
Raport științific

Proiect PN-III-P1-1.1-PD-2016-0127

Prelucrarea eficientă a semnalelor biomedicale folosind
codarea rară (SCEBIOS)

Etapă 2, 2019

dr.ing. Nicolae Cleju

Cuprins

1	Introducere	3
1.1	Sumar al rezultatelor științifice din anul 2019	3
2	Metode de identificare a patologiilor inimii	5
2.1	Identificarea patologiilor inimii folosind codarea rară și tehnica Bag-of-Words . .	5
2.1.1	Tehnica Bag-of-Words	5
2.1.2	Codarea rară a semnalelor	6
2.1.3	Etape ale procesului de clasificare	7
2.1.4	Baza de date “PhysioNet Computing in Cardiology Challenge 2017” . . .	13
2.1.5	Rezultate de simulare	13
3	Metode îmbunătățite de codare rară, cu aplicații în codarea semnalelor de tip ECG	18
3.1	Metoda proiecțiilor afine iterative (Iterative Affine Projections)	18
3.1.1	Introducere	18
3.1.2	Algoritmul Iterative Hard Thresholding	20
3.1.3	Algoritmul proiecțiilor afine iterative (Iterative Affine Projection)	21
3.1.4	Garanții teoretice pentru succesul algoritmului IAP în cazul reconstrucției fără zgomot	24
3.1.5	Avantajele algoritmului IAP	26
3.1.6	Considerații referitoare la implementare	28
3.1.7	Rezultate ale simulărilor	29
3.2	Metode de fuziune între algoritmii Orthogonal Matching Pursuit și Least Squares Pursuit	36
3.2.1	Introducere	37
3.2.2	Strategii euristice de fuziune a algoritmilor OMP și LSP	40
3.2.3	Rezultate de simulare	42
4	Concluzii	47
	Bibliografie	48

1 Introducere

Prezentul raport științific și tehnic prezintă rezultatele proiectului de cercetare postdoctorală (PD) PN-III-P1-1.1-PD-2016-0127 intitulat “Prelucrarea eficientă a semnalelor biomedicale folosind codarea rară (SCEBIOS)”, etapa 2 / 2019.

1.1 Sumar al rezultatelor științifice din anul 2019

Un sumar al rezultatelor obținute este prezentat mai jos:

- Lucrare în conferință internațională indexată ISI:
N. Cleju, I.Ciociu, “Spectrum Insensitive Sparse Recovery with Iterative Affine Projections”, Proc. 27th European Signal Processing Conference EUSIPCO 2019, 2019, A Coruna, Spain, pp.1-4, ISBN 978-9-0827-9702-2
 - Lucrarea este prezentată *in extenso* în secțiunea 3.1
 - Codul sursă aferent lucrării este disponibil la adresa https://github.com/nikleju/code_paper_IterativeAffineProjections
- Lucrare în conferință internațională indexată ISI:
N. Cleju, I.Ciociu, “Fusion of Orthogonal Matching Pursuit and Least Squares Pursuit for Robust Sparse Recovery”, Proc International Symposium on Signals, Circuits and Systems (ISSCS) 2019, Iasi, Romania, 1-4
 - Lucrarea este prezentată *in extenso* în secțiunea 3.2
 - Algoritmii dezvoltați sunt disponibili la adresa https://github.com/nikleju/code_paper_FusionOMPLSP
- Două lucrări în pregătire:
 - O lucrare tip conferință cu titlul provizoriu “Atrial Fibrillation Classification in ECG signals using Bag-of-Words Sparse Coding Approach”, bazată pe demersul din secțiunea 2.1, aflată în lucru

- O lucrare destinată unei reviste ISI, cuprinzând versiunea extinsă și mai dezvoltată a lucrării “Spectrum Insensitive Sparse Recovery with Iterative Affine Projections” de mai sus, aflată în pregătire
- Algoritmii aferenți lucrărilor publicate deja sunt disponibili la adresele indicate mai sus
- Pagina web a proiectului este disponibilă la adresa www.etti.legacy.tuiasi.ro/ncleju/scebios

2 Metode de identificare a patologiilor inimii

2.1 Identificarea patologiilor inimii folosind codarea rară și tehnica Bag-of-Words

În această secțiune prezentăm o metodă de clasificare a patologiilor inimii din semnale de tip ECG, folosind tehnicile de codare rară și cea Bag-of-Words, ce are la baza metoda folosită în [1].

2.1.1 Tehnica Bag-of-Words

Una din soluțiile inovatoare utilizate în clasificarea semnalelor este tehnica denumită “Bag-of-Words”.

Tehnica “Bag-of-Words” este un model de analiză și reprezentare a datelor bazat pe numărarea trăsăturilor distincte dintr-un set de date. Aceste date sunt cunoscute sub numele de “cuvinte” (“words”). Metoda provine din domeniul procesării textelor, de unde și terminologia aferentă.

Dat fiind un text de lungime mare, tehnica “Bag-of-Words” analizează și stochează numărul de apariții a fiecărui cuvânt. Se obține astfel o histogramă a frecvențelor de apariție a fiecărui cuvânt, ceea ce reprezintă o metodă simplă și eficace de descriere a vocabularului folosit. Texte de natură diferită, cum ar fi texte literare sau jurnalistice, texte cu limbaj modern sau învechit sau texte ale unor autori diferiți, tind să aibă un vocabular diferit, ceea ce înseamnă că histogramele obținute pot fi utilizate ca element caracteristic în analiza și clasificarea datelor. Desigur, “cuvintele” contorizate pot fi nu doar cuvinte, și litere individuale, expresii sau alte trăsături numărabile din textele considerate.

Aceeași metodă se poate aplica în mod similar la orice semnal de tip serie de timp, atât timp cât se pot extrage trăsături localizate ale semnalelor care să reprezinte “cuvintele”. Pentru semnalele biomedicale analizate, în speță pentru semnale de tip electrocardiografic (ECG), aceste trăsături sunt obținute aplicând o fereastră alunecătoare asupra semnalului, segmentele extrase astfel fiind urmate de codare rară în raport cu un dicționar optimizat de trăsături.

2.1.2 Codarea rară a semnalelor

Codarea rară (“*sparse coding*”) este o metodă modernă de reprezentare și analiză a semnalelor, care a cunoscut o largă utilizare în ultimul deceniu. Principiul de funcționare al metodei este prezentat în continuare, pentru semnale unidimensionale.

Dat fiind un semnal $x \in \mathbb{R}^n$ și un dicționar $D \in \mathbb{R}^{n \times N}$, conținând N vectori d_i (numiți *atomi* ai dicționarului), codarea rară presupune găsirea unei reprezentări a semnalului x ca o combinație liniară a unui număr redus de atomi. În forma sa cea mai simplă, acest deziderat se poate exprima ca o problemă de optimizare de tipul următor:

$$\gamma^* = \arg \min_{\gamma} \|\gamma\|_0 \text{ cu constrângerea } x = D \cdot \gamma. \quad (2.1)$$

Subliniem că D reprezintă, de obicei, un dicționar *supracomplet*, adică numărul de atomi depășește dimensiunea spațiului în care se lucrează, $N > n$. Acest lucru face ca, în general, să existe multiple reprezentări admisibile ale semnalului x folosind atomi din D . Obiectivul codării rare este de a găsi cea mai rară (*sparse*) dintre acestea, adică cea care utilizează cel mai mic număr de atomi, întrucât în acest fel reprezentarea va conține în special componentele dominante care compun semnalul x .

În practică, condiția unei reprezentări exacte a semnalului nu este realistă atunci când se consideră semnale afectate de zgomot, și de aceea condiția $x = D \cdot \gamma$ este de obicei înlocuită cu o variantă mai relaxată, și anume $\|y - D \cdot \gamma\| \leq \epsilon$, care permite o eroare de aproximare în limita unui factor ϵ . În plus, utilizarea normei ℓ_0 pentru a cuantifica raritatea unei reprezentări γ conduce la probleme ne-convexe, de tip NP-hard [2], și de aceea ea poate fi înlocuită cu norma ℓ_1 , care încurajează raritatea reprezentării dar în același timp asigură convexitatea problemei. Problema de optimizare care se obține astfel, cunoscută sub numele de *LASSO* sau *Basis Pursuit*, poate fi formulată în diferite moduri. În algoritmul descris în cele ce urmează, se folosește formularea neconstrânsă de mai jos:

$$\gamma^* = \arg \min_{\gamma} \frac{1}{2} \|x - D \cdot \gamma\|_2^2 + \lambda \|\gamma\|_1.$$

Sub această formă, problema de optimizare urmărește minimizarea simultană a erorii de reprezentare $\|x - D \cdot \gamma\|_2$ dar și a normei ℓ_1 a reprezentării, care încurajează obținerea unei reprezentări rare. Parametrul λ controlează raportul dintre cei doi termeni. Problema poate fi rezolvată în mod eficient utilizând algoritmi iterativi de optimizare. În algoritmul implementat aici, utilizăm cu precădere algoritmul din [3], implementat în biblioteca [4].

Elementul fundamental în orice problemă de codare rară este dicționarul D utilizat. În aplicații uzuale, se pot folosi cu succes fie dicționare bazate pe transformări standard de semnal (de ex. bazate pe transformata *wavelet*), fie dicționare *învățate*, atomii cărora sunt optimizați pentru reprezentarea compactă a unui număr mare de semnale x reprezentative. În literatura de

specialitate există un număr mare de algoritmi de învățare a dicționarilor, de ex. [5], [6]. În această lucrare utilizăm dicționare învățate, în principal pe baza algoritmului *online* descris în [6], și implementat în [4].

Corespondența dintre codarea rară și tehnica Bag-of-Words este următoarea. Atomii dicționarului corespund cuvintelor din modelul Bag-of-Words, și formează astfel întreg vocabularul disponibil al modelului. Pentru un segment de lungime redusă dintr-un semnal x , codarea rară indică care atomi reprezintă segmentul cel mai eficient, ceea ce corespunde “cuvintelor” aflate în segmentul respectiv. Dacă semnalul x este de durată mare, codarea rară se aplică pe segmente extrase succesiv din semnal folosind o fereastră alunecătoare. Cumulând reprezentările rare ale fiecărui segment se obține o descriere globală a semnalului, sub forma unei histograme care reflectă utilizarea medie fiecărui atom în întreg semnalul analizat.

2.1.3 Etape ale procesului de clasificare

În această secțiune este descris în detaliu procesul de analiză și prelucrare a semnalelor ECG. Acesta este sumarizat în Fig.2.1, și constă din următoarele etape:

1. Extragerea de trăsături primare din semnalul ECG;
2. În faza de antrenare: antrenarea unui dicționar din trăsăturile primare;
3. Obținerea trăsăturilor secundare prin codarea rară a trăsăturilor primare cu dicționarul utilizat;
4. Operația de “Pooling” pentru a obține histograma caracteristică a semnalului;
5. În faza de antrenare: antrenarea unui clasificator tip SMV pentru clasificarea histogramei semnalelor
6. Clasificarea histogramei semnalului cu clasificatorul SVM

În continuare, vom prezenta fiecare etapă constitutivă a procesului.

2.1.3.1 Extragerea de trăsături primare

Tehnicile de tip “Bag-of-Words” se bazează pe identificarea unor trăsături locale, la nivelul “cuvintelor” care compun o secvență de semnal.

În această lucrare, utilizăm două tipuri de trăsături:

- trăsături primare: sunt obținute direct din analiza semnalelor de tip serie de timp
- trăsături secundare: sunt obținute din trăsăturile primare, și sunt utilizate în clasificarea semnalelor

Pentru extragerea trăsăturilor primare, folosim o fereastră alunecătoare pentru a parcurge o înregistrare mai lungă a unui semnal ECG, din fiecare fereastră extrăgând un set de trăsături.

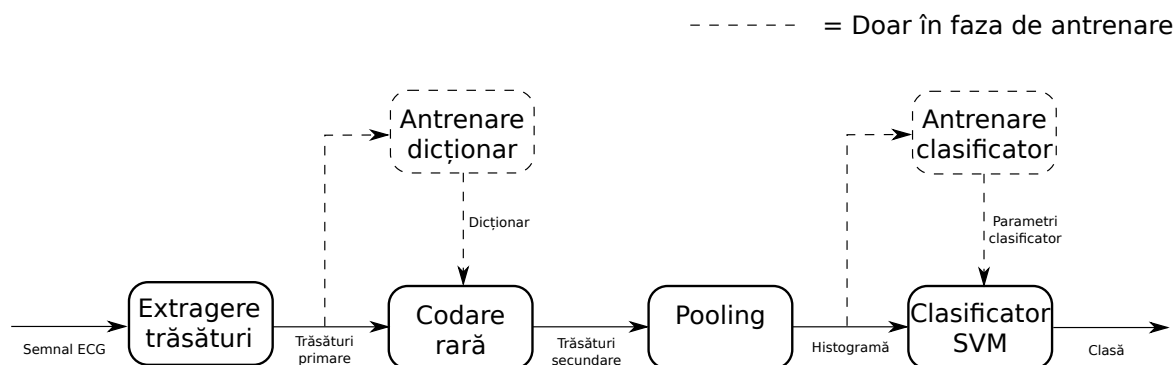


Figure 2.1: Schema bloc a procesului de clasificare a patologiilor din semnale ECG, folosind codarea rară și tehnica Bag-of-Words. Călea reprezentată punctat reprezintă operații care se fac doar în faza de antrenare.

Principiul extragerii de trăsături pe baza unei ferestre alunecătoare este ilustrat în Fig. 2.2. Semnalul, care poate avea o lungime considerabilă, este analizat în segmente de lungime fixă (ferestre). Semnalul $x[n]$ este înmulțit cu o fereastră $w[n]$ care păstrează doar o porțiune din semnalul original. Segmentul astfel obținut este apoi furnizat algoritmului de extragere de trăsături, care extrage un vector de trăsături (reprezentat în Fig. 2.2 sub forma coloanelor din matricea de trăsături). După aceea, fereastra $w[n]$ este deplasată cu un pas fix, pentru a selecta porțiunea următoare din semnal, iar procesul se repetă până la acoperirea întregului semnal.

În general, ferestrele utilizate în acest proces pot avea forme diferite. Fereastra rectangulară, cea mai simplă, furnizează un segment nealterat din semnalul original. Alte tipuri de ferestre (de ex. fereastra Hamming) ponderează valorile de la marginile segmentelor, pentru a atenua artefactele care pot apărea la capetele segmentelor, de exemplu în urma transformărilor frecvențiale. În această lucrare utilizăm fereastra rectangulară, procesul fiind controlat de doi parametri:

- lungimea ferestrei rectangulare
- pasul de deplasare

Întrucât din fiecare fereastră a semnalului se obține un vector de trăsături, la finalul întregii operații se obține o matrice de trăsături, care grupează trăsăturile obținute de la fiecare fereastră succesivă, în ordine.

În literatura de specialitate, nenumărați algoritmi au fost utilizați în scopul extragerii de trăsături dintr-un semnal. Aceștia pot fi împărțiți, în general, în trei categorii:

- algoritmi generali bazați pe transformări de semnal uzuale: trăsăturile reprezintă coeficienți obținuți prin transformarea DCT, FFT, *wavelet* etc. (de multe ori utilizând transformări liniare de semnal)
- algoritmi generali bazați pe procesări neliniare de semnal: trăsăturile sunt obținute prin

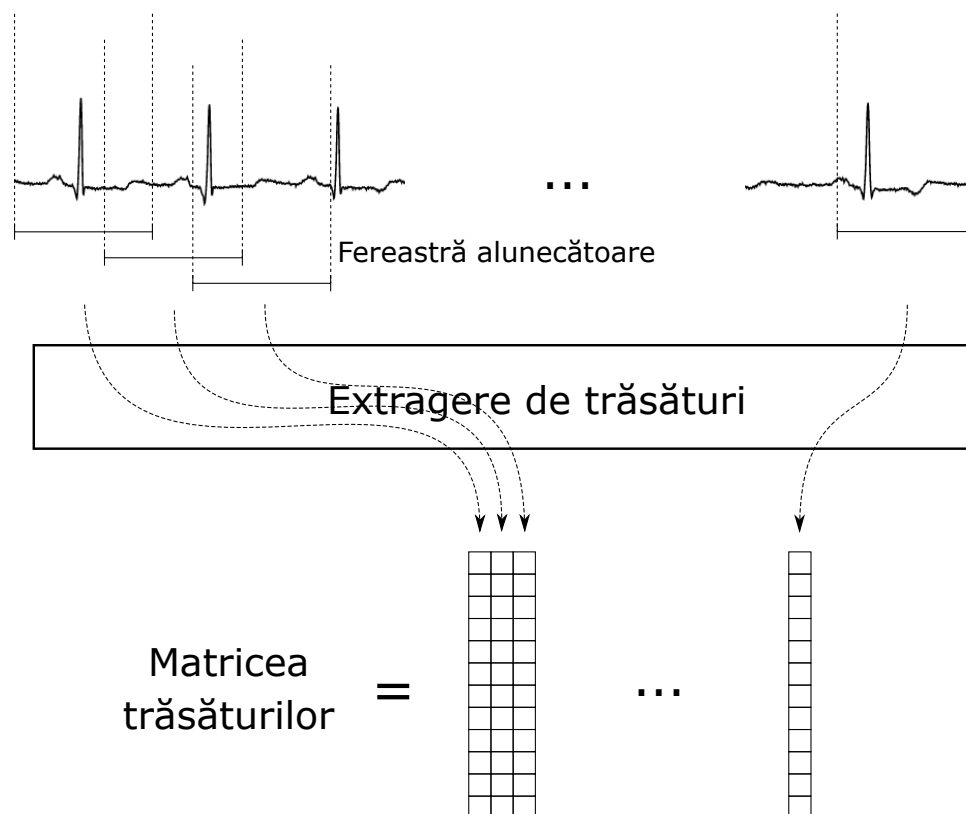


Figure 2.2: Extragerea trăsăturilor dintr-o serie de timp folosind o fereastră alunecătoare. Din fiecare fereastră de semnal se obține un vector de trăsături, în final obținându-se o matrice de trăsături pentru întregul semnal.

algoritmi mai complicați de tip codare rară, rețele neurale etc.

- algoritmi specifici unei anumite probleme sau categorii de semnale: furnizează trăsături care au o relevanță particulară în anumite scenarii. De exemplu, pentru detecția diferitelor tipuri de aritmii din semnale ECG sau PPG, se cunoaște importanța trăsăturilor care măsoară variabilitatea intervalului RR.

În această lucrare, pentru extragerea trăsăturilor primare optăm pentru algoritmi din prima categorie, din următoarele motive:

- vom utiliza algoritmi de codare rară (mai complecși) pentru trăsăturile secundare, așadar putem utiliza algoritmi mai simpli pentru trăsăturile primare
- pentru a nu reduce generalitatea metodei, nu apelăm la trăsături particulare adaptate unor anumite patologii speciale

În lumina acestor considerente, utilizăm în principal trăsături primare obținute prin transformarea *wavelet*, cu diverse nivele de rezoluție. Trăsăturile reținute corespund nivelelor inferioare de

rezoluție, coeficienții de detaliu de la nivelele superioare fiind ignorați, ceea ce aduce un beneficiu suplimentar sub forma robusteții la zgomot (eliminarea coeficienților de detaliu ai transformării *wavelet* fiind deseori utilizată ca metodă de filtrare de zgomot a semnalelor).

2.1.3.2 Generarea dicționarului

Din trăsăturile primare se obțin, la pasul următor, trăsăturile secundare. Acestea se obțin prin codarea rară a vectorilor de trăsături primare, în raport cu un dicționar fix. Acest dicționar este învățat din datele disponibile, într-o etapă de antrenare, urmând a fi utilizat fără modificări în faza de operare propriu-zisă.

Dicționarul utilizat în codarea rară este obținut cu unul din algoritmi de învățare a dicționarelor disponibil în literatură. O scurtă prezentare a unor algoritmi fundamentali din această familie a fost făcută în Raportul Științific aferent anului 2018, unde s-au trecut în revistă principalii algoritmi din domeniul codării rare. Ca atare, nu vom mai prezenta în detaliu natura problemei.

În prezenta lucrare, folosim în principal unul dintre cei trei algoritmi de mai jos:

- algoritmul de antrenare continuă Mairal *et.al.* [6]
- algoritmul Discriminative K-SVD (D-KSVD)[7], care extinde algoritmul clasic K-SVD prin învățarea unui clasificator liniar împreună cu dicționarul
- algoritmul Label-Consistent K-SVD (LC-KSVD)[8], care extinde algoritmul D-KSVD prin adăugarea posibilității de a împărți atomii dicționarului învățat în grupe alocate în mod special unor anumite categorii de semnale

Datele de intrare pentru algoritmul de învățare a dicționarului reprezintă vectorii de trăsături primare obținuți dintr-un număr de înregistrări ECG care formează setul de antrenare. Rezultatele finale și evaluarea algoritmilor disponibili sunt în curs de finalizare.

Menționăm că antrenarea dicționarului pentru codare rară este în multe privințe echivalentă cu o cuantizare vectorială, des întâlnită în algoritmi clasici de procesare de semnal. La fel cum în cuantizarea vectorială se aproximează vectorii de intrare cu un vector dintre vectorii de cuantizare disponibili (care trebuie în prealabil găsiți printr-un proces de antrenare), în codarea rară vectorii de intrare se aproximează cu o combinație liniară rară a atomilor din dicționar (care trebuie în prealabil găsiți printr-un proces de antrenare). Diferența constă, așadar, într-o generalizare a modelului de aproximare, de la aproximare cu un vector la aproximare cu o combinație liniară de vectori.

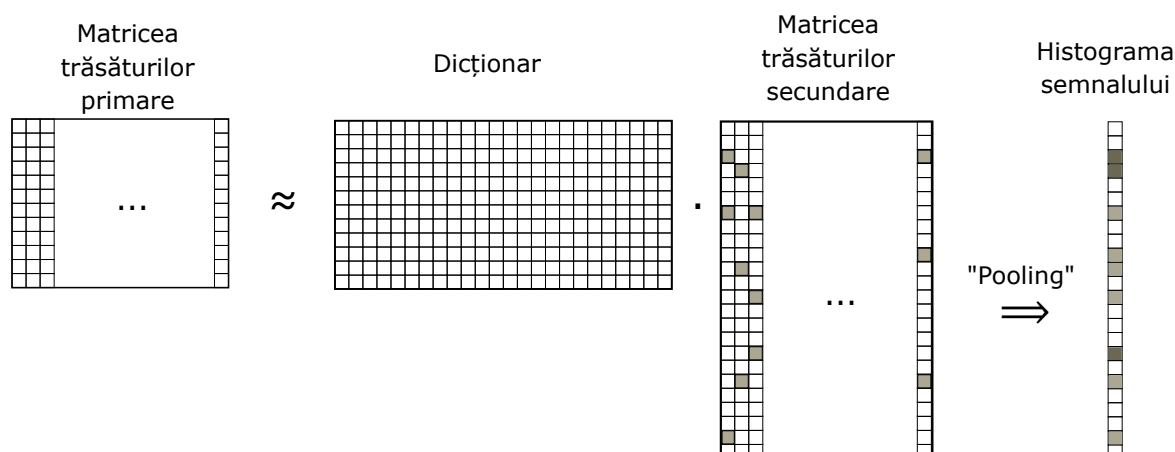


Figure 2.3: Extragerea trăsăturilor secundare din cele primare, folosind codarea rară. Din matricea trăsăturilor secundare se obține apoi o histogramă a utilizării atomilor din dicționar, care caracterizează întreg semnalul.

2.1.3.3 Codarea rară și obținerea trăsăturilor secundare

Odată disponibil un dicționar, trăsăturile secundare se obțin prin codarea rară a vectorilor de trăsături primare în raport cu dicționarul respectiv. Reamintim faptul că dintr-o înregistrare ECG se obține o matrice de trăsături primare, conform Fig.2.2. Fiecare vector de trăsături primare (fiecare coloană) va fi descompus într-o reprezentare rară în raport cu dicționarul. Trăsăturile secundare sunt reprezentate de vectorii de reprezentare rară obținuți în urma codării, după cum este prezentat în Fig.2.3

Algoritmii de codare rară utilizați sunt esențiali în găsirea atomilor care compun trăsăturile primare, rolul lor fiind critic în algoritm. În acest sens, în secțiunea 3 prezentăm doi algoritmi de codare rară cu performanțe îmbunătățite față de algoritmii *state-of-the-art* din literatura de specialitate. Rezultatele evaluărilor acestor algoritmi pentru extragerea de trăsături secundare în problema clasificării patologiilor de semnal ECG sunt în curs de finalizare.

2.1.3.4 Histograma de tip “Bag-of-Words”

Odată finalizată extragerea trăsăturilor secundare din cele primare, prin codare rară, se va obține o matrice de trăsături secundare pentru fiecare înregistrare ECG analizată. Pentru a putea fi utilizate într-o etapă de clasificare, trăsăturile secundare sunt supuse operației de “pooling” (comasare)¹.

Operația de “pooling” are următoarele obiective:

¹Vom prefera utilizarea termenului englezesc “pooling”, încetățenit în literatura de specialitate de limba engleză, în locul traducerii în limba română sub forma “grupare”.

- reducerea dimensionalității matricii de trăsături secundare, care poate avea o dimensiune prohibitivă (egală cu dimensiunea dicționarului \times numărul de ferestre cu care a fost analizat întregul semnal)
- obținerea unui vector final de dimensiune fixă, indiferent de lungimea înregistrării ECG analizate (matricile de trăsături secundare pot avea lungimi diferite, dacă semnalele ECG analizate au lungimi diferite, întrucât numărul de ferestre necesare pentru analiza unui semnal depinde de lungimea acestuia)
- păstrarea informației relevante din ansamblul trăsăturilor secundare, pentru a se putea face o clasificare ulterioară.

În această lucrare, optăm în principal pentru o metodă de tip “sum-pooling”, în care toți vectorii de trăsături secundare se adună, în modul, pentru a produce un vector final de dimensiune fixă. Practic, matricea de trăsături secundare în valori absolute se adună “pe linii”, rezultatul fiind un vector de dimensiune fixă, egală cu dimensiunea dicționarului, care sumarizează gradul de utilizare al fiecărui atom din dicționar în semnalul respectiv.

Vectorul obținut astfel are semnificația unei histogramme, care ilustrează importanța și gradul de utilizare al atomilor din dicționar în reprezentarea unui semnal ECG. Dacă un atom apare în multe trăsături secundare cu coeficienți de valoare semnificativă, atunci coeficientul său din histograma finală va avea o valoare ridicată. Dimpotrivă, un atom care nu este utilizat decât arareori în trăsăturile secundare va avea un coeficient redus în histograma finală.

Histograma obținută astfel se înscrie în logica tehnicilor tip “Bag-of-Words”, care se bazează pe astfel de histogramme pentru a sintetiza frecvența de apariție a diverselor cuvinte / trăsături. În metoda descrisă aici, “cuvintele” reprezintă atomii dicționarului, iar histograma obținută sintetizează frecvența și importanța fiecărui atom în întreaga înregistrare ECG.

Menționăm că în literatura de specialitate sunt utilizate și alte metode de “pooling”, de exemplu metoda “max-pooling” utilizată frecvent în arhitecturi de rețele neurale convoluționale, prin care reține doar coeficientul de valoare maximă (în cazul nostru, coeficientul de valoare absolută maximă de pe liniile matricii de trăsături secundare). Totuși, în aplicația propusă aici soluția “max-pooling” ar crește sensibilitatea la zgomot, care este un considerent important în ceea ce privește semnalele biomedicale. O porțiune atipică de semnal ECG ar risca în acest caz să producă o reprezentare rară cu coeficienți mari în dreptul unor atomi care nu sunt relevanți pentru restul semnalului, iar tehnica “max-pooling” ar prelua aceste valori maxime, afectând rezultatul final.

2.1.3.5 Antrenarea unui clasificator SVM

Ultima etapă din lanțul de procesare a datelor în vederea clasificării patologiilor o reprezintă utilizarea unui clasificator pentru a clasifica histograma semnalului analizat într-una din categoriile

considerate.

În acest sens, utilizăm un clasificator clasic de tip SVM [9], care este antrenat pe histogramele înregistrărilor ECG din setul de antrenare. Alte soluții posibile includ un clasificator de tip rețea neurală, dar întrucât dimensiunea setului de date avut la dispoziție este redus am evitat utilizarea rețelelor neurale, cunoscute ca necesitând baze de date de dimensiune mare pentru a fi antrenate eficient.

Variantele de SVM utilizate includ SVM fără *kernel*, precum și variante cu *kernel* de tip intersecție, respectiv χ^2 ("chi pătrat").

2.1.4 Baza de date "PhysioNet Computing in Cardiology Challenge 2017"

Pentru a evalua metoda descrisă mai sus, utilizăm semnalele ECG disponibile în cadrul competiției "PhysioNet Computing in Cardiology Challenge 2017" [10], [11], care are ca obiectiv clasificarea fibrilațiilor atriale din semnale ECG cu un canal, de lungime redusă.

În cadrul competiției "PhysioNet Computing in Cardiology Challenge 2017" au fost puse la dispoziția comunității științifice un număr de peste 10000 de înregistrări ECG achiziționate direct de la utilizatori (nu într-un cadru medical), de lungime scurtă (între 30 și 60 secunde, în general) care au fost adnotate de către cadre medicale și clasificate în patru categorii:

1. Normal: semnale cu aspect normal (ritm normal);
2. Fibrilație Atrială: semnale care prezintă fibrilație atrială;
3. Altele: semnale cu alt tip de ritm cardiac (alte categorii de fibrilații, de ex.fibrilație ventriculară);
4. Zgomot: semnale prea afectate de zgomot pentru a putea fi clasificate.

Obiectivul principal este de a detecta fibrilația atrială, o afecțiune comună a inimii care afectează aproximativ 1-2 % din populația lumii, asociată cu riscuri ridicate de activitate cardiacă, pornind de la semnale ECG de lungime mică care pot fi achiziționate cu dispozitive mobile, direct de către utilizatori. Fibrilația atrială se evidențiază în semnalul ECG prin anumite caracteristici, printre care absența undei P sau prezența unor componente oscilante în segmentul TQ.

Un exemplu de înregistrări ECG din fiecare categorie este prezentată în Fig. 2.4 [11].

2.1.5 Rezultate de simulare

În continuare prezentăm rezultatele inițiale obținute aplicând metoda propusă mai sus. Menționăm că rezultatele nu sunt finale, fiind în curs de îmbunătățire până a se ajunge aproape de nivelul rezultatelor *state-of-the-art* în domeniu.

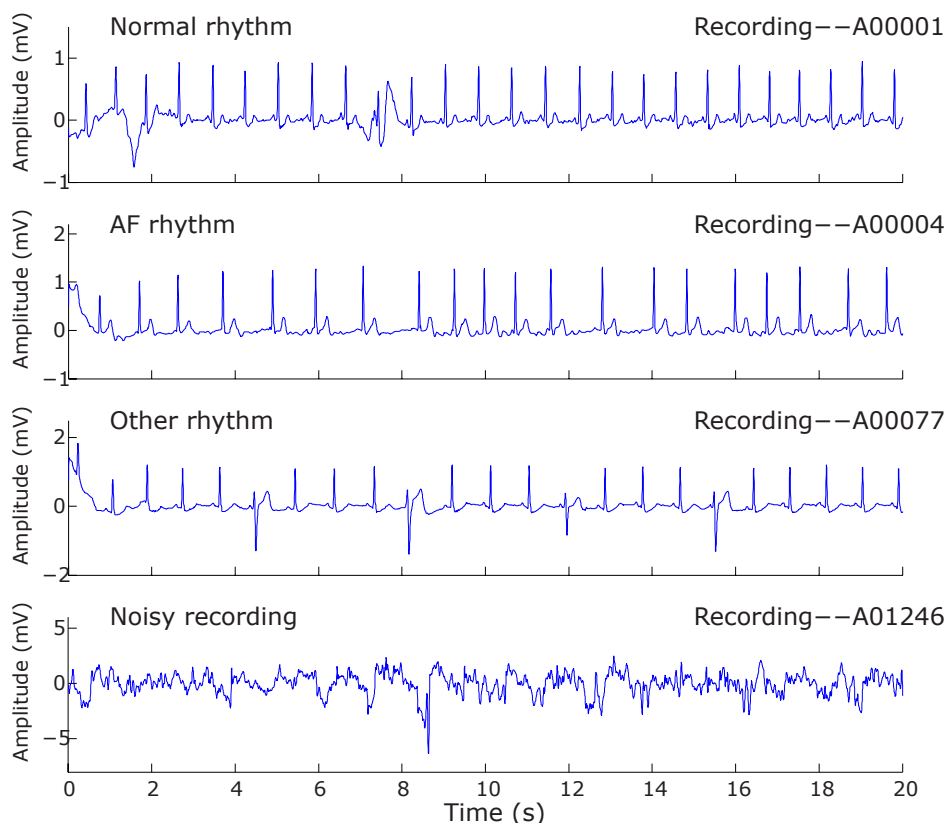


Figure 2.4: Exemplu de semnal ECG din fiecare clasă de semnal din setul de date "PhysioNet Computing in Cardiology Challenge 2017" (imagine din [CinC2017Web])

Pentru evaluarea metodei, folosim cele 8527 înregistrări ECG disponibile în setul de antrenare pus la dispoziție de baza de date "PhysioNet Computing in Cardiology Challenge 2017". Întrucât numărul de semnale disponibile din fiecare categorie variază, cel mai mic număr fiind al semnalelor din clasa Zgomot, cu 284 de semnale, păstrăm același număr de 284 semnale și din restul claselor, alese în mod aleator. Construim astfel un set de date echilibrat, pe care îl împărțim apoi într-un set de antrenare a dicționarului și a clasificatorului SVM, compus din 184 de semnale din fiecare categorie, și un set de evaluare a rezultatelor, compuse din 100 de semnale din fiecare categorie.

Din semnalele astfel selectate sunt extrase trăsăturile primare, din ferestre de lungime între 128 și 512. După extragerea trăsăturilor primare, se antrenează un dicționar folosind toate trăsăturile primare ale tuturor semnalelor ECG folosite la antrenare. Numărul de atomi ai dicționarului poate fi în plaja de valori 1000 - 10000, o valoare mai mare ducând la creșterea rapidă a timpului necesar procesului de antrenare, dar și a dimensiunii trăsăturilor secundare și deci a procesului de clasificare ulterior. Trăsăturile secundare se obțin apoi prin codarea rară a trăsăturilor primare în raport cu dicționarul.

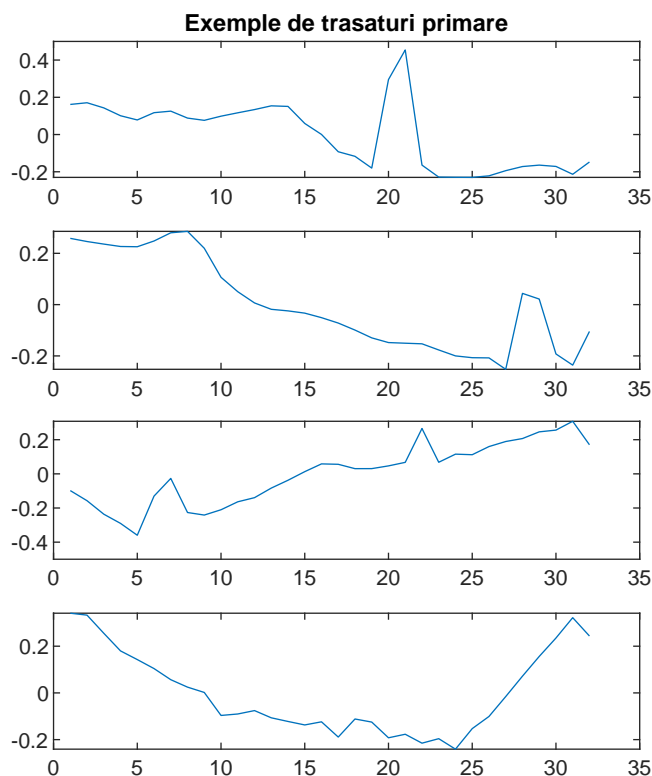


Figure 2.5: Exemplu de trăsături primare de lungime 32, extrase din câte fereastră provenind din patru semnale ECG diferite.

Un exemplu de trăsături primare de lungime 32 coeficienți, extrase din ferestre de lungime 256, este prezentat în Fig.2.5.

Un exemplu de atomi dintr-un dicționar cu 2000 de atomi este prezentat Fig.2.6.

Un exemplu de histogramme ale semnalelor ECG, obținute prin operația de “pooling” (sumare în valori absolute) aplicată trăsăturilor secundare, este prezentat Fig.2.7.

După antrenare, algoritmul este evaluat prin clasificarea semnalelor ECG din setul de test, comparând clasa prezisă de algoritm cu cea corectă a semnalelor. Evaluarea este repetată de un număr de ori (de exemplu de 10 ori) și rezultatele sunt mediate, pentru a evita un caz excepțional cauzat de o împărțire favorabilă a semnalelor în setul de antrenare și de test. Rezultatele evaluării nu sunt finale, și depind de parametrii tuturor etapelor de procesare a datelor (dimensiunea ferestrei alunecătoare, dimensiunea și tipul trăsăturilor primare, numărul de atomi ai dicționarului, parametrii algoritmului de antrenare a dicționarului, parametrii și algoritmul de codare rară, tipul de clasificator SVM folosit, etc.)

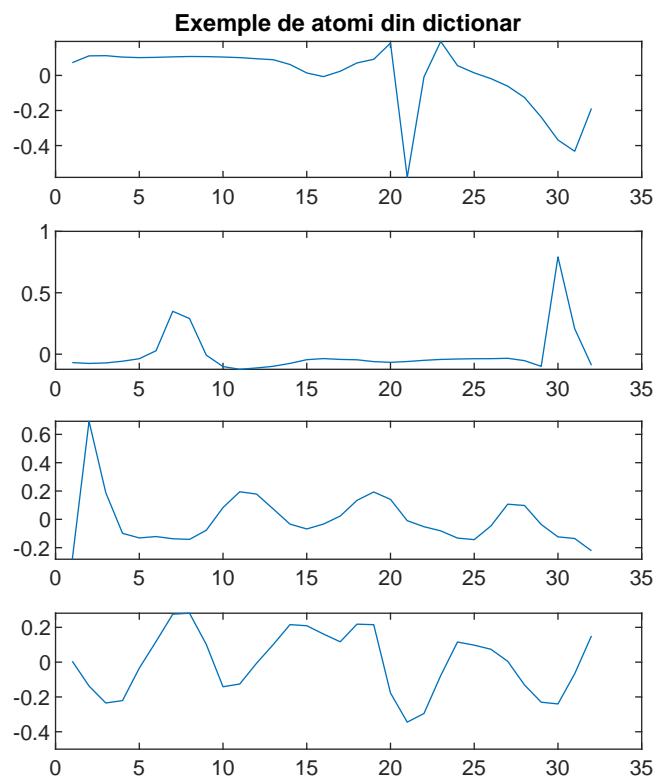


Figure 2.6: Exemplu de atomi de lungime 32, obținuți prin antrenarea unui dicționar din trăsăturile primare extrase din întreg detul de antrenare utilizat

În prezent, obținem în mod curent o rată de clasificare corectă în jurul valorii de 60 %, dar aceasta trebuie îmbunătățită până în jurul valorilor de 75 - 80 %, care sunt obținute cu algoritmi cei mai buni la momentul actual. Printre ideile considerate în momentul de față pentru îmbunătățirea rezultatelor se numără:

- filtrarea semnalelor utilizate în antrenare, pentru a elimina segmentele afectate de zgomot (“outliers”) care pot afecta procesul de antrenare;
- algoritmi diferiți antrenare a dicționarului, precum și tehnici îmbunătățite (de exemplu, antrenarea unui dicționar pentru fiecare clasă de semnal în parte [12]);
- alegerea unor trăsături primare mai adecvate semnalelor ECG și particularităților acestora.

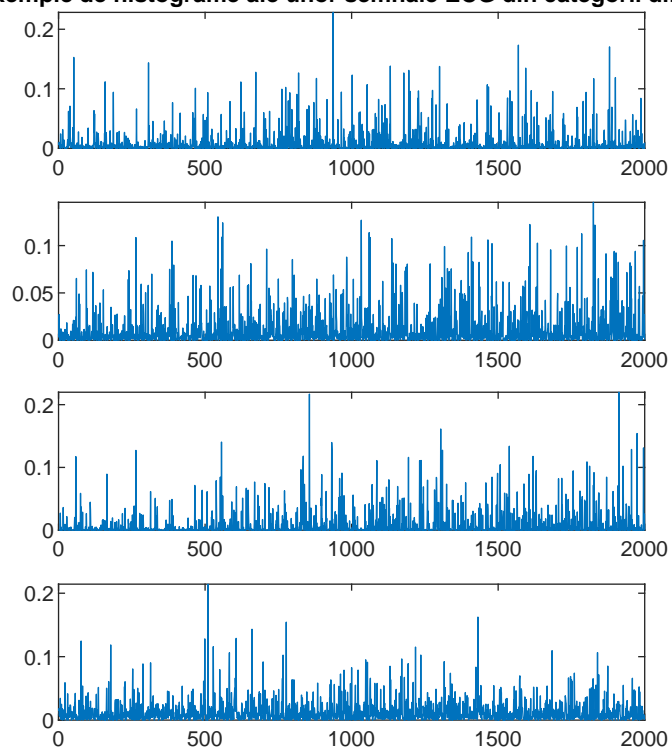
Exemple de histograme ale unor semnale ECG din categorii diferite

Figure 2.7: Exemplu de histograme obținute pentru patru semnale ECG diferite, câte unul din fiecare clasă de semnal. Aceste histograme reprezintă intrarea în clasificatorul SVM, care va prezice apoi clasa semnalului evaluat.

3 Metode îmbunătățite de codare rară, cu aplicații în codarea semnalelor de tip ECG

3.1 Metoda proiecțiilor afine iterative (Iterative Affine Projections)

3.1.1 Introducere

Codarea rară a semnalelor [13], [14] este o problemă de optimizare binecunoscută și studiată amănunțit în ultimele decenii. Dat fiind un semnal rar $\mathbf{x} \in \mathbb{R}^n$, $\|\mathbf{x}\|_0 = s$, obiectivul este de a recupera semnalul dintr-un număr redus de m măsurători liniare, exprimate sub forma multiplicării cu o matrice de achiziție $A \in \mathbb{R}^{m \times n}$, cu rezultatul posibil afectat de zgomot aditiv \mathbf{e} :

$$\mathbf{y} = A\mathbf{x} + \mathbf{e}$$

Semnalul \mathbf{x} poate fi recuperat din vectorul de măsurători \mathbf{y} de lungime mai mică, în general, printr-o problemă de optimizare care urmărește găsirea celei mai rare soluții a sistemului de ecuații implicat de procesul de achiziție:

$$\mathbf{x} = \arg \min_x \|\mathbf{x}\|_p \text{ cu constrângerea } \|\mathbf{y} - A\mathbf{x}\|_2 \leq \epsilon.$$

Problema de reconstrucție poate fi formulată sub diverse forme, toate exprimând obiectivul fundamental de a găsi cea mai rară reprezentare a vectorului \mathbf{y} în raport cu dicționarul determinat de matricea A . De obicei, norma ℓ_0 sau ℓ_1 sunt utilizate pentru a cuantifica raritatea vectorului soluție.

În cazul unui scenariu de achiziție fără zgomot, constrângerea de tip pătratic

$$\|\mathbf{y} - A\mathbf{x}\|_2 \leq \epsilon$$

degenerează într-o constrângere exactă

$$\mathbf{y} = A\mathbf{x}.$$

Dacă se utilizează norma ℓ_0 pentru cuantificarea rarității soluției, problema de optimizare devine:

$$\mathbf{x} = \arg \min_x \|\mathbf{x}\|_0 \text{ cu constrângerea } \mathbf{y} = A\mathbf{x}. \quad (3.1)$$

Problema reconstrucției semnalului se pune în același mod și în cazul în care semnalul \mathbf{x} nu este el însuși rar (cu alte cuvinte, nu este rar în baza canonică), dar are în schimb o reprezentare rară γ într-o altă bază sau într-un dicționar supracomplet D cunoscute, cu alte cuvinte $\mathbf{x} \approx D\gamma$. caz în care avem $\mathbf{y} \approx AD\gamma$. În această situație, se găsește mai întâi descompunerea rară γ ca soluție a problemei de optimizare echivalente, în care produsul $A_{ef} = A \cdot D$ reprezintă un dicționar efectiv, sau o matrice echivalentă de achiziție pentru descompunerea rară γ . Semnalul \mathbf{x} va fi găsit apoi într-o a doua etapă, din vectorul de reprezentare γ , prin înmulțire cu D . Întrucât problema de optimizare este în esență aceeași, în cele ce urmează considerăm ambele cazuri ca fiind acoperite de ecuația (3.1), unde \mathbf{x} reprezintă semnalul rar inițial și A reprezintă dicționarul considerat, fie cel inițial fie cel echivalent, în funcție de natura problemei.

În general, reconstrucția semnalului are succes doar dacă coloanele matricii A (atomii) satisfac o anume proprietate de incoerență [15]. Proprietatea de izometrie restrânsă (Restricted Isometry Property, RIP) [16] este cea mai utilizată metodă din literatură pentru a caracteriza aceste condiții. Spunem că matricea A satisface proprietatea RIP de ordin s cu constanta δ_s dacă, pentru toți vectorii x cu raritatea cel mult s , este satisfăcută relația următoare:

$$(1 - \delta_s)\|x\|_2^2 \leq \|A\mathbf{x}\|_2^2 \leq (1 + \delta_s)\|x\|_2^2$$

Această relație poate fi exprimată alternativ ca o limită impusă normei spectrale a tuturor submatricilor A_s obținute prin selectarea a doar s coloane din A [17]:

$$\|I - A_s^T A_s\|_{2 \rightarrow 2} \leq \delta_s,$$

unde notația $\|\cdot\|_{2 \rightarrow 2}$ desemnează norma spectrală a unei matrici. Pentru o matrice cu o constantă RIP de valoare mică, toate submatricile formate din s coloane sunt aproape de a fi ortonormale, ceea ce face ca norma termenului $\|A\mathbf{x}\|_2$ să fie apropiată de cea a lui \mathbf{x} . O valoare suficient de mică a constantei RIP a dicționarului reprezintă o condiție suficientă pentru a garanta succesul majorității algoritmilor de reconstrucție din literatură [17].

În literatură au fost propuse numeroase metode pentru rezolvarea ecuației (3.1). O importanță semnificativă în contextul acestei lucrări o vom acorda algoritmului *Iterative Hard Thresholding* (IHT) [18]. Algoritmul IHT se bazează pe repetarea următoarei operații:

$$\mathbf{x}^{k+1} = H_s(\mathbf{x}^k + \mu \cdot A^T(x - A\mathbf{x}^k)), \quad (3.2)$$

unde $H_s(\cdot)$ denotă operatorul de “*hard thresholding*,”¹ care păstrează doar cele mai mari s valori absolute ale unui vector, transformând toate celelalte în zero.

La o analiză mai atentă, iterația (3.2) poate fi descompusă în două operații succesive, constând în (i) actualizarea soluției urmată de (ii) o etapă de prăguire (“thresholding”). Ecuația de actualizare a soluției

$$\mathbf{v}^k = \mathbf{x}^k + \mu \cdot A^T(x - A\mathbf{x}^k) \quad (3.3)$$

reprezintă în fapt o actualizare de tip coborâre după gradient (Gradient Descent) de dimensiune μ , pornind de la soluția curentă \mathbf{x}^k și orientat înspre minimizarea termenului de eroare $\|y - A\mathbf{x}^k\|_2^2$. Vectorul \mathbf{v}^k obținut astfel este apoi supus operației de “prăguire”, ceea ce conduce la o soluție tot de raritate egală cu s dar având eroarea de reprezentare potențial mai mică. Procesul este apoi repetat până la convergență.

Un dezavantaj cunoscut al algoritmului IHT îl reprezintă sensibilitatea la valoarea pasului de actualizare μ , alegerea unei valori potrivite fiind necesară pentru a se asigura convergența algoritmului. De asemenea, metode adaptive, dar mai complexe, de alegere a pasului μ au fost propuse în literatură [19].

În cele ce urmează, prezentăm un nou algoritm de codare rară, cu rezultate îmbunătățite comparativ cu algoritmul IHT.

3.1.2 Algoritm Iterative Hard Thresholding

Metoda propusă pornește de la observația că în algoritmul IHT operatorul de prăguire (“hard thresholding”) $H_s(\cdot)$ are un dublu rol:

1. Prin această operație se găsește suportul prezumtiv al soluției rare de la iterația curentă, notat în continuare T^k , care conține pozițiile celor mai mari s valori în valoare absolută ale vectorului \mathbf{v}^k .
2. Prin “hard thresholding” are loc proiecția ortogonală a vectorului \mathbf{v}^k pe subspațiul I_{T^k} generat de vectorii bazei canonice corespunzători suportului prezumtiv aflat mai sus, T^k .

Cele două operații au roluri distincte. Actualizarea suportului prezumtiv al reprezentării rare în cadrul unei iterații este o operație necesară la majoritatea algoritmilor de codare rară, cum ar fi de exemplu algoritmul Orthogonal Matching Pursuit (OMP), diferența constând mai degrabă în modul cum se face selecția propriu-zisă a atomilor din suport. De exemplu, la OMP, la fiecare

¹Preferăm să lăsam anumiți termeni tehnici din limba engleză în original, atunci când nu există un termen echivalent consacrat în limba română, sau când traducerea termenului nu poate păstra semnificația originală.

De exemplu, în acest caz, termenul de “*hard thresholding*” s-ar putea traduce prin *prăguire rigidă*, dar considerăm că traducerea ar fi ușor forțată și, în fapt, nu clarifică sensul expresiei mai mult decât termenul original din limba engleză, netradus.

atom se adaugă un nou atom la suportul deja existent, în timp ce la IHT întreg suportul este reestimat la fiecare iterație prin selecția celor mai mari valori absolute din \mathbf{v}^k .

În schimb, proiecția ortogonală pe care o realizează operația de “hard-thresholding” derivă din cu totul alte considerente. În general, odată avut un suport prezumtiv T^k al soluției de la pasul k , calitatea unei reprezentări \mathbf{x}^k este determinată de doi termeni de eroare:

$$E(\mathbf{x}^k) = \|\mathbf{y} - A\mathbf{x}^k\|_2^2 + \lambda \|\mathbf{x}_{T_c^k}^k\|_2^2 \quad (3.4)$$

Primul dintre aceștia îl reprezintă eroarea de reprezentare $\|\mathbf{y} - A\mathbf{x}^k\|_2^2$, care măsoară cât de precis este reconstruit semnalul y prin reprezentarea \mathbf{x}^k . Al doilea termen cuantifică raritatea reprezentării, măsurând cât anume din reprezentarea \mathbf{x}^k se extinde în afara suportului prezumtiv. Mulțimea T_c^k din (3.4) reprezintă *co-suportul* semnalului, adică setul de atomi care nu aparțin suportului T^k , iar notația $\mathbf{x}_{T_c^k}^k$ reprezintă vectorul \mathbf{x}^k restrâns la elementele sale care corespund acestui co-suport. În cele ce urmează, ne vom referi la acești doi termeni ca fiind “eroarea de reprezentare” și “eroarea de raritate” ale reprezentării \mathbf{x}^k . Factorul λ reprezintă importanța relativă atribuită celor doi termeni de eroare în procesul de reconstrucție. Subliniem faptul că o reprezentare exactă \mathbf{x}^* a semnalului y , dacă există, ar avea ambii termeni de eroare egali cu zero. În acest sens, algoritmul IHT urmărește obținerea unei soluții cu eroarea totală cât mai mică, ceea ce înseamnă o reprezentare mai bună.

În algoritmul IHT, operația de “hard thresholding” forțează eroarea de raritate la valoarea 0, prin reținerea doar a primilor s coeficienți în valoare absolută, ceea ce înseamnă de fapt o proiecție ortogonală pe subspațiul canonic generat de suportul T^k . În schimb, eroarea de reprezentare este tolerată, fiind utilizată pentru actualizarea soluției printr-un pas de tip “coborâre după gradient”. Astfel, gradientul erorii de reprezentare $\|\mathbf{y} - A\mathbf{x}^k\|_2^2$ este:

$$\mathbf{g} = \frac{d}{d\mathbf{x}^k} \left(\|\mathbf{y} - A\mathbf{x}^k\|_2^2 \right) = -2 \cdot A^T \cdot (\mathbf{y} - A\mathbf{x}^k),$$

care reprezintă exact termenul de actualizare a soluției din (3.3).

În lumina celor prezentate mai sus, algoritmul IHT poate fi înțeles ca o succesiune de trei operații: (i) reducerea erorii de reprezentare printr-un pas de tip coborâre după gradient, (ii) selecția suportului prezumtiv prin alegerea celor mai mare s valori absolute și (iii) eliminarea erorii de raritate prin proiecție ortogonală. Algoritmul IHT astfel descris este rezumat ca Algoritmul 1.

3.1.3 Algoritmul proiecțiilor afine iterative (Iterative Affine Projection)

Algoritmul propus are la baza ideea de a schimba rolul celor doi termeni de eroare din IHT. În loc de a elimina eroarea de raritate la fiecare iterație, tolerând eroarea de reprezentare, vom proceda în schimb la eliminarea erorii de reprezentare, $\|\mathbf{y} - A\mathbf{x}^k\|_2^2 = 0$, tolerând eroarea de

Algorithm 1 Iterative Hard Thresholding (IHT)

Intrare: A = matricea sistemului (dicționarul)**Intrare:** y = semnalul de intrare

- 1: $k \leftarrow 0$
- 2: $\mathbf{x}^0 \leftarrow 0\mathbf{y}$
- 3: **repetă**
- 4: Coborâre după gradient pentru reducerea erorii de reprezentare:

$$\mathbf{v}^k = \mathbf{x}^k + \lambda \cdot A^T(\mathbf{y} - A\mathbf{x}^k)$$

- 5: Selecția suportului și proiecție ortogonală pentru eliminarea erorii de raritate:

$$\mathbf{x}^{k+1} = H_s(\mathbf{v}^k)$$

- 6: **până când** soluția converge

- 7: **returnează** soluția \mathbf{x}^{k+1}
-

raritate. Eliminarea erorii de reprezentare la fiecare iterație se face tot prin proiecție ortogonală, dar de această dată proiecția se face pe planul afin care conține soluțiilor ecuației $\mathbf{y} = A\mathbf{x}$. Se va obține astfel o reprezentare exactă, dar care nu este rară, având o eroare de raritate $\|\mathbf{x}_{T_c^k}^k\|_2^2$ nenulă. Aceasta va fi redusă la fiecare iterație printr-un pas de tip coborâre după gradient.

Se observă astfel că algoritmul propus conține în aceleași operații ca și IHT: (i) selecția suportului, (ii) coborâre după gradient și (iii) proiecție ortogonală, dar acestea se aplică termenilor de eroare inversați. Prin analogie cu IHT, algoritmul este denumit algoritmul proiecțiilor afine iterative (Iterative Affine Projections) (IAP). Descrierea sa este furnizată sub forma Algoritmului 2.

Fiecare pas al algoritmului IAP este descris în cele ce urmează:

1. **Coborâre după gradient.** Soluția \mathbf{x}^k de la începutul fiecărei iterații satisface condiția de reprezentare în mod exact (are eroare de reprezentare nulă), motiv pentru care trebuie să aibă o eroare de raritate nulă, respectiv nu formează un vector de raritate s (în caz contrar, dacă ambii termeni de eroare sunt nuli, înseamnă că s-a ajuns deja la soluția finală). Din acest punct se realizează un pas de actualizare de tip coborâre după gradient, de lungime μ , pentru reducerea erorii de raritate $\|\mathbf{x}_{T_c^k}^k\|_2^2$. Întrucât gradientul termenului $\|\mathbf{x}_{T_c^k}^k\|_2^2$ este $2\mathbf{x}_{T_c^k}^k$, se obține astfel ecuația:

$$\mathbf{v}^k \leftarrow \mathbf{x}^k - \mu \cdot \mathbf{x}_{T_c^k}^k.$$

Întrucât eroarea de raritate este determinată de cele mai mici $(N - s)$ valori din \mathbf{x}^k , prin această operație acestea sunt reduse suplimentar cu un factor μ , în timp ce cele mai mari s valori absolute rămân nemodificate. Se observă faptul că pentru $\mu = 1$ se obține exact operația de “hard thresholding” asupra vectorului \mathbf{x}^k , dar subliniem că valoarea lui μ poate

Algorithm 2 Iterative Affine Projection (IAP)**Intrare:** A = matricea sistemului (dicționarul)**Intrare:** N = bază ortonormală a spațiului nul al matricii A **Intrare:** y = semnalul de intrare1: $k \leftarrow 0$ 2: $\mathbf{x}^0 \leftarrow A^\dagger \mathbf{y}$ 3: **repetă**

4: Gradient descent step (shrink) to reduce sparsity error

$$\mathbf{v}^k \leftarrow \mathbf{x}^k - \mu \cdot \mathbf{x}_{T_c^k}^k$$

5: Project back on affine solution space

$$\mathbf{x}^{k+1} \leftarrow \mathbf{x}^0 + N^T N \mathbf{v}^k$$

6: **până când** soluția converge7: **returnează** soluția \mathbf{x}^{k+1}

diferi, întrucât operația nu trebuie înțeleasă ca o proiecție aleatoare ci ca o coborâre după gradient.

2. **Proiecție ortogonală.** A doua operație a algoritmului are rolul de a aduce vectorul \mathbf{v}^k înapoi în spațiul afin al soluțiilor, prin proiecție ortogonală:

$$\mathbf{x}^{k+1} \leftarrow \mathbf{x}^0 + N^T N \mathbf{v}^k,$$

Aici, $\mathbf{x}^0 = A^\dagger \mathbf{y}$ reprezintă soluția de energie minimă (“least-squares”) a sistemului, N desemnează o matrice ale cărei linii formează o bază a spațiului nul al matricii A , iar notația † reprezintă pseudo-inversa Moore-Penrose a unei matrici. Pentru simplitate, considerăm liniile matricii B ca fiind ortonormale, astfel încât termenul $N^T N$ reprezintă operatorul de proiecție ortonormală pe spațiul nul al matricii A . Reamintim că spațiul soluțiilor ecuației $\mathbf{y} = A\mathbf{x}$ este un spațiu afin, elementele sale fiind suma dintre soluția de energie minimă \mathbf{x}^0 și elementele din spațiul nul.

Înlocuind \mathbf{v}^k cu relația sa de definiție conduce la o formulare a algoritmului constând dintr-o singură relație, care acoperă ambii pași prezentați anterior:

$$\begin{aligned} \mathbf{x}^{k+1} &\leftarrow \mathbf{x}^0 + N^T N (\mathbf{x}^k - \mu \cdot \mathbf{x}_{T_c^k}^k) \\ &\leftarrow \mathbf{x}^k - \mu \cdot N^T N \mathbf{x}_{T_c^k}^k. \end{aligned} \tag{3.5}$$

S-a folosit aici faptul că vectorul \mathbf{x}^k aparține el însuși spațiului afin al soluțiilor, astfel încât $\mathbf{x}^k = \mathbf{x}^0 + N^T N \mathbf{x}^k$.

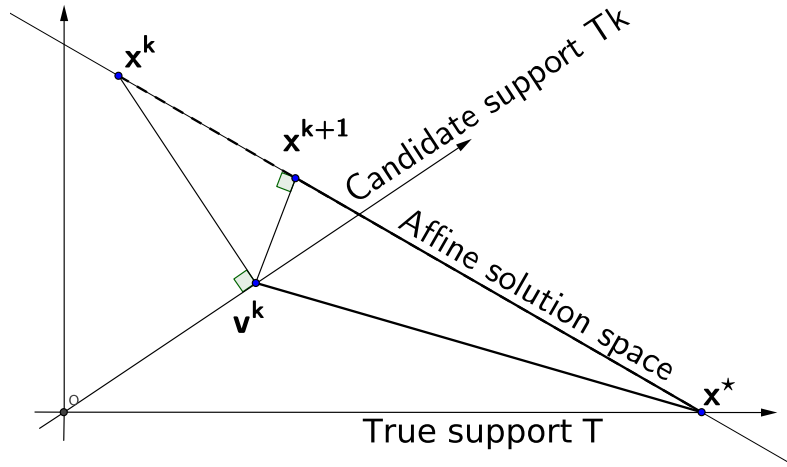


Figure 3.1: Reprezentare grafică a unei iterații a algoritmului IAP, pentru $\mu = 1$, într-un spațiu 3D. Pasul 1: Soluția \mathbf{x}^k este supusă unui operații de "hard thresholding", obținându-se \mathbf{v}^k . Pasul 2: \mathbf{x}^k suferă o proiecție ortogonală pe spațiul afin al soluțiilor, obținându-se următoarea soluție

O reprezentare grafică de principiu a unei iterații a algoritmului IAP este prezentată în Fig.3.1, pentru cazul când pasul $\mu = 1$. Soluția \mathbf{x}^k la iterația k aparține spațiului afin al soluțiilor ecuației $\mathbf{y} = \mathbf{A}\mathbf{x}$. Prin reducerea componentelor corespunzătoare cosuportului se tinde către subspațiul corespunzător suportului prezumtiv T^k , care poate fi diferit de suportul adevărat T . În această figură, $\mu = 1$ ceea ce face ca rezultatul \mathbf{v}^k să fie exact de raritate s , fiind așadar situat în subspațiul generat de suportul T^k , dar pentru o valoare diferită a pasului μ vectorul \mathbf{v}^k poate fi în afara acestui subspațiu. Soluția următoare \mathbf{x}^{k+1} este proiecția lui \mathbf{v}^k pe subspațiul afin al soluțiilor, avansând către soluția exactă \mathbf{x}^* , care se găsește atât în spațiul afin, dar este și de raritate s , fiind așadar la intersecția dintre spațiul afin și un subspațiu corespunzător suportului real de dimensiune s .

Subliniem faptul că fiecare iterație a algoritmului IAP pornește de la o reprezentare \mathbf{x}^k care se găsește în spațiul afin al soluțiilor ecuației $\mathbf{y} = \mathbf{A}\mathbf{x}$, și conduce la o altă reprezentare \mathbf{x}^{k+1} îmbunătățită, care face parte din același spațiu afin. Așadar, algoritmul avansează către soluția finală croindu-și drum prin subspațiul afin, toate reprezentările din secvența de soluții generată de algoritm satisfacând ecuația $\mathbf{y} = \mathbf{A}\mathbf{x}$.

3.1.4 Garanții teoretice pentru succesul algoritmului IAP în cazul reconstrucției fără zgomot

Fie un semnal \mathbf{y} care are o reprezentare exactă de raritate s , $\mathbf{y} = \mathbf{A}\mathbf{x}^*$, $\|\mathbf{x}^*\|_0 = s$. În cele ce urmează vom demonstra o condiție teoretică care garantează găsirea corectă a reprezentării \mathbf{x}^* prin algoritmul IAP, în cazul fără zgomot, bazată pe constanta RIP de ordin $2s$. Particularitatea

esențială este că nu e vorba de constanta nu este a matricii dicționarului, cum se întâmplă de obicei la algoritmi din această familie (inclusiv la IHT), ci a matricii singulare dreaptă.

În această demonstrație, considerăm valoarea pasului de coborâre după gradient ca fiind egală cu $\mu = 1$. Demonstrația este inspirată din analiza excelentă din [17].

Teorema 1. *Fie \mathbf{x}^* un vector necunoscut de dimensiune n , și de raritate egală cu s , $\|\mathbf{x}^*\|_0 = s$. Fie A o matrice de dimensiune $m \times n$, $m \leq n$, astfel încât*

$$\mathbf{y} = A\mathbf{x}.$$

Considerăm o descompunere SVD a matricii A de forma $A = USV^T$, unde V^T este în forma redusă (are dimensiunea $m \times n$).

Dacă constanta RIP de ordin $2s$ a matricii V^T satisface condiția:

$$\delta_{2s}^{V^T} < \frac{1}{2}, \quad (3.6)$$

atunci algoritmul IAP converge linear către soluția adevărată \mathbf{x}^ .*

Demonstrație. Fie soluția prezumtivă de la iterația k notată cu \mathbf{x}^k , vectorul obținut din acesta prin coborâre după gradient către un vector de raritate s , \mathbf{v}^k , următoarea soluție a algoritmului de la pasul $(k + 1)$, \mathbf{x}^{k+1} , și soluția adevărată \mathbf{x}^* . Facem referire la reprezentarea grafică din Fig. 3.1 pentru ilustrarea acestor vectori.

Demonstrația se bazează pe compararea lungimilor următorilor vectori: (i) $\mathbf{x}^{k+1} - \mathbf{x}^*$, (ii) $\mathbf{v}^k - \mathbf{x}^*$, și (iii) $\mathbf{x}^k - \mathbf{x}^*$. În particular, ne concentrăm pe demonstrarea inegalităților

$$\|\mathbf{x}^{k+1} - \mathbf{x}^*\|_2 < \|\mathbf{v}^k - \mathbf{x}^*\|_2 < \|\mathbf{x}^k - \mathbf{x}^*\|_2.$$

Prima inegalitate rezultă din observația că diferența $\mathbf{x}^* - \mathbf{x}^{k+1}$ reprezintă proiecția vectorului diferență $\mathbf{x}^* - \mathbf{v}^k$ pe spațiul nul, din moment ce \mathbf{x}^{k+1} este proiecția ortogonală a lui \mathbf{v}^k pe spațiul afin al soluțiilor, iar \mathbf{x}^* aparține aceluiași spațiu afin. Avem așadar

$$\begin{aligned} \|\mathbf{x}^{k+1} - \mathbf{x}^*\|_2 &= \|N^T N(\mathbf{v}^k - \mathbf{x}^*)\|_2 \\ &= \|(I - A^\dagger A)(\mathbf{v}^k - \mathbf{x}^*)\|_2, \end{aligned}$$

unde am rescris operatorul de proiecție ortogonală pe spațiul nul $N^T N$ sub forma mai convenabilă $(I - A^\dagger A)$.

Considerând descompunerea SVD a matricii A de forma $A = USV^T$, putem rescrie $A^\dagger A = VV^T$. Observăm în continuare faptul că ambii vectori care apar în partea dreaptă, \mathbf{v}^k și \mathbf{x}^* , au raritatea s . Dacă definim reuniunea dintre suporturile celor doi vectori ca fiind $L = \text{supp}(\mathbf{x}^k) \cup \text{supp}(\mathbf{x}^*)$,

$|L| \leq 2s$, în partea dreaptă a ecuației putem restrânge matricea V la coloanele corespunzătoare setului L , după cum urmează:

$$\|\mathbf{x}^{k+1} - \mathbf{x}^*\|_2 = \|(I - V_L V_L^T)(\mathbf{v}^k - \mathbf{x}^*)\|_2,$$

Folosind relația de echivalență $\delta_{|L|}^{V^T} = \|(I - V_L V_L^T)\|_{2 \rightarrow 2}$, putem rescrie ecuația sub forma:

$$\begin{aligned} \|\mathbf{x}^{k+1} - \mathbf{x}^*\|_2 &= \|(I - V_L V_L^T)(\mathbf{v}^k - \mathbf{x}^*)\|_2 \\ &\leq \|(I - V_L V_L^T)\|_{2 \rightarrow 2} \|\mathbf{v}^k - \mathbf{x}^*\|_2 \\ &\leq \delta_{2s}^{V^T} \|\mathbf{v}^k - \mathbf{x}^*\|_2 \end{aligned}$$

Cea de-a doua inegalitate considerată se poate demonstra după cum urmează: întrucât \mathbf{v}^k este obținut prin operația de "hard thresholding" aplicată asupra lui \mathbf{x}^k , va fi mai apropiat de \mathbf{x}^k decât orice alt vector de raritate s , incluzându-l pe \mathbf{x}^* . Ca urmare, avem $\|\mathbf{x}^k - \mathbf{v}^k\|_2 \leq \|\mathbf{x}^k - \mathbf{x}^*\|_2$. Considerând apoi inegalitatea triunghiului în triunghiul $(\mathbf{x}^k, \mathbf{v}^k, \mathbf{x}^*)$, lungimea laturii $\mathbf{v}^k - \mathbf{x}^*$ este mai mică decât suma celorlalte două laturi, și întrucât dintre acestea două latura $\|\mathbf{x}^k - \mathbf{x}^*\|_2$ este mai mare, avem:

$$\|\mathbf{v}^k - \mathbf{x}^*\|_2 \leq 2\|\mathbf{x}^k - \mathbf{x}^*\|_2$$

Combinând ambele inegalități obținem:

$$\begin{aligned} \|\mathbf{x}^{k+1} - \mathbf{x}^*\|_2 &\leq \delta_{2s}^{V^T} \|\mathbf{v}^k - \mathbf{x}^*\|_2 \\ &\leq 2\delta_{2s}^{V^T} \|\mathbf{x}^k - \mathbf{x}^*\|_2 \end{aligned}$$

Prin urmare, dacă V^T are o constantă RIP

$$\delta_{2s}^{V^T} < \frac{1}{2},$$

atunci eroarea dintre noua soluție la pasul $k+1$, \mathbf{x}^{k+1} , și soluția adevărată \mathbf{x}^* descrește la fiecare iterație cu un factor cel puțin constant, așadar algoritmul converge linear către soluția adevărată \mathbf{x}^* . \square

3.1.5 Avantajele algoritmului IAP

Condiția (3.6) arată că algoritmul IAP depinde de proprietățile RIP ale unei matrici tip "tight frame", V^T , formată din cei mai semnificativi vectori singulari dreapta ai matricii dicționar A . Ca urmare, considerând a descompunere SVD de forma $A = USV^T$, algoritmul IAP nu este afectat de spectrul de valori singulare S , cu alte cuvinte **algoritmul IAP nu este afectat de proprietățile spectrale a dicționarului**. Această "insensibilitate spectrală" este benefică

atunci când se lucrează cu dicționare învățate, care pot fi destul de coerente, și sunt de obicei destul de dezavantajoase din punct de vedere al numărului de condiție" (au valorile singulare într-o plajă largă, ceea ce face ca numărul de condiție, adică raportul dintre cea mai mare și cea mai mică valoare singulară, să aibă o valoare ridicată). Acest rezultat este verificat prin testele 2 și 3 din secțiunea 3.1.7.

În sens mai larg, acest avantaj al algoritmului IAP comparativ cu IHT se poate explica și în felul următor. Ambii algoritmi urmăresc minimizarea, la fiecare iterație, a celor doi termeni de eroare, respectiv eroarea de reprezentare $\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2$ și eroarea de raritate $\|\mathbf{x}_{T_c^k}\|_2^2$. Unul dintre aceștia este redus printr-un pas de tip coborâre după gradient, în timp ce cel de-al doilea este eliminat prin proiecție ortogonală. Acești doi termeni sunt de formă pătratică, dar pot diferi esențial în ceea ce privește numărul de condiție. Pentru eroarea de reprezentare, matricea \mathbf{A} poate avea un număr de condiție de valoare mare, care nu este sub controlul utilizatorului, întrucât derivă din dicționarul utilizat. Acest lucru este valabil mai ales atunci când dicționarul este obținut prin algoritmi de învățare, cum ar fi [6], întrucât datele înseși pot aparține unui domeniu îngust din spațiul de date. În schimb, pentru cel de-al doilea termen de eroare, numărul de condiție are întotdeauna valoarea 1, valoarea optimă. Acest lucru se poate vedea dacă rescriem termenul $\mathbf{x}_{T_c^k}$ sub forma

$$\mathbf{x}_{T_c^k} = I_{T_c^k} \cdot \mathbf{x},$$

unde $I_{T_c^k}$ este formată din liniile matricii unitate I_n care corespund co-suportului T_c^k , selectând astfel elementele din reprezentarea \mathbf{x} care corespund acestuia. Liniile matricii $I_{T_c^k}$ sunt ortonormale (fiind un subset al liniilor matricii unitate), și formează așadar ceea ce se numește "tight frame", pentru care numărul de condiție este egal cu unitatea, toate valorile singulare fiind de fapt egale cu 1.

Algoritmul IHT operează coborârea după gradient asupra erorii de reprezentare, iar proiecția ortogonală asupra erorii de raritate. Or, este binecunoscut faptul că algoritmi de coborâre după gradient nu sunt eficienți în probleme în care numărul de condiție are valoare mare. Acest lucru se explică prin forma foarte alungită a funcției de cost în problemele cu număr de condiție de valoare mare, care face ca gradientul să nu fie orientat înspre punctul de minim real, iar coborârea după gradient să converge foarte încet. De exemplu, în antrenarea rețelelor neurale de tip "deep learning", unde funcția de cost este de asemenea extrem de defavorabil condiționată, acest fapt motivează utilizarea algoritmilor rapizi de tip ADAM [ADAM2015] pentru antrenare, în locul celor uzuali de tip coborâre după gradient. Conchidem așadar că algoritmul IHT este, prin coborârea după gradient, sensibil la numărul de condiție al dicționarului, așa cum se va vedea din simulările care urmează.

Din contră, algoritmul propus IAP efectuează coborârea după gradient asupra termenului de eroare de raritate, care are numărul de condiție egal cu 1, ceea ce înseamnă că gradientul este

orientat direct către punctul optim. Eroarea de reprezentare este eliminată în schimb prin proiecție ortogonală, care însă este robustă la valori mari ale numărului de condiție (până la eventuale erori numerice). Faptul că eroarea de raritate este o funcție pătratică cu numărul de condiție egal cu 1 are două consecințe:

- Coborârea după gradient este mult mai eficientă la algoritmul IAP decât la IHT, IAP fiind în fapt imun la numărul de condiție al dicționarului.
- Pentru eliminarea totală a erorii de raritate la fiecare iterație este suficient a se alege o valoare convenabilă pentru pasul μ , întrucât gradientul este îndreptat direct către punctul de minim al funcției, și deci printr-o simplă scalare a sa se poate atinge acest punct.

Practic, prin alegerea pasului $\mu = 1$, eroarea de raritate este eliminată complet la fiecare iterație, soluția \mathbf{x}^k a fiecare iterația suferind exact o operație de “hard-thresholding”. În acest sens, algoritmul IAP cu $\mu = 1$ poate fi înțeles ca un algoritm de tip proiecții alternative (“Alternating Projections”), în care se alternează proiecțiile ortogonale pe spațiul afin al soluțiilor (pentru reducerea erorii de reprezentare) cu proiecții ortogonale pe subspațiul canonic de dimensiune s al suportului estimat T^k de la fiecare iterație (pentru asigurarea rarității).

Inversarea celor doi termeni de eroare face așadar ca algoritmul IAP să fie **imun la numărul de condiție al dicționarului**, fapt care se verifică și prin simulări, și care este reflectat și prin condiția teoretică demonstrată mai sus, care nu depinde de spectrul dicționarului.

Menționăm că nu cunoaștem în literatură rezultate care să descrie proprietățile RIP ale matricilor singulare dreapta ale unor matrici aleatoare, analoge rezultatelor binecunoscute referitoare la proprietățile RIP ale matricilor aleatoare, dar prezumăm că proprietățile lor sunt similare.

3.1.6 Considerații referitoare la implementare

Pentru compararea costului computațional implicat de algoritmul IAP, în comparație în special cu IHT, vom distinge între costuri pe iterație și costuri pe problemă de optimizare.

În termeni de costuri pe iterație, algoritmul IAP este similar cu IHT. Pentru ambii algoritmi, costul unei iterații este dominat de multiplicările cu A^* și A (pentru IHT), respectiv cu N^T și N (pentru IAP). Întrucât A are dimensiunea $m \times n$, în timp ce N are dimensiunea $(n - m) \times n$, se poate argumenta că IHT este ușor mai eficient pentru dicționare “late”, cu factor de supradimensionare n/m mai mare decât 2, în timp ce IAP este mai eficient pentru dicționare supracomplete dar cu factor de supradimensionare mai mic decât 2. În raport cu numărul de atomi n al dicționarului, ambii algoritmi au aceeași complexitate.

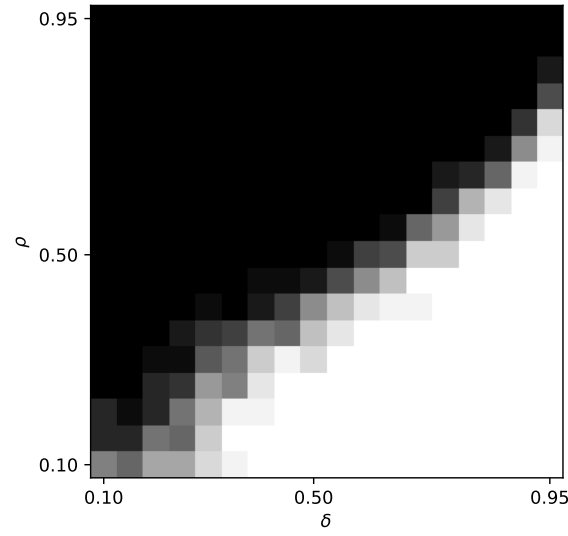
IAP poate avea un ușor dezavantaj în ceea ce privește costurile pe problemă de optimizare. De obicei, într-o problemă de codare rară se cunoaște de la bun început dicționarul A , dar IAP necesită în prealabil calculul unei baze N a spațiului nul (sau orice altă formă de a defini

operatorul de proiecție pe spațiul nul al matricii A), plus calculul soluției inițiale de energie minimă. Dacă dicționarul este același pentru un număr mare de codări rare, baza N pentru spațiul nul trebuie calculată o singură dată, ceea ce este mai eficient. În schimb, soluția de energie minimă trebuie calculată la începutul oricărei probleme de codare rară.

3.1.7 Rezultate ale simulărilor

În cele ce urmează, vom verifica performanțele algoritmului IAP în comparație cu alți algoritmi iterativi care utilizează de asemenea norma ℓ_0 pentru selecția atomilor: Iterativa Hard Thresholding (IHT) în două variante (cu pasul $\mu = 1$, și cu pasul adaptiv - AIHT[[19]]), Orthogonal Matching Pursuit (OMP) [20], [21], și algoritmul Approximate Message Passing (AMP) [22].

Prima simulare realizată folosește dicționare de dimensiune $m \times n$ cu elemente aleatoare, generate de o distribuție Gaussiană, ceea ce le face apropiate de matrici de tip “tight frame”. În urma simulărilor, reprezentăm grafic imaginile de tip tranziție de fază indicând procentul de semnale rare reconstruite exact, în funcție de rata de compresie $\delta = \frac{m}{n}$ și de raritatea relativă a problemei $\rho = \frac{s}{m}$. Pentru fiecare pereche de valori (δ, ρ) generăm un dicționar aleator Gaussian A , precum și 20 de semnale rare \mathbf{x} de raritate s variabilă, cu dimensiune $n = 200$. Se urmărește apoi refacerea acestor semnale rare \mathbf{x} din vectorii de măsurători $\mathbf{y} = A\mathbf{x}$, folosind fiecare algoritm testat. Se va lua în calcul procentul de semnale reconstruite în mod exact, ceea ce înseamnă că eroarea relativă de reconstrucție este mai mică decât 10^{-6} din norma semnalului \mathbf{x} original. Procentul de semnale reconstruite exact este indicat prin tonuri de gri, unde alb înseamnă 100%, negru 0%, iar diversele nuanțe de gri reprezintă valori intermediare.



(a) IAP

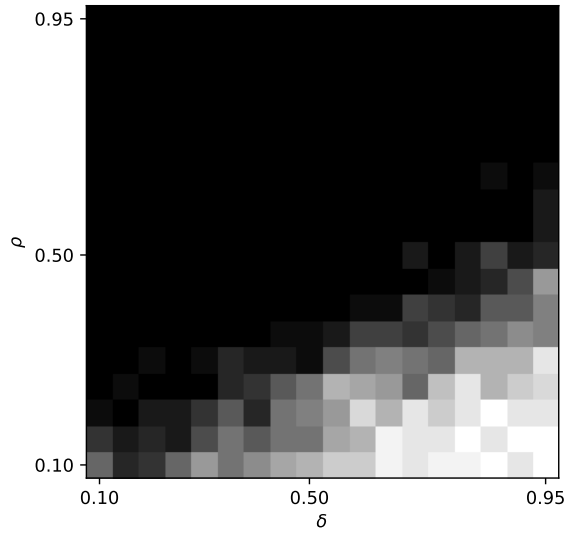
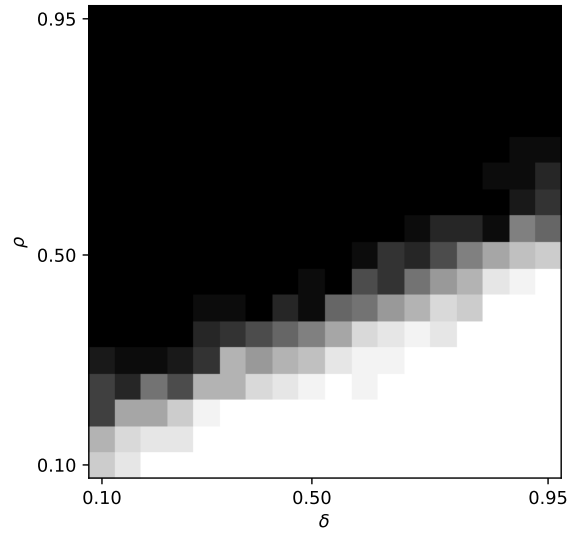
(b) IHT($\mu = 1$)(c) IHT (μ adaptiv)

Figure 3.2: Test 1 (a, b, c). Procentul de semnale rare reconstruite în mod exact cu diverși algoritmi de reconstrucție, folosind dicționare aleatoare Gaussiene: (a) algoritmul Iterative Affine Projection propus, (b) Iterative Hard Thresholding cu $\mu = 1$, (c) Iterative Hard Thresholding cu μ adaptiv, (d) Orthogonal Matching Pursuit și (e) Approximate Message Passing. Albul indică 100% din semnale reconstruite, iar negru 0%.

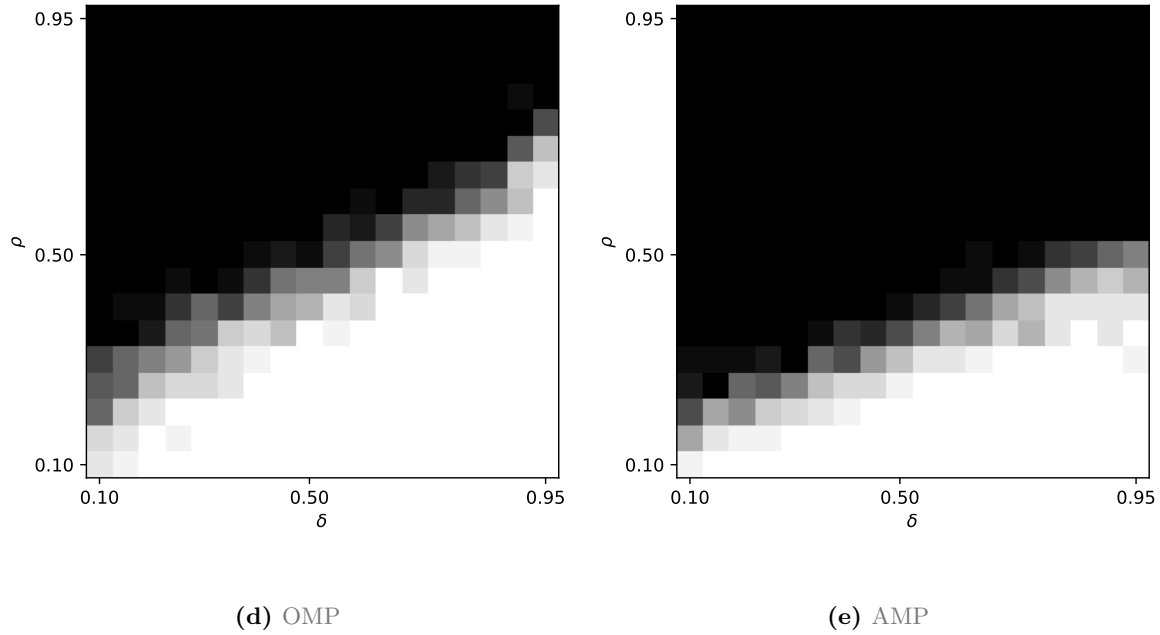
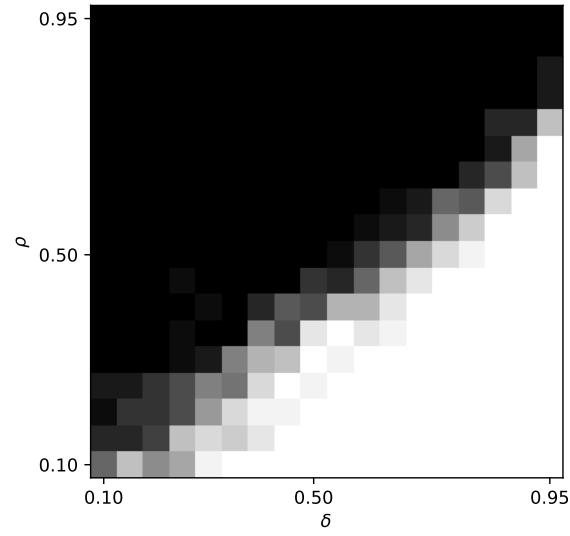


Figure 3.2: Test 1 (d, e). Procentul de semnale rare reconstruite în mod exact cu diverși algoritmi de reconstrucție, folosind dicționare aleatoare Gaussiene: (a) algoritmul Iterative Affine Projection propus, (b) Iterative Hard Thresholding cu $\mu = 1$, (c) Iterative Hard Thresholding cu μ adaptiv, (d) Orthogonal Matching Pursuit și (e) Approximate Message Passing. Albul indică 100% din semnale reconstruite exact, iar negru 0%.

Fig.3.2 înfățișează tranzițiile de fază obținute cu IAP, IHT, OMP și AMP. Pentru IAP se folosește pasul $\mu = 1$, pentru IHT se folosesc două variante, una cu pasul fix $\mu = 1$ și cealaltă cu pas adaptiv [19]. Rezultatele arată faptul că IAP are cele mai bune performanțe, însă toți algoritmii au în general rezultate bune.

Al doilea test este destinat să verifice imunitatea algoritmului IAP la dicționare cu număr de condiție mare, într-un scenariu de achiziție comprimată. Pentru fiecare pereche de parametri (δ, ρ) generăm un dicționar aleator Gaussian D de dimensiune $n \times n = 200 \times 200$, dar căruia i se înlocuiește apoi spectrul valorilor singulare cu o funcție exponențial descrescătoare reprezentată în Fig.3.4. Atomii sunt apoi re-normalizați. Generăm semnale aleatoare cu reprezentare de raritate s prin combinații lineare aleatoare ale atomilor din D , $\mathbf{x} = D\gamma$, $\|\gamma\|_0 = s$, care sunt apoi multiplicare cu o matrice de achiziție aleatoare P , de asemenea Gaussiană, de dimensiune $m \times n$. Se încearcă apoi cu algoritmii testați reconstrucția reprezentărilor rare γ , din care se reconstruiesc semnalele inițiale \mathbf{x} . Spunem că un semnal este refăcut în mod exact dacă eroarea de reconstrucție relativă este sub 10^{-6} din norma semnalului original \mathbf{x} .



(a) IAP

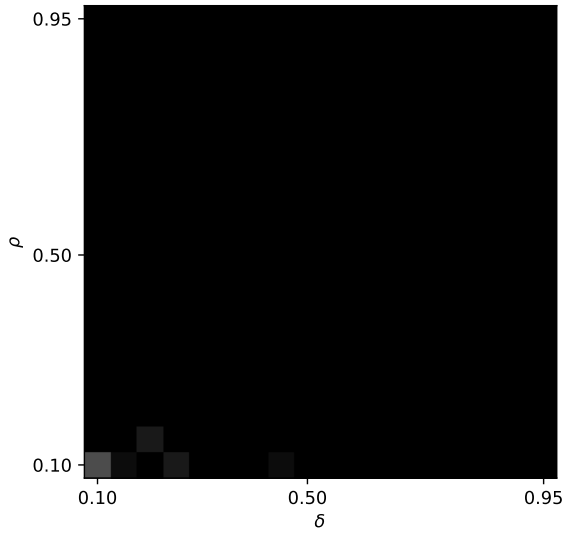
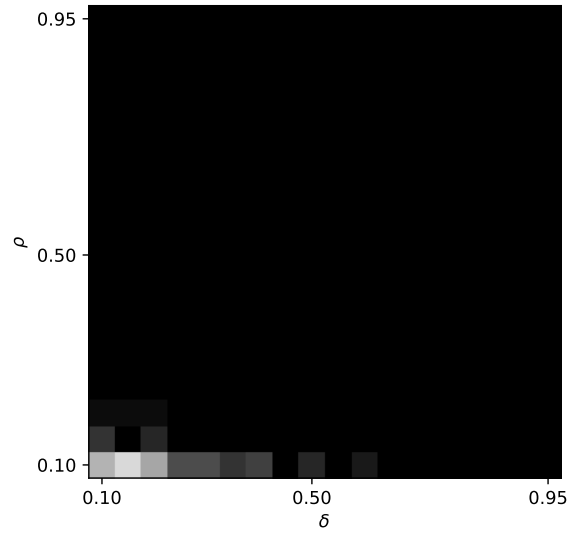
(b) IHT($\mu = 1$)(c) IHT (μ adaptiv)

Figure 3.3: Test 2 (a, b, c). Procentul de semnale rare reconstruite în mod exact cu diverși algoritmi de reconstrucție, folosind dicționare aleatoare Gaussiene cu spectrul valorilor singulare exponențial descrescător: (a) algoritmul Iterative Affine Projection propus, (b) Iterative Hard Thresholding cu $\mu = 1$, (c) Iterative Hard Thresholding cu μ adaptiv, (d) Orthogonal Matching Pursuit și (e) Approximate Message Passing. Albul indică 100% din semnale reconstruite exact, iar negru 0%.

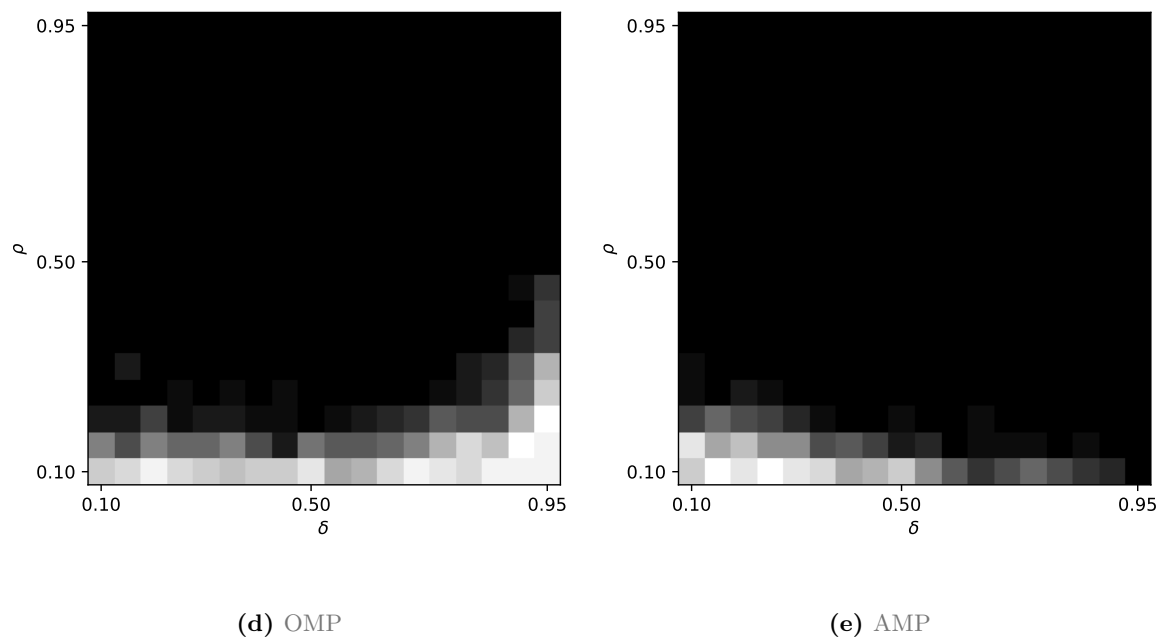


Figure 3.3: Test 2 (d, e). Procentul de semnale rare reconstruite în mod exact cu diverși algoritmi de reconstrucție, folosind dicționare aleatoare Gaussiene cu spectrul valorilor singulare exponențial descrescător: (a) algoritmul Iterative Affine Projection propus, (b) Iterative Hard Thresholding cu $\mu = 1$, (c) Iterative Hard Thresholding cu μ adaptiv, (d) Orthogonal Matching Pursuit și (e) Approximate Message Passing. Albul indică 100% din semnale reconstruite exact, iar negru 0%.

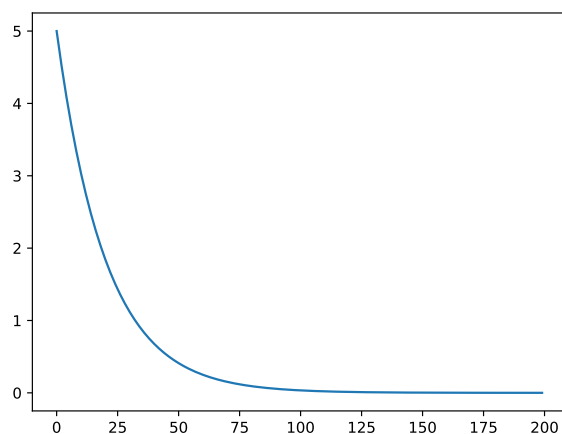


Figure 3.4: Spectrul exponențial descrescător de valori singulare folosit pentru dicționarele aleatoare din Testul 2

Rezultatele obținute sunt prezentate în Fig.3.3. Se observă în mod evident că algoritmul IAP este practic imun la numărul de condiție puternic defavorabil al dicționarelor, în timp ce toți ceilalți algoritmi sunt afectați în mod semnificativ.

În cel de-al treilea test efectuat, în locul dicționarelor aleatoare folosim un dicționar compus din porțiuni de imagini de dimensiune 8×8 pixeli, învățat cu algoritmul K-SVD [5] din baza de date “USC-SIPI Miscellaneous”. Acest dicționar ilustrează mai bine dicționarele întâlnite în aplicații reale, unde pot exista atomi puternic corelați. Dimensiunea dicționarului este 64×80 . Ca mai înainte, pe baza dicționarului generăm semnale cu reprezentare rară prin combinații liniare aleatoare ale atomilor, apoi verificăm posibilitățile de reconstrucție din proiecții aleatoare cu algoritmi testați.

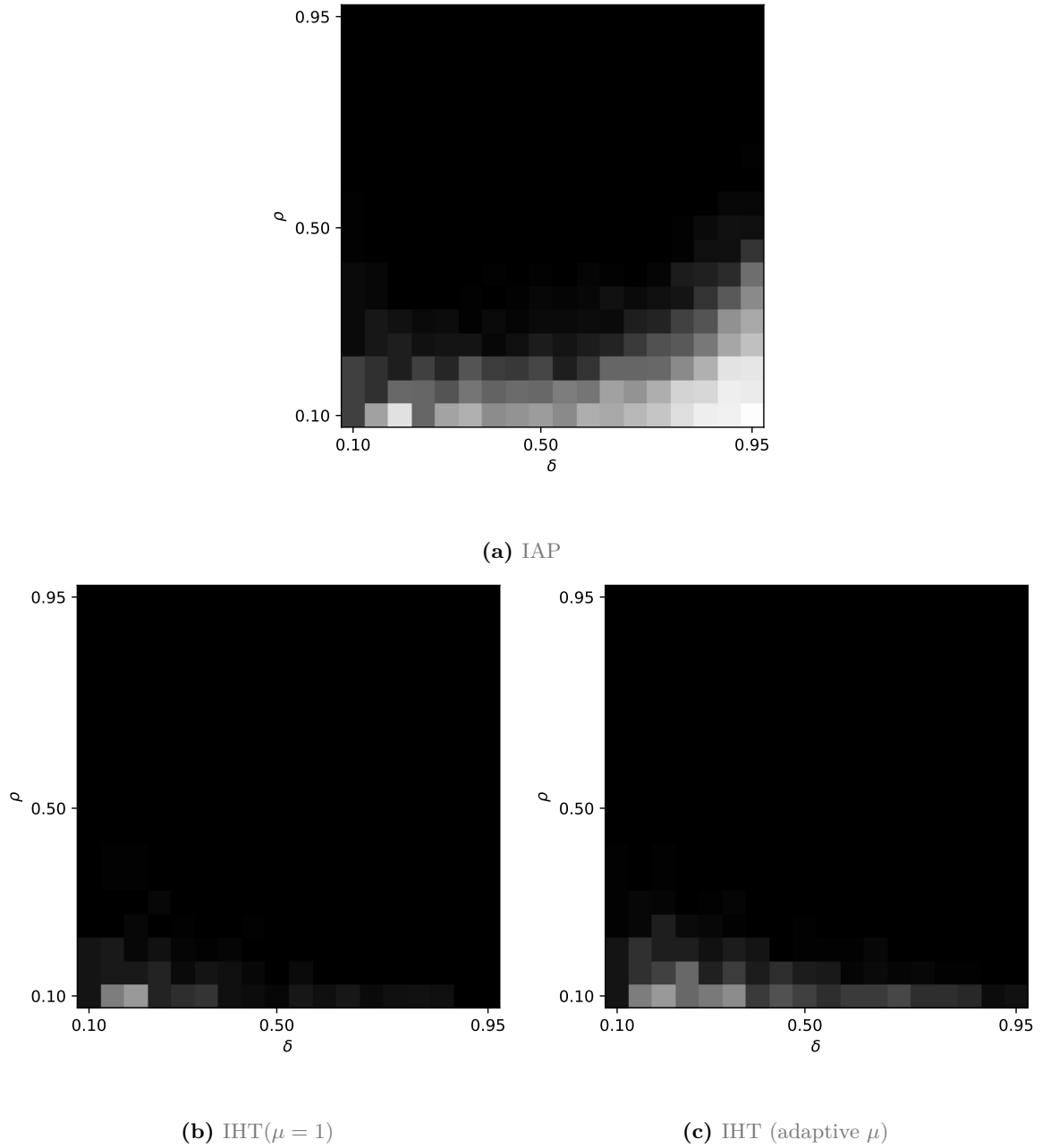


Figure 3.5: Test 3 (a, b, c). Procentul de semnale rare reconstruite în mod exact cu diverși algoritmi de reconstrucție, folosind un dicționar compus din porțiuni de imagini, învățat cu algoritmul K-SVD: (a) algoritmul Iterative Affine Projection propus, (b) Iterative Hard Thresholding cu $\mu = 1$, (c) Iterative Hard Thresholding cu μ adaptiv, (d) Orthogonal Matching Pursuit și (e) Approximate Message Passing. Albul indică 100% din semnale reconstruite exact, iar negru 0%.

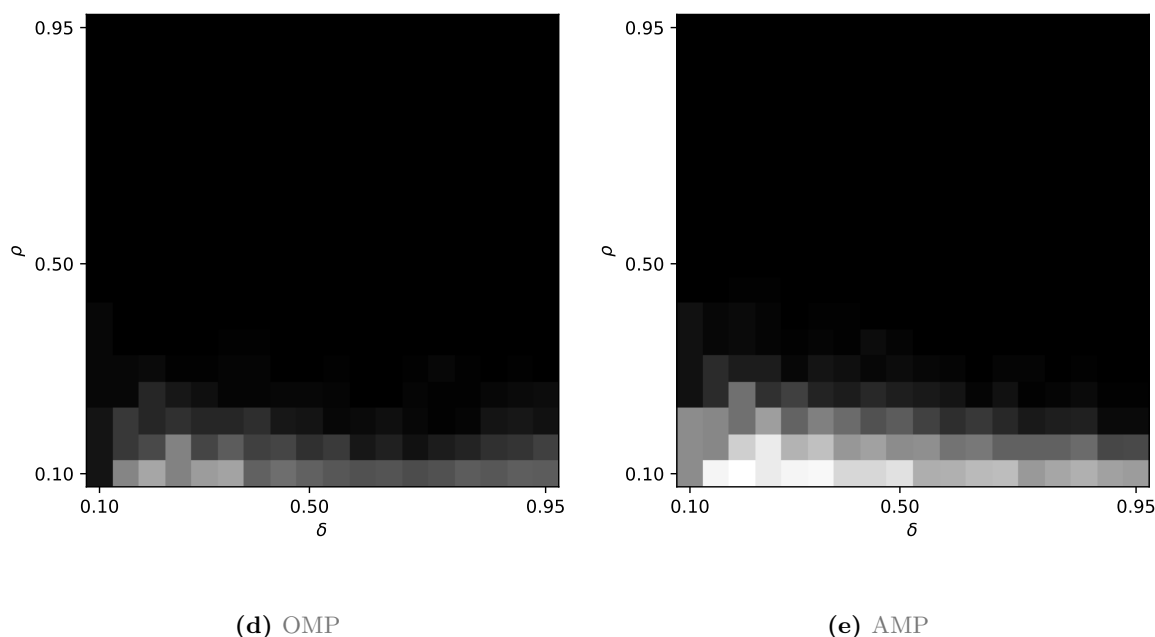


Figure 3.5: Test 3 (d, e). Procentul de semnale rare reconstruite în mod exact cu diverși algoritmi de reconstrucție, folosind un dicționar compus din porțiuni de imagini, învățat cu algoritmul K-SVD: (a) algoritmul Iterative Affine Projection propus, (b) Iterative Hard Thresholding cu $\mu = 1$, (c) Iterative Hard Thresholding cu μ adaptiv, (d) Orthogonal Matching Pursuit și (e) Approximate Message Passing. Albul indică 100% din semnale reconstruite exact, iar negru 0%.

Rezultatele grafice prezentate în Fig.3.5 arată că refacerea semnalelor cu reprezentare rară folosind acest dicționar învățat este mult mai dificilă. Totuși, atunci când se compară algoritmi testati, IAP conduce la performanțe vizibil mai bune comparativ cu IHT și OMP, superioare de asemenea și algoritmului AMP dar cu un ecart mai redus.

3.2 Metode de fuziune între algoritmi Orthogonal Matching Pursuit și Least Squares Pursuit

O a doua direcție de cercetare pentru îmbunătățirea soluțiilor de codare rară are în vedere combinarea inovativă a doi algoritmi existenți în literatură, Orthogonal Matching Pursuit [20], [21] Least Squares Pursuit [23], pentru a se obține o variantă mai robustă. Au fost studiate două metode de fuziune a celor doi algoritmi, ce exploatează similaritatea dintre structurile acestor algoritmi. O prima abordare are la bază medierea vectorilor gradient din cei doi algoritmi, la fiecare iterație. A doua abordare propune alegerea atomului selectat de algoritmul cu un

coeficient mai ridicat de încredere în alegerea făcută.

Rezultatele simulărilor indică performanțe mai robuste, inclusiv în situații care sunt dezavantajoase pentru fiecare dintre algoritmi originali.

3.2.1 Introducere

Pentru a explica metoda propusă în această secțiune, vom prezenta mai întâi cei doi algoritmi originali, care stau la baza soluțiilor propuse.

Considerăm problema standard de codare rară ce urmărește minimizarea normei ℓ_0 a reprezentării rare, de forma:

$$\gamma^* = \arg \min \|\gamma\|_0 \text{ cu constrângerea } y = A\gamma, \quad (3.7)$$

unde y este vectorul de măsurători, A este dicționarul utilizat, iar γ este vectorul reprezentării rare. Doi dintre algoritmi din literatură pentru rezolvarea acestei probleme sunt Orthogonal Matching Pursuit (OMP) [20], [21] și o variantă înrudită cu acesta, Least Squares Pursuit [23]. Cei doi algoritmi au în comun aceeași structură iterativă, diferențele constând în modul în care constrângerile de reprezentare exactă și de raritate sunt utilizate în cadrul unei iterații.

Bazându-ne pe asemănarea dintre cei doi algoritmi, în cele ce urmează investigăm două metode euristice de a îi combina într-o variantă mai robustă, care beneficiază de avantajele amândurora. Pentru aceasta, este necesară mai întâi prezentarea celor doi algoritmi.

3.2.1.1 Algoritmul Orthogonal Matching Pursuit

Orthogonal Matching Pursuit (OMP) este un algoritm foarte popular pentru rezolvarea problemelor de codare rară, care lucrează într-o manieră iterativă, prin selectarea la fiecare iterație a unui nou atom și adăugarea sa la suportul estimat. Suportul estimat T^k crește cu câte un atom la fiecare iterație, până când devine suficient de mare pentru a se găsi o soluție care să reprezinte întreg vectorul de intrare. Selecția unui nou atom are loc după cum urmează:

1. Se proiectează ortogonal vectorul de intrare y pe suportul estimat de la iterația curentă, T^k . În acest fel se obține un vector candidat γ^k care este rar (suportul T^k fiind restrâns), dar care are o eroare de reprezentare nenulă, $\|y - A\gamma^k\|_2^2 > 0$ (dacă eroarea de reprezentare este egală cu zero, atunci γ^k este soluția finală și algoritmul se oprește).
2. Se calculează gradientul erorii de reprezentare:

$$g = A^T(y - A\gamma^k),$$

3. Se selectează atomul corespunzător celui mai mare element din g , în valoare absolută, care

Algorithm 3 Orthogonal Matching Pursuit (OMP)

- 1: Iterația $k \leftarrow 0$
- 2: $T^0 \leftarrow \emptyset$
- 3: **repetă**
- 4: Actualizarea soluției prin proiecție pe suporul estimat:

$$\gamma^k \leftarrow \arg \min_{\gamma} \|y - A\gamma\|_2^2 \text{ a.î. } \gamma_{T_c^{(k)}} = 0$$

- 5: Calcularea gradientului erorii de reprezentare:

$$\mathbf{g} = A^T(y - A\gamma^k)$$

- 6: Actualizarea suportului cu atomul corespunzător valorii absolute maxime din gradient:

$$T^{k+1} \leftarrow T^k \cup \arg \max_i |\mathbf{g}_i|$$

- 7: **până când** criteriu de oprire **returnează** γ^k

promite cea mai mare descreștere a termenului de eroare.

Pașii 2 și 3 pot fi interpretați drept calculul corelației dintre reziduul de reprezentare, $(\mathbf{y} - A\gamma^k)$, și toți atomii din dicționarul A , urmată de selecția atomului cel mai similar cu reziduul respectiv.

Algoritmul este descris sub forma Algoritmului 3. Facem observația că algoritmul este descris într-un mod ușor diferit decât în literatura de specialitate, prin aceea că o iterație începe de la suportul estimat T^k și se termină prin obținerea suportului actualizat T^{k+1} pentru iterația următoare, calculul soluției candidat γ^k fiind doar un pas intermediar. Acest lucru este necesar pentru a evidenția asemănarea cu algoritmul Least Squares Pursuit, care urmează o procedură similară.

3.2.1.2 Algoritmul Least Squares Pursuit

Algoritmul Least Squares Pursuit (LSP) [23] reprezintă o alternativă recentă la algoritmul OMP, cu o metodă diferită de selecție a atomilor.

Considerând suportul estimat T^k de la începutul iterației k , soluția candidat γ^k de la iterația respectivă se poate calcula și altfel decât prin proiecție ortogonală pe setul atomilor din suportul T^k , ca în cazul OMP. Pentru aceasta, observăm că soluția γ^k de la iterația curentă caută să minimizeze doi termeni de eroare distincți:

- eroarea de raritate, cuantificată prin energia coeficienților corespunzători atomilor din afara suportului considerat T^k :

$$\|\gamma_{T_c^{(k)}}^k\|_2^2$$

Algorithm 4 Least Squares Pursuit

-
- 1: Iteration $k \leftarrow 0$
 - 2: $T^0 \leftarrow \emptyset$
 - 3: **repetă**
 - 4: Update solution via projection on the affine set

$$\gamma^k \leftarrow \arg \min_{\gamma} \|\gamma_{T_c^{(k)}}\|_2^2 \text{ subject to } y = A\gamma$$

- 5: Compute the gradient of the sparsity error

$$\mathbf{g} \leftarrow \gamma_{T_c^{(k)}}$$

- 6: Add to the support the atom corresponding to largest absolute value of the gradient

$$T^{k+1} \leftarrow T^k \cup \arg \max_i |\mathbf{g}_i|$$

- 7: **până când** stop criterion
 - 8: Output γ^k
-

- eroarea de reprezentare, reprezentată de energia erorii de reconstrucție:

$$\|y - A\gamma\|_2^2$$

În expresiile de mai sus, $\gamma_{T_c^k}^k$ desemnează vectorul γ^k restricționat la elementele sale corespunzătoare co-suportului T_c^k .

Privind din perspectiva celor doi termeni de eroare astfel definiți, proiecția ortogonală operată în cadrul unei iterații a algoritmului OMP înseamnă de fapt operația prin care se asigură minimizarea erorii de reprezentare, sub constrângerea de a păstra eroarea de raritate strict egală cu zero. Eroarea de raritate este nulă pentru că proiecția se face doar pe subspațiul generat de atomii din suport, iar eroarea de reprezentare este minimizată ca urmare a caracterului ortogonal al proiecției.

Algoritmul LSP propune interschimbarea celor doi termeni de eroare: minimizarea erorii de raritate, sub constrângerea menținerii erorii de reprezentare strict egală cu zero. Acest lucru se obține prin proiecția pe spațiul afin al soluției ecuației de reprezentare $\mathbf{y} = A\gamma$. În continuare, selecția următorului atom se face similar, alegându-se cel atomul corespunzător celei mai mari valori absolute a gradientului erorii. Diferența este că la OMP se consideră gradientul erorii de reprezentare, în timp ce în cazul LSP se consideră gradientul erorii de raritate. Algoritmul este prezentat ca și Algoritmul 4.

Menționăm că motivația algoritmului LSP este comună cu cea care stă în spatele algoritmului IAP prezentat în Secțiunea 3.1.3, și anume că termenul erorii de raritate are numărul de condiție

egal cu 1, în timp ce eroarea de reprezentare poate avea o valoare defavorabilă, ceea ce face ca gradientul utilizat erorii de raritate utilizat în LSP să fie mai robust decât gradientul erorii de reprezentare din OMP.

3.2.2 Strategii euristice de fuziune a algoritmilor OMP și LSP

Pornind de la cei doi algoritmi prezentați mai sus, vom investiga strategii de fuziune a lor într-o soluție mai robustă, motivați fiind de următoarele două observații:

- Algoritmii OMP și LSP respectă aceeași procedură iterativă, după cum se observă și din descrierile lor din Algoritmul 3 și Algoritmul 4. Diferența între ei apare doar în operațiile din *interiorul* unei iterații, unde cei doi termeni de eroare sunt interschimbați. Din acest motiv, se justifică încercarea de a combina regulile de selecție a atomilor din cei doi algoritmi, păstrând însă același cadru iterativ comun, respectiv faptul că fiecare iterație k începe de la suportul estimat T^k și se finalizează prin obținerea suportului următor T^{k+1} , cu un atom suplimentar.
- Condițiile teoretice care garantează convergența celor doi algoritmi sunt de natură diferită, ceea ce face ca cei doi algoritmi să se comporte diferit în practică. În particular, o condiție cunoscută din literatură care este suficientă pentru a garanta succesul codării rare cu algoritmul OMP este următoarea [24]:

$$\|A_{T^c}^*(A_T^*)^\dagger\|_{\infty, \infty} < 1. \quad (3.8)$$

O condiție similară pentru algoritmul LSP este cea de mai jos [23]:

$$\|N_{T^c}^\dagger N_T\|_{\infty, \infty} < 1, \quad (3.9)$$

unde N este o matrice de tip “tight frame” transpusă, ale cărei linii constituie o bază pentru spațiul nul al dicționarului A .

O diferență semnificativă este aceea că în cazul algoritmului LSP condiția depinde numai de spațiul nul al dicționarului, și astfel este mai robustă la dicționare cu spectru variat al valorilor singulare. În schimb, condiția pentru algoritmul OMP depinde de valorile dicționarului însuși. Astfel, cei doi algoritmi pot avea în practică rezultate diferite pentru anumite tipuri de semnale, motiv pentru care se justifică înglobarea lor într-o soluție globală, mai robustă.

Cele două strategii de fuziune a algoritmilor sunt prezentate mai jos.

3.2.2.1 Fuziune prin medierea vectorului gradient

O primă metodă de fuziune se bazează pe selectarea atomului care este cel mai potrivit din perspectiva **ambilor** algoritmi simultan. Fie \mathbf{g}_{OMP} vectorul gradient utilizat într-o iterație a algoritmului OMP, iar \mathbf{g}_{LSP} vectorul gradient corespunzător din algoritmul LSP (a se vedea Algoritmul 3 și Algoritmul 4).

Se definește gradientul mediu ca fiind o medie ponderată a celor doi vectori:

$$\mathbf{g}^{avg} = \lambda \cdot \mathbf{g}_{OMP} + (1 - \lambda) \cdot \mathbf{g}_{LSP}, \quad (3.10)$$

unde λ este un parametru de pondere a celor doi termeni.

La fiecare iterație, atomul selectat pentru includerea în setul suport va fi cel corespunzător celei mai mari valori absolute din vectorul gradient, considerând însă de această dată gradientul mediu definit mai sus:

$$T^{k+1} \leftarrow T^k \cup \arg \max |\mathbf{g}_i^{avg}|$$

Aceasta este aceeași regulă de selecție ca la algoritmii OMP și LSP, dar aplicată asupra gradientului mediu. Atomul selectat în acest fel nu este neapărat cel care ar fi fost selectat de OMP sau de LSP, dacă ar fi rulat independent. De exemplu, dacă alegerea algoritmului OMP este un candidat slab pentru algoritmul LSP și vice-versa, atunci atomul selectat în final ar putea fi cel de-al doilea atom candidat din perspectiva ambilor algoritmi, atât timp cât acesta este o alegere rezonabilă în ambii algoritmi simultan. În acest fel se crește robustețea procesului de selecție.

3.2.2.2 Fuziune prin alegerea candidatului cu încredere mai ridicată

A doua strategie de fuziune a celor doi algoritmi se bazează pe alegerea unuia dintre cei doi atomi propuși de OMP și LSP, în funcție de algoritmul care are cel mai ridicat grad de “încredere” în alegerea făcută.

Se definește **coeficientul de încredere** al algoritmilor OMP și LSP la iterația k ca fiind raportul dintre prima și a doua cea mai mare valoare absolută a vectorului gradient:

$$C = \frac{\arg \max |\mathbf{g}_i|}{\arg \max_{i \neq \arg \max |\mathbf{g}_i|} |\mathbf{g}_i|}$$

Atomul selectat în cadrul unei iterații va fi cel propus fie de OMP fie de LSP, în funcție de care algoritm are un coeficient de încredere de valoare mai mare:

$$T^{k+1} \leftarrow T^k \cup \begin{cases} \arg \max |\mathbf{g}_i^{OMP}|, & \text{dacă } C^{OMP} > C^{LSP} \\ \arg \max |\mathbf{g}_i^{LSP}|, & \text{dacă } C^{OMP} \leq C^{LSP} \end{cases}$$

Metoda poate aduce un beneficiu atunci când cea mai mare valoare absolută a vectorului gradient este apropiată de valoarea următoare, caz în care se poate presupune că regula de selecție va fi mai puțin robustă la erori aleatoare ale vectorului de intrare, ale dicționarului, sau dacă vectorul de intrare are o reprezentare doar aproximativ rară în dicționarul considerat. Din contră, dacă valoarea cea mai mare a gradientului are o margine semnificativă față de valoarea imediat următoare, atunci se poate presupune că selecția va fi mai puțin influențată de eventualele erori.

Coeficientul de încredere definit mai sus surprinde exact dimensiunea marginii de siguranță a selecției, coeficientul având valori în intervalul de la 1 (în cazul unei selecții cu grad mic de încredere, când valoarea maximă este foarte apropiată de cea următoare) la ∞ (selecție cu grad foarte mare de încredere, când valoarea următoare este neglijabilă comparativ cu valoarea maximă), o valoare mai mare semnificând o încredere mai ridicată.

3.2.3 Rezultate de simulare

Pentru a verifica performanțele de codare rară ale metodelor propuse, acestea vor fi supuse unor teste, în două scenarii diferite. Cele două scenarii de test sunt alese astfel încât să mai fie defavorabile, pe rând, atât algoritmului OMP cât și algoritmului LSP, scopul urmărit fiind de a observa dacă metodele propuse reușesc să evite dezavantajele fiecărui algoritm constitutiv în parte, împrumutând caracteristicile algoritmului optim în fiecare scenariu.

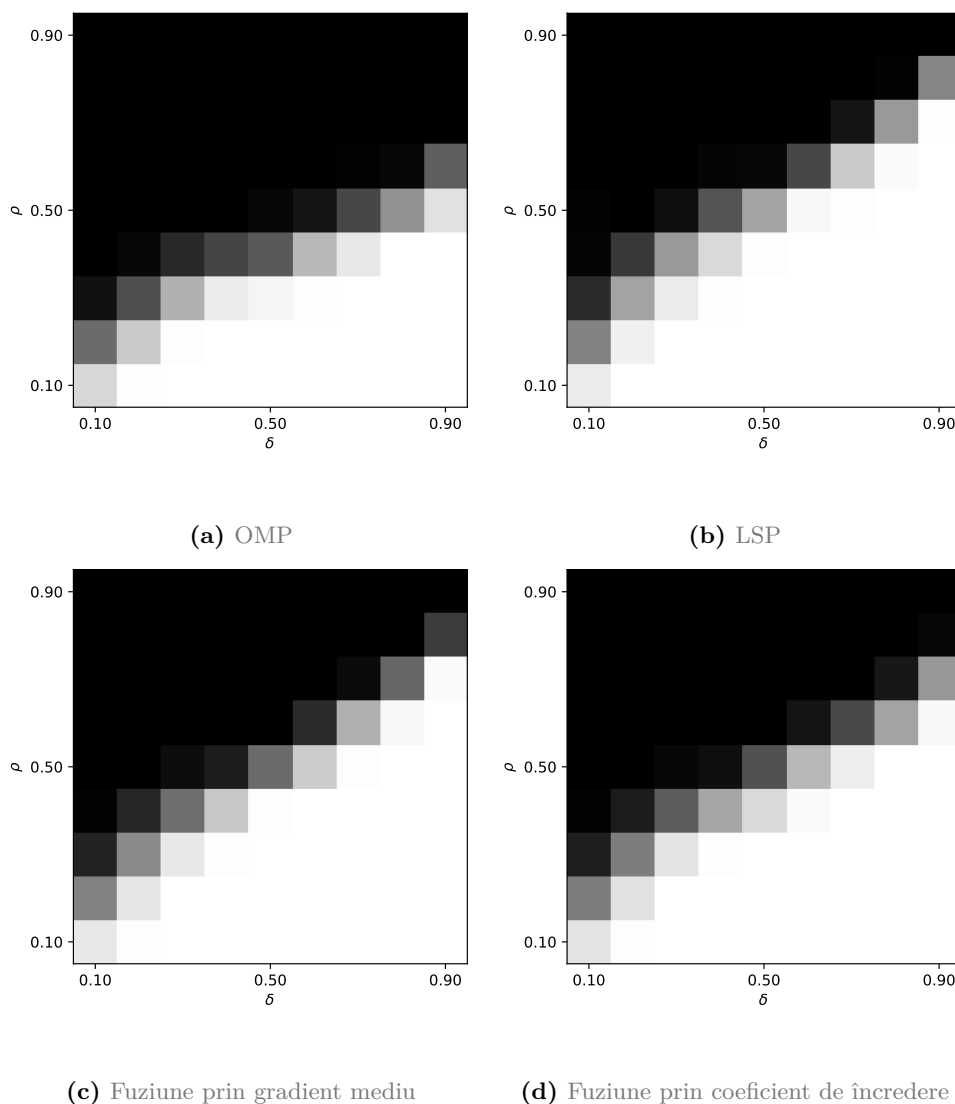


Figure 3.6: Test 1. Procentul semnalelor rare reconstruite în mod exact cu diferiți algoritmi de reconstrucție, folosind dicționare aleatoare Gaussiene: (a) Orthogonal Matching Pursuit, (b) Least Squares Pursuit, (c) Fuziune prin gradient mediu, (d) Fuziune prin coeficient de încredere. Albul indică un procent de 100% semnale reconstruite, negrul 0%, iar nivelurile de gri valori intermediare.

Primul scenariu de test, care este mai favorabil algoritmului LSP decât OMP, folosește dicționare aleatoare, cu elemente independente și identic distribuite conform unei distribuții Gaussiene, de dimensiune 200×200 . Semnale sintetice cu reprezentare rară sunt create prin combinații aleatoare ale atomilor din dicționar. Semnalele sunt apoi achiziționate comprimat folosind o matrice de achiziție de asemenea Gaussiană. Din vectorii de măsurători astfel obținuți se încearcă

apoi reconstruirea semnalelor originale cu algoritmii testați. Rezultatele sunt prezentate în Fig.3.6, în funcție de doi factori: factorul de compresie al problemei de codare rară, reprezentat de $\delta = \frac{m}{n}$, și raritatea relativă a reprezentării rare, $\rho = \frac{k}{m}$, unde k este raritatea semnalelor generate, m este numărul de măsurători (numărul de linii al matricei de achiziție), iar $n = 200$ reprezintă dimensiunea dicționarelor utilizate. Pentru fiecare pereche de valori (δ, ρ) sunt generate un număr de 100 semnale rare, care sunt apoi achiziționate și reconstruite. Se consideră ca un semnal a fost reconstruit în mod exact dacă eroarea relativă a reconstrucției sale este mai mică decât pragul de $1e^{-6}$. Procentul semnalelor reconstruite exact, pentru fiecare pereche de parametri (δ, ρ) , este reprezentat în imagine sub forma unei nuanțe de gri: albul indică un procent de 100%, negrul 0%, iar nivelurile de gri indică valorile intermediare.

Pentru strategia de fuziune prin medierea gradientilor, factorul de pondere λ utilizat în teste are valoarea $\lambda = 0.5$ (medie aritmetică între cei doi vectori gradient), dar rezultate similare se obțin pentru valori dintr-o plajă mai largă în jurul valorii de 0.5.

Rezultatele sunt prezentate sub forma unei aproximări a unei imagini tip tranziție de fază [25], folosind un set discret de valori pentru parametrii δ , ρ , acoperind plaja de valori $[0, 1]$ cu un pas de 0.1.

După cum se poate observa în Fig. 3.6, metodele de fuziune propuse conduc la rezultate îmbunătățite față de algoritmul OMP, metoda cea mai bună fiind cea bazată pe medierea gradientilor, care atinge practic performanțele algoritmului LSP.

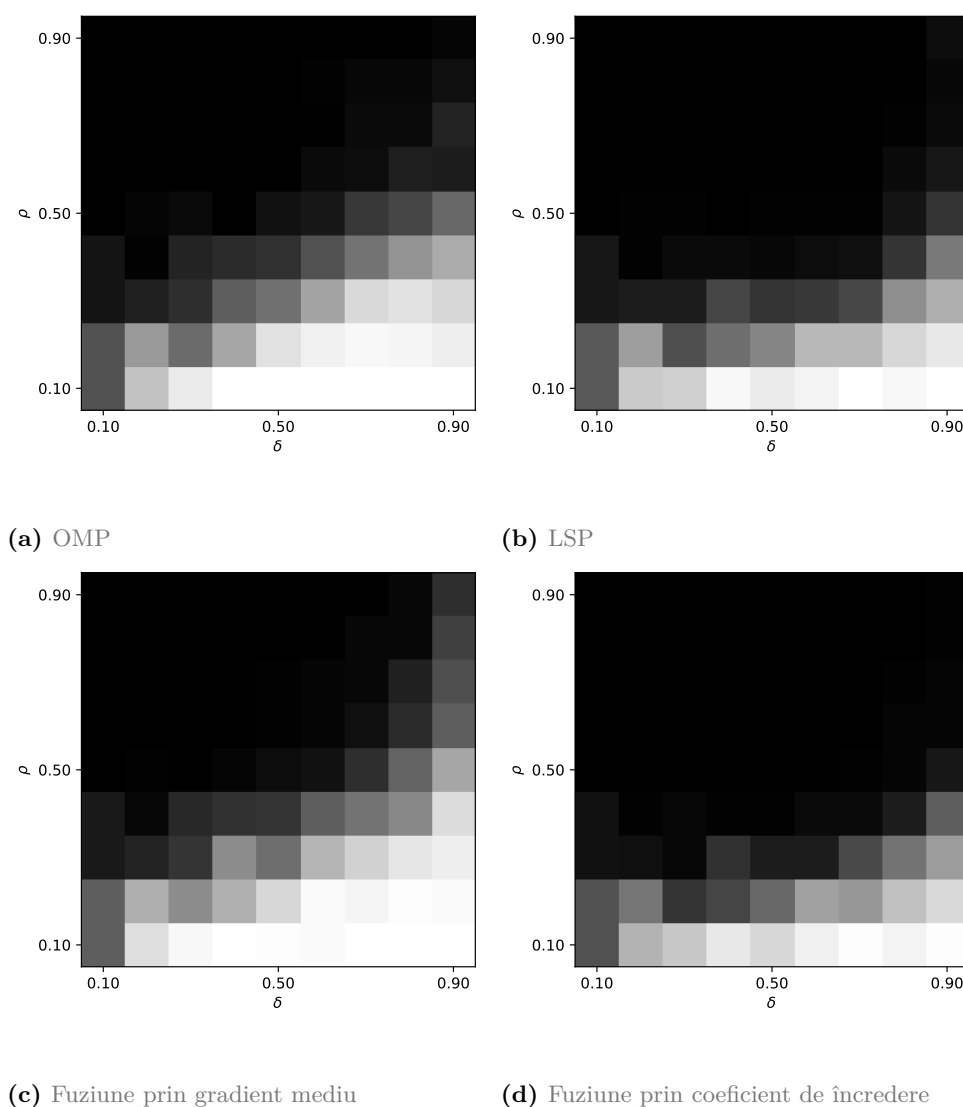


Figure 3.7: Test 2. Procentul semnalelor rare reconstruite în mod exact cu diferiți algoritmi de reconstrucție, folosind dicționare cu atomi clonați: (a) Orthogonal Matching Pursuit, (b) Least Squares Pursuit, (c) Fuziune prin gradient mediu, (d) Fuziune prin coeficient de încredere. Albul indică un procent de 100% semnale reconstruite, negrul 0%, iar nivelurile de gri valori intermediare.

Pentru cel de-al doilea test, sunt generate semnale cu reprezentare rară pe baza unor dicționare în care unii atomi sunt clonați, respectiv apar de mai multe ori în dicționar, pentru a ilustra cazul unor dicționare cu coerențe ridicată între atomi. Aceste dicționare au un impact negativ asupra algoritmului LSP, din următorul motiv: dacă un atom apare de mai multe ori în cadrul dicționarului, atunci, atât timp cât nici una dintre copiile sale nu este selectată în setul suport,

minimizarea erorii de raritate va distribui coeficienți de valoare mică și egali între ei pentru toate copiile atomului respectiv. Acest lucru se întâmplă deoarece prin minimizarea normei ℓ_2 se obțin coeficienți de valori mici și de valoare redusă. În acest fel, acești atomi identici vor avea șanse mai reduse de a fi aleși în suport în comparație cu alți atomi care nu sunt clonați, ceea ce poate impacta în mod negativ procesul de selecție al atomilor. Pe de altă parte, algoritmul OMP nu este afectat de prezența în dicționar a mai multor clone ale aceluiași atom, pentru că toate vor avea același coeficient de corelație cu reziduul de la fiecare iterație, iar semnalul va fi reconstruit în mod identic oricare ar fi clona selectată în suport. În cazul ambilor algoritmi, odată ce un atom este selectat în suport toate clonele sale devin irelevante în desfășurarea ulterioară a algoritmului, întrucât etapa de proiecție ortogonală din ambii algoritmi elimină posibilitatea ca o clonă a unui atom deja selectat să poată fi selectată la o iterație ulterioară.

Rezultatele acestui test sunt prezentate în Fig.3.7, pentru un dicționar aleator de dimensiune 60×1060 , care conține 60 de atomi distincți, dintre care un număr de 30 sunt individuali iar ceilalți 30 sunt clonați de aproximativ 33 de ori fiecare. După cum am argumentat mai sus, pentru acest tip de dicționar algoritmul LSP este mai afectat decât OMP, obținând rezultate mai slabe.

La fel ca în primul test, observăm că algoritmul de fuziune bazat pe medierea vectorilor gradient este mai robust la natura defavorabilă a dicționarului, obținând performanțe similare cu OMP. În acest scenariu, cea de-a doua strategie de fuziune propusă nu conduce la rezultate superioare.

4 Concluzii

Acest raport științific și tehnic prezintă rezultatele activității de cercetare desfășurată în cadrul proiectului de cercetare postdoctorală PN-III-P1-1.1-PD-2016-0127 intitulat “Prelucrarea eficientă a semnalelor biomedicale folosind codarea rară (SCEBIOS)”, etapa 2 / 2019.

Activitatea de cercetare s-a concretizat în două direcții principale, prezentate pe larg în raport, respectiv:

- O soluție de identificare a patologiilor inimii din semnale ECG folosind codarea rară și tehnica Bag-of-Words, descrisă în secțiunea 2.1. În această direcție există o lucrare în lucru, cu rezultate parțiale disponibile, dar care trebuie îmbunătățite în vederea publicării;
- Algoritmi îmbunătățiți de codare rară, prezentați în secțiunile 3.1 și 3.2. În această direcție au fost publicate două lucrări la conferințe internaționale, și se află în pregătire cel puțin o lucrare mai extinsă destinată publicării într-o revistă de specialitate.

Algoritmii aferenți lucrărilor deja publicate sunt disponibile pe Internet în regim *open-source*.

Bibliografie

- [1] I. B. Ciocoiu, “Comparative analysis of bag-of-words models for ecg-based biometrics,” *IET Biometrics*, vol. 6, no. 6, pp. 495–502, 2017.
- [2] B. K. Natarajan, “Sparse approximate solutions to linear systems,” *SIAM J. Comput.*, vol. 24, no. 2, pp. 227–234, Apr. 1995.
- [3] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, “Least angle regression,” *The Annals of Statistics*, vol. 32, no. 2, pp. 407–451, 2004.
- [4] “SPAMS toolbox.”
- [5] M. Aharon, M. Elad, and A. Bruckstein, “K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation,” *IEEE Transactions on signal processing*, vol. 54, no. 11, p. 4311, 2006.
- [6] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, “Online dictionary learning for sparse coding,” in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 689–696.
- [7] Q. Zhang and B. Li, “Discriminative K-SVD for dictionary learning in face recognition,” in *The twenty-third IEEE conference on computer vision and pattern recognition, CVPR 2010, san francisco, ca, usa, 13-18 june 2010*, 2010, pp. 2691–2698.
- [8] Z. Jiang, Z. Lin, and L. S. Davis, “Label consistent k-svd: Learning a discriminative dictionary for recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2651–2664, Nov. 2013.
- [9] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: Data mining, inference and prediction*, 2nd ed. Springer, 2009.
- [10] G. D. Clifford *et al.*, “AF classification from a short single lead ecg recording: The physionet/computing in cardiology challenge 2017,” in *2017 computing in cardiology (cinc)*, 2017, pp. 1–4.
- [11] “AF classification from a short single lead ecg recording - the physionet computing in cardiology challenge 2017.”
- [12] M. Fira, L. Goras, C. Barabasa, and N. Cleju, “On ECG Compressed Sensing using Specific Overcomplete Dictionaries,” *aece*, vol. 10, no. 4, pp. 23–28, 2010.

- [13] E. J. Candès, J. K. Romberg, and T. Tao, “Stable signal recovery from incomplete and inaccurate measurements,” vol. 59, pp. 1207–1223, 2006.
- [14] D. L. Donoho, “Compressed sensing,” *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.
- [15] E. J. Candes and J. Romberg, “Sparsity and incoherence in compressive sampling,” *Inverse Problems*, vol. 23, no. 3, pp. 969–985, Jun. 2007.
- [16] E. J. Candès, “The restricted isometry property and its implications for compressed sensing,” *Comptes Rendus Mathématique*, vol. 346, pp. 589–592, 2008.
- [17] S. Foucart, “Sparse recovery algorithms: Sufficient conditions in terms of restricted isometry constants,” in *Approximation Theory XIII: San Antonio 2010*, 2012, pp. 65–77.
- [18] T. Blumensath and M. E. Davies, “Iterative hard thresholding for compressed sensing,” *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 265–274, 2009.
- [19] T. Blumensath, “Accelerated iterative hard thresholding,” *Signal Processing*, vol. 92, no. 3, pp. 752–756, 2012.
- [20] Y. C. Pati, R. Rezaifar, and P. S. Krishnaprasad, “Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition,” in *Proc. 27th annual asilomar conf. On signals, systems, and computers*, 1993, pp. 40–44.
- [21] G. Davis, S. Mallat, and M. Avellaneda, “Adaptive greedy approximations,” *Constructive approximation*, pp. 57–98, 1997.
- [22] D. L. Donoho, A. Maleki, and A. Montanari, “Message-passing algorithms for compressed sensing,” *Proceedings of the National Academy of Sciences*, vol. 106, no. 45, pp. 18914–18919, 2009.
- [23] N. Cleju, “Least squares pursuit for sparse signal recovery,” in *2017 International Symposium on Signals, Circuits and Systems (ISSCS)*, 2017, pp. 1–4.
- [24] J. Tropp, “Greed is good: Algorithmic results for sparse approximation,” *Information Theory, IEEE Transactions on*, vol. 50, no. 10, pp. 2231–2242, 2004.
- [25] D. Donoho and J. Tanner, “Observed universality of phase transitions in high-dimensional geometry, with implications for modern data analysis and signal processing,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 367, no. 1906, pp. 4273–4293, 2009.