Algoritmi de Reprezentare Rară, Învățarea Dicționarelor și Reconstrucție Rară: Detectarea și Clasificarea Aritmiilor EKG pe 12 Derivații

Tudor Pistol și Teofil Simiraș

January 16, 2025

Contents

1 A. TITLU ȘI CUPRINS)

Titlu propus pentru proiect:
„Algoritmi de Reprezentare Rară, Învățarea Dicționarelor și Reconstrucție Rară: Detectarea și Clasificarea Aritmiilor EKG pe 12 Derivații”

Cuprins (planificat pentru prezentare):

1. Introducere și Context
2. Problema Abordată
3. Justificarea Problemei
4. Abordare Tehnică Propusă
5. Tehnologii și Biblioteci Folosite

6. Rezultate Așteptate și Metodologie de Evaluare
7. Concluzii și Direcții Potențiale de Extindere

2 B. PROBLEMA ABORDATĂ ÎN PROIECT

În cadrul acestui proiect, **intenționăm să detectăm și să clasificăm aritmiile cardiace** (cum ar fi fibrilația atrială, extrasistole etc.) în semnalele EKG cu **12 derivații**. Se știe că:

- Datele EKG de 12 derivații oferă o imagine foarte completă a activității electrice a inimii, dar sunt și mai voluminoase.
- Analiza clasică necesită implicarea intensivă a specialiștilor (cardiologi), iar variabilitatea umană poate duce la erori.

Obiectiv principal: Să dezvoltăm (în următoarea perioadă) un *pipeline automat* care, după preprocesarea și segmentarea bătăilor cardiace, să aplice **Reprezentarea Rară (Sparse Coding)** pentru extragerea de caracteristici și apoi să folosească un **clasificator** (SVM, Logistic Regression etc.) pentru a identifica automat bătăile normale față de cele care prezintă aritmii cardiace.

3 C. JUSTIFICAREA PROBLEMEI ABORDATE

1. De ce e importantă?

- Aritmiile cardiace sunt frecvente și pot fi critice dacă nu sunt depistate la timp.
- Un sistem semi-automat sau automat pentru detecția aritmiilor cardiace reduce timpul de diagnostic și crește acuratețea.

2. Context și ce problemă rezolvă?

- În spitale se adună zilnic sute/mii de EKG-uri. Un algoritm robust ajută la trierea rapidă a pacienților care au nevoie de investigații suplimentare.
- În dispozitive portabile (Holter EKG, wearables), un algoritm cu cost computațional relativ scăzut poate alerta medicul sau pacientul în timp real.

3. Unde poate fi folosit?

- Clinici, centre de cardiologie, laboratoare de cercetare care lucrează cu analiza semnalelor cardiace.
- În aplicații de telemedicină și monitorizare la distanță (conectate la cloud).

4 D. CUM ESTE ABORDATĂ PROBLEMA TEHNIC (DESCRIERE MATEMATICĂ, PLAN DE IMPLEMENTARE)

În această secțiune, descriem **modul în care vom implementa** proiectul, pas cu pas, fără a intra (încă) în toate detaliile codului, ci arătând *ce urmează să facem* tehnic.

1. Colectarea Datelor și Structurarea lor

- **Set de date**: Planificăm să utilizăm *A large scale 12-lead electrocardiogram database for arrhythmia study*, care conține o varietate de înregistrări EKG pe 12 derivații, cu adnotări pentru diferite tipuri de aritmii cardiace.
- Vom organiza datele în *fișiere* (.mat, .csv) astfel încât fiecare fișier să conțină un EKG pe 12 derivații și eventual etichete la nivel de bătăi cardiace (normal/aritmie).

2. Preprocesare (Filtrare, Normalizare, Segmentare)

(a) **Filtrare**:

- Aplicăm un filtru band-pass (0.5 – 40 Hz) pentru a reține componentele relevante din EKG.
- Eliminăm bruiajul de rețea (50/60 Hz) cu un filtru notch.
- Stabilim o metodă de înlăturare a baseline wander (ex. high-pass la 0.5 Hz sau spline interpolation).

(b) **Normalizare**:

- Vom normaliza fiecare derivată la intervalul [-1, 1], pentru a asigura consistența amplitudinii între pacienți diferiți.

(c) **Segmentare pe bătăi cardiace**:

- Folosim un algoritm de detecție QRS (Pan-Tompkins sau alt detectator) pentru a identifica peak-urile R.
- Extragem segmente tip (ex. 100 ms înainte + 300 ms după R-peak).
- Rezultatul: o colecție de bătăi cardiace ($\mathbf{x}_i$) de aceeași lungime L, pentru fiecare dintre cele 12 derivații.

3. Reprezentarea Rară (Sparse Coding) – Plan de Implementare

- **Veci flatten vs. patch-uri**:
  - Cel mai simplu mod: aplatizăm (flatten) fiecare bătaie (12 derivații $\times$ L eșantioane) într-un vector $\mathbf{x} \in \mathbb{R}^{12 \cdot L}$.
  - În viitor, putem extinde cu patch-uri 2D sau sub-patch-uri temporale.
- **Dictionary Learning (ex. K-SVD sau Online)**:

- Vom antrena un dicționar $\mathbf{D}$ de dimensiune $(d \times K)$, unde $d = 12 \cdot L$, iar K este numărul de "atomi".
- Metode posibile:
  * **K-SVD** (classică, implementare în scikit-learn/ SPAMS).
  * **Online Dictionary Learning** (mai rapid, util când avem multe date).
- Vom decide un **nivel de raritate** (ex. $s = 10$–20 atomi nenuli la reconstrucție) sau vom folosi o penalizare L1 ($\lambda \|\boldsymbol{\alpha}\|_1$).

- **Extragerea coeficienților**:
  - Fiecare nouă bătaie cardiacă este reprezentată sub forma unui vector de coeficienți rari $\boldsymbol{\alpha}_i$.
  - Acești coeficienți devin *feature*-urile de intrare în clasificator.

4. Clasificare – Diferite Căi pe care le Vom Explora

(1) **SVM (kernel RBF sau liniar)**:

- Este robust la date cu dimensiuni medii.
- Vom încerca inițial RBF (care prinde relații nelineare) și vom face grid search pentru parametrii C și γ.

(2) **Regresie Logistică**:

- Metodă simplă, interpretabilă, utilă pentru a vedea rapid dacă vectorii de coeficienți au putere de discriminare.
- Timp de antrenare redus, face debugging mai ușor.

(3) **Random Forest sau XGBoost**:

- Metode de tip ensemble, pot fi eficiente în clasificarea aritmiilor cardiace.
- Vor fi testate pentru a vedea dacă oferă un plus de acuratețe sau stabilitate.

(4) **Abordare bazată pe Rețele Neurale (opțional / extensie)**:

- Dacă timpul permite, vom explora o arhitectură MLP (Fully-Connected) care primește ca input tot vectorul de coeficienți rari.
- Avantaj: poate capta interacțiuni nelineare mai subtile între coeficienți.

5. Metodologie de Testare

- Vom separa datele în *train-validation-test* (ex. 70%–15%–15%).
- Vom calcula metrici precum: **Acuratețe**, **Sensibilitate (Recall)**, **Specificitate**, **F1-score**, **ROC/AUC**.
- Vom face experimente cu diverși hiperparametri (dimensiune dicționar, nivel de raritate, parametri de clasificare etc.) și vom documenta parametrii care dau cele mai bune rezultate.

5 E. TEHNOLOGIILE FOLOSITE

(1) **Python**

- Deja un standard în cercetare și dezvoltare rapidă.

(2) **Biblioteci de bază**

- **NumPy, SciPy**: pentru calcule matriciale, funcții de filtrare semnal.
- **matplotlib / seaborn**: vizualizarea semnalelor EKG pre și post-filtrare, graficarea metricilor de clasificare.

(3) **Biblioteci pentru Sparse Coding și Dictionary Learning**

- **scikit-learn** (`DictionaryLearning`, `SparseCoder`) – ușor de integrat, are implementări decente.
- (Opțional) **SPAMS** – pachet specializat pe K-SVD, Lasso, OMP, dacă avem nevoie de performanțe superioare.
- **PyTorch / TensorFlow** (opțional, dacă abordăm rețele neurale, sau un autoencoder sparse).

(4) **Clasificare**

- `sklearn.svm` (SVC) pentru SVM,
- `sklearn.linear_model` (LogisticRegression),
- `sklearn.ensemble` (RandomForestClassifier),
- `xgboost` (dacă testăm XGBoost).

De ce folosim aceste tehnologii?

- Ecosistemul Python + scikit-learn ne oferă implementări rapide pentru experimente multiple (SVM, RF, LR).
- SPAMS sau implementările custom K-SVD sunt mai specializate, pot fi testate dacă avem timp/ nevoie de performanță ridicată.
- Matplotlib/Seaborn sunt esențiale pentru a arăta clar evoluția procesării semnalelor, distribuția coeficienților, curbe ROC etc.

6 F. REZULTATE AȘTEPTATE ȘI METODOLOGIE DE EVALUARE

Cum plănuim să ne evaluăm rezultatele?

1. Scenariu de Antrenare/Test

- Vom împărți datele în "bătăi normale" și "bătăi cu aritmii cardiace" (posibil grupate pe tipuri de aritmii).
- Pe setul de test, vom rula clasificatorul și vom nota cât de des bătăile cu aritmii sunt recunoscute corect (sensibilitate) și cât de des bătăile normale sunt clasificate corect (specificitate).

2. Metrici

- **Acuratețe**
- **Sensibilitate (Recall)** și **Precizie (Precision)**
- **F1-score**
- **ROC/AUC**

3. Rezultate așteptate

- Un salt față de metoda de bază (features brute) – ne dorim o acuratețe de peste 90%.
- O îmbunătățire a sensibilității în detectarea aritmiilor cardiace (unde e critic să nu ratăm cazurile patologice).
- Demonstrarea că reprezentarea rară aduce un plus (prin compararea cu scenariul "fără sparse coding").

4. Plan de Documentare

- Vom prezenta (în raport final și în repo GitHub) grafice cu learning curves, confusion matrix, ROC curves etc.
- Vom detalia parametrii folosiți (dimensiunea dicționarului, numărul de iterații, tip de clasificator etc.) și modul în care au influențat performanța.

7 G. CONCLUZII ȘI DIRECȚII POTENȚIALE DE EXTINDERE

1. Concluzia Principală (ce dorim să demonstrăm)

- Reprezentarea rară (Sparse Coding) poate fi un mod eficient de a extrage feature-uri reprezentative dintr-un semnal EKG complex de 12 derivații, ducând la o clasificare mai robustă a bătăilor cardiace și la detecția aritmiilor cardiace.

2. Posibile Extensii / Căi pe care am putea merge ulterior

- Abordare multi-clasă (detectarea specifică a tipurilor de aritmii: fibrilație atrială, flutter, SVT etc.).
- Incorporarea altor tipuri de semnale fiziologice (ex. PPG, BP) într-o metodă multimodală.
- Sparse Autoencoder: în loc de K-SVD, am putea folosi o rețea neuronală cu constrângeri de raritate (L1) la nivelul straturilor ascunse.
- Transfer Learning: dacă avem un dicționar antrenat pe un set mare, îl putem aplica pe alt set nou cu efort minim.

- Implementare Embedded (pe un dispozitiv wearable sau Holter) pentru detecție în timp real.
- Îmbunătățirea preprocesării: teste cu diverse algoritmi de filtrare adaptivă, wavelet filtering etc.
- Explorarea altor modele: SVM cu kernel polinomial, rețele neurale cu LSTM (pentru secvențe), modele tip Transformer specializate pentru semnale EKG.

3. Beneficiile Proiectului

- Are un **impact practic** (potențial medical).
- Oferă **flexibilitate**: se pot testa mai multe variante de dicționare și clasificatori.
- Deschide **noi direcții de cercetare** și permite integrarea altor tehnologii (deep learning, big data etc.).

Mențiuni Finale

- **Proiectul este în fază inițială**: vom prezenta planul de implementare (ce am descris mai sus) și motivația.
- **Codul sursă** va fi structurat în scripturi separate pentru preprocesare, antrenare dicționar, extragere coeficienți, clasificare și evaluare.
- **Repo GitHub**: va conține tot codul și rezultatele intermediare (ploturi, fișiere JSON cu parametri etc.) pentru transparență.
- Pe parcurs, vom documenta toate încercările, inclusiv eșecurile sau abordările care nu dau rezultate bune, pentru a arăta clar procesul de învățare.

Rezumat Final – Ce Urmează Să Facem

(1) **Colectăm și preprocesăm semnalele EKG de 12 derivații**: filtru band-pass, eliminare baseline, segmentare bătăi.

(2) **Antrenăm un dicționar (K-SVD sau Online) pentru Sparse Coding**: obținem o matrice $\mathbf{D}$.

(3) **Fiecare bătaie cardiacă** (aplatizată) este reprezentată de un vector de coeficienți rari $\boldsymbol{\alpha}$.

(4) **Testăm mai multe clasificatoare** (SVM, Logistic Regression, RandomForest, eventual MLP) pentru a compara performanța.

(5) **Evaluăm** pe un set de test, calculăm metrici (acuratețe, recall, specificity, F1-score).

(6) **Comparăm** cu metode fără Sparse Coding și cu parametri diferiți.

(7) **Tragem concluzii** și **indicăm direcții viitoare** (extensii, îmbunătățiri).

Prin acest proiect, ne propunem să demonstrăm că **reprezentarea rară** poate oferi un plus de acuratețe și robusteză în **detecția automată** a aritmiilor cardiace EKG pe 12 derivații, contribuind astfel la dezvoltarea de instrumente de analiză cardiacă mai eficiente.

Bibliografie

- [**1**] *A large scale 12-lead electrocardiogram database for arrhythmia study*, `https://physionet.org/content/ecg-arrhythmia/1.0.0/`