

Take Home Exam “Dynamic Optimization and Reinforcement Learning” Winter 2020/2021

Lingens/Masuhr/Trede

March 2, 2021

Exercise 1:

You are hired as a junior consultant to run one of the branches of Jane’s car rental. Your duty is to plan every evening how many cars you want to order from the national branch which then are delivered to you overnight.


When ordering cars, you have to pay a service fee of 20 € and 5 € per every car ordered. You rent out cars every day for 7 € each. Because it is your first month on the job, Jane gave you an exact booking list of how many cars are going to be rented out every day over the next 30 days. The booking list is available on the Learnweb site of the course (bookings.csv). If there are fewer cars than booked you can renege on the bookings without any costs. Customers always return cars to the national branch so you need not take returned cars into consideration.


The maximum capacity of your rental is 25 cars and you discount future cash flows at a daily rate of $\beta = 0.99$. In the morning of day 1 there are no cars in the rental.

- (a) Write down the contemporaneous profit of a business day as a function of the number of cars that you order and the number of cars present on the previous evening. Hint: To simplify the time structure of the problem, it helps to pretend that you decide about the order in the morning and the cars are then delivered immediately.
- (b) Write down the present value of the profit that you generate over the month. Also write down the associated Bellman equation. What is the state and the control of this problem. What is the trade-off that you face?
- (c) The value function is represented by a matrix of dimension 30×26 . Suppose row 5 of the matrix looks like this (these values are not the

true value function):

[5 7 11 16 22 27 28 29 ... 85 90]


 **Explain** the meaning of these matrix entries.

- (d)  Solve **numerically** for the value function. Derive the optimal ordering path for $t = 1, \dots, 30$. **Plot** the optimal path. **Calculate** the total profit in the month. **Analyse** how this changes if the service fee per order is drastically decreased to 1 €.

Exercise 2:

You have managed successfully Jane's car rental for the first month. Now your duty is to run it for the rest of the time. However, you lack the exact rental information. Jane tells you that she thinks that demand follows a simple Poisson process with the arrival rate λ . Based on this information you plan your orders from the national branch.

- (a) **Write down** the Bellman equation for this problem. Which role does the uncertainty concerning rentals play?

-  (b) Solve this problem **numerically** using value function iteration. Derive the policy function. **Plot** the policy function for $\lambda = 2$, $\lambda = 9$ and $\lambda = 15$.

Exercise 3:

Jane is going to open another branch of her rental at Palma de Mallorca. To learn about the dynamics of rental requests she acquired three years of data from another company. Using this data set, she suspects **three main factors** to describe the current number of rental requests:

- Weather (temperature) forecasts: one week before up to the day before the current day,
- previous rental requests: one week before up to the day before the current day,
- the current day of week.

The data set `janestate.RData` contains all above mentioned states as **normalized values**, i.e. they have mean 0 and standard deviation 1, the data set `janerentals.RData` contains a vector of three years of rental data (**non normalized**). The Mallorcan main branch is very flexible and does not charge

Jane any costs per order, so she only pays the variable cost of 5€ per ordered car. On the other, hand parking space in Palma is very expensive so she'll be charged 6€ for each car that is left at her rental in the evening. Jane orders cars in the evening and they'll be delivered the next morning.

- (a) Use the `load()` command to load `janestate.RData` into R; it should look like this:

```
rentals_lag1 rentals_lag2 rentals_lag3 rentals_lag4 rentals_lag5 rentals_lag6 rentals_lag7 fc_1_wins1 fc_1_wins2 fc_1_wins3 fc_1_wins4 fc_1_wins5 fc_1_wins6 fc_1_wins7 day
[1] 2.5200057 1.3523978 -0.3848238 -0.9756340 -1.5551511 -0.7796991 0.1889173 -0.4380412 -0.4369913 -0.4025015 -0.4779864 -0.4098175 -0.3368312 -0.3163870 -1.498928660
[2] -0.7823208 2.5172134 1.3523978 -0.3848238 -0.9756340 -1.5551511 -0.7805164 -0.4845797 -0.2421869 -0.6177876 -0.2140961 -0.1248941 -0.2402085 -0.1719692 -0.997038901
[3] -0.9760155 -0.7823208 2.5172134 1.3523978 -0.3848238 -0.9756340 -1.5560793 -0.4520205 -0.6057628 -0.7315151 -0.4578106 -0.5776476 -0.1638382 -0.7801892 -0.492155146
[4] -1.3652031 -0.9772386 -0.7823208 2.5172134 1.3523978 -0.3848238 -0.9744071 -0.3025703 -0.4598703 -0.1748843 -0.4438110 -0.3692050 -0.6246600 -0.1722466 -0.002736632
[5] -1.5365002 -1.3652031 -0.9772386 -0.7823208 2.5172134 1.3523978 -0.3025703 -0.4598703 -0.1748843 -0.4438110 -0.3692050 -0.6246600 -0.1722466 -0.002736632
[6] -0.9760155 -1.3652031 -0.9772386 -0.7823208 2.5172134 1.3523978 -0.3025703 -0.4598703 -0.1748843 -0.4438110 -0.3692050 -0.6246600 -0.1722466 -0.002736632
```

Now, have a look at `janestate`, it shows all demand side states. Which supply side state would you need to add in order to achieve the complete set of states? How many input neurons do you need to represent the action value function?

- (b) Create an artificial neural network with an input layer of size according to (a), a single hidden layer of 50 *swish* neurons and a linear output layer.
- (c) Janes car rental itself is obviously non episodic, but since we only have data for three years, we just work as if it was an episodic task, i.e. after we reach the end of the data set, we start another epoch of learning, beginning with the first day.

We always start an epoch at the evening of day one with no cars in stock and decide to order no car for the next day. Use on policy SARSA to let your ANN learn the action value function according to the following (hyper)parameters:

- Number of training epochs: $N = 200$, number of days per epoch: $TT = 1096$
- Cost of requesting cars: 0 Euro per order plus 5€ per car
- Cost of parking cars overnight: 6€ per car
- Revenue: 10€ per car
- $\gamma = 0.9$
- Maximum number of cars at the rental: 25
- Action set: $\{0, 1, 2, \dots, 25\}$
- $\epsilon = \min\left(0.05, \frac{2}{\ln(n+1)^3}\right)$
- $\alpha = \frac{0.00005}{\sqrt{\ln(n+2)}}$

Store the rewards of all iterations of an epoch and compute the average rewards over each epoch. Store the ANN with the highest average reward over an epoch.

Hints:

- When computing the output of the net and the gradient do not use the raw state *cars evening* but work with $\frac{CarsEvening - maxCars/2}{10}$, instead.
 - When computing the output of the net and the gradient do not use the raw action but work with $\frac{action - maxCars/2}{10}$, instead.
- (d) Compare the average reward using the ANN to this simple rule of thumb: "Request cars until the number of cars tomorrow morning equals the number of rentals, today"