# Sampling based fitted value-iteration
Report

Lampridis, Konstantinos
10753915

Wang, Yikang
student number

Wang, Yikang
student number

Wang, Yikang
student number

May 22, 2015

## 1 Introduction

This report briefly explains our approach for implementing the continuous predator-prey environment using the Sampling based fitted value-iteration algorithm, as well as the challenges we faced and exposes our findings while tweaking different parameters. We simulate the familiar predator-prey environment and gather measurements after each episode and we post the different measurements gathered for the average number of time steps and the average duration per episode over one hundred episodes. This process is repeated for a plethora of different settings governed by altering a set of parameters that affect performance, accuracy and efficiency.

## 2 Implementation

To begin, we reimplemented the framework of the environment in order to keep it more simple compared to the initial version of Autonomous Agents 1. We opted to experiment on two different state representations: a $[1 \times 2]$ vector that stores the absolute differences of the $x$ and $y$ coordinates of the agents and a scalar representation as the euclidean distance between them. In both cases, we exploited the fact that we were able to treat the actions of the predator as deterministic since the movement of the prey is just a zero mean Gaussian noise. To be more specific, in our case [Lang and Slavkovik2013].

$$s_{t+1} = f(s_t, a_t) + \epsilon_t$$

where $f$ is the deterministic function of the state and $\epsilon_t$ is the zero mean Gaussian corresponding to the dynamics of the prey. These facts leaded us to

$$E_{s'}[V(s')] \approx V(E_{s'}[s']) = V(f(s,a))$$

Digging into the fitted value-iteration algorithm, we overcome the continuity of the state and action space by sampling whenever needed. We experimented with different number of samples (both for states and actions) but also with different values for the $\gamma$ parameter of the bellman backup and the threshold that determines whether the approximation of $V$ has converged. All the values that we report in the next section are averages over one hundred of episodes.

## 3 Experiments

## 4 Conclusions

## References

[Lang and Slavkovik2013] Jrme Lang and Marija Slavkovik. Judgment aggregation rules and voting rules. In Patrice Perny, Marc Pirlot, and Alexis Tsoukis, editors, *Algorithmic Decision Theory*, volume 8176 of *Lecture Notes in Computer Science*, pages 230–243. Springer Berlin Heidelberg, 2013.