Εργασία Βιοϊατρικής

Χειμερινό εξάμηνο 2018-2019

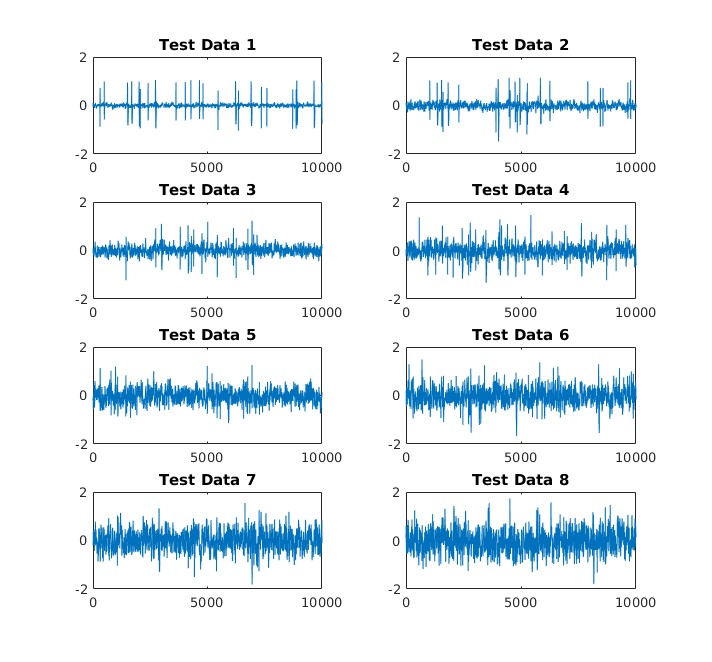
Γέροντας Αλέξανδρος, ..., [agerontas@auth.gr](mailto:agerontas@auth.gr)

Κατωμέρης Νικόλαος, 8551, [ngkatomer@auth.gr](mailto:ngkatomer@auth.gr)

Πετρίδης Σταύρος, ..., [spetridis@auth.gr](mailto:spetridis@auth.gr)

# Θέμα 1ο

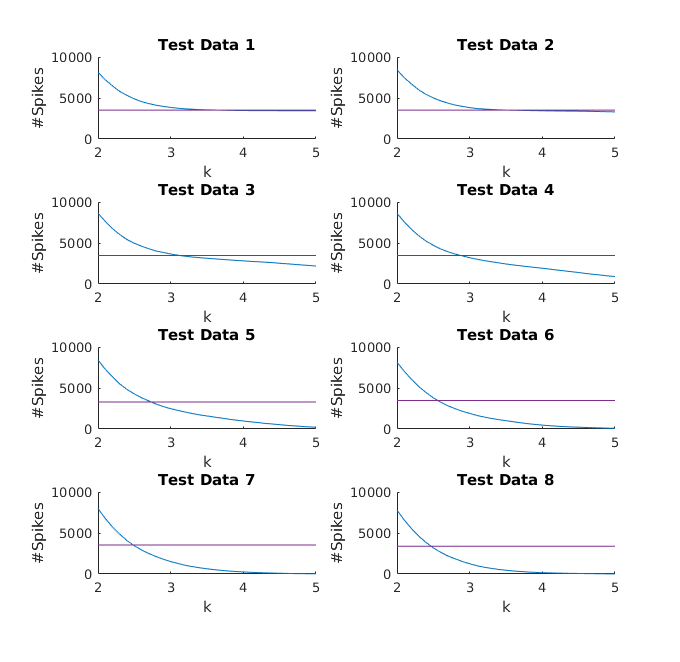
## Ερώτημα 1.1

Στην παραπάνω εικόνα παρουσιάζονται τα 10000 πρώτα δείγματα των σημάτων Data\_Test\_x.

Παρατηρούμε ότι υπάρχει διαφορά στον θόρυβο που περιέχεται στα σήματα. Συγκεκριμένα, ο θόρυβος των δεδομένων Data\_Test\_1 είναι λιγότερος από τον θόρυβο στα Data\_Test\_2 που με τη σειρά του είναι λιγότερος από αυτόν των Data\_Test\_3 κ.ο.κ.

Η διαπίστωση αυτή επαληθεύεται από τον αριθμό κορυφών που, ενώ είναι παρόμοιος για όλα τα δεδομένα, κοιτάζοντάς τα αυτό δεν είναι εμφανές λόγω της μεγάλης παρουσίας θορύβου των τελευταίων Data\_Test.

## Ερώτημα 1.2

Στο καθένα από τα παραπάνω διαγράμματα φαίνεται με οριζόντια ευθεία το σημείο του πραγματικού αριθμού κορυφών του κάθε δείγματος και η καμπύλη των ανιχνευόμενων κορυφών για τις μεταβαλλόμενες τιμές του k.

Παρατηρούμε ότι η ιδανική τιμή του k βρίσκεται σε όλες τις περιπτώσεις στο διάστημα [2, 4] ενώ μειώνεται από δείγμα σε δείγμα.

Επιπλέον, παρατηρούμε ότι στα δεδομένα που ο θόρυβος είναι χαμηλός, μπορεί να προκύψει εκτίμηση των κορυφών για αρκετά μεγάλα εύρη τιμών στο k πέρα από ένα χαμηλό όριο. Αντίθετα, όταν υπάρχει πολύς θόρυβος στα δεδομένα, η επιλογή του k μεταβάλλει σε μεγάλο βαθμό την εκτίμηση και μικρή αλλαγή στο k μπορεί να δώσει πολύ λανθασμένα αποτελέσματα.

## Ερώτημα 1.3

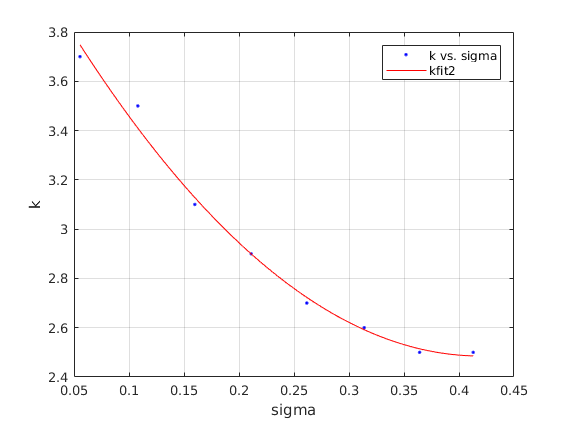
Για την εύρεση ενός εμπειρικού κανόνα για την τιμή του k, από τα 8 δείγματα παρατηρήθηκε συσχέτιση της ιδανικής τιμής του k με την εκτίμηση της τυπικής απόκλισης του θορύβου, η οποία υπολογίστηκε ως εξής:

σn = median(|singal|) / 0.6745

(Donoho, D., & Johnstone, I. M. (1994). Ideal spatial adaptation by wavelet shrinkage. Biometrika, 81, 425–455.)

Σχεδιάστηκαν τα ιδανικά k σε σχέση με τα σn των 8 περιπτώσεων και βρέθηκε εμπειρικός κανόνας που μπορεί να δώσει τιμή στο k με βάση οποιοδήποτε σn. Ο κανόνας που βρέθηκε παρούσιάζεται στο παρακάτω διάγραμμα και πρόκειται για την καμπύλη:

y = 9.518x2 - 7.99x + 4.162



Όπως φαίνεται, η καμπύλη αυτή προσεγγίζει ικανοποιητικά τις ιδανικές τιμές των k για τις 8 περιπτώσεις που ήταν διαθέσιμες.

# Θέμα 2ο

## Ερώτημα 2.1

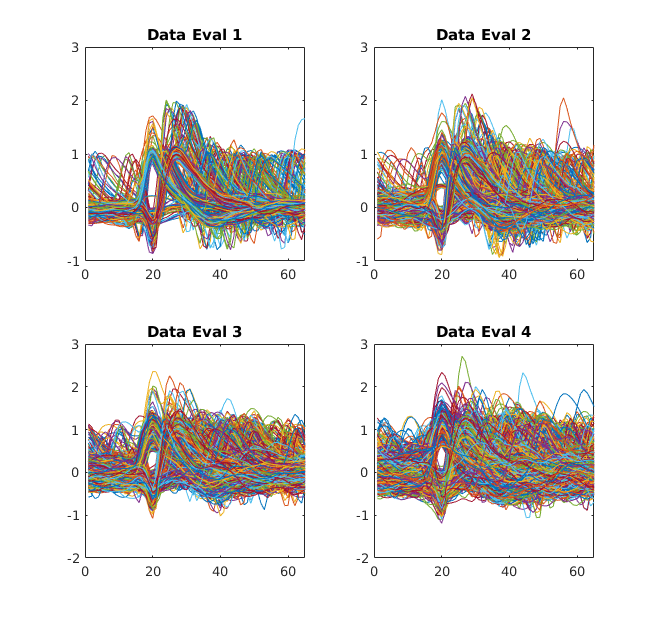
Χρησιμοποιώντας των εμπειρικό κανόνα του ερωτήματος 1.3 στα δεδομένα Data\_Eval\_E\_x

βρέθηκαν:

|  |  |
| --- | --- |
| ***Σήμα*** | ***Αριθμός κορυφών*** |
| Data\_Eval\_E\_1 | 3424 |
| Data\_Eval\_E\_2 | 3621 |
| Data\_Eval\_E\_3 | 3746 |
| Data\_Eval\_E\_4 | 4132 |

## Ερώτημα 2.2

Οι κορυφές που βρέθηκαν στο προηγούμενο ερώτημα στοιχίστηκαν και απεικονίζονται παρακάτω:



Στα διαγράμματα παρατηρούμε ότι οι κορυφές προήλθαν από διαφορετικούς νευρώνες καθώς μετά τη στοίχισή τους σχηματίζονται κάποια πρότυπα. Τα πρότυπα αυτά είναι περισσότερο εμφανή στα δεδομένα Data\_Eval\_E\_1 και λιγότερο στα επόμενα κατά σειρά, όπου ξεχωρίζουν περισσότερο 2 ομάδες κορυφών και η 3η γίνεται λιγότερο εμφανής.

## Ερώτημα 2.3

Έγινε στοίχιση των εντοπισμένων κορυφών στις πραγματικές. Για τις κορυφές που εντοπίστηκαν ισχύουν τα παρακάτω:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Δεδομένα*** | ***#Εντοπισμένες κορυφές*** | ***#Πραγματικές κορυφές*** | ***Ποσοστό αντιστοίχισης*** |
| Data\_Eval\_E\_1 | 3424 | 3410 | 98.36% |
| Data\_Eval\_E\_2 | 3621 | 3520 | 93.73% |
| Data\_Eval\_E\_3 | 3746 | 3511 | 81.39% |
| Data\_Eval\_E\_4 | 4132 | 3526 | 67.84% |

Κι εδώ, μπορεί να φανεί ότι στα δεδομένα με τον περισσότερο θόρυβο (Data\_Eval\_E\_4) η εκτίμηση του k έδωσε αρκετά περισσότερες κορυφές που στην πραγματικότητα αποτελούν θόρυβο κι όχι χρήσιμο σήμα.

Όπως σχολιάστηκε και προηγουμένως, αυτό δεν σημαίνει ότι το ιδανικό k απέχει πολύ από το k που έδωσε το μοντέλο μας, αλλά επαληθεύει ότι η αύξηση του θορύβου οδηγεί σε λάθος εκτιμήσεις για τις κορυφές ακόμη και με μία σχετικά καλή επιλογή του παράγοντα k.

Φυσικά, αν ο θόρυβος ξεπεράσει κάποιο όριο, καμιά επιλογή k δεν θα μπορεί να διαχωρίσει το χρήσιμο σήμα απ’ αυτόν.

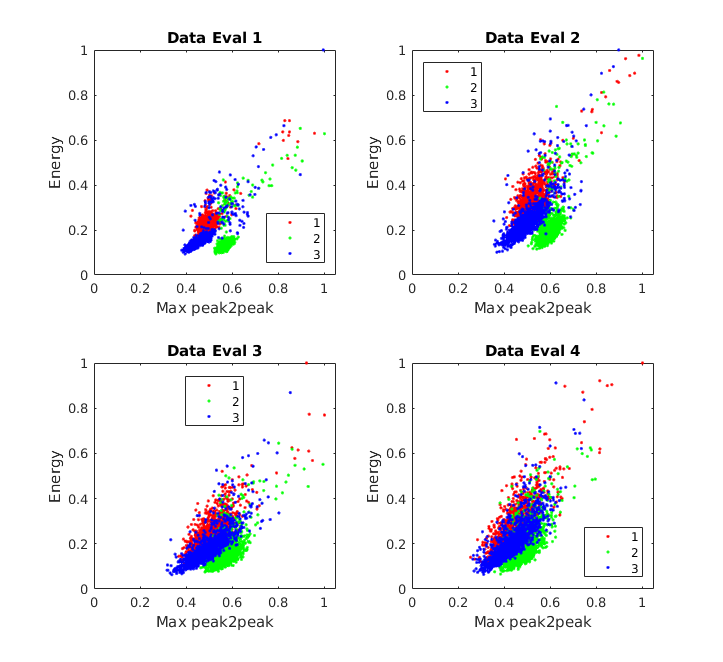
Στην προκειμένη περίπτωση μπορούμε να υποθέσουμε ότι κι εδώ τα δεδομένα Data\_Eval\_E\_4 περιέχουν πολύ περισσότερο θόρυβο από τα Data\_Eval\_E\_1.

## Ερώτημα 2.4

Για μια αρχική ομαδοποίηση των κορυφών επιλέχθηκαν τα εξής χαρακτηριστικά:

* Η ενέργεια των κορυφών.
* Το μέγιστο πλάτος των κορυφών.

Τα χαρακτηριστικά αυτά κανονικοποιήθηκαν και παρουσιάζονται παρακάτω για τις κορυφές που αντιστοιχήθηκαν σε πραγματικές κορυφές έτσι ώστε να μπορούν να χρωματιστούν με την γνωστή κλάση της κάθε πραγματικής κορυφής.



Παρατηρούμε ότι οι κορυφές της κάθε κλάσης -κατά πλειοψηφία- μπορούν να διαχωριστούν γραμμικά μεταξύ τους κι έτσι αναμένουμε σχετικώς καλά αποτελέσματα από τον αλγόριθμο διαχωρισμού.

Και πάλι, βλέπουμε ότι οι ομάδες των κορυφών των διαφορετικών σημάτων είναι περισσότερο εμφανείς για τα δεδομένα Data\_Eval\_E\_1, λιγότερο για τα Data\_Eval\_E\_2 κι ακόμη λιγότερο για τα επόμενα που περιέχουν περισσότερο θόρυβο.

## Ερώτημα 2.5

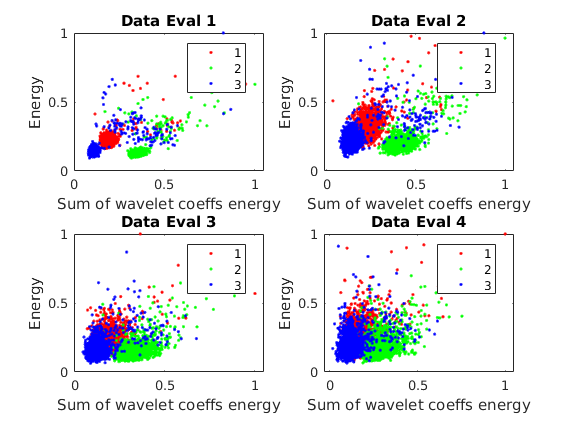
Εκτελώντας την MyClassify στα δεδομένα, με τα δύο χαρακτηριστικά που αναφέρθηκαν στο προηγούμενο ερώτημα, επιτεύχθηκαν τα εξής ποσοστά ορθής ταξινόμησης:

|  |  |
| --- | --- |
| ***Δεδομένα*** | ***Ποσοστό ορθής ταξινόμησης*** |
| Data\_Eval\_E\_1 | 87.29% |
| Data\_Eval\_E\_2 | 82.08% |
| Data\_Eval\_E\_3 | 73.52% |
| Data\_Eval\_E\_4 | 66.37% |

Η αύξηση του αριθμού των χαρακτηριστικών δεν οδηγεί απαραίτητα σε καλύτερη ταξινόμηση.

Σημαντικό ρόλο στην βελτίωση του ποσοστού ταξινόμησης έχει η επιλογή χαρακτηριστικών που οδηγούν σε καλύτερο διαχωρισμό των κορυφών που ανήκουν σε διαφορετικές ομάδες και καλύτερη “σύνδεση” των κορυφών που ανήκουν στην ίδια ομάδα.

Για παράδειγμα, η επιλογή δύο διαφορετικών χαρακτηριστικών οδήγησε σε καλύτερη ομαδοποίηση όπως φαίνεται στην επόμενη εικόνα:



Τα ποσοστά της παραπάνω ταξινόμησης, καθώς και ποσοστά ταξινομήσεων με περισσότερα χαρακτηριστικά παρουσιάζονται στον επόμενο πίνακα.

Στον πίνακα αυτό βλέπουμε ότι γενικώς η αύξηση των χαρακτηριστικών που χρησιμοποιούνται στην ταξινόμηση, οδηγεί σε καλύτερη ταξινόμηση. Όμως, επίσης γίνεται εμφανές ότι το παραπάνω συμπέρασμα δεν είναι απόλυτο και μπορεί να διαφέρει ανάλογα με τα δεδομένα, τον θόρυβο σ’ αυτά αλλά και τον αλγόριθμο της ταξινόμησης.

|  |  |  |  |
| --- | --- | --- | --- |
| ***Δεδομένα*** | ***Ποσοστό ορθής ταξινόμησης*** | | |
| ***2 χαρακτηριστικά*** | ***4 χαρακτηριστικά*** | ***6 χαρακτηριστικά*** |
| Data\_Eval\_E\_1 | 90.25% | 89.26% | 91.72% |
| Data\_Eval\_E\_2 | 85.03% | 84.94% | 85.03% |
| Data\_Eval\_E\_3 | 77.95% | 84.94% | 81.69% |
| Data\_Eval\_E\_4 | 71.65% | 74.19% | 75.76% |

# Παράρτημα

Ο κώδικας που χρησιμοποιήθηκε στην παρούσα εργασία παρουσιάζεται παρακάτω:

1ο θέμα

%% Read the test data

% Number of cases

N = 8;

% Number of points

M = 1440000;

% Vars for storing the data.

spike\_data = zeros(N, M);

spike\_num = zeros(N, 1);

for i = 1:N

filename = sprintf("Data\_Test\_%d.mat", i);

load(filename);

if length(data) ~= M

disp("Data sizes not consistent. Check the M value");

break;

end

spike\_data(i, :) = data;

spike\_num(i) = spikeNum;

clear data spikeNum;

end

%% Display the signals

sample\_size = 10000;

figure('Name', "Head of the test signals");

for i = 1:8

subplot(4,2,i)

plot(spike\_data(i, 1:sample\_size))

title(sprintf("Test Data %d", i))

end

%% Noise standard deviation estimation

sigma = median(abs(spike\_data), 2) ./ 0.6745;

%% Find spikes

ks = 2:0.1:5;

k\_size = size(ks, 2);

peak\_num = zeros(N,k\_size);

for i = 1:N

cnt = 1;

for k = ks

peak\_num(i,cnt) = sum(findpeaks(spike\_data(i,:)) > k\*sigma(i));

cnt = cnt + 1;

end

end

%% Plot spikes

figure('Name', "Spikes vs k")

for i = 1:8

subplot(4,2,i);

hold on;

plot(ks, peak\_num(i, :))

xlabel('k')

ylabel('#Spikes');

title(sprintf("Test Data %d", i))

plot(ks, spike\_num(i) \* ones(k\_size))

end

%% Best ks

k\_best = zeros(N, 1);

for i = 1:N

[~, idx] = min(abs(peak\_num(i,:)-spike\_num(i)));

k\_best(i) = ks(idx);

end

%% Data for creating models

model\_sigmas = sigma;

model\_k = k\_best;

%% Create model for k

k\_model = kfit2(model\_sigmas, model\_k);

clear sigma k\_best

2ο θέμα

%% Read the new data

% Number of cases

N = 4;

% Number of points

M = 1440000;

% Vars for storing the data.

spike\_data = zeros(N, M);

spike\_class = cell(N, 1);

spike\_times = cell(N, 1);

for i = 1:N

filename = sprintf("Data\_Eval\_E\_%d.mat", i);

load(filename);

if length(data) ~= M

disp("Data sizes not consistent. Check the M value");

break;

end

spike\_data(i, :) = data;

spike\_class{i} = spikeClass;

spike\_times{i} = spikeTimes;

clear data spikeClass spikeTimes;

end

%% Noise standard deviation estimation

sigma = median(abs(spike\_data), 2) ./ 0.6745;

%% Evaluate new data

k\_predicted = zeros(N,1);

for i=1:N

k\_predicted(i) = feval(k\_model, sigma(i));

end

%% Find spikes

spikeNumEst = zeros(N,1);

spikeTimesEst = cell(N, 1);

spikeHeigh = cell(N, 1);

for i = 1:N

[peaks, locs] = findpeaks(spike\_data(i,:)); %, 'MinPeakDistance', 64);

threshold = k\_predicted(i) \* sigma(i);

valid\_locs = peaks > threshold;

spikeHeigh{i} = peaks(valid\_locs);

spikeTimesEst{i} = locs(valid\_locs);

spikeNumEst(i) = size(spikeHeigh{i}, 2);

end

clear valid\_locs threshold

%% Plot spikes

j = 1;

spikeEst = cell(N, 1);

figure('Name', "Sorted peaks")

for j=1:N

subplot(2,2,j)

spikeEst{j} = zeros(spikeNumEst(j), 65);

for i=1:spikeNumEst(j)

search\_range = spikeTimesEst{j}(i) - 30 : spikeTimesEst{j}(i) + 30;

[~, min\_idx] = min(spike\_data(j, search\_range));

[~, max\_idx] = max(spike\_data(j, search\_range));

center = min(min\_idx, max\_idx) + spikeTimesEst{j}(i) - 30;

spikeEst{j}(i,:) = spike\_data(j, center-20:center+44);

end

plot(spikeEst{j}')

title(sprintf("Data Eval %d", j))

xlim([0 65])

end

clear min\_idx max\_idx center search\_range

%% Find spike-estimated spike pairs

spike\_pairs = cell(4, 1);

for j=1:N

cnum = size(spike\_times{j}, 2);

next\_id = 1;

spike\_pairs{j} = zeros(spikeNumEst(j), 1, 'uint32');

for i=1:cnum

if next\_id > spikeNumEst(j)

break;

end

while next\_id < spikeNumEst(j) && spike\_times{j}(i) > spikeTimesEst{j}(next\_id)

next\_id = next\_id + 1;

end

if spikeTimesEst{j}(next\_id)-spike\_times{j}(i) < 33

spike\_pairs{j}(next\_id) = i;

next\_id = next\_id + 1;

end

end

end

clear next\_id

%% Percentage of spike-pairing

pairing\_pc = zeros(N, 1);

for i=1:N

dif = sum(spike\_pairs{i} == 0) + spikeNumEst(i) - size(spike\_times{i}, 2);

pairing\_pc(i) = (spikeNumEst(i) - dif) / spikeNumEst(i);

end

%% spike classification

maxA = cell(N, 1);

zc = cell(N, 1);

WLenergy = cell(N, 1);

means = cell(N, 1);

energy = cell(N, 1);

cycles = cell(N, 1);

for j = 1:N

maxA{j} = zeros(spikeNumEst(j), 1);

zc{j} = zeros(spikeNumEst(j), 1);

WLenergy{j} = zeros(spikeNumEst(j), 1);

means{j} = zeros(spikeNumEst(j), 1);

energy{j} = zeros(spikeNumEst(j), 1);

cycles{j} = zeros(spikeNumEst(j), 1);

for i = 1:spikeNumEst(j)

spike = spikeEst{j}(i, :);

maxA{j}(i) = max(spike)-min(spike);

zc{j}(i) = sum(spike(1:end-1) <= 0 & spike(2:end) > 0) + sum(spike(1:end-1) > 0 & spike(2:end) > 0);

means{j}(i) = mean(spike);

F = fft(spike);

WL = cwt(spike);

energy{j}(i) = sum(F.\*conj(F));

WLenergy{j}(i) = sum(sum(WL.\*conj(WL)));

cycles{j}(i) = mean(diff(spike));

end

end

clear spike

%% Extract characteristics

Data = cell(N, 1);

group = cell(N, 1);

for j = 1:N

valid\_idx = spike\_pairs{j} ~= 0;

found = spike\_pairs{j}(valid\_idx);

Data{j}(:,1) = zc{j}(valid\_idx);

Data{j}(:,2) = maxA{j}(valid\_idx);

Data{j}(:,3) = means{j}(valid\_idx);

Data{j}(:,4) = WLenergy{j}(valid\_idx);

Data{j}(:,5) = energy{j}(valid\_idx);

Data{j}(:,6) = cycles{j}(valid\_idx);

% Normalize

Data{j} = Data{j} ./ max(Data{j});

group{j}(:,1) = spike\_class{j}(found);

end

%% Scatter plot

figure('Name', 'Scatter plots')

for j = 1:N

subplot(2,2, j)

gscatter(Data{j}(:,4), Data{j}(:,5), group{j})

title(sprintf("Data Eval %d", j));

xlabel("Sum of wavelet coeffs energy")

ylabel("Energy")

end

%% Classification. Modify [1:6] to select specific data.

pcs = zeros(4, 1);

for i=1:N

pcs(i) = MyClassify(Data{i}(:, [1:6]), group{i});

end

Συνάρτηση για την εύρεση του κανόνα για το k

function [fitresult, gof] = kfit2(model\_sigmas, model\_k)

%KFIT2(MODEL\_SIGMAS,MODEL\_K)

% Create a fit.

%

% Data for 'k\_fit2' fit:

% X Input : model\_sigmas

% Y Output: model\_k

% Output:

% fitresult : a fit object representing the fit.

% gof : structure with goodness-of fit info.

%

% See also FIT, CFIT, SFIT.

% Auto-generated by MATLAB on 29-Dec-2018 15:46:05

%% Fit: 'k\_fit2'.

[xData, yData] = prepareCurveData( model\_sigmas, model\_k );

% Set up fittype and options.

ft = fittype( 'poly2' );

% Fit model to data.

[fitresult, gof] = fit( xData, yData, ft );

% Plot fit with data.

figure( 'Name', 'kfit2' );

h = plot( fitresult, xData, yData );

legend( h, 'k vs. sigma', 'kfit2', 'Location', 'NorthEast' );

% Label axes

xlabel sigma

ylabel k

grid on