Deep Utility Network Algorithm

Kesler Isoko
EMG Team
Bionic Society
Universtiy of Sheffield

Abstract

The project focused on designing an algorithm to develope a versatile policy to configure the behaviour of a bionic arm
Key words: AES, heat transfer coefficient, distillation column

I. Introduction

The following reinforcement learning algorithm was developed in order to train a robotic arm to move as desired gievn an electrical signal from the EMG sensor. This is because by training the bionic arm on a dataset using a Neural Network we would get an overfitted model that would not proide the versatility that is desired when it comes to the patern recognigtion algorithm of a bionic arm. As such we want to train an agent to recognise which signal correspond to which movement and develop a policy through interacting with an environment that returns a reward given an action that the agent takes and updates the state.

II. Problem Specification

In order to solve this problem we have to minimise the difference between the actual movement intended and the predicted behaviour from the model. This was done by attaching gyro sensors on both hands of the amputee and an emg sensor on the missing hand to generate predictions. In order to minimise the difference we have to first specify our variables:

- Agent
- Environment
- State
- Reward
- Action

A. Agent

Within this problem the agent is the bionic arm that will take actions and receive rewards from the environemnt that will subject the bionic arm to the state.

B. Environment

The environment on this problem is the algorithm itself as it will compute all the other variables to train the agent

C. State

In this case we can let the state of the system to be the distance between the bionic arm and the actual arm. These states will provide a value for the utility and the reward.

$$s \approx [U, R] \tag{1}$$

D. Action

The action will be the adjusted predicted value of gyro from the agent. This will translate into a probability distribution as actions are the probabilities that the agent will transition from one state to another.

$$a \in A(s) \tag{2}$$

where A(s) is a probability distribution that each value of the gyro is the desired value of the gyro.

E. Reward

The reward is the result from the classification carried out by the SVM which given a certain threshold is either the correct movemnnt or is not as such the agent will be rewarded either 1 or -1.

F. Policy

The purpose of algorithm is to develope a robust policy that can be implemnted in given different scenarios to allow the agent to act in a more versatil way. This can be done by favouring those actions that will ive the agent a higher utility. This is done using a probability distirbution that can be descriped by formula 3

$$\pi^*(s) = argmax_{a \in A(s)} \Sigma(P(s'|s,a)U(s')) \tag{3}$$

III. Method

To classify whether the difference between the two actions where significant or not an SVM was implemented that would categorise each gyro response and either assign it to class 1 which correspond to a reward of R = +1 and class 0 which would correspond to a reward of R = -1. The classification was carried out by solving the following optimization:

$$w^* = argmax_w(min_n(d_H(\phi(x_n)))) \tag{4}$$

Using Langrangian multipliers we can obtain the dual form of the optimisation problem that can be solved using quadratic programming

$$max(W(\alpha) = \Sigma\alpha_i - \frac{1}{2}\Sigma y^{(i)} y^{(j)} \alpha_i \alpha_j K < x_m, x_n >) \tag{5}$$

subject to the following constraints:

$$0 \leq \alpha_i \leq C, \tag{6}$$

for i=1,2,3...,n

$$\Sigma \alpha y^{(i)} = 0 \tag{7}$$

The action can be computed from the kernel which is a similarity measure in this case a gaussian kernel was implemented as shown in equation 8 from [1].

$$K(x_i, x) = exp(-\gamma ||x_i - x_j||^2) \tag{8}$$

the learning process can be carried out by using dynamic programming and by implementing the bellmans equation to get utilities values that can be implemented to favour some actions rather then others.

$$U = R(s_i) + \gamma[max_{a \in A(s)}(\Sigma P(s'|s,a)U(s'))] \tag{9}$$

Consider the following markov decision process:

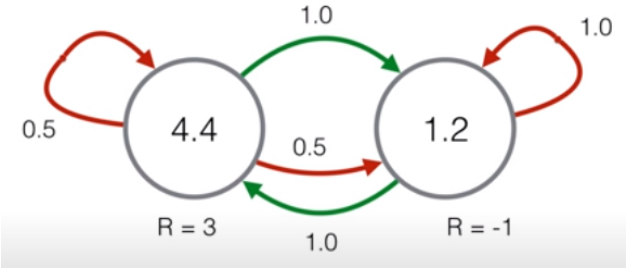

Fig. 1. shows a decision markov process with converging utilities

REFERENCES

[1] Shili Peng, Qinghua Hu, Yinli Chen, and Jianwu Dang. Improved support vector machine algorithm for heterogeneous data. *Pattern Recognition*, 48(6):2072–2083, 2015.