CS 181 Machine Learning
Practical 4 Report, Team *la Dernière Dame M*

(Jeremiah) Zhe Liu[1], (Vivian) Wenwan Yang[2], and Jing Wen[1]

[1]Department of Biostatistics, Harvard School of Public Health
[2]Department of Computational Science and Engineering, SEAS

May 5, 2015

1 Exploratory Analysis

1.1 Q learning

The state space is discretised over the horizontal and vertical distance from the next lower pipe. The Action is to either jump or not. Firstly, we need to find the distance from nearest pipe. After calculating the distance, we have our states defined, and the action is implemented via a static boolean variable, which defines the action performed by the monkey. As the monkey learns by performing actions randomly, it leans the q-value for the states it visits. After sufficient learning, based on the exploration rate, it decides what action is to be performed at a particular rate.Expanding the difinition of the Q function from the Bellman equations, we have:

$$Q(s,a) = R(s,a) + \gamma \sum P(s'|s,a) max_{a' \in A} Q(s',a')$$

$$= R(s,a) + \gamma E_{s'}[max_{a' \in A} Q(s',a')]$$

$$= E_{s'}[R(s,a) + \gamma max_{a' \in A} Q(s',a')]$$

The goal is to estimate Q(s,a) as the expectation, where s' is drawn from $P(s'|s,a)$, of $R(s,a) + max_{a'} Q(s',a)$. Each time we actually take action a from state s we observe a transition to s' and receive a reward r. This could give us the sample from $P(s'|s,a)$. We could use this sample to update the old estimate Q(s,a). The details of the algorithm is shown below:

Initialize Q(s,a) arbitrarily

Repeat (for each episode):

Initialize s

Repeat (for each step of episode):

Choose a from s using policy derived from Q

Take action a, observe r, s'

$Q(s,a) \leftarrow Q(s,a) + \alpha \times [r + \gamma max'_a Q(s',a') - Q(s,a)]$

$s \leftarrow s'$

until s is terminal

$0 < \gamma < 1$ is the discount factor and $0 < \alpha < 1$ is the learning rate. The learning rate α determining how much of an effect the new sample has on the current estimate. If α is large, we will adjust quickly but may not converge. What we did is to decrease α gradually as the number of samples of $Q(s,a)$ increases.

1.2 Exploration vs Exploitation

The monkey has to determine what action to take even while it is learning. It receives rewards and punishments even as it is learning. The monkey has to spend time to learn the pipes, but we will expect the monkey to start being productive before the monkey has learned everything there is to know about the pipe. The monkey needs to decide what to do considering both the effect of its action on its immediate rewards and future state, and the need to learn for the future. This issue is known as the exploration-exploitation tradeoff. Q-learning has the following two theoretical properties:

i If every state-action pair(s,a) is visited an unbounded number of times and the learning rate α is "eventually small enough" then the Q-values converge to the limit.

ii If we exploit the Q-values in the limit, then the policy converges to the optimal policy in the limit.

In order to achieve these two requirements, we define the distinct learning rate for each state/action pair and have that rate be $\alpha_k(s,a) = 1/k$ where k is the number of times action α has been taken from state s. We adopt the "$\varepsilon - greedy$" policy that the optimal action is taken with probability $1-\varepsilon$, but with probability ε, a uniformly random action is taken to induce exploration.We took $\varepsilon = 1/t$, where t is the number of time periods that the monkey has experience.

2 Method

2.1 Rationale on Model Choice

2.2 Estimation

2.3 Numerical Challenges & Further Modification

2.3.1 Parameter Selection

3 Discussion & Possible Directions

Reference

1. Ricci F, Rokach L, Shapira B et al. (2010) **Recommender Systems Handbook**. *Springer*.

2. Koren Y, Bell R, Volinsky C. (2009) **Matrix factorization techniques for recommender systems**. *IEEE Computer* Aug 2009, 42-49.

3. Srebro N, Jaakkola T.(2003) **Weighted low-rank approximations**. *Proceedings of the Twentieth International Conference* 720727.

4. R Salakhutdinov, A Mnih. (2008) **Probabilistic Matrix Factorization**. *Advances in Neural Information Processing Systems* Vol. 20

5. Koren, Y. (2008) **Factorization Meets the Neighborhood: a Multifaceted Collaborative Filtering Model**, *Proc. 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.