UNIVERSITATEA DIN BUCUREȘTI

FACULTATEA DE
MATEMATICĂ ȘI INFORMATICĂ

SPECIALIZAREA INFORMATICĂ

Proiect Procesarea Semnalelor

PROBLEMA RECUPERARII SEMNALULUI. METODE DE RECONSTRUCTIE RARA

Absolvent

Neacsu Mihnea-Valentin

Coordonator științific

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. In aceasta lucrare vom prezenta cateva metode de reconstructie rara a unui semnal pe baza unui dictionar, precum si cateva metode de generare a dictionarelor.

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. 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.

Cuprins

1 Introducere

1.1 Prezentarea problemei

Problema recuperarii semnalului consta in recuperarea unui semnal $y' \in \mathbb{R}^n$ dintr-o masurare zgomotoasa $y \in \mathbb{R}^n$. O metoda de a face acest lucru 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.

1.2 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 [4].

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 [2]

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 [2], $sgn(z_i) + \max(z_i - t\lambda, 0)$, 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 $\tau_{i+1} = \frac{\sqrt{4\tau_i^2+1}+1}{2}$.

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: este foarte posibil ca siloutia problemei de optimizare, pentru 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 potrivit atom" la fiecare pas. Asemenea algoritmi sunt descrisi in [3], [4].

Pentru un asemenea set de atomi S, definim "rezidualul":

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

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.

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 $_i$. Vom face asta cu metoda celor mai mici patrate (least-squares).

Complexitatea acestui algoritm este $O(sM(N + s^2))$, deoarece calculul celui mai bun atom este $O(NM)$, iar rezolvarea problemei least-squares este $O(Ms^)$. 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 [2], putem sa inmultim matricele pe blocuri de fiecare data cand adaugam un atom nou.

3 Alegerea dictionarului

3.1 Dictionare predefinite

Un caz particular al problemei recuperarii semnalului, anume cel cu

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

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

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 ”extindem o transformare”. Mai jos avem un exemplu, atomii unui DCT extins.

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. (Exemplu imagine before/after)

3.2 Dictionare antrenate

TBA

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

Concluzii

4 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 $\mathrm{DWT}(y) = Ay$, 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 $\mathrm{DWT}(b_1), \mathrm{DWT}(b_2) \cdots$, pentru ca $\mathrm{DWT}(v)$ 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] Bogdan Dumitrescu și Paul Irofti, „Sparse Representations", în Dictionary Learning Algorithms and Applications, Springer International Publishing, 2018, pp. 1–23, ISBN: 9783319786742, DOI: 10.1007/978-3-319-78674-2_1.

[3] 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.

[4] 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.