Advanced AI Techniques (WS04)

Exercise sheet 9
Christmas sheet
Deadline: Tuesday, 11 Jan 05

From the Foundations of AI lecture, you know the minimax algorithm for determining a strategy in a finite game with alternating moves. This algorithm yields the best strategy under the assumption that all players play rational in the sense that your opponent chooses the move that minimizes your reward, and you choose the move that maximizes it. In the lecture, we saw that not in all cases of two person games, this assumption holds.

In the first exercise, you will see how difficult it can be in real life to be sure about one's best strategy. You are asked to discuss differences between alternative strategies.

In the Advanced AI Techniques lecture, some algorithms for learning a strategy were presented. In the second exercise, you are asked to implement such a strategy and discuss the result.

With reinforcement learning, it is necessary to find the balance between exploitation moves, in which you expect to gain a high immediate reward, and exploration moves, in which you expect to learn. It is not easy to find a good balance between both types of action selection (indeed the action selection policy should depend on your time horizon).

The ϵ-greedy action selection method fixes a constant probability ϵ for choosing an exploratory move, whereas the Gibbs method starts with a high probability for exploration and gradually reduces it. In the third exercise, you are asked to implement and compare different action selection methods.

For the implementations, you can use any programming language you like. Please send me the program code via e-mail, and a listing of your results. On the AAIT web page, we have linked an implementation of a function that calculates a normal distribution with mean value 0 and standard deviation 1 (randomGauss). You need it for the implementation of exercise 3, but you do not need to analyze this function (except if you use another programming language).

Exercise 1 (4 points)

The christmas cookie game is defined as follows. On the christmas market, you buy a bag of christmas cookies (your move) and then offer its cookies to your friend. Your friend chooses one cookie (his/her move). You then eat all remaining cookies (your reward). [1]

The cookies [2] *have the following values for you:*

Abbreviation	*cookie (English)*	*cookie (German)*	*value*
A	*Anise cookies*	*Anisplätzchen*	*1*
B	*Brownies*	*Brownies*	*2*
C	*Cinnamon stars*	*Zimtsterne*	*3*
G	*Gingerbread wafers*	*Lebkuchen*	*4*
O	*Old Grandma's secret recipe*	*Omas Geheimrezept*	*5*
R	*Rum Balls*	*Rumkugeln*	*6*
V	*Vanilla crescents*	*Vanillekipferl*	*8*

In your move, you have the choice between the following three bags:

- $X = \langle G, G, G, G \rangle$ *(4 pieces of gingerbread wafers)*
- $Y = \langle A, B, R, V \rangle$ *(4 assorted pieces)*
- $Z = \langle A, O, O, O \rangle$ *(3 pieces of old grandma's and 1 anise cookie)*

The reward you get is the sum of the values of the three remaining cookies after your friend's move.

You do not know what the values of the different types of cookies are for your friend. Display the 2-level decision tree, find the optimal strategy, and calculate the expected reward for you under the following assumptions:

1. *Your friend has the same preference order and wants to maximize his/her reward.*
2. *Your friend chooses randomly one of the 4 cookies with equal probability.*
3. *Your friend has a preference for a specific type of cookie and selects one of the preferred type. You have no clue which preference your friend has.* [3]

[1] This not unfair. You spent the money, right?

[2] For recipes, you can look at the pages
http://www.christmas-cookies.com/recipes/
http://german.about.com/library/blrezepte02b.htm/
http://southernfood.about.com/cs/cookierecipes/a/aa121497.htm

[3] This is modeled in the following way. Assume there are 3 different types A,B, C in the bag. Then P(a cookie of type A is taken) = P(a cookie of type B is taken) = P(a cookie of type C is taken) = $\frac{1}{3}$, no matter how many cookies of each type the bag contains.

4. *Your friend is very nice to you (It's Christmas, and it's your friend!) and leaves the best cookies to you.*

Compare the last result with the reward you would get with the minimax algorithm. If you followed the minimax strategy, what share (in %) of your possible reward would you lose?

Remark (just to think about it):
This is a typical situation in ordinary life. Often it is not clear which strategy (if any) your fellows follow. In our case, if you do not know which strategy your friend uses, you can learn a rewarding behaviour strategy by going to the christmas market repeatedly and playing the christmas cookie game with the same friend again and again. You then can use reinforcement learning. What are the advantages and disadvantages of reinforcement learning in such real life situations?

Exercise 2 (4 points) *(A binary bandit problem)*

Assume you want to give a nice christmas present to your uncle. You have the choice to either give him an after shave (action $a = A$) or a book (action $a = B$). If he is happy with the present, he will smile (this makes you happy: $r = success$), otherwise frown (reward $r = failure$).

Implement the supervised algorithm for learning (over the next 50 years) whether it is better to give him an after shave or a book as a christmas present. Assume that the uncle's taste does not change, i. e. the success probabilities $p_A = P(r_t = success|A)$ and $p_B = P(r_t = success|B)$ are independent of the time point t.

Trigger 100 runs (each run simulating 50 plays) for each of the following binary bandit tasks and determine (for each task) in how many cases the intuitively better action was "learned" as desired action. Explain the problems of learning with this model of "supervised learning".

1. $p_A = 0.1,\ p_B = 0.8$
2. $p_A = 0.2,\ p_B = 0.1$
3. $p_A = 0.5,\ p_B = 0.5$
4. $p_A = 0.8,\ p_B = 0.9$

Exercise 3 (4 points)

On January 6th, you and two friends want to earn some money as star singers. You have the choice between three different districts: Alt-Stühlinger, Betzenhausen and City. The reward distributions (relative figures) are normal Gauss distributions with a standard deviation of 1 and the following mean values:

district (action a)	*mean expected reward $E(r\|a)$*
Alt-Stühlinger	*5*
Betzenhausen	*4*
City	*2*

In order to find out which is the best district to sing in, implement the action evaluation method "sample average". Start with an initial estimated quality of 100 for each district (i. e. assume a prior "experience" of once having been rewarded 100 for each action). Apply the following three action selection methods:

- *greedy,*
- *ϵ-greedy with $\epsilon = 0.01$,*
- *ϵ-greedy with $\epsilon = 0.1$*

in different runs of your program. For each run, repeat 3000 action selections. Report your total reward.
Compare your final quality estimation for the three actions with the real values. Which of the three runs yielded the best action quality estimation and why?

Modify your program so that you implement the Gibbs action selection method for some value τ. Test different values of τ. What is the effect of increasing τ? What is a good value for τ in order to maximize your total reward (again, stop after 3000 action selections per run).