Machine Learning: Assignment #2

Due on Wednesday, Aban 24th , 1393

Prof. Majid Nili Ahmad Abadi

Nima Mohammadi

(nima.mohammadi@ut.ac.ir)

1- The dataset for this part has been created by concatenating the reward/loss schedule table of 40 trials in Bechara et al. (1994), as instructed by the assignment.

Various functions and values have been tested for learning rate which are plotted and depicted in Figure 1. It includes the proposed functions in the assignment (with α=.3 as the constant). Also I've tested 5/(5+k) as learning rate of my choice for the sake of comparison. I've found $\alpha(t) = C\alpha(0) / (C + t)$ in a book titled "Process Modeling Using the Self-Organizing Map" written by Jaakko Hollmén. I set the arbitrary parameter C to 5 that seems to result in comparatively good performance. This specific value was based on my empirical results.

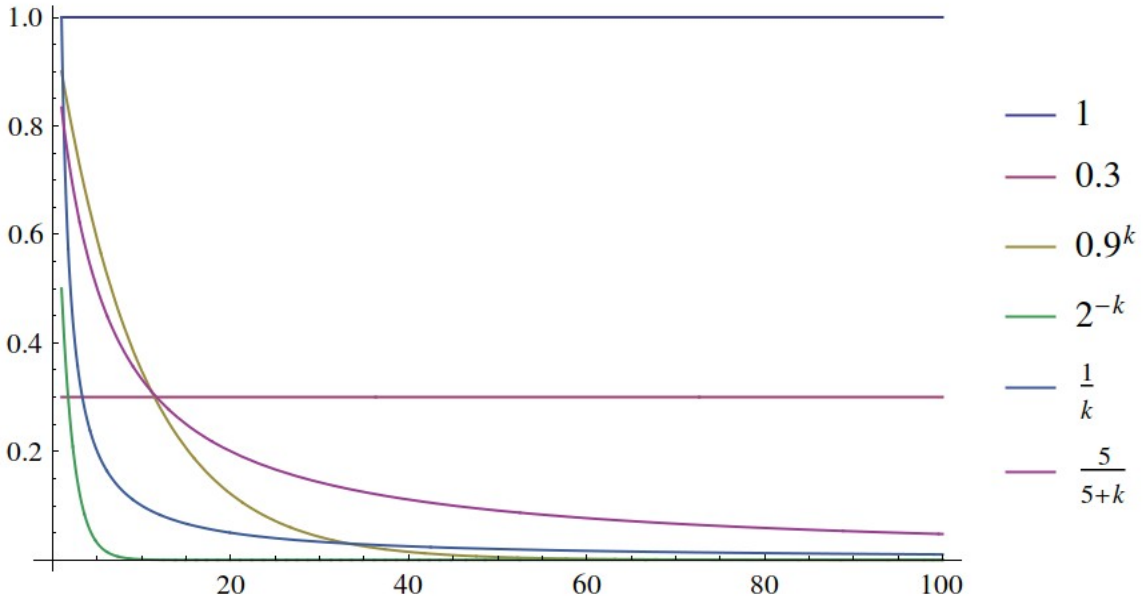


Fig. 1: Various functions taken as learning rate [Plotted by Wolfram Mathematica]

At first we evaluate the performance of Epsilon-greedy policy. This method chooses randomly between exploration and exploitation strategy regarding the value of ε. The higher the value of ε, the higher is the chance that we randomly select an action, hence higher exploration. Therefore we opt to choose the value of ε to be near one, and gradually moving its value toward zero which corresponds to more and more exploitation.

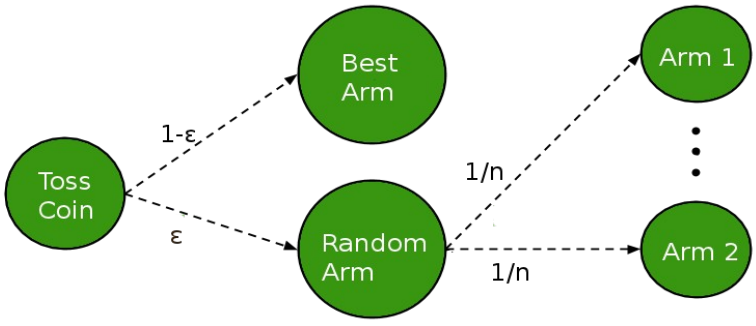
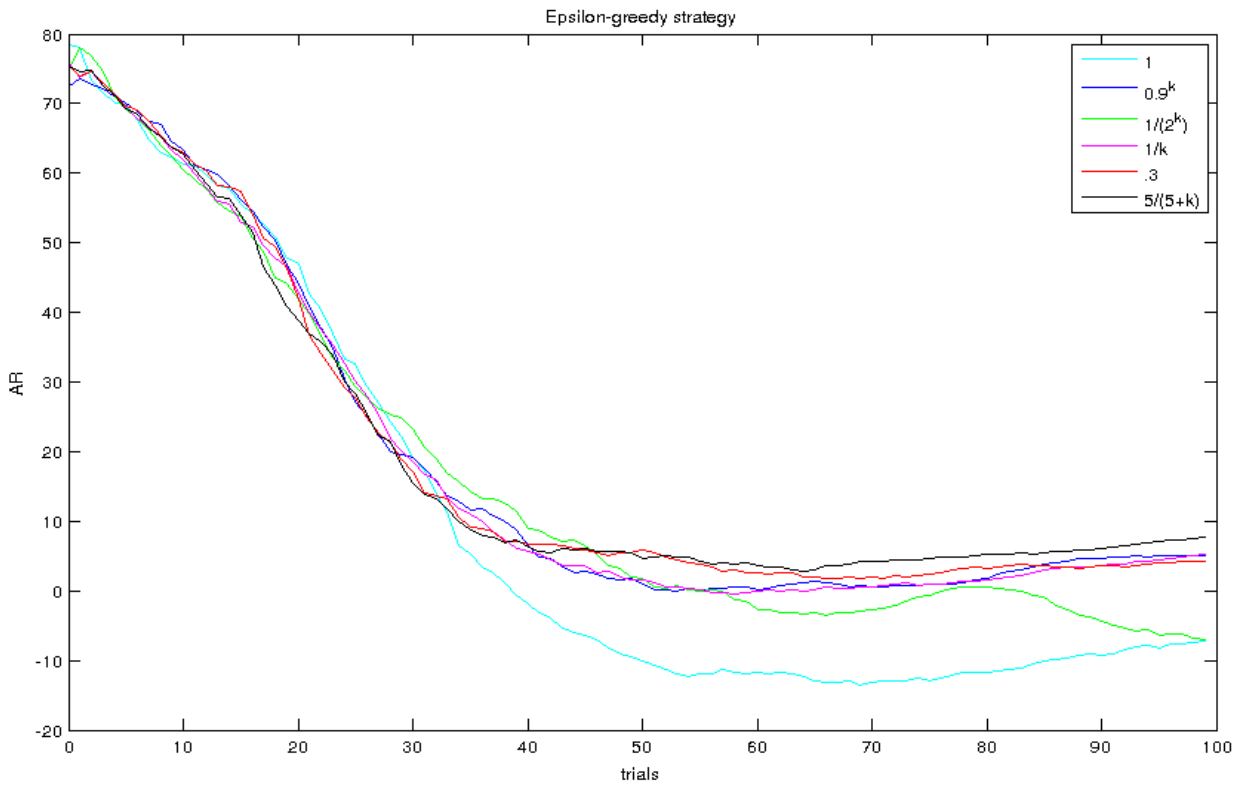


Fig. 2: Scheme of Epsilon-greedy strategy

The means for average reward of 100 runs for Epsilon-greedy over this dataset are plotted in Fig 3. The curves in these plots are calculated by taking the mean of average reward value for 100 previous runs. You may notice that the average rewards at the ending trials are in fact much less than the starting trials! This is due to the fact the assignment requires the averaging window to be of length 100. Looking at the table in Bechara et al., you will realize that some draws from deck A and especially deck B result in enormous losses. The steep decline of the curves actually happens at around those trials. Since the length of the averaging window is relatively large, the effect of those huge losses stays in AR of the next trials.

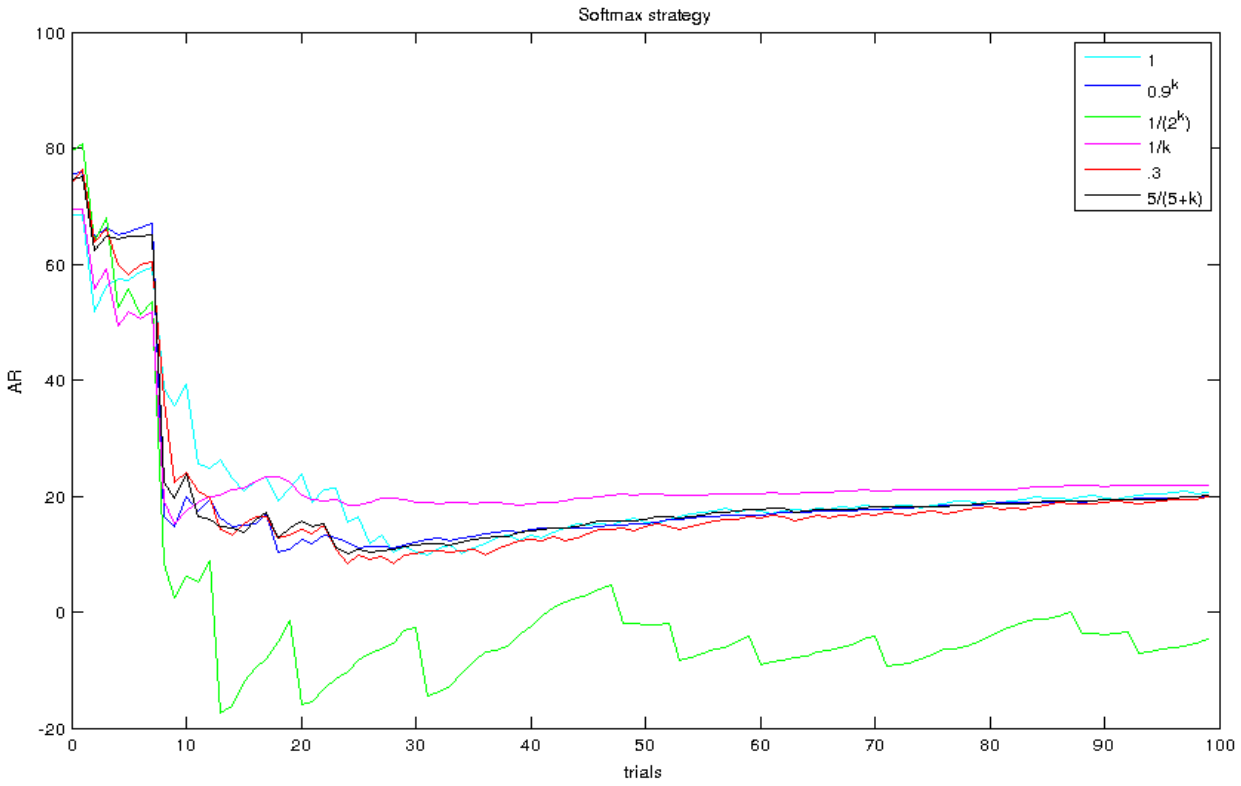
As can be seen in Fig. 3, in Epsilon-greedy, my own learning rate has outperformed the other learning rates. The reason for this, as suggested by Fig. 1, comparing different learning rates, might be because of the less steeper decline in learning rate for 5/(5+k). It could be said that Epsilon-greedy needs more time for learning (i.e. longer exploration period) which based on the AR of different runs is apparently provided by 5/(5+k). This explanations holds for the two next high performing learning rates which are 0.9^k and 1/k. Their performance correspond to how late their curves "fall". The two worst learning rates which expectedly had the worst performance are $\alpha=1$ and $\alpha=2^{-k}$. In $\alpha=1$, being constants of relatively high value, its corresponding model has less "sense of time", so to speak, compared to the other learning rates. The estimation error would have a high influence over Q-value and there is not decrease in value of α, therefore this configuration ($\alpha=1$) has higher exploration compared to the others. Although $\alpha=.3$ is also a constant, because it is smaller, the contribution of estimation error is more acceptable, and therefore it performs better than $\alpha=1$. At the end, in case of $\alpha=2^{-k}$, it soon moves α to zero and more and more fails to take take into account its recent estimation error; not much like a sane decision maker!



(Fig. 3: Performance of Epsilon-greedy strategy for different learning rates)

Fig. 4 shows the AR for the second approach, action-value method with Boltzmann softmax policy. It is a bit of an enhancement over the previous approach. You can see that five of our choices for learning rates acted better than the best performance of Epsilon-greedy. Hands down, softmax is the winner! The problem with Epsilon-greedy is that it explores different actions at random without any concern over their merits. For this strategy, we chose temperature (τ) to be initially 10 and gradually decreased it to 0.001 in 100 episodes.

At first glance, it can be noticed that all of the learning rates (except one) performed very much alike, with 1/k being the winner with slightly better AR. It may be due to the fact that softmax converges much faster than Epsilon-greedy (i.e. during 10 first trials). And as Fig. 1 shows, in these five learning rates, the value of alpha (α) is high enough during the first 10 trials. Unlike those, 2^{-k} very soon moves toward zero, so the contribution of estimation error is less effective which causes this configuration to fail to compete with the others. One reason that some failing learning rates in Epsilon-greedy perform relatively well in Softmax is that Softmax does not do the exploration willy-nilly and has a 'planned' exploration based on the estimated values of the actions which in turn avoids wasting precious trials on inferior actions.



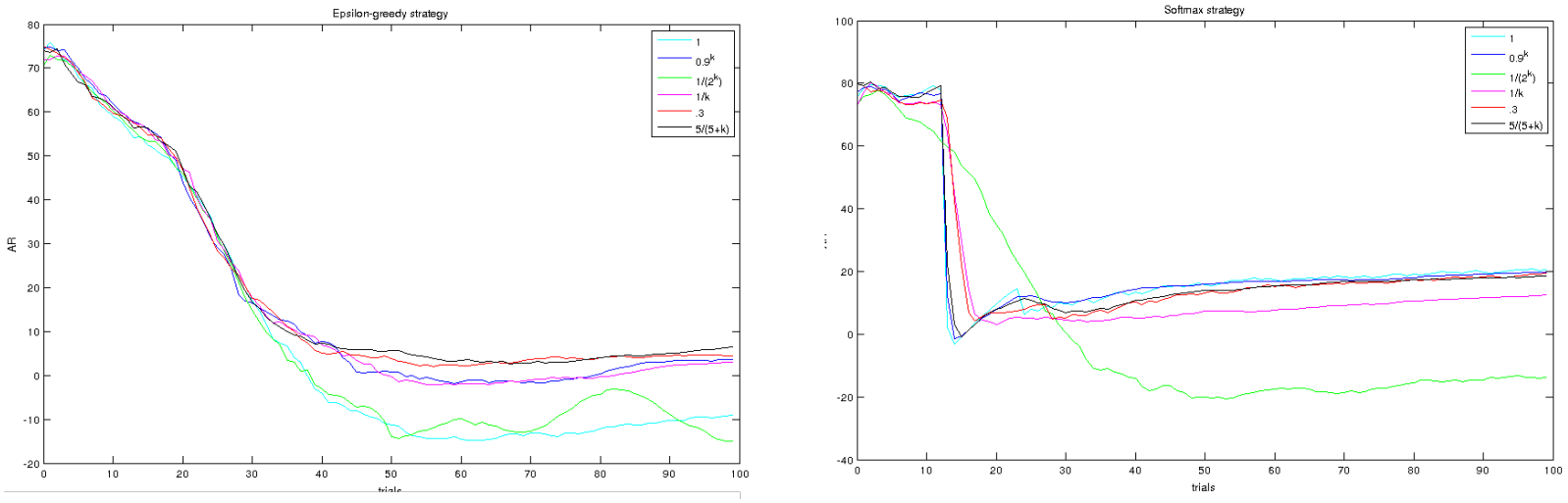
(Fig. 4: Performance of Softmax strategy for different learning rates)

	Epsilon-greedy		Softmax	
α	**Decks A & B**	**Decks C & D**	**Decks A & B**	**Decks C & D**
1	44.7	55.2	0	100
0.9^k	18.9	81	0.7	99.2
1/(2^k)	68.5	31.4	61.4	38.5
1/k	13.4	86.5	0.4	99.5
.3	11.9	88	0.3	99.6
5/(5+k)	10.7	89.2	0.8	99.1

(Table 1: The percentage of cards drawn from different decks in last 40 trials)

Table 1 lists the percentage of chosen decks in the **last 40 trials** of 100 runs. We know that Decks A & B give higher gains but as their losses (punishments) are also greater, they cancel each other. Decks C & D, although give gains of smaller size, yet their losses are also smaller. The net profit of Decks C & D, which are equal, are higher than Decks A & B, and therefore are the optimal actions. Notice that the percentage of the chosen actions in Table 1 should be taken with a grain of salt, and AR is indeed a better criterion for comparison. However this table can tell us about the convergence of the two approaches toward the optimal decks. Table 1 confirms that the learning rate proposed by me is in fact the best one for Epsilon-greedy and is one of the best ones for Softmax.

As for the third part of the question, optimistic initial values, instead of setting the initial values to zero, we set them all to five. This so-called "optimism" encourages the methods to explore. The experiments are repeated and the results are depicted in Fig. 5.
As for Epsilon-greedy, we can not see much of a difference. It only caused 2^{-k} to fail even worse. It is because higher initial Q-values gives a false sense of high value for inferior decks, which 2^{-k} learning rate can not compensate since its learning soon stops. As for Softmax, the final ARs apparently stay the same. However at starting trials, ARs of leading learning rates are higher compared to the non-optimistic case. It is probably since softmax converges fast, and optimistic initial values postpone the convergence and stress more on exploration. The performance of the worst learning rate for Softmax, which is 2^{-k} initially is more promising which once again confirms that 2^{-k} suffers from premature exploitation which could be alleviated by increasing exploration rate.



(Fig. 5: Epsilon-greedy and Softmax with optimistic initial values)

2- The two models of PVL and WSLS have been implemented and their free parameters have been estimated by Maximum Likelihood Estimation. The log likelihoods of the models were used as an objective function and patternsearch() function of MATLAB has been employed to find the set of parameters maximizing the objective function. The initial values 0.3 and 0.6 have been, respectively, chosen for p(stay|win) and p(shift|loss) for WSLS. Also 0.1, 4, 0.5 and 0.3 are set as initial points for Alpha, Lambda, A and c in PVL parameter estimation.

For the first part of the question we need to compare the parameters of the two models. For this purpose, we use inferential stats (here t-test) to compare the average value of parameters between these two groups. We employ the ttest2() function of MATLAB for independent two-sample t-test analysis.

	P(stay|win)	**P(shift|loss)**
Group1	0.51	0.68
Group2	0.42	0.77
h	0	0
p-value	0.13	0.11

Table 2: average of estimated parameters of WSLS model for each group followed by t-test

Feeding values for both parameters to ttest2(), h would be zero for both with p-values 0.13 and 0.11 respectively for P(stay|win) and P(shift|loss) which indicates failure to reject the null hypothesis that the parameters are from populations with equal means, against the alternative that the means are unequal. Therefore we can not reject the null hypothesis at the 0.05 significance level.

	Apha	**Lambda**	**A**	**C**
Group1	0.015	0.004	0.16	2.01
Group2	0.04	0.004	0.109	1.89
h	1	0	0	1
P-value	0.025	0.50	0.072	0.02

Table 3: average of estimated parameters of PVL model for each group followed by t-test

For PVL, the result of t-test indicates that Alpha and C estimated for the two group significantly differ. Higher value of C for Group1 suggests that those participants increase the degree to which they select higher-valued actions as the number of trials increases, faster than Group2. Higher value of Alpha for Group2 also indicates that Group2 scale rewards to a higher value compared to Group1.

Based on estimated values for WSLS, it can be said that human beings are more likely to change their action once they stumble upon a bad situation, than to stay on their winning action.

It is of interest that looking at the low value of Lambda for PVL, it suggests that, contrary to our expectations, people do not have much aversion toward loss. The decay parameter A which determines how much people value their past experiences is relatively low, which might suggest an unfortunate conclusion that people do not learn much from their past.

So if it was up to me, I would choose a higher value for Lambda, ~3.0, which would make the decision maker to have more aversion toward loss, and not that high, as to avoid the decision maker to be very conservative and afraid of experiencing other actions. Setting C to be around 2 seems to be fair, as the common sense states that being greedy and ambitious does not usually result in well-being.

3- For this question, Akaike information criterion (AIC) which is a measure of the relative quality of a statistical model is used to asses and select the proper model. It gives us a trade-off between how well the model can fit to the data and how complex it is.

	AIC
WSLS	33.2454
PVL	37.2454

Tabel 4: Average of AIC for two models

The AIC for each method on each participant of the two groups are computed. The method with the minimum AIC, which here is WSLS, is the preferred model. The average of AIC for the two models are listed in Table 4.

* The codes are sent with this document. The codes have been tested on MATLAB R2010b over a 64-bit linux machine.