Table of Contents

..... 1
Load Files, Prepare Data 1
Evaluate Transmission Rates, Find Optimum for Each County 1
Export Results 2


```
% CSCE 666
% COVID-19 Project
% Iterative parameter estimation
%
% The SIRD model estimates deaths from an epidemic using an ODE.
% The solution to the ODE is parameterized by:
% r0 : reproduction number
%
% This script finds the transmission rate which best fits the SIRD
 model
% to real data for 210 counties in the USA using an iterative
 approach.
%
% The resulting r0 for each county is saved in the file 'r0_list.csv'
% for further analysis and feature correlation. Counties with poor
% statistics are omitted.

clear; clc;
```

Load Files, Prepare Data

```
population   = csvread('us-county-population_2.csv');
num_counties = length(population);
FIPS = population(1,:);

cases_data = readtable('us-county-cases_2.csv');
mortality_rates = csvread('Mortality_Rate_Summary.csv');

% Specify the SAA options for in-training monitoring
options = optimoptions('simulannealbnd','PlotFcns',...
          {@saplotbestx,@saplotbestf,@saplotx,@saplotf});
options.MaxStallIterations = 500;
```

Evaluate Transmission Rates, Find Optimum for Each County

Create a variable to store parameter values for each county

```
params = zeros(4, num_counties);
params(1,:) = FIPS;
```

```
r0_list = linspace(1.0, 3.5, 50);
trials = zeros(length(r0_list), num_counties);
for county = 1:num_counties-1
    % Population of county
    d = mortality_rates(2, county);
    pop = population(2, county);
    init_cases = cases_data.(county);
    Iinit = init_cases(2);

    fprintf('\nCounty: %g\nFIPS:   %g\nM.R.:   %f\n', [county,
 FIPS(county), d])
    % If ONLY optimizing transmission rate:
    for i = 1:length(r0_list)
        x = [r0_list(i) d county pop Iinit];
        trials(i, county) = mse_sir(x);
    end
    [~,ind] = min(trials(:, county));
    fprintf('r0:     %f\n', r0_list(ind))
end
```

Export Results

```
r0_eval = zeros(3, num_counties);
r0_eval(1, :) = FIPS;
for county = 1:num_counties
   [mser,ind] = min(trials(:, county));
   r0_eval(2, county) = r0_list(ind);
   r0_eval(3, county) = mser;
end

tbd = [];
for county = 1:num_counties-1
    if r0_eval(2, county) == 1
        tbd = [tbd county];
    elseif r0_eval(2, county) == 2
        tbd = [tbd county];
    end
end

r0_eval(:,tbd) = [];
csvwrite('r0_list.csv', r0_eval)
```

Published with MATLAB® R2018a