Inteligencja obliczeniowa i jej zastosowania

Metody uczenia nienadzorowanego

Dawid Mikowski 251674 Piotr chorościn 228937

2020

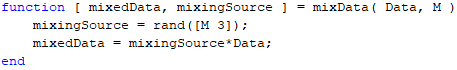
# Zadanie 1

Zaimplementować w Matlabie algorytm AMUSE I zastosować go do separacji sygnałów zmieszanych x(t), uzyskanych na podstawie modelu x(t) = As(t), gdzie s(t) są sygnałami źródłowymi z plików „Shannon\_Hurley.mat” oraz „Music\_Ozerov.mat”.

Przyjąć: M = 3 i M = 10. Oceń jakość estymowanych sygnałów oraz macierzy mieszającej, stosując miarę SIR.

## Implementacja

Stworzono funkcję pozwalającą na mieszanie danych w prosty sposób jako parametry przyjmuje ona macierz źródłową oraz M a zwraca macierz zmieszaną oraz mieszającą:



Zaimplementowano algorytm „AMUSE” w języku matlab zgodnie z instrukcją zawartą w wykładzie.

1. Na początek zapisano w zmiennej T rozmiar drugiego wymiaru macierzy sygnału czyli ilości próbek



1. Kolejnym krokiem było obliczenie macierzy scentralizowanej poprzez odjęcie od każdej kolumny średniej kolumny całej macierzy.



1. Obliczono macierz korelacji według wzoru z wykładu



1. Następnie wykonano EVD aby uzyskać macierz cech oraz macierz wartości własnych



1. Wykorzystując wcześniej uzyskane zmienne wykonano wybielanie



1. Obliczono macierz autokorelacji



1. Aby uniknąć przypadku macierzy niesymetrycznej wykonano symetryzację



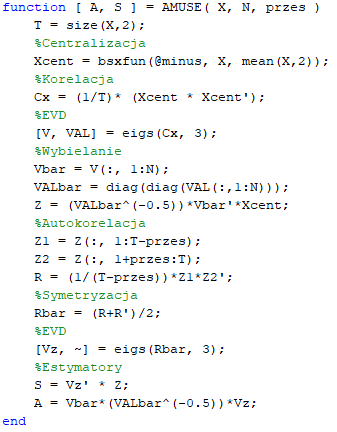
1. Ponownie wykonano EVD



1. Obliczono potrzebne estymatory



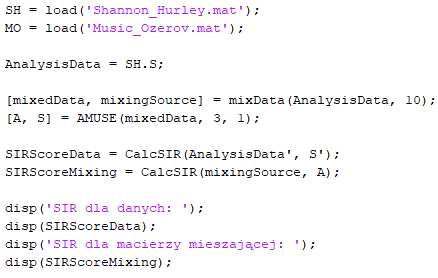
Cały algorytm prezentuje się następująco



## Testy

Następnie przetestowano algorytm na różnych plikach oraz z rożną wartością M. Testy wykonano stosując miarę SIR. Podobnie jak w poprzednich laboratoriach została użyta funkcja obliczająca miarę SIR pobrana ze strony <https://github.com/andrewssobral/TDALAB/blob/master/CalcSIR.m>

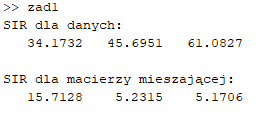
Do wykonania testów napisano skrypt wykorzystujący obie napisane funkcje a następnie obliczające ich efektywność wykorzystując miarę SIR.



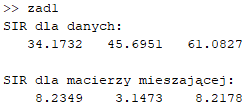
Przesunięcie pozostawiamy dla wszystkich testów równe 1.

## Wyniki

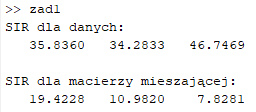
1. „Music Ozerov” M = 3



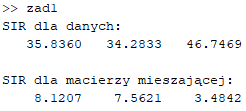
1. „Music Ozerov” M = 10



1. „Shannon Hurley” M = 3



1. „Shannon Hurley” M = 10



Jak widać na powyższych wynikach algorytm radzi sobie tak samo jeżeli chodzi o estymowaną macierz sygnałów źródłowych niezależnie od rozmiaru macierzy mieszającej. Lepsze wyniki daje analiza pliku Music Ozerov. Wyniki SIR dla estymowanej macierzy mieszającej zmniejszają się znacznie w przypadku zmiany rozmiaru tej macierzy z 3 na 10.

# Zadanie 2

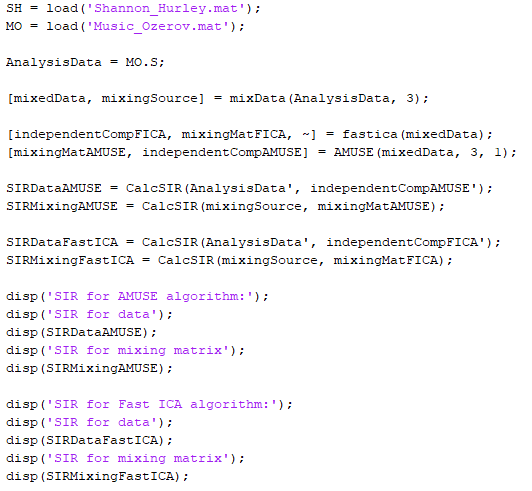
Zaimplementuj dowolną wersję algorytmu maksymalizacji kurtozy (Fast ICA), zastosuj go do sygnałów z zadania 1, porównaj wyniki z algorytmu AMUSE.

## Implementacja

Implementację algorytmu fast ICA pobrano z internetu ze strony „https://research.ics.aalto.fi/ica/fastica/code/dlcode.shtml” a następnie wykorzystano ją tak jak w zadaniu pierwszym porównując przy tym wyniki.

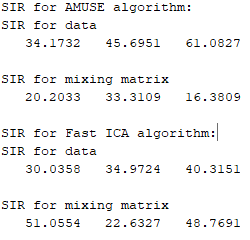
## Testy

Testy wykonano podobnie jak w zadaniu pierwszym jednak dodano obliczenie estymatorów za pomocą metody Fast ICA co pozwala na porównanie efektywności obu algorytmów.

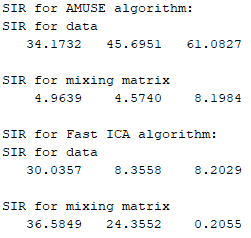


## Wyniki

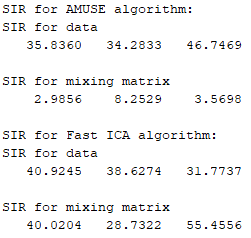
1. „Music Ozerov” M = 3



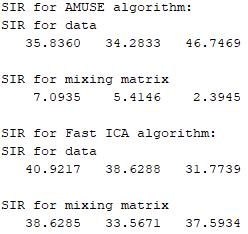
1. „Music Ozerov” M = 10



1. „Shannon Hurley” M = 3



1. „Shannon Hurley” M = 10



Analiza wyników pozwala wywnioskować że algorytm fast ICA pobrany ze strony internetowej sprawdza się dużo lepiej jeżeli naszym priorytetem jest uzyskanie poprawnej macierzy mieszającej. W takim przypadku jest on nawet 5 razy bardziej efektywny. Wyniki dla estymowanych sygnałów źródłowych pokazują, że lepszy jest algorytm AMUSE jednak nie są to różnice znaczne. Patrząc na wszystkie wyniki można podjąć decyzje że do rozwiązania problemu separacji sygnałów dużo lepiej przygotowany jest algorytm FAST ICA.