Programming Session: Exercise

Exercise 1: FA with a GMM

Algorithm 1 The GMMFA Algorithm

Initialize the GMM with 1 Gaussian.
loop
  Get observation $(\mathbf{x}_t, y_t)$
  Calculate the activation $w_{t,i}$ of each Gaussian in $(\mathbf{x}_t, y_t)$ (E step)
  Update the parameters of the GMM, $\mathbf{\Theta} = \{\alpha_i, \mu_i, \Sigma_i\}, i = 1, ..., K$ (M step)
  Calculate $\mu(y|\mathbf{x}_t)$
  Calculate the approximation error $e = (y_t - \mu(y|\mathbf{x}_t))^2$
  if $e \geq \text{thr}_{\text{error}}$ **then**
    Get density of samples $p(\mathbf{x}, y)$
    if $p(\mathbf{x}, y) \leq \text{thr}_{\text{density}}$ **then**
      Generate new Gaussian
    end if
  end if
end loop

Implement the algorithm GMMFA with Gaussian generation and apply it to the approximation of the target function y=sin(x) in the interval x=[-5,5]. Training samples should generated consecutively in such interval, going back and forth in [-5, 5] (biased sampling), with a sample interval of 0.1.

After every swap of the domain (back or forth), estimate the mean square error (MSE) of the approximation in the entire domain using equally distributed samples with an interval of 0.1.

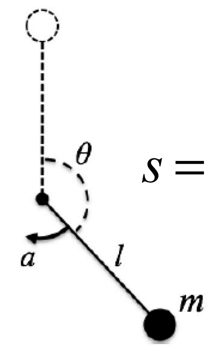
At the end of the training (after a small enough MSE is reached), plot the evolution of the MSE during the learning and compare the approximation done with respect to the target function. Also indicate in the latter plot the one standard deviation at each point.

Exercise 2: Q-Learning with a GMM

Implement Q-Learning with a GMM for the control of the inverted pendulum using the GMMFA algorithm implemented in the previous exercise.

Use the simulation of the inverted pendulum of Lecture 3. Perform training episodes of 500 iterations starting with the pendulum in the downwards position. After each training episode, perform a test episode of 500 iterations only using the policy learned so far (no learning) and memorize the accumulated reward.

Execute a total of 100 training episodes (or more if needed). After learning, present a plot with the evolution of the outcome of the test episodes and compare the results with those obtained in lecture 3 of Q-Learning with variable resolution (i.e. present both results in the same plot).



$$s = \{\theta, \dot{\theta}\}, a = \tau$$

$$r(\theta, \dot{\theta}) = -|\theta|$$

Reward r(s')

Programming Session

Implement the code in Matlab. Save all the implemented files in a folder `L6_surnames`. Implement a script called `test.m` that executes the implemented functions and presents the requested results.

Send the folder compressed (.zip) by email to alejandro.agostini@tum.de with the subject RLRWS20_L6_surnames.

Cart-pole Scenario Set-up (Advance Hands-on Proj.)

Use equations and model parameters from Deisenroth, M. P. (2010), Appendix C.2.

$$s = \{x, \dot{x}, \theta, \dot{\theta}\}$$

Deisenroth, M. P. (2010). Efficient reinforcement learning using Gaussian processes. KIT Scientific Publishing.

Ranges of variation:

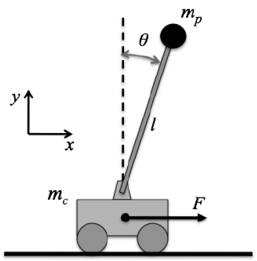
$$x = [-6, 6]$$

$$\dot{x} = [-10, 10]$$

$$\theta = [-\pi, \pi]$$

$$\dot{\theta} = [-10, 10]$$

$$F = [-10, 10]$$



Simulation interval 0.01 seconds.
Action interval of 0.1 seconds.

Reward function:

$$r(s,a) = -(1 - exp(-0.5\,(j - j_{target})\,T^{-1}\,(j - j_{target})'));$$

$$T^{-1} := A^2 \begin{bmatrix} 1 & l & 0 \\ l & l^2 & 0 \\ 0 & 0 & l^2 \end{bmatrix}$$

l = length of pendulum

$$A = 1$$

$$j = \left(x, \sin(\theta), \cos(\theta)\right)$$

$$j_{\text{target}} = (0, 0, 1)$$