,,,,

UNIVERSITATEA DIN BUCUREȘTI

FACULTATEA DE
MATEMATICĂ ȘI INFORMATICĂ

SPECIALIZAREA INFORMATICĂ

Proiect Procesarea Semnalelor

PROBLEMA RECUPERARII SEMNALULUI. METODE DE RECONSTRUCTIE RARA

Student

Neacsu Mihnea-Valentin

Coordonator

Prof. univ. dr. Rusu Cristian

București, ianuarie 2026

Rezumat

Problema recuperarii semnalului consta in a reconstrui un semnal $y' \in \mathbb{R}^n$ dintr-o masurare zgomotoasa $y \in \mathbb{R}^n$. Pentru aceasta, vom folosi metode bazate pe dictionare. Vom aplica aceste metode pe semnale sintetice (serii de timp) si fisiere audio.
In aceasta lucrare vom prezenta cateva metode de reconstructie rara a unui semnal pe baza unui dictionar, precum si o metoda de generare a dictionarelor.
Vom arata ca FISTA reconstituie fidel semnalul original, cu conditia sa fie suficient de simplu, si este mai rapid ca OMP, si OMP est mult mai bun la capturarea structurii complexe.

Abstract

The signal recovery problem is about recovering a signal $y' \in \mathbb{R}^n$ from a noisy measurement $y \in \mathbb{R}^n$. We will accomplish this goal using dictionary-based methods. We will apply these methods on 2 kind of input: synthetic time series and music files. In this paper, we will present a few methods for the sparse reconstruction of a signal based on a dictionary, as well as a few methods of generating dictionaries. We will prove that FISTA recinstruct the original signal faster and more accurately, but only if it is simple enough. OMP is much better at capturing complex structure.

Cuprins

1 Introducere

1.1 Prezentarea problemei

Problema recuperarii semnalului consta in recuperarea unui semnal $y' \in \mathbb{R}^n$ dintr-o masurare zgomotoasa $y \in \mathbb{R}^n$. Aplicatii ale acestei probleme sunt in imagistica medicala [2], in reprarea imaginilor distruse(inpainting), sau chiar compresie [3]. IN aceasta lucrare, vom aplica solutia bazata pe dictionare pentru repararea semnalelor audio corupte.

1.2 State-of-the-art in domeniu

La ora actuala, una din cele mai bune solutii pentru problema presupune aplicarea unei retele neurale adanci pe intreg semnalul, numita BiL-DCAE (Bidirectional LSTM-based Denoising Convolutional Autoencoders), [6]. Aceasta intoarce un semnal "filtrat", printr-o reducere de dimensiune, similar cu un autoencoder clasic.
Mai exact, [6] descrie reteaua astfel: la inceput un encoder convolutional, constand in L aplicari ale functiei:

$$h^{(i)} = \text{ReLU}(W^{(i)} * h^{(i-1)} + b^{(i)}), \text{cu} i = 1, 2, \cdots L$$

, unde L este numarul de straturi convolutionale din encoder, $W^{(i)}$ si $b^{(i)}$ sunt kernelul si biasul fiecarui strat, $*$ este operatorul de convolutie, iar $h^{(0)}$ este, prin conventie, inputul. Pe output actioneaza 2 straturi BiLSTM, succesiv. Fiecare aplica LSTM pe $(h_i, \overrightarrow{h_{i-1}})$ si $(h_i, \overleftarrow{h_{i-1}})$, unde h_i si h_{i-1} sunt starile curenta si anterioara ale retelei, iar sagetile indica ordinea in care citim semnalul ($\rightarrow$ e de la stanga la dreapta , $\leftarrow$ e de la dreapta la stanga). Apoi, recuperam rezultatul aplicand un strat idientic cu primul, doar inlocuim ReLU cu alta functie de activare ϕ, care este liniara.

1.3 Solutia noastra

Solutia noastra se bazeaza pe faptul ca semnalul cautat, y', are o structura simpla, si se poate scrie ca o combinatie liniara de semnale cunoscute. In acest sens, vom incerca sa scriem y sub forma $Dx + v$, unde D este un "dictionar" de semnale de baza, x este reprezentarea lui y, iar v este un termen zgomot pe care vrem sa-l eliminam.

In aceasta scriere, vom considera ca $y, v \in \mathbb{R}^n, D \in \mathbb{R}^{n \times n}$ si $x \in \mathbb{R}^m$, unde $m \gg n$. Coloanele lui D reprezinta semnale de baza, pe care le vom numi, de aici inainte, atomi. Conditia $M > N$ este necesara pentru ca sistemul de ecuatii $y' = Dx$ trebuie sa aiba solutii, dintre care sa cautam una cat mai rara.

1.4 Reprezentarea ca problema de optimizare

Pornind de la presupunerea ca solutia x cautata este rara, putem exprima problema de optimizare in 2 moduri, in functie de cum definim raritatea. Prima cale este sa calculam

$$\min_{x \in \mathbb{R}^m} \|y - Dx\| \quad \text{a.i.} \quad \|x\|_0 \leq S \tag{1.1}$$

,unde prin $\|x\|_0$ intelegem numarul de valori diferite de 0 din x.

O alta cale ar fi sa calculam

$$\min_{x \in \mathbb{R}^m} \|y - Dx\| + \lambda \|x\|_1 \tag{1.2}$$

,unde $\|x\|_1$ este norma Manhattan a lui x, iar λ este un hiperparametru.

A doua abordare are marele avantaj ca functia-obiectiv este convexa, dar nu este derivabila pe tot domeniul de definitie, deci putem aplica un algoritm numit Fast Iterative Shrinkage Threshold Algorithm (FISTA) [1]. Rata sa de convergenta este $O(k^{-2})$, adica a k-a aproximare a solutiei este, pe medie, de k^2 mai apropiata de original decat prima. Mai mult, putem controla si "raritatea" solutiei.

Din nefericire, in aceasta varianta exista posibilitatea ca solutia sa includa zgomot, adica nu obtinem ce ne-am propus. Mai mult, convergenta tinde sa dureze. Asadar, in general preferam sa rezolvam prima problema. Acolo, putem aplica un algoritm greedy, unde alegem cate un atom potrivit la fiecare iteratie. Un exemplu este in [7].

In continuare, vom prezenta mai detaliat cei doi algoritmi.

2 Prezentarea principalilor algoritmi

2.1 ISTA si FISTA

Un algoritm simplu pentru a rezolva problema

$$\min_{x\in\mathbb{R}^m} \|y - Dx\| + \lambda\|x\|_1 \tag{2.1}$$

ar fi o coborare clasica pe gradient: incepem cu o solutie x_0 si definim $x_{i+1} = x_i - t\nabla f(x_i)$, pentru $i \geq 0$,unde t este un hiperparametru, iar $f(\mathrm{x}) = \|y - Dx\| + \lambda\|x\|_1$. Repetam pana cand suntem multumiti de aproximare.

Calculul acesta se poate rescrie echivalent:

$$\min_{x\in\mathbb{R}^m} f(x_i) + (x - x_i)^T\nabla f(x_k) + \frac{1}{2t}\|x - x_i\|^2 \tag{2.2}$$

conform [3]

Ecuatia 2.2 se generalizeaza natural pentru cazul problemei noastre, la

$$\min_{x\in\mathbb{R}^m} (f(x_i) + (x - x_i)^T\nabla f(x_k) + \frac{1}{2t}\|x - x_i\|^2 + g(x)) \tag{2.3}$$

unde prin $g(x)$ intelegem $\lambda\|x\|_1$, iar t este un hiperparametru prin care controlam lungimea pasului (nu vrem pasi excesiv de lungi, am putea rata minimul). Sau, mai departe, restrangand in patrat,

$$\min_{x\in\mathbb{R}^m} \frac{1}{2t}\|x - x_i + t\nabla f(x_i)\|^2 + g(x) \tag{2.4}$$

Solutia noii probleme de optimizare 2.4 este, conform aceluiasi [3], $sgn(z_i) + \max(z_i - t\lambda, 0)$, aplicata pe componente, unde $z_i = x_i - t\nabla f(x_i)$.

Acum, daca inlocuim gradientul in expresia lui z_i, obtinem:

$$x_{i+1} = sgn(x_i^{'}) + \max(x_i^{'} - t\lambda, 0) := \phi(x) \tag{2.5}$$

unde $x_i^{'} = x_i - 2tD^T(Dx_i - y)$.

Aplicarea repetata a relatiei 2.5 este cunoscuta sub numele de ISTA (Iterative Shrinkage Thresholding Algorithm).

FISTA(Fast ISTA), algoritmul descris in [1], se bazeaza pe aceeasi idee, dar introduce si un termen impuls: in loc sa genereze o solutie doar din solutia precedenta, el foloseste si diferenta intre ultimele 2 solutii Formula de generare a noilor termeni este:

$$x_{i+1} = \phi(x_i) + \frac{\tau_i - 1}{\tau_{i+1}}(\phi(x_i) - \phi(x_i - 1))$$

, unde $(\tau_i)_{i \geq 0}$ controleaza influenta termenului impuls. El este definit in [1] recursiv: $\tau_0 = 1$ si $\tau_{i+1} = \frac{\sqrt{4\tau_i^2+1}+1}{2}$.

Complexitatea sa este pur si simplu $O(steps * N * M)$, deoarece la fiecare pas efectuam o inmultire matrice-vector, care domina complexitatea.

2.2 OMP

Metodele ISTA si FISTA, cu toata eleganta lor, au dezavantajul ca nu controleaza raritatea solutiei: ar putea gasi un raspuns de forma

$$y = \begin{bmatrix} 10^{-3} \\ 10^{-3} \\ \vdots \\ 10^{-3} \end{bmatrix} \tag{2.6}$$

, ceea ce incalca presupunerea pe care ne bazam. De asemenea, "controlul fin" propus de ele nu este mereu foarte fin, si putem obtine o solutie cu foarte mult zgomot. De aceea, putem alege sa rezolvam problema:

$$\begin{aligned} &\min_x \|y - Dx\|_2 \\ &\text{s.t. } \|x\|_0 \leq S \end{aligned} \tag{2.7}$$

Pentru aceasta problema, functioneaza bine algoritmi Greedy, care aleg "cel mai po-

trivit atom" la fiecare pas. Asemenea algoritmi sunt descrisi in [4], [7].
Pentru un asemenea set de atomi S, definim "reziduallul":

$$e := y - \sum_{i=1}^{|S|} \alpha_j D_j$$

Algoritmii acestia au forma generala:

1. Alege "cel mai bun" atom.
2. Calculeaza cea mai buna aproximare bazata pe atomii alesi pana acum.
3. Calculeaza ce ramane din semnalul original.
4. Repeta pana se verifica conditia de raritate.

OMP este un asemenea algoritm. La fiecare pas, el alege atomul cel mai corelat cu e. Partea a doua a problemei ne cere sa gasim $\alpha_1, \alpha_2 \cdots \alpha_k$ astfel incat

$$\|y - \sum_{i=1}^{k} \alpha_i D_i\|$$

sa fie minima. Aceasta problema este una clasica: vrem sa proiectam y pe subspatiul generat de D_i. Vom face asta cu metoda celor mai mici patrate (least-squares).

Reamintim: alegem s atomi, dintr-un dictionar cu M atomi, de lungime N. Complexitatea acestui algoritm este $O(sM(N + s^2))$, deoarece calculul celui mai bun atom este $O(NM)$, iar rezolvarea problemei least-squares este $O(Ms^2)$. Pentru s mic, este net mai rapid ca FISTA.

2.3 OMP optimizat

Desi mai rapid ca FISTA, OMP poate fi destul de lent, in special pe semnale lungi. Astfel, cautam o cale de a-l accelera. Cea mai "lenta" parte a algoritmului este calculul proiectiei semnalului y pe subspatiul generat de atomii curenti, la fiecare pas, ceea ce implica o problema least-squares, de complexitate $O(Ms^2)$.

Ideea este urmatoarea: daca am nota cu A atomii selectati la un moment dat, noi trebuie sa calculam inversa matricei $A^T A$. Matricea aceasta este mereu simetrica si pozitiv semi-definita, deci admite mereu o factorizare Cholesky, fie ea LL^T. Inmultirea cu cele 2 matrice L si L^T este mai rapida, ele fiind triunghiulare.

Mai mult, nu este nevoie sa calculam factorizarea Cholesky la fiecare pas: conform unui truc din [3], putem sa inmultim matricele pe blocuri de fiecare data cand adaugam un atom nou.

Mai exact, notand A augmentat cu d ca $\tilde{A}$, avem:

$$\tilde{A}^T \tilde{A} = \begin{bmatrix} A^T A & A^T \mathrm{d} \\ \mathrm{d}^T A & \mathrm{d}^T \mathrm{d} \end{bmatrix} = \begin{bmatrix} L & 0 \\ \mathrm{w}^T & 1 \end{bmatrix} \begin{bmatrix} L^T & \mathrm{w} \\ 0 & 1 \end{bmatrix} = \tilde{L}\tilde{L}^T$$

unde w este solutie a sistemului de ecuatii $Lw = A^T d$. Duap aceea, noul set de coeficienti pentru atomi, x, se calculeaza ca $\tilde{L}\tilde{L}^T x = b$, unde b este lista produselor scalare dintre fiecare atom selectat si semnalul original. Astfel, in loc sa rezolvam 2 sisteme generale, rezolvam 3 sisteme triunghiulare, care se rezolva mai repede: $O(s^2)$ fata de $O(s^3)$ Complexitatea finala este $O(s(sM + MN + s^2))$, deoarece calculul celui mai corelat atom este tot $O(MN)$, solutia primului sistem este $O(sm)$ si solutia ultimelor 2 sisteme este $O(s^2)$.

2.4 Nota de implementare

Toti algoritmii din prezenta lucrare au fost implementati in Python/NumPy, folosind np.arrays pentru a creste viteza. Nu a fost nevoie de structuri de date mai speciale.

3 Alegerea dictionarului

3.1 Dictionare predefinite

Un caz particular al problemei recuperarii semnalului, anume cel cu

$$D = \begin{bmatrix} 1 & 1 & 1 & ... & 1 \\ 1 & \omega & \omega^2 & ... & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & ... & \omega^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & ... & \omega^{(N-1)(N-1)} \end{bmatrix} \tag{3.1}$$

(unde $\omega = e^{\frac{2\pi \mathrm{i}}{N}}$)

este Discrete Fourier Transform (DFT), care ne permite sa aproximam un semnal cu sinusoide pure. Dar, in acest caz, nu este nevoie de OMP: aici functioneaza FFT, care gaseste o descompunere in $O(N \log N)$.

Acelasi algoritm se aplica si pe alte transformari ortogonale. Dictionarul D fiind redundant, o prima idee pe care o putem incerca este sa "simulam" niste semnale simple, cum ar fi o serie de unde cosinus.

Cu toate astea, forta OMP incepe sa se vada atunci cand combinam semnalele clasice cu altele, cum ar fi cele din matricele wavelet.[2] Algoritmul lucreaza pe dictionare oricat de mari si oricat de variate, deci putem captura multe semnale, cum vom vedea in sectiunea urmatoare.

3.2 Dictionare antrenate

TBA

[2]Pentru mai multe detalii, a se vedea Apendicele.

Testare

4 Nota

Pe tot parcursul testarii, am folosit un dictionar mixt, compus din N unde cosinus si o matrice wavelet de tip DB4. FISTA este rulat cu $\lambda = 10$, pret de 7 iteratii pe audio, si cu $\lambda = 30$ pret de 40 de iteratii pe serii de timp clasice.

5 Rezultatele testarii

5.1 Serii de timp

Pe serii de timp, OMP, cu dictionar de cosinus si wavelet, s-a descurcat de minune, obtinand MSE-uri mici, de ordinul lui 0.1, pe serii de timp artificiale. Asta era de asteptat, deoarece seriile de timp nu au o structura deosebit de complexa. FISTA, in schimb, are tendinta de a "aproxima" si semnalul original, indiferent cum am alege hiperparametrii, si are MSE-uri mai mari, dar nu cu mult (cel mai bun este 0.2, pentru 40 de pasi, si nu trece de 0.3)
Includem mai jos 2 grafice, unul cu un semnal recuperat cu OMP, folosind maxim 10 atomi, si unul recuperat cu FISTA. Ambele sunt suprapuse peste cel de input.

5.2 Fisiere audio

Vom testa pe 4 fisiere: unul cu muzica pop, unul cu muzica simfonica, unul cu rock/opera si unul cu jazz, de cate 10 secunde fiecare. Ele se afla pe https://github.com/neomihnea21/ice-queen, deoarece nu pot fi inserate in PDF. Peste ele, introducem "zgomot roz" [8], adica zgomot in care orice interval de o octava are aceeasi energie. Il generam aplicand un filtru trece-jos pe un zgomot alb, la frecventa de esantionare a muzicii originale. Mai jos este un tabel cu rezultatele.

Observam ca pe cele mai complexe dintre liniile melodice, cea de jazz si cea simfonica, algoritmii au avut cel mai mic MSE. Intre celelalte nu se observa diferente notabile.

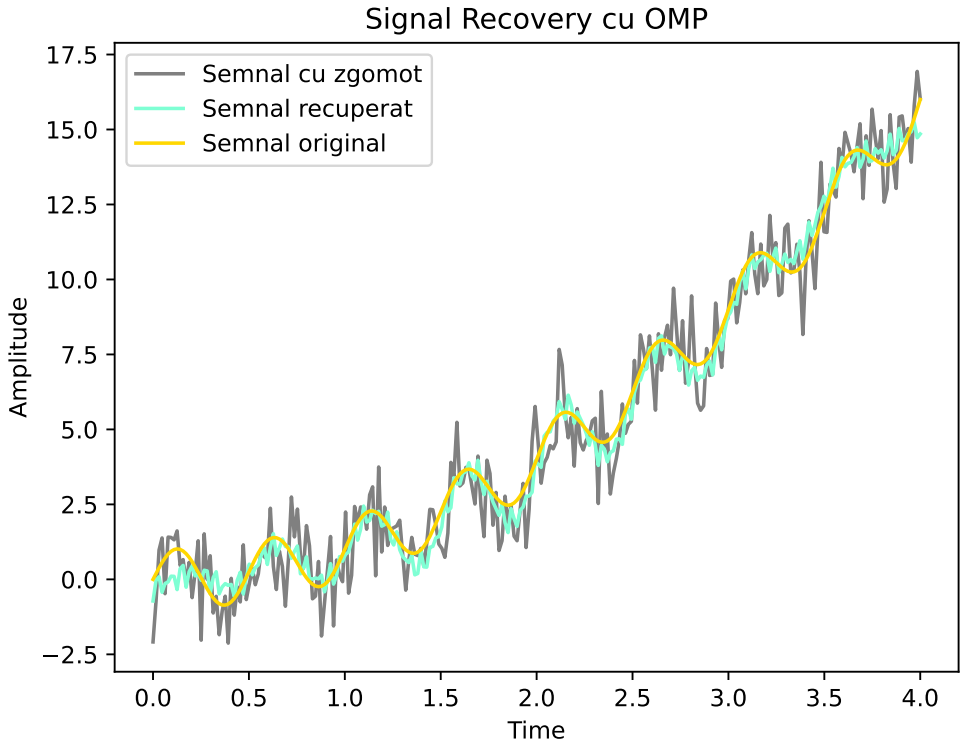

Figura 1: Recuperarea semnalului folosind OMP (maxim 10 atomi).

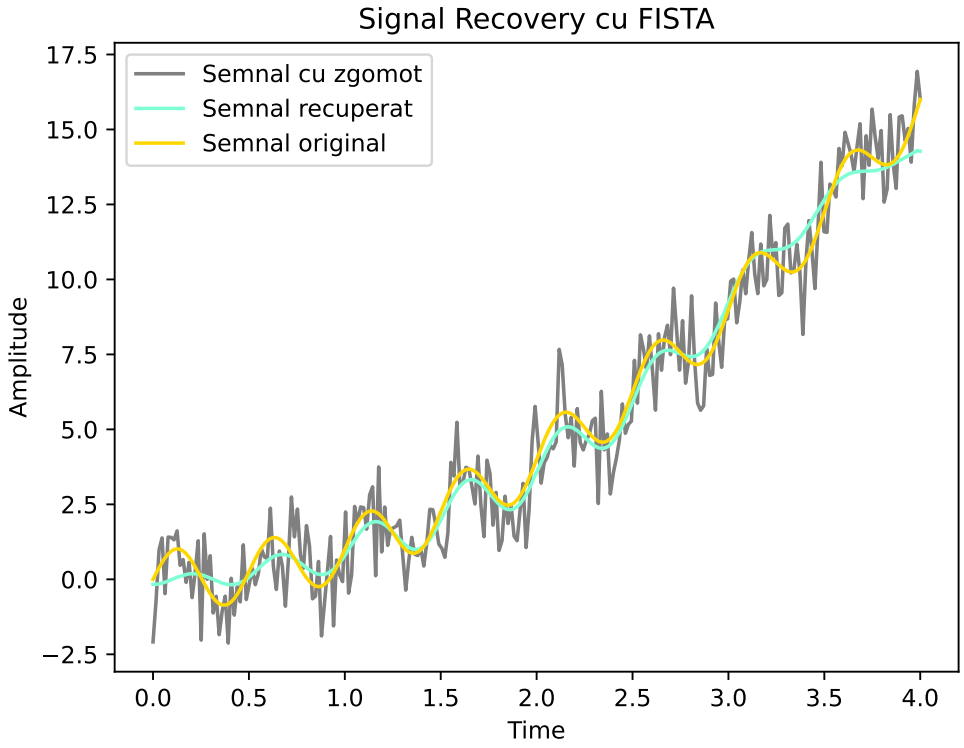

Figura 2: Recuperarea semnalului folosind FISTA.

Cantece	OMP time(s)	FISTA time(s)	OMP error	FISTA
Closing Time(Semisonic, pop)	61.4126	1.2844	0.0019	0.000
De Vei Pleca(Iris + Felicia Filip, rock/opera)	63.6477	1.3480	0.0006	1.1524e
Unit Structures (Cecil Taylor, jazz)	65.1244	1.3374	1.2600e-05	3.2754e
Nunta lui Figaro (W.A.Mozart, simfonica)	60.7197	1.3277	4.1281e-05	1.5743e

Tabela 1: Timpi si erori pentru fisierele audio.

De asemenea, observam ca, desi FISTA are MSE si SNR mai mici, eroarea se afla ”unde nu trebuie”. Daca ascultam rezultatele sale, nu se disting cantecele, ci doar un zgomot alb.

Asadar, MSE si SNR nu sunt metrici de incredere si nu reflecta strict calitatea recuperarii.

Concluzii

In concluzie, desi FISTA este de 7.5 ori mai rapida, in medie, decat OMP, nu se descurca deloc pe semnale cu structura complexa, cum ar fi cele muzicale, dar pe semnale cu structura simpla este net superioara.

De asemenea, observam cu stupoare ca pe cea mai complexa linie melodica, recuperarea facuta de OMP a fost cea mai fidela.

Mai mult, ambele metode sunt net superioare, ca resurse de timp si memorie folosite, fata de [6].

5.3 Directii de dezvoltare

In viitor, intentionam sa implementam dictionare unicat, care nu sunt doar wavelets / cosinusuri clasice. In acest sens, vom obtine dictioarele prin invatare nesupervizata, folosind metode clasice ca MOD si K-SVD. Mai mult, avem in vedere si implementarea de metode moderne, bazate pe autoencodere[5], pentru antrenarea dictionarului. Vom compara aceste metode intre ele si, eventual, cu reteaua neurala din [6].

A Generarea matricelor wavelet

Spre deosebire de alte transformari clasice ale semnalelor, cum ar fi DFT sau DCT, transformata wavelet (DWT) nu are o matrice $A \in \mathbb{R}^{n \times n}$ clasica, cu proprietatea ca $\text{DWT}(y) = Ay$, pentru un semnal $y \in \mathbb{R}^n$ pe care o putem scrie in avans - trebuie sa o generam. Pentru a o genera, ne folosim de "filtrele" specifice functiei wavelet folosite.

Mai concret, fie e_i al i-lea vector din reperul canonic al lui $\mathbb{R}^n$. Acestuia ii aplicam o transformata wavelet, prin downsampling $\downarrow 2$ succesiv si filtrare, si concatenam rezultatele. Aceasta este a i-a coloana din matricea transformarii wavelet.

Metoda functioneaza, deoarece, in ipoteza ca DWT este o transformare liniara, este suficient sa stim $\text{DWT}(b_1), \text{DWT}(b_2) \cdots$, pentru ca $\text{DWT}(v)$ este o combinatie liniara a lor, pe care o putem calcula folosind matricea.

Referințe

[1] Amir Beck și Marc Teboulle, „A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems", în SIAM Journal on Imaging Sciences 2.1 (Ian. 2009), pp. 183–202, DOI: 10.1137/080716542, URL: http://dx.doi.org/10.1137/080716542.

[2] Pingmei Cai, Guinan Wang, Shiwei Yu, Hongjuan Zhang, Shuxue Ding și Zikai Wu, „Sparse electrocardiogram signals recovery based on solving a row echelon-like form of system", în IET Systems Biology 10.1 (Feb. 2016), pp. 34–40, ISSN: 1751-8857, DOI: 10.1049/iet-syb.2015.0002, URL: http://dx.doi.org/10.1049/iet-syb.2015.0002.

[3] Bogdan Dumitrescu și Paul Irofti, Dictionary Learning Algorithms and Applications, Springer International Publishing, 2018, ISBN: 9783319786742, DOI: 10.1007/978-3-319-78674-2.

[4] S. G. Mallat și Zhifeng Zhang, „Matching pursuits with time-frequency dictionaries", în IEEE Transactions on Signal Processing 41.12 (1993), pp. 3397–3415, ISSN: 1053-587X, DOI: 10.1109/78.258082.

[5] Anish Mudide, Joshua Engels, Eric J. Michaud, Max Tegmark și Christian Schroeder de Witt, Efficient Dictionary Learning with Switch Sparse Autoencoders, 2024, arXiv: 2410.08201 [cs.LG], URL: https://arxiv.org/abs/2410.08201.

[6] Yun Pan, Quan Luo, Yiyou Fan, Haoming Chen, Donghua Zhou, Hongsheng Luo, Wei Jiang și Jinshan Su, „Deep Learning-Based Denoising of Noisy Vibration Signals from Wavefront Sensors Using BiL-DCAE", în Sensors 25.16 (Aug. 2025), p. 5012, ISSN: 1424-8220, DOI: 10.3390/s25165012, URL: http://dx.doi.org/10.3390/s25165012.

[7] Y. C. Pati, R. Rezaiifar și P. S. Krishnaprasad, „Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition", în Proceedings of 27th Asilomar Conference on Signals, Systems and Computers, vol. 1, Nov. 1993, pp. 40–44.

[8] Richard F. Voss și John Clarke, „"1/f noise" in music: Music from 1/f noise", în The Journal of the Acoustical Society of America 63.1 (1978), pp. 258–263, DOI: 10.1121/1.381721.