Artificial Intelligence
Uppsala University – Autumn 2014
Report for Assignment 2 by Team 1

Sander Cox   Martin Kjellin   Malin Lundberg
Sverrir Thorgeirsson

October 13, 2014

1 Introduction

This report describes our program to compete on the Where's Croc game. The goal of the game is to find the crocodile Croc, that can be located at any of 35 waterholes in Wollomunga national park. Hopefully, he will be found before he eats the two backpackers that roam the park. The search is performed using a *hidden Markov model* (HMM).

2 Overall algorithm

The overall algorithm that we use to find Croc is as follows. An infinite series of game sessions, each consisting of 100 games, is run. In the beginning of each new game, each waterhole is assigned a probability of $\frac{1}{35}$ of being Croc's current location. Then, the following series of steps is repeated until Croc has been found.

The probability of Croc being currently located at each waterhole is estimated using the so-called *state estimation algorithm* (described in section 3). When the estimation is finished, the probabilities are adjusted using information about the locations of the backpackers. If any such location is a positive integer (which means that the backpacker is not being eaten), we know that Croc is currently not located at that waterhole. Consequently, the probability for that waterhole is set to zero. If, on the other hand, the location of a backpacker is a negative integer (which means that the backpacker is being eaten), we know for certain that Croc is located at that waterhole. Consequently, the probability for the corresponding waterhole is set to one, while the probabilities for all other waterholes are set to zero.

In general, we would like to move towards the currently most probable waterhole. However, if no waterhole has a probability that is particularly high in comparison to the others, it is hard to know for sure which waterhole is actually the most probable. Therefore, we decided that if the maximum probability over all the waterholes is lower than 0.15, we don't bother moving towards the corresponding waterhole. Instead, we will move towards the closest (before the latest pair of moves) of three waterholes that are located at the center of the national park (as indicated by the graph structure). Moving in this way means that we will be fairly close to all waterholes. This will likely be more beneficial than moving towards the currently most probable one, since that waterhole might be located far away from the one that will be considered the most probable one in the next iteration. We indicate the decision to move towards one of the central waterholes by setting it as the most probable waterhole.

Then, a standard breadth-first search is performed in order to find a shortest path from our current location to the currently most probable waterhole. If the length of the path (in number of waterholes) is 1, our current location is the most probable waterhole, so we search directly. The probability of our current location is set to zero, since we know that if Croc is not found during this search and another iteration must thus be performed, Croc's correct location is not our current one. If the length of the path is two, we make one move along the shortest path before we search and set the probability of that waterhole to zero. If the length of the path is more than two, we just move two steps along the shortest path in order to come closer to Croc. If Croc has not been found (as indicated by the return value of moves and searches), the game normally continues with a new probability estimation for each of the waterholes. However, if the average score of the current session is above a certain threshold that depends on the number of played games, the session is aborted (as it will probably not improve the record and thus is a waste of time) and a new one is started.

3 Estimating Croc's Location

In a HMM, the state of the world at time t is described by a state variable S_t. We cannot observe this state directly. Instead, we have access to a number of observation (evidence) variables O_t^x which indirectly provide us with information about the state of the world.

There are two types of probabilities in a HMM: transition probabilities and observation probabilities. Given that the world was in state S_{t-1} at time $t-1$, it will be in state S_t at time t with transition probability $P(S_t|S_{t-1})$. Given that the world is in state S_t, we will observe O_t^x with observation probability $P(O_t^x|S_t)$.

In this case, the state of the world is Croc's current location (waterhole number), so $S_t \in \{1, 2, \ldots, 35\}$. The observation variables O_t^c, O_t^s, and O_t^a are the current readings (of calcium, salinity, and alkalinity, respectively) from the sensor that Croc is carrying.

We want to calculate $P(S_t|S_{t-1}, O_t^c, O_t^s, O_t^a)$ for each state, that is, the probability of Croc currently lurking at that waterhole given his previous location and the current sensor readings. This probability is proportional to

$$P(O_t^c|S_t)P(O_t^s|S_t)P(O_t^a|S_t)P^*(S_t|S_{t-1})$$

where

$$P^*(S_t|S_{t-1}) = \sum_{i=1}^{35} P(S_t|S_{t-1} = i)P(S_{t-1} = i)$$

(that is, a sum over all possible previous states).

The recursive character of the transition probabilities calls for a base case, that is, a value S_0 for each state. As mentioned in section 2, we set $P(S_0) = \frac{1}{35}$

for all waterholes. That is, we assume that the probability is equally distributed over all waterholes.

The observation probabilities must be calculated from the sensor readings. Since we know, for each waterhole, the expectations μ and the standard deviations σ of the distributions from which the readings were drawn, we can calculate the value of the probability density function $f(O_t^x)$ for each observation O_t^x using the formula

$$f(O_t^x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(O_t^x-\mu)^2}{2\sigma^2}}$$

In a slight abuse of probability theory, these values can then be used as observation probabilities. That is, we assume that

$$P(O_t^x|S_t) = f(O_t^x)$$

The values calculated as

$$P(O_t^c|S_t)P(O_t^s|S_t)P(O_t^a|S_t)P^*(S_t|S_{t-1})$$

are not probabilities, that is, their sum is not 1. Therefore, they are normalized by being divided by the sum of the calculated values for all states. The results of the normalization are the desired probabilities $P(S_t|S_{t-1}, O_t^c, O_t^s, O_t^a)$.

4 Discussion

Our program achieves a fairly good result. As of the time of writing, our record score is 11.53. We think, however, that there is room for improvements:

- Weighing the probabilities for different waterholes using path length might be a good idea, at least if the difference in probabilities between the most current location and the others is small. That is, if Croc's currently most probable location is very distant from our current location, we might better decide to move towards some closer, but slightly less probable, location, thereby hopefully reducing the risk of making expensive mistakes.

- Since the graph that we deal with is small and its structure fixed, and since we use standard breadth-first search, there is no real need to repeatedly search the graph for shortest paths. Instead, the shortest path between each pair of nodes could be calculated and stored once and for all. This would speed up the program, making it possible to play more games during a certain period of time.