

# ChainMDP description

Team 4

June 8th 2022

## 1 Default Parameters

These are default parameters for our agent. All values are provided in file `chain_test.py`. There is no initial weight.

```
sa_list = []

for i in range(env.n):
    for j in range(2):
        sa_list.append((i, j))

agent_params = { 'gamma'           : 0.9,
                  'kappa'          : 1.0,
                  'mu0'            : 0.0,
                  'lamda'          : 4.0,
                  'alpha'          : 3.0,
                  'beta'           : 3.0,
                  'max_iter'       : 100,
                  'sa_list'        : sa_list }
```

## 2 Initialization

Initialize agent by calling agent.

```
agent = agent(agent_params).
```

## 3 Training method

Below is the code used in training for k episodes. Just modify number in `training(k)` for training for k episodes.

```

def training(k):

    for episode in range(k):
        s = env.reset()
        done = False

        while not done:
            a = agent.take_action(s, 0)

            # Step environment
            s_, r, done, t = env.step(a)
            agent.observe([t, s, a, r, s_])
            agent.update_after_step(10, True)

            # Update current state
            s = s_

```

## References

- [1] Ian Osband, Daniel Russo, and Benjamin Van Roy. (more) efficient reinforcement learning via posterior sampling, 2013.
- [2] E. Markou and C. E. Rasmussen. Bayesian methods for efficient Reinforcement Learning in tabular problems. In *NeurIPS Workshop on Biological and Artificial RL*, 2019.
- [3] E. Markou. <https://github.com/stratisMarkou/sample-efficient-bayesian-rl>, 2019.