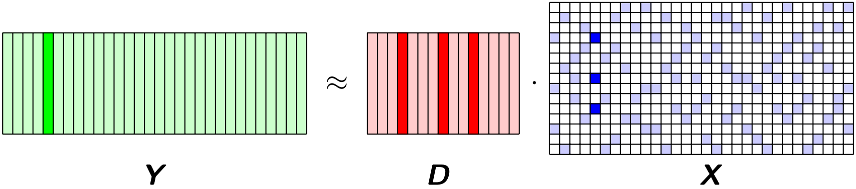
Exemplu: Problema de antrenare

Aproximarea $\boldsymbol{Y} \approx \boldsymbol{DX}$ trebuie să fie cât mai bună.



Exemplu: Problema de antrenare

Contribuția unui singur semnal



Calitatea aproximării

Eroarea de reprezentare $\boldsymbol{E}$ este

$$\boldsymbol{E} = \boldsymbol{Y} - \boldsymbol{D}\boldsymbol{X} \tag{11}$$

Iar obiectivul (10) poate fi rescris drept

$$\|\boldsymbol{E}\|_F = \sqrt{\sum_{i=1}^{m} \sum_{\ell=1}^{N} e_{i\ell}^2}. \tag{12}$$

sau partiționat pe semnale

$$\|\boldsymbol{E}\|_F^2 = \sum_{\ell=1}^{N} \|\boldsymbol{e}_\ell\|^2 = \sum_{\ell=1}^{N} \|\boldsymbol{y}_\ell - \boldsymbol{D}\boldsymbol{x}_\ell\|^2, \tag{13}$$

Subprobleme

Având în vedere că (10) este o problemă biliniară, în practică problema este împărțită în două.

Problema de reprezentare (sau de codare rară)

$$\begin{aligned} \min_{\boldsymbol{X}} \quad & \|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{X}\|_F^2 \\ \text{s.t.} \quad & \|\boldsymbol{x}_\ell\|_0 \leq s, \ \ell = 1 : N \end{aligned} \tag{14}$$

și problema actualizării dicționarului

$$\begin{aligned} \min_{\boldsymbol{D},(\boldsymbol{X})} \quad & \|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{X}\|_F^2 \\ \text{s.t.} \quad & \boldsymbol{X}_{\Omega^c} = 0 \\ & \|\boldsymbol{d}_j\| = 1, \ j = 1 : n \end{aligned} \tag{15}$$

unde Ω reprezintă pozițiile cu elemente nenule din $\boldsymbol{X}$, iar Ω^c setul complementar.

Schemă de calcul

Algorithm 3: Optimizare alternativă

Data: semnale $\boldsymbol{Y} \in \mathbb{R}^{m \times N}$
raritate s
dicționar inițial $\boldsymbol{D} \in \mathbb{R}^{m \times n}$
număr de iterații K

Result: dicționar antrenat $\boldsymbol{D}$

for $k = 1 : K$ **do**
- Codificare: păstrând $\boldsymbol{D}$ fixat, rezolvăm (14) pentru a calcula reprezentările rare $\boldsymbol{X}$
- Actualizarea dicționarului: păstrând tiparul nenul Ω fixat, rezolvăm (15) pentru a calcula dicționarul nou $\boldsymbol{D}$; matricea $\boldsymbol{X}$ poate fi actualiată sau nu
- Normarea atomilor, dacă nu a fost efectuată deja: $\boldsymbol{d}_j \leftarrow \boldsymbol{d}_j / \|\boldsymbol{d}_j\|$, $j = 1 : n$

Coordinate Descent (CD)

Observăm că produsul $\boldsymbol{DX}$ poate fi scris în funcție de fiecare atom

$$\boldsymbol{DX} = \sum_{i=1}^{n} \boldsymbol{d}_i \boldsymbol{x}_i^T. \tag{16}$$

deci contribuția atomului j poate fi separată

$$\|\boldsymbol{Y} - \boldsymbol{DX}\| = \left\| \boldsymbol{Y} - \sum_{i \neq j} \boldsymbol{d}_i \boldsymbol{x}_i^T - \boldsymbol{d}_j \boldsymbol{x}_j^T \right\|. \tag{17}$$

unde partea fixă este

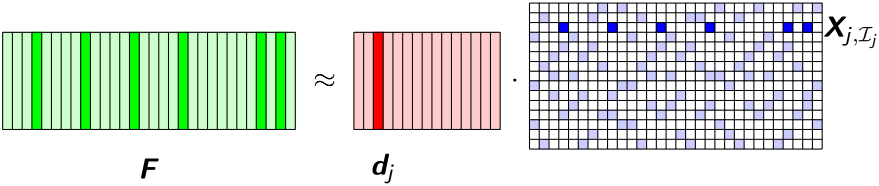
$$\boldsymbol{F} = \left[\boldsymbol{Y} - \sum_{i \neq j} \boldsymbol{d}_i \boldsymbol{x}_i^T \right]_{\mathcal{I}_j}, \tag{18}$$

iar actualizarea atomului $\boldsymbol{d}_j$ implică rezolvarea problemei

$$\min_{\boldsymbol{d}_j} \quad \left\| \boldsymbol{F} - \boldsymbol{d}_j \boldsymbol{X}_{j,\mathcal{I}_j} \right\|_F^2 \tag{19}$$

Exemplu: actualizarea atomului $\boldsymbol{d}_j$

Problema aproximării (19) este



Soluție CD

Soluția problemei (19) este

$$\boldsymbol{d} = \frac{\boldsymbol{F}\boldsymbol{x}}{\|\boldsymbol{x}\|^2}. \tag{20}$$

Demonstrație folosind trasa și proprietatea $\mathrm{tr}(\boldsymbol{ABC}) = \mathrm{tr}(\boldsymbol{BCA})$.

$$\begin{aligned}
\left\|\boldsymbol{F} - \boldsymbol{d}\boldsymbol{x}^T\right\|_F^2 &= \mathrm{tr}[(\boldsymbol{F}^T - \boldsymbol{x}\boldsymbol{d}^T)(\boldsymbol{F} - \boldsymbol{d}\boldsymbol{x}^T)] \\
&= \mathrm{tr}(\boldsymbol{F}^T\boldsymbol{F}) - 2\mathrm{tr}(\boldsymbol{F}^T\boldsymbol{d}\boldsymbol{x}^T) + \mathrm{tr}(\boldsymbol{x}\boldsymbol{d}^T\boldsymbol{d}\boldsymbol{x}^T) \\
&= \|\boldsymbol{F}\|_F^2 - 2\boldsymbol{x}^T\boldsymbol{F}^T\boldsymbol{d} + \|\boldsymbol{x}\|^2\,\boldsymbol{d}^T\boldsymbol{d}.
\end{aligned} \tag{21}$$

Normalizare

Soluția problemei

$$\begin{aligned} \min_{\boldsymbol{d}} \quad & \left\| \boldsymbol{F} - \boldsymbol{d}\boldsymbol{x}^T \right\|_F^2 \\ \text{s.t.} \quad & \|\boldsymbol{d}\| = 1 \end{aligned} \tag{22}$$

este

$$\boldsymbol{d} = \frac{\boldsymbol{F}\boldsymbol{x}}{\|\boldsymbol{F}\boldsymbol{x}\|}. \tag{23}$$

Această soluție mai este soluția problemei CD?

Abordarea K-SVD

- cel mai popular algoritm de antrenare de dicționar
- actualizează simultan atomul și reprezentările ce îl folosesc

$$\begin{aligned} \min_{\boldsymbol{d},\boldsymbol{x}} \quad & \left\| \boldsymbol{F} - \boldsymbol{d}\boldsymbol{x}^T \right\|_F^2 \\ \text{s.t.} \quad & \|\boldsymbol{d}\| = 1 \end{aligned} \tag{24}$$

- poate fi privit drept o formă bloc a CD

Soluția K-SVD

Fie descompunerea SVD a matricei $\boldsymbol{F}$

$$\boldsymbol{F} = \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T, \tag{25}$$

atunci soluția problemei (24) este

$$\boldsymbol{d} = \boldsymbol{u}_1, \quad \boldsymbol{x} = \sigma_1 \boldsymbol{v}_1. \tag{26}$$

Algoritmul K-SVD

Algorithm 4: Actualizare K-SVD

Data: semnalele $\boldsymbol{Y} \in \mathbb{R}^{m \times N}$
dicționarul actual $\boldsymbol{D} \in \mathbb{R}^{m \times n}$
reprezentările $\boldsymbol{X} \in \mathbb{R}^{n \times N}$

Result: dicționarul actualizat $\boldsymbol{D}$

Eroarea de calcul $\boldsymbol{E} = \boldsymbol{Y} - \boldsymbol{D}\boldsymbol{X}$
for $j = 1$ **to** n **do**
    Modificarea erorii: $\boldsymbol{F} = \boldsymbol{E}_{\mathcal{I}_j} + \boldsymbol{d}_j \boldsymbol{X}_{j,\mathcal{I}_j}$
    Calculul primei valori singulare σ_1 a lui $\boldsymbol{F}$ împreună cu vectorii singulari asociați $\boldsymbol{u}_1$ and $\boldsymbol{v}_1$
    Actualizarea atomului și a reprezentărilor:
    $\boldsymbol{d}_j = \boldsymbol{u}_1, \quad \boldsymbol{X}_{j,\mathcal{I}_j} = \sigma_1 \boldsymbol{v}_1^T$
    Recalcularea erorii: $\boldsymbol{E}_{\mathcal{I}_j} = \boldsymbol{F} - \boldsymbol{d}_j \boldsymbol{X}_{j,\mathcal{I}_j}$

Aplicații

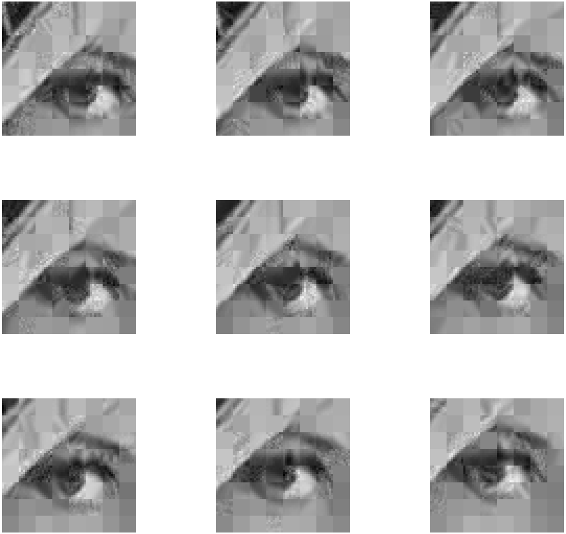
- eliminarea zgomotului (*denoising*)
- completarea unei imagini (*inpainting*)
- compresie
- clasificare
- rezumarea colecților de imagini (*image collection summarization*)
- rezumarea video (*video summarization*)
- albume foto (*photo albuming*)

Exemplu: eliminarea zgomotului

Original

Noisy

Cleaned



Overlapping Patches