% The second chapter script

%% Load data

load(“Afghanistan.mat");

%% Cleaning and save of cleaned data

Pop = 229,263; % population

start = 100; % Minimal number of infected to consider start

data(data < start) = [];

data = data / Pop;

save(“Afghnistan.mat", 'data');

%% Load previously saved normalised data

load(“AfghanistanNorm.mat");

%% Draw original data

figure;

plot(data);

xlabel("Days from beginning of epidemic");

ylabel("Fraction of infected population");

%% Draw Logarithm data

figure;

plot(log(data));

xlabel("Days from beginning of epidemic");

ylabel("Log of fraction of infected population");

%% Initial conditions

R0 = 28 / Pop;

I0 = data(1) - R0;

S0 = 1 - I0 - R0;

%% Parameters estimation

from = 4 %1; % 1 22

to = 83; % 19 46

waveEnd = 270; %176; % 255

b = 0.1;

% Linear regression

mdl = fitlm(from:to, log(data(from:to)));

r = table2array(mdl.Coefficients(2, 1));

% c = table2array(mdl.Coefficients(1, 1));

% The last coefficient

a = r + b;

%% Integrate system of ODE [1, length(data)] 1:length(data)

[t, y] = ode45(@(t, y) sir(t, y, a, b), 1:length(data), [S0, R0]);

% Accuracy estimation

p = 1 - y(:, 1);

MSE = sum((data(1:waveEnd) - p(1:waveEnd)) .^ 2) / waveEnd;

%% Draw figure

figure;

plot(data);

lims = ylim();

hold on;

plot(p);

legend('Observed', 'Predicted', 'Location', 'northwest');

ylim(lims);

xlabel("Days from beginning of epidemic");

ylabel("Fraction of infected population");

%% SIR graph

figure();

yyaxis left

plot(y(:,1),'linewidth',2);

hold on

yyaxis right

plot(y(:,2), 'b');

plot(1 - y(:,1) - y(:,2),'g');

legend('S', 'R', 'I');

xlabel("Days from beginning of epidemic");

ylabel("Fraction of infected population");

%% Predicted vs observed graph

figure;

plot(data, p(1:length(data)), 'r-');

hold on;

mi = min([data, p(1:length(data))]);

ma = max([data, p(1:length(data))]);

plot([mi, ma], [mi, ma], 'b-');

xlabel("Observed fraction of infected population");

ylabel("Predicted fraction of infected population");

%% Play with original conditions

figure

plot(data);

hold on;

lims = ylim();

labs = {'Observed'};

——————————————————————

%% Parameters estimation

from = 99 %1; % 1 22

to = 395; % 19 46

waveEnd = 270; %176; % 255

b = 0.1;

% Linear regression

mdl = fitlm(from:to, log(data(from:to)));

r = table2array(mdl.Coefficients(2, 1));

% c = table2array(mdl.Coefficients(1, 1));

% The last coefficient

a = r + b;

——————————————————————

%% Parameters estimation

from = 4 %1; % 1 22

to = 395; % 19 46

waveEnd = 270; %176; % 255

b = 0.1;

% Linear regression

mdl = fitlm(from:to, log(data(from:to)));

r = table2array(mdl.Coefficients(2, 1));

% c = table2array(mdl.Coefficients(1, 1));

% The last coefficient

a = r + b;

——————————————————————————————————————————————————————————————————————————————

% The second chapter script

%% Load data

load(“’China.mat");

%% Cleaning and save of cleaned data

Pop = 99,320,723; % population

start = 100; % Minimal number of infected to consider start

data(data < start) = [];

data = data / Pop;

save(“China.mat", 'data');

%% Load previously saved normalised data

load(“ChinaNorm.mat");

%% Draw original data

figure;

plot(data);

xlabel("Days from beginning of epidemic");

ylabel("Fraction of infected population");

%% Draw Logarithm data

figure;

plot(log(data));

xlabel("Days from beginning of epidemic");

ylabel("Log of fraction of infected population");

%% Initial conditions

R0 = 28 / Pop;

I0 = data(1) - R0;

S0 = 1 - I0 - R0;

%% Parameters estimation

from = 1; %1; % 1 22

to = 34; % 19 46

waveEnd = 270; %176; % 255

b = 0.1;

% Linear regression

mdl = fitlm(from:to, log(data(from:to)));

r = table2array(mdl.Coefficients(2, 1));

% c = table2array(mdl.Coefficients(1, 1));

% The last coefficient

a = r + b;

%% Integrate system of ODE [1, length(data)] 1:length(data)

[t, y] = ode45(@(t, y) sir(t, y, a, b), 1:length(data), [S0, R0]);

% Accuracy estimation

p = 1 - y(:, 1);

MSE = sum((data(1:waveEnd) - p(1:waveEnd)) .^ 2) / waveEnd;

%% Draw figure

figure;

plot(data);

lims = ylim();

hold on;

plot(p);

legend('Observed', 'Predicted', 'Location', 'northwest');

ylim(lims);

xlabel("Days from beginning of epidemic");

ylabel("Fraction of infected population");

%% SIR graph

figure();

yyaxis left

plot(y(:,1),'linewidth',2);

hold on

yyaxis right

plot(y(:,2), 'b');

plot(1 - y(:,1) - y(:,2),'g');

legend('S', 'R', 'I');

xlabel("Days from beginning of epidemic");

ylabel("Fraction of infected population");

%% Predicted vs observed graph

figure;

plot(data, p(1:length(data)), 'r-');

hold on;

mi = min([data, p(1:length(data))]);

ma = max([data, p(1:length(data))]);

plot([mi, ma], [mi, ma], 'b-');

xlabel("Observed fraction of infected population");

ylabel("Predicted fraction of infected population");

%% Play with original conditions

figure

plot(data);

hold on;

lims = ylim();

labs = {'Observed'};

I00 = I0 / 4;

MNSs = zeros(1, 5);

for k = 1:5

S0 = 1 - I00 - R0;

[t, y] = ode45(@(t, y) sir(t, y, a, b), 1:length(data), [S0, R0]);

p = 1 - y(:, 1);

MNSs(k) = mean(sum((p-data) .^ 2));

plot(p);

labs = [labs, sprintf("I(1) = %.4g", I00)];

I00 = I00 \* 2;

end

legend(labs, 'Location', 'northwest');

ylim(lims);

xlabel("Days from beginning of epidemic");

ylabel("Fraction of infected population");

figure

plot([1:5], MNSs);

%% Play with original conditions with optimisation

figure

plot(data);

hold on;

lims = ylim();

labs = {'Observed'};

I00 = I0 / 2;

for k = 1:5

S0 = 1 - I00 - R0;

aOpt = fminbnd(@(x) forSearch(x, 160, b, [S0, R0], data), b, 3 \* b);

[t, y] = ode45(@(t, y) sir(t, y, aOpt, b), 1:length(data), [S0, R0]);

p = 1 - y(:, 1);

plot(p);

labs = [labs, sprintf("I(1) = %.4g, a = %.4f", I00, aOpt)];

I00 = I00 \* 2;

end

legend(labs, 'Location', 'northwest');

ylim(lims);

xlabel("Days from beginning of epidemic");

ylabel("Fraction of infected population");

%% Search optimal parameters

init = [S0, R0];

%aOpt = fminbnd(@(x) forSearch(x, waveEnd, b, init, data), 0.1, 0.2);

aOpt = fminbnd(@(x) forSearch(x, length(data), b, init, data), 0.1, 0.2);

% Tests

%[t, y] = ode45(@(t, y) sir(t, y, aOpt, b), 1:length(data), [S0, R0]);

[t, y] = ode45(@(t, y) sir(t, y, aOpt, b), 1:2000, [S0, R0]);

% Accuracy estimation

p = 1 - y(:, 1);

MSE = sum((data(1:waveEnd) - p(1:waveEnd)) .^ 2) / waveEnd;

%% Further prediction

bt = 0.8;

aOpt = fminbnd(@(x) forSearch(x, length(data), bt, init, data), 0.1, 1);

[t, y] = ode45(@(t, y) sir(t, y, aOpt, bt), 1:2000, [S0, R0]);

figure;

plot(1 - y(:, 1));

hold on;

plot(data);

xlabel("Days from beginning of epidemic");

ylabel("Fraction of infected population");

%% Search of all parameters simultaneously

% We will use a, b, and I0 to minimise MSE

% vector x contains [a, b, I0]

x = fminsearch(@(x) toMinimise(x, @sir, data), [a, b, I0]);

% Calculate accuracy

[~, y] = ode45(@(t, y) sir(t, y, x(1), x(2)), 1:length(data),...

[1 - 5.407493240633449e-07 - x(3) , 5.407493240633449e-07]);

p = 1 - y(:, 1);

err = mean((p - data) .^ 2);

%% Search of all parameters simultaneously

% We will use a and b to minimise MSE

% vector x contains [a, b]

[x, val] = fminsearch(@(x) toMinimise2(x, @sir, data), [a, b]);

%

[~, y] = ode45(@(t, y) sir(t, y, x(1), x(2)), 1:length(data),...

[1 - 5.407493240633449e-07 - 2.471996910003863e-06 , 5.407493240633449e-07]);

p = 1 - y(:, 1);

err = mean((p - data) .^ 2);

%% functions

function err = toMinimise(x, sir, data)

[~, y] = ode45(@(t, y) sir(t, y, x(1), x(2)), 1:length(data),...

[1 - 5.407493240633449e-07 - x(3) , 5.407493240633449e-07]);

p = 1 - y(:, 1);

err = mean((p - data) .^ 2);

end

function err = toMinimise2(x, sir, data)

[~, y] = ode45(@(t, y) sir(t, y, x(1), x(2)), 1:length(data),...

[1 - 5.407493240633449e-07 - 2.471996910003863e-06 , 5.407493240633449e-07]);

p = 1 - y(:, 1);

err = mean((p - data) .^ 2);

end

function dydt = sir(t, y, a, b)

% y(1) = S(t)

% y(2) = R(t)

I = 1 - sum(y);

dydt = [ - a \* y(1) \* I;

b \* I];

end

function err = forSearch(a, wEnd, b, y0, data)

[t, y] = ode45(@(t, y) sir(t, y, a, b), 1:wEnd, y0);

p = 1 - y(:, 1);

err = sum((data(1:wEnd) - p(1:wEnd)) .^ 2) / wEnd;

end

———————————————————

%% Parameters estimation

from = 42; %1; % 1 22

to = 81; % 19 46

waveEnd = 270; %176; % 255

b = 0.1;

% Linear regression

mdl = fitlm(from:to, log(data(from:to)));

r = table2array(mdl.Coefficients(2, 1));

% c = table2array(mdl.Coefficients(1, 1));

% The last coefficient

a = r + b;

———————————————————

%% Parameters estimation

from = 1; %1; % 1 22

to = 81; % 19 46

waveEnd = 270; %176; % 255

b = 0.1;

% Linear regression

mdl = fitlm(from:to, log(data(from:to)));

r = table2array(mdl.Coefficients(2, 1));

% c = table2array(mdl.Coefficients(1, 1));

% The last coefficient

a = r + b;

——————————————————————————————————

————————————————————————————————————-

% The second chapter script

%% Load data

load("India.mat");

%% Cleaning and save of cleaned data

Pop = 45,002,179; % population

start = 100; % Minimal number of infected to consider start

data(data < start) = [];

data = data / Pop;

save("India.mat", 'data');

%% Load previously saved normalised data

load(“IndiaNorm.mat");

%% Draw original data

figure;

plot(data);

xlabel("Days from beginning of epidemic");

ylabel("Fraction of infected population");

%% Draw Logarithm data

figure;

plot(log(data));

xlabel("Days from beginning of epidemic");

ylabel("Log of fraction of infected population");

%% Initial conditions

R0 = 28 / Pop;

I0 = data(1) - R0;

S0 = 1 - I0 - R0;

%% Parameters estimation

from = 1; %1; % 1 22

to = 38; % 19 46

waveEnd = 270; %176; % 255

b = 0.1;

% Linear regression

mdl = fitlm(from:to, log(data(from:to)));

r = table2array(mdl.Coefficients(2, 1));

% c = table2array(mdl.Coefficients(1, 1));

% The last coefficient

a = r + b;

%% Integrate system of ODE [1, length(data)] 1:length(data)

[t, y] = ode45(@(t, y) sir(t, y, a, b), 1:length(data), [S0, R0]);

% Accuracy estimation

p = 1 - y(:, 1);

MSE = sum((data(1:waveEnd) - p(1:waveEnd)) .^ 2) / waveEnd;

%% Draw figure

figure;

plot(data);

lims = ylim();

hold on;

plot(p);

legend('Observed', 'Predicted', 'Location', 'northwest');

ylim(lims);

xlabel("Days from beginning of epidemic");

ylabel("Fraction of infected population");

%% SIR graph

figure();

yyaxis left

plot(y(:,1),'linewidth',2);

hold on

yyaxis right

plot(y(:,2), 'b');

plot(1 - y(:,1) - y(:,2),'g');

legend('S', 'R', 'I');

xlabel("Days from beginning of epidemic");

ylabel("Fraction of infected population");

%% Predicted vs observed graph

figure;

plot(data, p(1:length(data)), 'r-');

hold on;

mi = min([data, p(1:length(data))]);

ma = max([data, p(1:length(data))]);

plot([mi, ma], [mi, ma], 'b-');

xlabel("Observed fraction of infected population");

ylabel("Predicted fraction of infected population");

%% Play with original conditions

figure

plot(data);

hold on;

lims = ylim();

labs = {'Observed'};

I00 = I0 / 4;

MNSs = zeros(1, 5);

for k = 1:5

S0 = 1 - I00 - R0;

[t, y] = ode45(@(t, y) sir(t, y, a, b), 1:length(data), [S0, R0]);

p = 1 - y(:, 1);

MNSs(k) = mean(sum((p-data) .^ 2));

plot(p);

labs = [labs, sprintf("I(1) = %.4g", I00)];

I00 = I00 \* 2;

end

legend(labs, 'Location', 'northwest');

ylim(lims);

xlabel("Days from beginning of epidemic");

ylabel("Fraction of infected population");

figure

plot([1:5], MNSs);

%% Play with original conditions with optimisation

figure

plot(data);

hold on;

lims = ylim();

labs = {'Observed'};

I00 = I0 / 2;

for k = 1:5

S0 = 1 - I00 - R0;

aOpt = fminbnd(@(x) forSearch(x, 160, b, [S0, R0], data), b, 3 \* b);

[t, y] = ode45(@(t, y) sir(t, y, aOpt, b), 1:length(data), [S0, R0]);

p = 1 - y(:, 1);

plot(p);

labs = [labs, sprintf("I(1) = %.4g, a = %.4f", I00, aOpt)];

I00 = I00 \* 2;

end

legend(labs, 'Location', 'northwest');

ylim(lims);

xlabel("Days from beginning of epidemic");

ylabel("Fraction of infected population");

%% Search optimal parameters

init = [S0, R0];

%aOpt = fminbnd(@(x) forSearch(x, waveEnd, b, init, data), 0.1, 0.2);

aOpt = fminbnd(@(x) forSearch(x, length(data), b, init, data), 0.1, 0.2);

% Tests

%[t, y] = ode45(@(t, y) sir(t, y, aOpt, b), 1:length(data), [S0, R0]);

[t, y] = ode45(@(t, y) sir(t, y, aOpt, b), 1:2000, [S0, R0]);

% Accuracy estimation

p = 1 - y(:, 1);

MSE = sum((data(1:waveEnd) - p(1:waveEnd)) .^ 2) / waveEnd;

%% Further prediction

bt = 0.8;

aOpt = fminbnd(@(x) forSearch(x, length(data), bt, init, data), 0.1, 1);

[t, y] = ode45(@(t, y) sir(t, y, aOpt, bt), 1:2000, [S0, R0]);

figure;

plot(1 - y(:, 1));

hold on;

plot(data);

xlabel("Days from beginning of epidemic");

ylabel("Fraction of infected population");

%% Search of all parameters simultaneously

% We will use a, b, and I0 to minimise MSE

% vector x contains [a, b, I0]

x = fminsearch(@(x) toMinimise(x, @sir, data), [a, b, I0]);

% Calculate accuracy

[~, y] = ode45(@(t, y) sir(t, y, x(1), x(2)), 1:length(data),...

[1 - 5.407493240633449e-07 - x(3) , 5.407493240633449e-07]);

p = 1 - y(:, 1);

err = mean((p - data) .^ 2);

%% Search of all parameters simultaneously

% We will use a and b to minimise MSE

% vector x contains [a, b]

[x, val] = fminsearch(@(x) toMinimise2(x, @sir, data), [a, b]);

%

[~, y] = ode45(@(t, y) sir(t, y, x(1), x(2)), 1:length(data),...

[1 - 5.407493240633449e-07 - 2.471996910003863e-06 , 5.407493240633449e-07]);

p = 1 - y(:, 1);

err = mean((p - data) .^ 2);

%% functions

function err = toMinimise(x, sir, data)

[~, y] = ode45(@(t, y) sir(t, y, x(1), x(2)), 1:length(data),...

[1 - 5.407493240633449e-07 - x(3) , 5.407493240633449e-07]);

p = 1 - y(:, 1);

err = mean((p - data) .^ 2);

end

function err = toMinimise2(x, sir, data)

[~, y] = ode45(@(t, y) sir(t, y, x(1), x(2)), 1:length(data),...

[1 - 5.407493240633449e-07 - 2.471996910003863e-06 , 5.407493240633449e-07]);

p = 1 - y(:, 1);

err = mean((p - data) .^ 2);

end

function dydt = sir(t, y, a, b)

% y(1) = S(t)

% y(2) = R(t)

I = 1 - sum(y);

dydt = [ - a \* y(1) \* I;

b \* I];

end

function err = forSearch(a, wEnd, b, y0, data)

[t, y] = ode45(@(t, y) sir(t, y, a, b), 1:wEnd, y0);

p = 1 - y(:, 1);

err = sum((data(1:wEnd) - p(1:wEnd)) .^ 2) / wEnd;

end

—————————————————————

%% Parameters estimation

from = 48; %1; % 1 22

to = 130; % 19 46

waveEnd = 270; %176; % 255

b = 0.1;

% Linear regression

mdl = fitlm(from:to, log(data(from:to)));

r = table2array(mdl.Coefficients(2, 1));

% c = table2array(mdl.Coefficients(1, 1));

% The last coefficient

a = r + b;

—————————————————————

%% Parameters estimation

from = 1; %1; % 1 22

to = 130; % 19 46

waveEnd = 270; %176; % 255

b = 0.1;

% Linear regression

mdl = fitlm(from:to, log(data(from:to)));

r = table2array(mdl.Coefficients(2, 1));

% c = table2array(mdl.Coefficients(1, 1));

% The last coefficient

a = r + b;