Contents

Lab 3: CuSum Algorithm for abrupt mean change

```
clear all;
clc;
close all;
```

Data generation

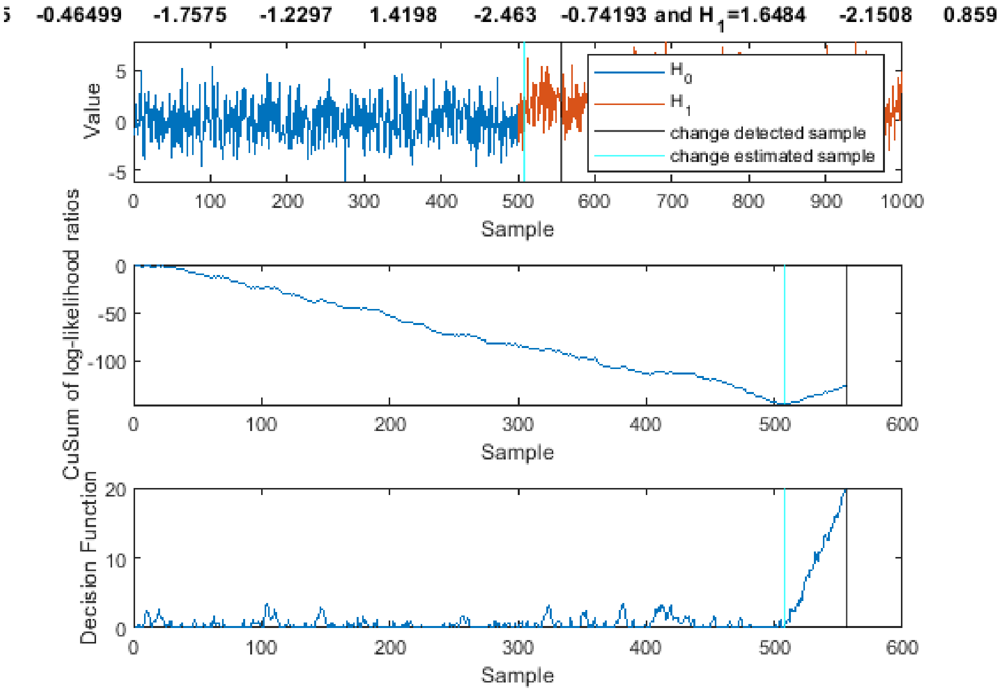
```
k = 1000;       % number of samples
sigma = sqrt(4);% variance
mu_0 = 0;       % H_0 average
mu_1 = 1.5;     % H_1 average
delta = abs(mu_1 - mu_0);   % difference between mu_0 and mu_1

H_0 = mu_0 + sigma * randn(1, k/2);   % H_0 samples
H_1 = mu_1 + sigma * randn(1, k/2);   % H_1 samples
```

Implementing CuSum Algo

```
x = [H_0, H_1];     % concatenate data for algorithm testing
threshold = 20;     % threshold above which a change will have been detected

% initialization
sample = 1;
S = zeros(1, k);    % cumulative sum
g = zeros(1, k);    % over threshold count
change_detected_sample = 0;
change_estimated_sample = 0;
while change_detected_sample == 0
    sample = sample + 1;
    % calculate instantaneous log-likelihood ratio
    inst_llr = (delta/(sigma^2)) * (x(sample) - mu_0 - (delta/2));
    % calculate "S", integration of log-likelihood ratios
    S(sample) = S(sample - 1) + inst_llr;
    % calculate "g", decision function
    g(sample) = max(g(sample - 1) + inst_llr, 0);
    % react if "g" is above threshold
    if g(sample) > threshold
        disp(['Change detected at sample number: ', num2str(sample)]);
        change_detected_sample = sample;
        [min_val, min_sample_index] = min(S);
        disp(['Estimated abrupt change occured at sample number: ', num2str(min_sample_index)]);
        change_estimated_sample = min_sample_index;
    end
end

figure
subplot(3, 1, 1);
plot((1:k/2), H_0, (k/2 + 1:k), H_1);
xlabel('Sample');
ylabel('Value');
xline(sample, 'k');
xline(min_sample_index, 'c');
```

```
legend('H_0', 'H_1', 'change detected sample', 'change estimated sample');
title(['Simulated samples with mean H_0=', num2str(H_0), ' and H_1=', num2str(H_1), ', and with sigma=', num2str(sigma)]);

x = (1:change_detected_sample);
subplot(3, 1, 2);
plot(x, S(1:change_detected_sample));
xline(sample, 'k');
xline(min_sample_index, 'c');
xlabel('Sample')
ylabel('CuSum of log-likelihood ratios');

subplot(3, 1, 3);
plot(x, g(1:change_detected_sample));
xline(sample, 'k');
xline(min_sample_index, 'c');
xlabel('Sample');
ylabel('Decision Function');
```

```
Change detected at sample number: 557
Estimated abrupt change occured at sample number: 508
```



Settings and Performance Analysis

```
FAR = 10^-3;    % set false alarm rate
% calculate threshold
threshold = abs(log(FAR))
E_H_1 = (delta^2) / (2  * sigma^2);
CADD = threshold / E_H_1
```

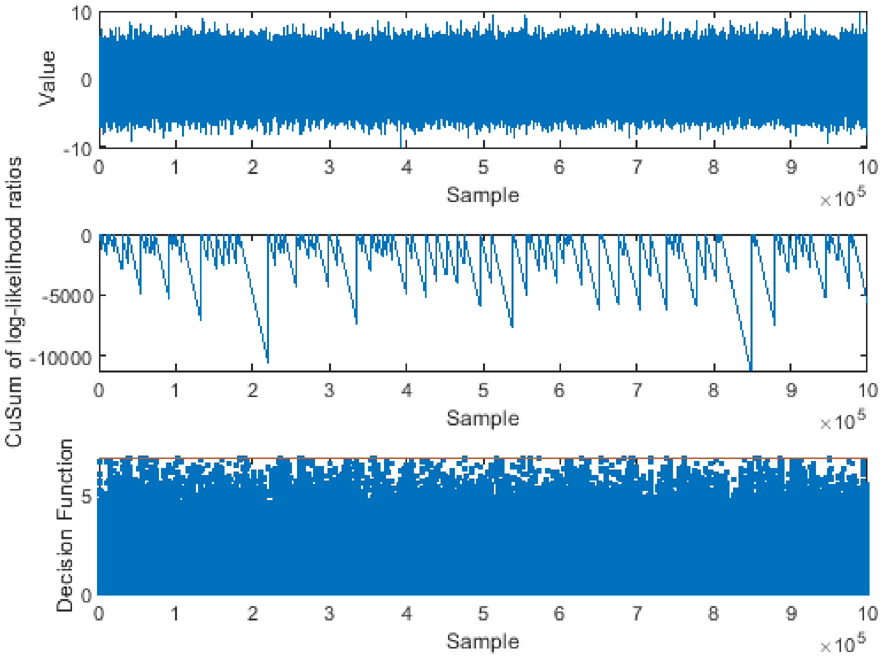
```
threshold =

    6.9078


CADD =
   24.5609
```

Experimental validation of FAR and CADD with calculated threshold

Experimental validation of false alarm rate

```
v = 1000*k;  % number of validation samples
x = mu_0 + sigma * randn(1, v);   % generate new H_0 samples

sample = 1;
changes_detected = 0;
S = zeros(1, v);    % cumulative sum
g = zeros(1, v);    % over threshold count
change_detected_sample = 0;
for i = (1:v - 1)
    sample = sample + 1;
    % calculate instantaneous log-likelihood ratio
    inst_llr = (delta/(sigma^2)) * (x(sample) - mu_0 - (delta/2));
    % calculate "S", integration of log-likelihood ratios
    S(sample) = S(sample - 1) + inst_llr;
    % calculate "g", decision function
    g(sample) = max(g(sample - 1) + inst_llr, 0);
    % react if "g" is above threshold
    if g(sample) > threshold
        % Reset algorithm if a false alarm was reached
        S(sample) = 0;
        g(sample) = 0;
        changes_detected = changes_detected + 1;
    end
end

disp(['Number of changes detected: ', num2str(changes_detected)]);

figure
subplot(3, 1, 1);
plot((1:v), x)
xlabel('Sample')
ylabel('Value');

subplot(3, 1, 2);
plot((1:v), S)
xlabel('Sample')
ylabel('CuSum of log-likelihood ratios');

subplot(3, 1, 3);
plot((1:v), g, '.', (1:v), threshold*ones(1,v));
xlabel('Sample');
ylabel('Decision Function');

FAR_exp = changes_detected / v
```

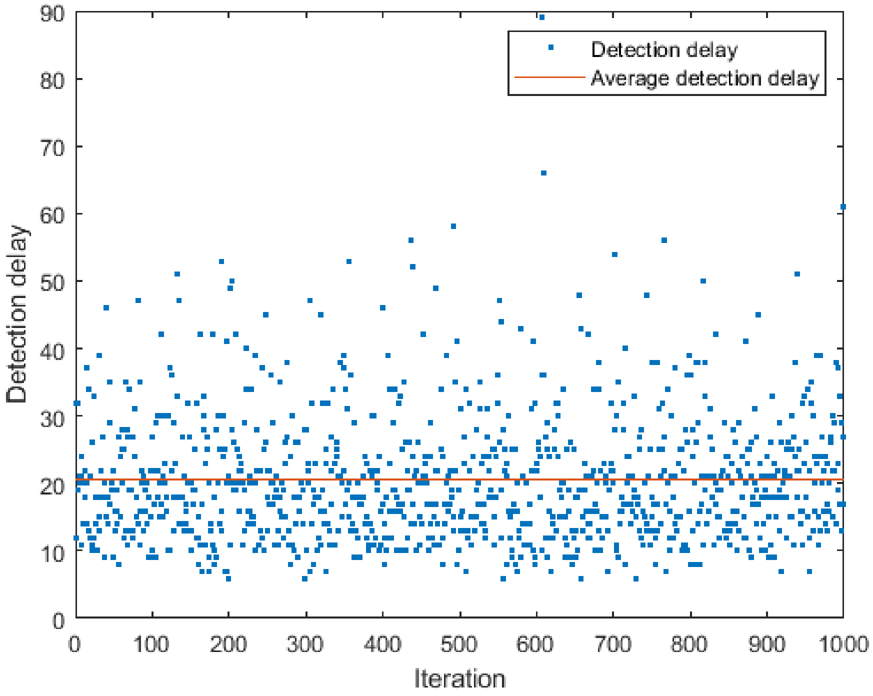
```
Number of changes detected: 108

FAR_exp =

   1.0800e-04
```



Experimental validation of CADD

```
iterations = 1000;
detection_delays = zeros(1, iterations);
for iter = (1:iterations)
    x = mu_1 + sigma * randn(1, k/2);   % generate new H_1 samples
    sample = 1;
    S = zeros(1, k/2);    % cumulative sum
    g = zeros(1, k/2);    % over threshold count
    change_detected_sample = 0;
    change_estimated_sample = 0;
    while change_detected_sample == 0
        sample = sample + 1;
        % calculate instantaneous log-likelihood ratio
        inst_llr = (delta/(sigma^2)) * (x(sample) - mu_0 - (delta/2));
        % calculate "S", integration of log-likelihood ratios
        S(sample) = S(sample - 1) + inst_llr;
        % calculate "g", decision function
        g(sample) = max(g(sample - 1) + inst_llr, 0);
        % react if "g" is above threshold
        if g(sample) > threshold
            % disp(['Change detected at sample #: ', num2str(sample)]);
            change_detected_sample = sample;
            [min_val, min_sample_index] = min(S);
            % disp(['Estimated abrupt change occured at sample #: ', num2str(min_sample_index)]);
            change_estimated_sample = min_sample_index;
            detection_delays(iter) = change_detected_sample - change_estimated_sample;
        end
    end
end
figure
CADD_exp = (sum(detection_delays) / iterations);
plot((1:iterations), detection_delays, '.', (1:iterations), CADD_exp*ones(1,iterations), '-')
xlabel('Iteration')
ylabel('Detection delay')
legend('Detection delay', 'Average detection delay')
disp(['Conditional average detection delay over ' num2str(iterations), ' iterations is ', num2str(CADD_exp)]);
```

```
Conditional average detection delay over 1000 iterations is 20.602
```



Published with MATLAB® R2018b