**Artificial Intelligence CE213 Assignment 2**

**By Alex Mezodi 1401665**

**Question 1**

A

The Monte-Carlo tree search which is a best-first search can be used on any game of finite length. The basis of its simulation is both the player and opponent playing random moves. By simulating a lot of random moves we can learn the game strategy. It selects future games states using: Selection, Expansion, Simulation, and Backpropagation.

Selecting the next action in the tree is used by applying exploitation (select action which leads to best result so far) and exploration (less promising action, still need to be explored).

Tree is expanded by one node each simulation. This is done by adding the first state which hasn't been found before.

In the Simulation stage action selected are random. Since this leads to a weak strategy we can use heuristic knowledge to weigh our actions and pick them accordingly.

And finally in Backpropagation after reaching the end of the simulation, we update the tree nodes with visit counts, and we also modify the win/loss ratio. The action executed is the one that was explored the most.

B

The classic approach is to use heuristic domain knowledge. Since it's hard to get strong enough game state, multiple Monte-Carlo based techniques appeared such as the Monte-Carlo tree search.

The quality is determined by the process Selection, Expansion, Simulation and Backpropagation. It expands nodes until one or more nodes are created, then a simulated game is played and the result is being back propagated in the tree.

C

Exploitation refers to selecting the action which yields the best result so far. Exploration can lead to less promising actions, but they still need to be evaluated. It's safe to say that overusing one can lead to less than satisfactory outcomes. If we use exploitation too much we wouldn't have enough new nodes to choose from, and exploration doesn't yield any good actions. Major exploration with minor exploitation would yield a good result and be a good balance, as we have many actions to choose from and exploitation can cherry-pick the best options out there.

D

When it comes to minimax search, we pass evaluations back up the search tree. When the opponent makes a move we pass back the minimum value; when we make a move we pass back the maximum value. Obviously the deeper we go down the search tree the better, and we can also utilize alpha-beta pruning to save effort.

As the paper suggests, alpha-beta pruning does wonders with old games such as chess or checkers, but when we have an inadequate evaluation function, or there is a high branching factor (such as modern board & video games), Monte-Carlo Tree Search is more advantageous.

E

There are key differences between modern board games (also known as Eurogames) such as Settlers of Catan, and classic board games such as chess. Modern board games often incorporate randomness, hidden information, multiple players and a difference between initial setups.

Talking about Settlers of Catan, MTCS outperforms minimax search and provides a difficult opponent to play against. As I mentioned before, minimax search depends on a good evaluation function and a low branching factor. With modern board games this is impossible; multiple players not to mention random factors make this very difficult.

F

The factors which need to be considered in order to calculate the number of simulations from a child node can be calculated with a formula. It helps maintaining a balance between exploitation and exploration.

The factors which need to be considered are: the number of wins after the ith move; the number of simulations after the ith move; exploration paramater; total number of simulations.

**Question 2**

**First we consider the initial information / uncertainty about class Play ball:**

p(yes) = 9/14; p(no) = 5/14

So for the whole set, Inf = -(9/14 log2(9/14) + 5/14 log2(5/14) = 0.940bits

Next, we choose the best attribute for building the branches from the root node of the decision tree.

We can choose Outlook, Temperature, Humidity, and Wind.

**Information Gain from Outlook Attribute**

Sunny examples/samples:

There are 5 of these. 2 yes and 3 no. (Play ball values)

So for this subset, p(yes) = 2/5 and p(no) = 3/5

Hence, Infsunny = -(2/5 log2(2/5) + 3/5 log2(3/5)) = **0.971**

Overcast examples

There are 4 of these. 4 yes and 0 no.

So for this subset, p(yes) = 1 and p(no) = 0

Hence, InfOvercast = -(1 x log2(1) + 0 x log2(0)) = **0**

Rain examples

There are 5 of these. 3 yes and 2 no.

So for this subset, p(yes) = 3/5 and p(no) 2/5

Hence, InfRain = -(3/5log2(3/5) + 2/5log2(2/5)) = **0.971**

Average Information about the class given Outlook:

5/14 Infsunny + 4/14 InfOvercast + 5/14 InfRain = 0.357 x 0.971 + 0.286 x 0 + 0.357 x 0.971 = 0.693bits

Hence, **Information Gain from Outlook** is Initial Information - Average Information given Outlook

= 0.940 – 0.693 = **0.247 bits**

**Information Gain from Temperature Attribute**

Hot examples/samples:

There are 4 of these. 2 yes and 2 no. (Play ball values)

So for this subset, p(yes) = 2/4 and p(no) = 2/4

Hence, Infhot = -(2/4 log2(2/4) + 2/4 log2(2/4)) = 1

Mild examples/samples:

There are 6 of these. 4 yes and 2 no.

So for this subset, p(yes) = 4/6 and p(no) = 2/6

Hence, Infmild = -(4/6 log2(4/6) + 2/6 log2(2/6)) = 0.919

Cool examples/samples:

There are 4 of these. 3 yes and 1 no.

So for this subset, p(yes) = 3/4 and p(no) = 1/4

Hence, Infcool = -(3/4 log2(3/4) + 1/4 log2(1/4)) = 0.811

Average Information about the class given Temperature:

4/14 Infhot + 6/14 Infmild + 4/14 Infcool = 0.286 x 1 + 0.429 x 0.919 + 0.286 x 0.811 = 0.912

Hence, **Information Gain from Temperature** is Initial Information - Average Information given Temp.

= 0.940 – 0.912= **0.028 bits**

**Information Gain from Humidity Attribute**

High examples/samples:

There are 7 of these. 3 yes and 4 no. (Play ball values)

So for this subset, p(yes) = 3/7 and p(no) = 4/7

Hence, InfHigh = -(3/7 log2(3/7) + 4/7 log2(4/7)) = 0.986

Normal examples/samples:

There are 7 of these. 6 yes and 1 no.

So for this subset, p(yes) = 6/7 and p(no) = 1/7

Hence, InfNormal = -(6/7 log2(6/7) + 1/7 log2(1/7)) = 0.592

Average Information about the class given Humidity:

7/14 InfHigh + 7/14 InfNormal = 0.5 x 0.986 + 0.5 x 0.592 = 0.789

Hence, **Information Gain from Humidity** is Initial Information - Average Information given Humidity.

= 0.940 – 0.789= **0.151 bits**

**Information Gain from Wind Attribute**

Strong examples/samples:

There are 6 of these. 3 yes and 3 no. (Play ball values)

So for this subset, p(yes) = 3/6 and p(no) = 3/6

Hence, InfStrong = -(3/6 log2(3/6) + 3/6 log2(3/6)) = 1

Weak examples/samples:

There are 8 of these. 6 yes and 2 no.

So for this subset, p(yes) = 6/8 and p(no) = 2/8

Hence, InfWeak = -(6/8 log2(6/8) + 2/8 log2(2/8)) = 0.811

Average Information about the class given Wind:

6/14 InfStrong + 8/14 InfWeak = 0.429 x 1 + 0.571 x 0.811 = 0.892

Hence, **Information Gain from Wind** is Initial Information - Average Information given Humidity.

= 0.940 – 0.892= **0.048 bits**

**CONCLUSION**

The information gain from Outlook is the highest and is therefore the best attribute.

Play Ball: Yes

Outlook

Hot, High, Weak, no

Hot, High, Strong, no

Mild, High, Weak, no

Cool, Normal, Weak, yes

Mild, Normal, Strong, yes

Sunny

Overcast

Mild High Weak, yes

Cool Normal Weak, yes

Cool Normal Strong, no

Mild Normal Weak, yes

Mild High Strong, no

Rain

Outlook

Hot, High, Weak, no

Hot, High, Strong, no

Mild, High, Weak, no

Cool, Normal, Weak, yes

Mild, Normal, Strong, yes

Sunny

Hot, High, Weak, yes

Cool Normal Strong, yes

Mild High Strong, yes

Hot Normal Weak, yes

Overcast

Mild High Weak, yes

Cool Normal Weak, yes

Cool Normal Strong, no

Mild Normal Weak, yes

Mild High Strong, no

Rain

Overcast can be simplified to play ball Yes. We need to extend the Sunny and Rain branches though.

The Sunny branch has 5 examples/samples: 2 yes and 3 no.

So p(yes) = 2/5 and p(no) = 3/5. From our previous example we know that the information/uncertainty is 0.971.

We also have to look at the Temperature, Humidity and Wind values for Sunny.

For the temperature:

Hot examples/samples:

We have 2 of these, 0 yes and 2 no.

From our previous example we know that the information gain is: 0bit.

Mild examples/samples:

We have 2 of these, 1 yes and 1 no.

Hence information gain is 1bit.

Cool examples/samples:

We have 1 of this, a no for play ball.

Information gain therefore is 0bit.

Average Information about the class given Temperature:

2/5 InfHot + 2/5 InfMild + 1/5 InfMild = 0.4 x 0 + 0.4 x 1 + 0.2 x 0 = 0.4bits

The information gain from temperature is not good enough, so we’ll have a look at Humidity.

High examples / samples:

We have 3 of these, 0 yes and 3 no.

The information gain is: 0bit.

Normal examples / samples:

We have 2 of these, 2 yes and 0 no.

The information gain is: 0bit.

Average Information about the class given Humidity:

3/5 InfHigh + 2/5 InfNormal = 0.6 x 0 + 0.4 x 0 = 0

Information gain from Humidity is 0.971, which is very close to perfect; therefore, humidity is a really good predictor of precipitation for this subset of examples.

Outlook

Sunny

Play Ball: Yes

Overcast

Mild High Weak, yes

Cool Normal Weak, yes

Cool Normal Strong, no

Mild Normal Weak, yes

Mild High Strong, no

Rain

Humidity

High

Normal

Play Ball: Yes

Play Ball: No

Now let’s have a look at the branch Rain.

It has 5 examples, 3 yes and 2 no. We know that the information/uncertainty is 0.971.

Let’s look at the Wind.

Strong wind examples:

We have 0 yes and 2 no.

The information gain is: 0bit.

Weak wind examples:

We have 3 yes and 0 no.

The information gain is: 0bit.

Using our previous example, we know that this equals 0, and the Information Gain from Wind is 0.971 which is very good and we update our table accordingly.

Outlook

Sunny

Play Ball: Yes

Overcast

Rain

Humidity

High

Normal

Play Ball: Yes

Play Ball: No

Wind

Weak

Strong

Play Ball: Yes

Play Ball: No

We can derive a set of IF-THEN rules from the Induction Tree:

**IF Overcast THEN play ball**

**IF Sunny AND High Humidity THEN Don’t play ball**

**IF Sunny AND Normal Humidity THEN play ball**

**IF Rainy AND Strong Wind THEN Don’t play ball**

**IF Rainy AND Weak Wind THEN play ball**

**Name two criteria that could be adopted to replace information gain for selecting best attribute in decision tree induction. Justify your answer.**

The two other criteria that could replace Information Gain for selecting the best attribute are Gini Impurity and Variance Reduction.

Gini Impurity measures how often a randomly chosen element in a set would be labelled incorrectly if it were labelled according to the distribution of labels in the given subset.

When the target variable is continuous we can employ Variance Reduction.

**Question 3**

**I, Calculate the similarities between each pair of soils**

First, we calculate the Euclidean Distance between the pairs of soils. I used this online calculator for reference: <http://calculator.vhex.net/calculator/distance/euclidean-distance>

Then, in the last column we calculate the similarities of two pairs of soils by dividing the Euclidean distance of the two soils by one.

|  |  |  |
| --- | --- | --- |
| Soil Pair | Distance | Similarity |
| S1, S2 | 20.62 | 0.048 |
| S1, S3 | 13.34 | 0.075 |
| S1, S4 | 31.56 | 0.032 |
| S1, S5 | 39.87 | 0.025 |
| S1, S6 | 12.25 | 0.080 |
| S2, S3 | 21.28 | 0.047 |
| S2, S4 | 14.18 | 0.071 |
| S2, S5 | 22.02 | 0.045 |
| S2, S6 | 11.18 | 0.089 |
| S3, S4 | 26.72 | 0.037 |
| S3, S5 | 34.12 | 0.029 |
| S3, S6 | 10.39 | 0.096 |
| S4, S5 | 8.37 | 0.119 |
| S4, S6 | 20.15 | 0.050 |
| S5, S6 | 28.28 | 0.035 |

S3

S6

S2

S1

S4

S5

**Ii, Draw the dendogram that would be produced by agglomerative hierarchical clustering**

Using the similarity chart we can construct the dendogram produced by agglomerative hierarchical clustering. (Above on the right.)

**Iii, Classify the six soils into two classes using the obtained dendogram**

**For 2 Classes:**

**S3, S6, S2 and S1 in class 1,**

**S4 and S5 in class 2.**