Tutorial for applying the eradication prioritisation methods from: *Prioritising the eradication of invasive species from island archipelagos with high reinvasion risk*

Tutorial authors: Viney Kumar, Andre Nunez, Kaitlyn Brown, Michael Bode

Introduction

This tutorial walks through the process of applying the methods described in the article "Prioritising the eradication of invasive species from island archipelagos with high reinvasion risk", available here. In particular, the tutorial files apply the technique to the 10-island Dampier archipelago, one of the two examples in the manuscript.

The goal of the method is to support the eradication of invasive species from archipelagos – multiple islands that are close enough for reinvasion to occur within the group. If eradications need to proceed sequentially, what is the best order to proceed with the eradication program?

The method is coded as a Matlab function, which defines the system parameters and runs the stochastic dynamic programming analyses and outputs the optimal eradication policy. Data about the archipelago, including information about the invasive species, the geography of the region, and details about the eradication program goals, are extracted from CSV files. The information contained in each file, and the required format, are detailed below. The approach we took to estimate these values for our tutorial exemplar are outlined, and then some basic approaches for new systems are briefly discussed.

Running the tutorial example

To calculate the optimal eradication schedule for the Dampier archipelago, the reader first needs to download the tutorial files from the Github repository. Once shese have been unpacked into a folder, open Matlab, then navigate to the folder that contains the files.

In the command window, run the Matlab script "Tutorial_SDP_solution.m". In the command window, you should see output like this:

```
Command Window
>> Tutorial_SDP_solution
Completed state # 50 out of 1024
Completed state # 100 out of 1024
Completed state # 150 out of 1024
Completed state # 200 out of 1024
Completed state # 250 out of 1024
Completed state # 300 out of 1024
Completed state # 350 out of 1024
Completed state # 400 out of 1024
Completed state # 450 out of 1024
Completed state # 500 out of 1024
Completed state # 550 out of 1024
Completed state # 600 out of 1024
Completed state # 650 out of 1024
Completed state # 700 out of 1024
Completed state # 750 out of 1024
Completed state # 800 out of 1024
Completed state # 850 out of 1024
Completed state # 900 out of 1024
Completed state # 950 out of 1024
Completed state # 1000 out of 1024
Completed action # 1
Completed action # 2
Completed action # 3
Completed action # 4
Completed action # 5
Completed action # 6
Completed action # 7
Completed action # 8
Completed action # 9
Completed action # 10
```

Figure 1: Expected MATLAB Command Window output upon running Tutorial_SDP_solution.m

The code will begin by estimating the state transition matrices using Monte Carlo simulation. It will then apply the SDP algorithm. If it has run correctly, you should see a large matrix printed into the command window, called "Optimal_island_to_eradicate".

In the sections below, we describe the parameters that must be estimated, how to apply the SDP algorithm and how to interpret the resultant matrix.

Defining the archipelago parameters

Details about the archipelago are critical concerns for decision-makers. Larger islands are harder to eradicate, islands that are in close proximity are likely to reinvade each other, islands close to the mainland are likely to be repeatedly reinvaded. The tutorial archipelago analyses the Dampier archipelago, comprising 10 islands, corresponding to the 10 rows of each parameter file.

	A	B	C
1	116.6	-20.477	
2	116.68	-20.513	
3	116.67	-20.52	
4	116.64	-20.578	
5	116.53	-20.598	
6	116.64	-20.553	
7	116.66	-20.615	
8	116.68	-20.653	
9	116.45	-20.659	
10	116.6	-20.7	
11			

Figure 2: Latitude and longitude of the centroids of the 10 islands in the Dampier archipelago. This data is contained in the CSV file "Tutorial_island_location.csv". Column A contains longitudinal coordinates and column B contains latitudinal coordinates.

	A	B
1	11.856	
2	2.3167	
3	1.0184	
4	20.686	
5	0.86866	
6	34.21	
7	10.511	
8	2.4447	
9	1.5669	
10	25.249	
11		

Figure 3: Area of the 10 islands in the Dampier archipelago in square kilometres, as used in the tutorial example. This data is contained in the CSV file "Tutorial_island_area.csv".

	A	B	C	D	E	F	G	H	I	J	K
1	0	7.212	6.0048	8.5925	8.1202	7.5337	12.502	19.126	22.325	19.79	
2	7.212	0	0.13207	2.5303	11.479	4.0407	7.8541	13.416	28.633	17.137	
3	6.0048	0.13207	0	2.2809	9.9231	2.5842	7.2792	13.478	27.078	16.483	
4	8.5925	2.5303	2.2809	0	2.3953	0.22379	0.83762	6.5319	16.859	6.8806	
5	8.1202	11.479	9.9231	2.3953	0	5.9725	7.1322	12.36	4.4903	8.268	
6	7.5337	4.0407	2.5842	0.22379	5.9725	0	3.9998	10.701	23.105	12.766	
7	12.502	7.8541	7.2792	0.83762	7.1322	3.9998	0	2.6587	20.466	4.5885	
8	19.126	13.416	13.478	6.5319	12.36	10.701	2.6587	0	24.26	3.7701	
9	22.325	28.633	27.078	16.859	4.4903	23.105	20.466	24.26	0	14.047	
10	19.79	17.137	16.483	6.8806	8.268	12.766	4.5885	3.7701	14.047	0	
11											

Figure 4: Minimum distance between the islands in the Dampier archipelago, as used in the tutorial example. This data is contained in the CSV file "Tutorial_distance_matrix.csv". Note that the columns A-J correspond to the 10 islands, as do rows 1-10. For example, the value in cell D2 represents the distance between island 2 (represented by row 2) and island 4 (represented by column D). Note that this is the same value as cell B4

	A	B
1	0.05	
2	0.05	
3	0.05	
4	0.05	
5	0.05	
6	0.05	
7	0.05	
8	0.05	
9	0.05	
10	0.05	
11		

Figure 5: Probability that the invasive species will go extinct on each island, without human intervention, in the Dampier archipelago tutorial example. These probabilities were assumed equal across all islands. This data is contained in the CSV file "Tutorial_natural_extinction.csv".

	A	B
1	0.087356	
2	0.44704	
3	1	
4	0.050065	
5	1	
6	0.030274	
7	0.098535	
8	0.42363	
9	0.66098	
10	0.041019	
11		

Figure 6: Probability that the invasive species will be successfully eradicated on each island, in the Dampier archipelago tutorial example, if an attempt is undertaken. This data was based on the equations in the Methods section of the manuscript, which assumed that larger areas were harder to eradicate from. This data is contained in the CSV file "Tutorial_eradication_probability.csv".

	A	B	C	D	E	F	G	H	I	J	K
1	0.5	0	0	0	0	0	0	0	0	0	
2	0	0.5	0.48312	0.25897	0	0.17487	0	0	0	0	
3	0	0.48312	0.5	0.27632	0	0.25537	0	0	0	0	
4	0	0.25897	0.27632	0.5	0.26823	0.47174	0.40215	0	0	0	
5	0	0	0	0.26823	0.5	0	0	0	0.15558	0	
6	0	0.17487	0.25537	0.47174	0	0.5	0.17674	0	0	0	
7	0	0	0	0.40215	0	0.17674	0.5	0.25047	0	0.15165	
8	0	0	0	0	0	0	0.25047	0.5	0	0.18761	
9	0	0	0	0	0.15558	0	0	0	0.5	0	
10	0	0	0	0	0	0	0.15165	0.18761	0	0.5	
11											

Figure 7: Probability that the population of invasive species from the island on the row will recolonise the island on the column, in the Dampier archipelago tutorial example. These values were estimated from the distance between islands, as predicted by the models contained in Lohr et al. (2017). "Predicting island biosecurity risk from introduced fauna using Bayesian Belief Networks". Science of the Total Environment, 601:1173-1181. This data is contained in the CSV file "Tutorial_recolonisation_probabilities.csv". Note that the columns A-J correspond to the islands represented by rows 1-10. For example, the value in cell C4 represents the probability that invasive species from island 4 (represented by row 4) will recolonise island 3 (represented by column 3).

	A	B
1	0	
2	0	
3	0	
4	0	
5	0	
6	0	
7	0	
8	0.34146	
9	0	
10	0.47734	
11		

Figure 8: Probability that the population of invasive species on the mainland will recolonise the island on the column, in the Dampier archipelago tutorial example. These values were based on the same models as Figure 7. This data is contained in the CSV file "Tutorial_recolonisation_probabilities_mainland.csv".

Defining the eradication program goals

As we describe in the main text of the article, the optimal approach to archipelago eradication depends on the goals of the decision-makers. One objective might be to maximise the total amount of invasive-free island area. Another objective might be to maximise the number of invasive-free islands.

In SDP, and in our Matlab code, the objective function is determined by the definition of the value vector, $V(i,t)$, which assigns a scalar value to each of the system states i.

```
% Define value vector

% This first value vector prefers states where a larger amount of island area is unoccupied by
% the invasive species
V = sum(repmat((Tutorial_IslandArea'),NumStates,1).*(States==0),2);

% This second value vector prefers states where there are a larger number of islands that are
% unoccupied by the invasive species, regardless of their area.
% V = sum(States==0,2);
```

The first value vector is currently operating in the Tutorial. This estimates the value on the basis of the island area. The second option is currently commented (i.e., is not being applied), and allows the user to estimate value on the basis of the number of uninvaded islands.

Running the SDP algorithm

Stochastic dynamic programming is a relatively simple algorithm to apply, once the transition matrices and value function for the problem have been defined. The code that applies SDP in the tutorial functions are here:

```
%% Apply SDP

Timesteps = 50;
ValueMatrix = zeros(NumStates,Timesteps+1);
ValueMatrix(:,end) = V;

discount_rate = -0.01; % Assign a discount rate to the future of 2.5%
for t = Timesteps:-1:1 % Backward step through all the years
    for n = 1:NumIslands+1 % Go through all the actions
        Action_value(:,n) = T_all(:,:,n)*ValueMatrix(:,t+1) + V.*exp(discount_rate.*t);
        RelVal(t) = exp(discount_rate.*t);
    end
    [ValueMatrix(:,t), Optimal_action(:,t)] = max(Action_value,[],2);
end
Optimal_island_to_eradicate = Optimal_action-1
```

The optimal eradication policy is contained in the matrix “Optimal_island_to_eradicate”. For the tutorial code, this matrix looks like this:

```
Command Window
>> whos Optimal_island_to_eradicate
  Name                             Size              Bytes  Class     Attributes

  Optimal_island_to_eradicate      1024x50           409600  double

>> Optimal_island_to_eradicate

Optimal_island_to_eradicate =

  Columns 1 through 33

     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     8     0     0     8     8     8     0     0     8     8     0     8     0     0     0     0     0     0     8     8     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
    10    10    10    10    10    10    10    10    10    10    10    10    10    10    10    10    10    10    10    10    10    10    10
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     8     8     8     8     8     8     8     8     8     8     8     8     8     8     8     8     8     8     8     8     8     8     8
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     8     0     8     0     0     8     0     8     0     0     8     0     0     0     8     0     0     0     0     8     0     0
     0     8     8     0     8     8     0     0     0     0     0     8     8     8     8     0     8     0     8     8     8     8     8
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     9     9     9     9     9     9     9     9     9     9     9     9     9     9     9     9     9     9     9     9     9     9     9
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
    10    10    10    10    10    10    10    10    10    10    10    10    10    10    10    10    10    10    10    10    10    10    10
     9     9     9     9     9     9     9     9     9     9     9     9     9     9     9     9     9     9     9     9     9     9     9
    10    10    10    10    10    10    10    10    10    10    10    10    10    10    10    10    10    10    10    10    10    10    10
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     8     0     0     0     0     8     0     0     0     0     0     0     0     0     8     0     0     0     0     8     0     0
     8     8     8     8     0     0     8     8     8     8     0     0     8     0     8     0     8     0     8     0     8     0     0
     9     9     9     9     0     8     8     8     8     8     8     8     0     8     8     8     0     8     0     0     0     0     8
     8     8     8     8     8     8     8     8     8     8     8     8     8     8     0     0     0     0     0     0     0     0     0
     8     8     8     8     8     8     8     8     8     8     8     8     8     8     8     8     8     8     8     8     8     8     8
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     8     8     8     8     8     8     8     8     8     8     8     8     8     8     8     8     8     8     8     8     8     8     8
     6     6     6     6     6     6     6     6     6     6     6     6     6     6     6     6     6     6     6     6     6     6     6
     7     7     7     7     7     7     7     7     7     7     7     7     7     7     7     7     7     7     7     7     7     7     7
     6     6     6     6     6     6     6     6     6     6     6     6     6     6     6     6     6     6     6     6     6     6     6
```

The matrix has dimensions 1024×50, where the 1024 rows correspond to the i invasion states that the system can be in (ranging from state 1: [0,0,0,0,0,0,0,0,0,0] where no islands are invaded, through to state 1024: [1,1,1,1,1,1,1,1,1,1] where all the islands are invaded). The 50 columns here correspond to the number of timesteps remaining until the objective function is calculated. Note that the matrix is much larger than the screenshot shown above.

To determine the optimal course of action, we consider the row of the matrix that corresponds to the current state of the archipelago. For example, if only island 10 is currently occupied by the invasive species (state 2: [0,0,0,0,0,0,0,0,0,1]), we consider the second row of the resultant matrix. The value recorded in the row corresponds to the island in the archipelago where eradication efforts should be focused (note that, if the matrix value is 0, the optimal action is to do nothing).

The optimal decisions can be visualised as a color image by running the following piece of code in the MATLAB command line:

>> imagesc(Optimal_island_to_eradicate), colorbar