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 4, 2015

1 Exploratory Analysis

2 Method

2.1 Rationale on Model Choice

2.2 Estimation

2.2.1 SVM

The SVM maps the input vectors into high-dimensional feature space and returns the maximum margin hyper plane. SVM algorithm with a linear kernal could be implemented by constructing the following problem:

$$min_{r,w,b} \frac{1}{2} ||w||^2$$
$$s.t. y^{(i)T}\left(w^T x^{(i)} + b\right) \geqslant 1, i = 1...n$$

Data:
The SVM is implemented in a static tree trunk setting, and only the first tree ahead of the monkey is considered. Given the command to jump, the current state is labeled as 1, otherwise current state is labeled as -1. Hence at each step, the training set includes a current state $x^{(i)} = ['vel', 'top', 'bot'] = [\Delta x, \Delta y, \Delta z]$ and related label $y^{(i)} \in \{-1, 1\}$.

We can then choose a high-dimension feature space. For the below example, nine features from fundamental analysis are selected by adding second and third order of the feature.

$$\mathbf{\Phi} = \left[\Delta x, \Delta y, \Delta z, \Delta x^2, \Delta y^2, \Delta z^2, \Delta x^3, \Delta y^3, \Delta z^3\right]$$
$$k(x, z) = \mathbf{\Phi}^T \mathbf{\Phi}$$

2.2.2 Model-based estimation

Optimization Algorithm
Value iteration could be implemented in order to find the value function for small MDPs. The value iteration is illustrated as follows:

For each state, initialize $V(s) := 0$.
Repeat until converge
{For each state:

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

}

After $V(s)$ and P_{sa} were learnt by the above algorithm, the optimal a is given by:

$$a^* = \max_{a \in A} P(s'|s, a) V(s)$$

2.2.3 Q-learning

2.2.4 exploitation vs. exploration

$e = 0.001$
$\epsilon = \frac{e}{k}$
k = # of actions taken from state s

2.3 **Numerical Challenges & Further Modification**

2.3.1 Parameter Selection

why $\gamma = 0.95$ $\alpha = 1/k$

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*.