Project 2: Where’s Croc

Artificial Intelligence (IDL340)

Basics

- This project should be performed alone or in pairs.
- You will use the R programming language.
- The use of third party code libraries/R packages is not permitted.
- The required code is available in the WheresCroc_1.2.1.zip file on student portal.
- See the R FAQ document on student portal for questions about using R.

Project Overview

You are a ranger in Wollomunga national park, in outback Australia. Crocodiles in this park have sensors attached that say where they are and the water conditions (salinity, phosphate and nitrogen levels) of the water the crocodile is current swimming in. The park consists of 40 waterholes, each of which is reachable only from its neighbors. The park has records of the calcium, salinity and alkalinity distributions in each waterhole. The sensor on one crocodile, 'Croc', has broken. It no longer says where he is, however it does still provide water condition readings. You need to find Croc. There are also two Swedish backpackers in the park, wandering around at random visiting waterholes. If they end up in the same waterhole as Croc, they will be eaten. You can move a ranger around the waterhole network and search for Croc at different locations. Your score is the number of moves it takes to find Croc.

Your task is to implement a control system to compete on the Where’s Croc game. You will use hidden Markov models and associated algorithms to work out where Croc is given the sequence of observable variables given to you in the game. Read the runWheresCroc help documentation in the R package for more details. The hidden Markov model will permit you to model the dynamic probabilities of Croc being at different water holes. This is important, but not the only important matter in performing well in the game. You will have to think about what else is important and how you can perform as well as possible.

You will provide

An R script with a function that can be passed to the runWheresCroc function.

Pass Requirements

Your function needs to:

- Equal or surpass par performance
- Meet the execution time requirement

See documentation on the testWC function in the WheresCroc package for details.

Important Functions

The important functions in the WheresCroc package are:

Function Name	Description
runWheresCroc	Runs one game of the Where's Croc game. The help documentation contains important information required for completing this project.
testWC	A simulation of the process that will be used to evaluate you (just with a different random seed), with par performance and time for this simulation given in the help documentation.

Example Control Functions

In addition, there are two control functions included, that you can pass to the runWheresCroc function:

1. randomWC
2. manualWC

Examine their help documentation for details. Note that none of these implement a HMM based solution.

Dangers

You only need to implement the forward algorithm for this project. The forward-backward and Viterbi algorithms are of no use to you.

The croc will move randomly, with a uniform distribution over moving to any adjacent waterholes or staying still.

The information you get each turn is both the readings, and the location (or death) or a tourist. Make use of both of these types of information.

Competition Hints (how to improve on the par function)

The obvious short-coming in the par function is that is it is entirely focused on the most probable location of the crocodile: The ranger heads as fast as possible there and searches. A possible improvement would be to also pause and search any high-probability waterholes found en route. A second improvement would be to plan the route taken so as to maximize the possibility of encountering such high-probability waterholes en route.

Remember: Implement the basic algorithm and ensure you have a passing implementation before trying to be ambitious!