Going Shopping

Tyler Gillette -- Eric Smrkovsky -- Vincent Weinberger

Our Group is Tyler, Eric, Vince, and Mike. We have created a scenario where we need X items and are going grocery shopping. We have worlds where you can either go to a standard store and do shopping for one item or you can go to a big box store like Costco and finish out your shopping. There are multiple worlds created with varying rewards for the states so that we can have a variety of outcomes. This ideology creates something closer to a real life outcome. We have chosen to create different MDP worlds setup as 2d grids. Each of these grids we have performed Value Iteration, and Q-learning. We have each algorithm setup as an experiment on a specific world. Each world is printed neatly to the console.

Part 1

Domain:

The goal of the domain is to travel to each store and get all X items with the minimum owed balance on your credit card, and least amount of time spent. There will be 21 states each with 1-2 possible actions, a reward, and store each with a possible exit.

There are possible actions of going to different stores such as a regular grocery store or a big box store like Costco. Any state can take an action to purchase the rest of the items by going to Costco and then going home. There is a 75% chance the item is in stock, and 25% chance it's not and we must drive to Costco to purchase the rest of the items there.

Rewards are a combination of cost to purchase an item and time spent, combined into one number we are treating as money. We considered a time cost for each action as a living reward but did not find an adequate way to implement it.

The requirements to head home or exit is that you must purchase all X items to take an exit. The goal is to exit with the least amount of debt owed. Which means that you will need to find the optimal route of buying groceries depending on the reward of either a regular store or Costco.

We chose to go with models where we know all the prices of each item at every store. This would fit the modern day approach to shopping where one can look up an item online and see where it is cheapest, or see what's on sale at any given time.

Part 2

Algorithm Details:

- Value Iteration
    - $U(s) = R(s) + max\ P(s' \square s,\ a)U(s')$
    - Value iteration is how you can compute an optimal MDP and the value.
    - Value iteration starts at the final value and works its way towards the first value, refining a V' value. This ends when the value if V converges.
- Q-Learning
    - $Q(s,a) = R(s) + P(s' \square s,\ a) maxQ(s',\ a')$
    - Q-Learning is a reinforcement algorithm that learns how good actions are.
    - Q-Learning takes the best policy based on the max expected value of the total reward.

Part 3

Problem Representation:

- Describe the model (Markov Decision Process) for your domain.
    - One of our Worlds for Example

(11, 1) Smart & Final Extra!
Action = EXIT
Reward = 10
	(11, 2) Costco
	Action = EXIT
	Reward = 0.001

(10, 1) Walmart Neighborhood	(10, 2) Costco

Going Shopping

Tyler Gillette -- Eric Smrkovsky -- Vincent Weinberger

Market
Action = North, East
Reward = 9
	Action = EXIT
	Reward = 5

(9, 1) Sprouts Farmers Market
Action = North, East
Reward = 8
	(9, 2) Costco
	Action = EXIT
	Reward = 10

(8, 1) WinCo Foods
Action = North, East
Reward = 7
	(8, 2) Costco
	Action = EXIT
	Reward = 15

(7, 1) FoodMaxx
Action = North, East
Reward = 6
	(7, 2) Costco
	Action = EXIT
	Reward = 19

(6, 1) Grocery Outlet
Action = North, East
Reward = 5
	(6, 2) Costco
	Action = EXIT
	Reward = 20

(5, 1) Aldi’s
Action = North, East
Reward = 4
	(5, 2) Costco
	Action = EXIT
	Reward = 30

(4, 1) Savemart
Action = North, East
Reward = 3
	(4, 2) Costco
	Action = EXIT
	Reward = 33

(3, 1) Vons
Action = North, East
Reward = 2
	(3, 2) Costco
	Action = EXIT
	Reward = 42

(2, 1) Albertsons Action = North, East	(2, 2) Costco Action = EXIT

Going Shopping

Tyler Gillette -- Eric Smrkovsky -- Vincent Weinberger

Reward = 1	Reward = 45

(1, 1) Walmart
Action = North, East
Reward = 0.001
	(1, 2) Costco
	Action = EXIT
	Reward = 50

- States:
    - 22 States
        - 11 Grocery Stores
            - Walmart
            - Albertsons
            - Vons
            - Aldi’s
            - Grocery Outlet
            - FoodMaxx
            - WinCo Foods
            - Sprouts Farmers Market
            - Walmart Neighborhood Market
            - Smart & Final Extra!
        - 11 Big Box Stores
            - Costco
- Actions:
    - 3 Available actions
        - North
        - East
        - Exit
- Rewards:
    - 22 Available rewards
        - $0.001
            - x2

    - $1.00
    - $2.00
    - $3.00
    - $4.00
    - $5.00
      - x2
    - $6.00
    - $7.00
    - $8.00
    - $9.00
    - $10.00
      - x2
    - $15.00
    - $19.00
    - $20.00
    - $30.00
    - $33.00
    - $42.00
    - $45.00
    - $50.00
  - Transition Model:
    - 75% probability going in desired direction
    - 25% probability going a non-desirable direction.

Part 4

Implementation:

- Tools
  - Visual Studio Code
    - Python Extension
    - Python Linter

  - PowerShell
  - Command Line
  - Jupyter Notebook
  - Discord
  - Github
  - Spotify
  - YouTube
  - W3School

- Programming Languages
  - Python 3.8.0
  - Python 3.9.0

- Key Algorithms
  - We focused on two algorithms in our implementation of the shopping worlds.
  - Value Iteration
    - $U(s) = R(s) + max\ P(s' \mid s,\ a)U(s')$
    - Value iteration is how you can compute an optimal MDP and the value.
    - Value iteration starts at the final value and works its way towards the first value, refining a V’ value. This ends when the value if V converges.
  - Q-Learning
    - $Q(s,a) = R(s) + P(s' \mid s,\ a)maxQ(s',\ a')$
    - Q-Learning is a reinforcement algorithm that learns how good actions are.
    - Q-Learning takes the best policy based on the max expected value of the total reward.

Going Shopping

Tyler Gillette -- Eric Smrkovsky -- Vincent Weinberger

Part 5

Experimental Results:

- **Experiment 1- Value Iteration on Midterm World**
    - Value iteration turned out great in the Midterm world. We ran into some issues when leaving the initial values of the states at 0, so we settled for initial values of 0.01 for our states which yielded the correct results.

```
Original Midterm World

  10   None
0.01   -10
0.01   1


Midterm World w/ Value Iteration


 --Step 0 --
0
0    0
0    0
delta:  10.0   gamma: 0.999

 --Step 1 --
10.0
0.01    -10.0
0.01    1.0
delta:  4.995   gamma: 0.999

 --Step 2 --
10.0
5.0    -10.0
0.76    1.0
delta:  3.2479987500000003   gamma: 0.999

 --Step 3 --
10.0
5.0    -10.0
4.01    1.0
delta:  0   gamma: 0.999
```

Going Shopping

Tyler Gillette -- Eric Smrkovsky -- Vincent Weinberger

- **Experiment 2 - Value Iteration on Shopping World**
    - Value iteration on the shopping world appeared to be very successful and we seemed to be getting correct results. Although we needed a more stochastic world to operate in to emulate the randomness of the real world. We explore this in Experiment 4.

```
Shopping World
Rewards of each state at the beginning.
  10     0.001
   9       5
   8      10
   7      15
   6      19
   5      20
   4      30
   3      33
   2      42
   1      45
0.001       50

State   Money Saved
(0,10)   39.88   Hit all 10 stores, NO COSTCO

(0,9)    40.13   Went to 9 stores then COSTCO

(0,8)    40.56   Went to 8 stores then COSTCO

(0,7)    41.14   Went to 7 stores then COSTCO

(0,6)    41.57   Went to 6 stores then COSTCO

(0,5)    41.14   Went to 5 stores then COSTCO

(0,4)    42.32   Went to 4 stores then COSTCO

(0,3)    42.95   Went to 3 stores then COSTCO

(0,2)    45.25   Went to 2 stores then COSTCO

(0,1)    47.19   Went to 1 store then COSTCO
```

```
 (0,0)    49.93   Went to directly to COSTCO

Best Policy Found
^  to COSTCO.
^  to COSTCO.
^  to COSTCO.
^  to COSTCO.
^  to COSTCO.
^  to COSTCO.
^  to COSTCO.
^  to COSTCO.
>  to COSTCO.
>  to COSTCO.
>  to COSTCO.
```

- **Experiment 3 - Value Iteration on Altered Shopping World**
    - Successfully print each state with possible directions and q values
    - Not practical for the midterm world because of the way rewards are set up

```
Shopping World
 Rewards of each state at the beginning.
   10      0.001
    9        5
    8       10
    7       15
    6       19
    5       50
    4       30
    3       33
    2       42
    1       45
0.001        40

 State   Money Saved
 (0,10)   39.88   Hit all 10 stores, NO COSTCO

 (0,9)    40.13   Went to 9 stores then COSTCO

 (0,8)    40.56   Went to 8 stores then COSTCO
 (0,7)    41.14   Went to 7 stores then COSTCO

 (0,6)    41.57   Went to 6 stores then COSTCO

 (0,5)    55.77   Went to 5 stores then COSTCO

 (0,4)    53.28   Went to 4 stores then COSTCO

 (0,3)    51.16   Went to 3 stores then COSTCO

 (0,2)    50.82   Went to 2 stores then COSTCO

 (0,1)    50.32   Went to 1 store then COSTCO

 (0,0)    47.69   Went to directly to COSTCO

Best Policy Found
^  to COSTCO.
^  to COSTCO.
^  to COSTCO.
^  to COSTCO.
^  to COSTCO.
>  to COSTCO.
^  to COSTCO.
^  to COSTCO.
^  to COSTCO.
^  to COSTCO.
^  to COSTCO.
```

- **Experiment 4 - Random World Value iteration**
    - In our 4th experiment, we wanted to add a more stochastic element to our world.
    - We added randomness to our rewards that were awarded for shopping at grocery stores that were not Costco.
    - They had a low end of -5, in which case our shopper did not find any items they were looking for and had wasted time and gas money getting to that particular store.
    - It went all the way up to a maximum of +30, which represents our shopper finding multiple items that they needed and at a greatly reduced cost. This

maximum reward declined as our shopper went to more and more grocery stores due to the fact that our stress level is ever increasing and the number of items we are looking for is likely decreasing as well; at least slightly.

- We also still have a 25% chance in each Grocery Store state to just give up on finding deals and go straight to Costco where we know we will find the rest of what we need. This emulates stress levels of an individual varying from day to day, or trip to trip.Sometimes we have the mental capacity to make it to most/all of the stores and sometimes we give up after X amount of stores and go to Costco.

```
 Shopping World
 Rewards of each state at the beginning.
0.001        50
   -2       45
   21        42
   10        33
   -1       30
    7       20
   -4       19
   -4       15
    2       10
   12         5
   -2     0.001

 State   Money Saved
 (0,10)   -2.0   Hit all 10 stores, NO COSTCO

 (0,9)    20.75   Went to 9 stores then COSTCO

 (0,8)    20.05   Went to 8 stores then COSTCO

 (0,7)    14.77   Went to 7 stores then COSTCO

 (0,6)    17.88   Went to 6 stores then COSTCO

 (0,5)    30.62   Went to 5 stores then COSTCO

 (0,4)    34.56   Went to 4 stores then COSTCO
 (0,3)    52.39   Went to 3 stores then COSTCO

 (0,2)    70.75   Went to 2 stores then COSTCO

 (0,1)    62.24   Went to 1 store then COSTCO

 (0,0)    59.13   Went to directly to COSTCO

Best Policy Found
.  to COSTCO.
>  to COSTCO.
^  to COSTCO.
^  to COSTCO.
>  to COSTCO.
>  to COSTCO.
>  to COSTCO.
>  to COSTCO.
^  to COSTCO.
^  to COSTCO.
^  to COSTCO.
```

- **Experiment 5 - QLearning on Midterm World**
    - In this experiment, we appeared to be getting results from our QLearning on the Midterm world to moderate success.

```
Executing trial  0
In state (0, 0) can go:
East
North
Q-Values of:
N:  0.0
S:  0.0
E:  0.01
W:  0.0

Executing trial  0
In state (0, 1) can go:
East
North
Q-Values of:
```

```
N:  0.0
S:  0.0
E:  0.0
W:  0.0

Executing trial  0
In state (1, 0) can go:
None available.
Q-Values of:
N:  0.0
S:  0.0
E:  0.0
W:  0.0

Executing trial  0
In state (1, 1) can go:
None available.
Q-Values of:
N:  0.0
S:  0.0
E:  0.0
W:  0.0

Executing trial  0
In state (0, 2) can go:
None available.
Q-Values of:
N:  0.0
S:  0.0
E:  0.0
W:  0.0

Executing trial  50
In state (0, 0) can go:
East
North
Q-Values of:
N:  0.1204
S:  0.0
E:  3.4233
W:  0.0

Executing trial  50
In state (0, 1) can go:
East
```

```
North
Q-Values of:
N:  0.0
S:  0.0
E:  0.9358
W:  0.0

Executing trial  50
In state (1, 0) can go:
None available.
Q-Values of:
N:  0.0
S:  0.0
E:  0.0
W:  0.0

Executing trial  50
In state (1, 1) can go:
None available.
Q-Values of:
N:  0.0
S:  0.0
E:  0.0
W:  0.0

Executing trial  50
In state (0, 2) can go:
None available.
Q-Values of:
N:  0.0
S:  0.0
E:  0.0
W:  0.0

Executing trial  100
In state (0, 0) can go:
East
North
Q-Values of:
N:  0.1204
S:  0.0
E:  1.3982
W:  0.0

Executing trial  100
```

```
In state (0, 1) can go:
East
North
Q-Values of:
N:  0.0
S:  0.0
E:  0.9358
W:  0.0

Executing trial  100
In state (1, 0) can go:
None available.
Q-Values of:
N:  0.0
S:  0.0
E:  0.0
W:  0.0

Executing trial  100
In state (1, 1) can go:
None available.
Q-Values of:
N:  0.0
S:  0.0
E:  0.0
W:  0.0

Executing trial  100
In state (0, 2) can go:
None available.
Q-Values of:
N:  0.0
S:  0.0
E:  0.0
W:  0.0

Executing trial  150
In state (0, 0) can go:
East
North
Q-Values of:
N:  0.1204
S:  0.0
E:  0.9051
W:  0.0
```

Going Shopping

Tyler Gillette -- Eric Smrkovsky -- Vincent Weinberger

```
Executing trial  150
In state (0, 1) can go:
East
North
Q-Values of:
N:  0.0
S:  0.0
E:  0.9358
W:  0.0
Executing trial  150

In state (1, 0) can go:
None available.
Q-Values of:
N:  0.0
S:  0.0
E:  0.0
W:  0.0

Executing trial  150
In state (1, 1) can go:
None available.
Q-Values of:
N:  0.0
S:  0.0
E:  0.0
W:  0.0

Executing trial  150
In state (0, 2) can go:
None available.
Q-Values of:
N:  0.0
S:  0.0
E:  0.0
W:  0.0

Executing trial  200
In state (0, 0) can go:
East
North
Q-Values of:
N:  0.1204
S:  0.0
```

```
E:  0.6734
W:  0.0

Executing trial  200
In state (0, 1) can go:
East
North
Q-Values of:
N:  0.0
S:  0.0
E:  0.9358
W:  0.0

Executing trial  200
In state (1, 0) can go:
None available.
Q-Values of:
N:  0.0
S:  0.0
E:  0.0
W:  0.0

Executing trial  200
In state (1, 1) can go:
None available.
Q-Values of:
N:  0.0
S:  0.0
E:  0.0
W:  0.0

Executing trial  200
In state (0, 2) can go:
None available.
Q-Values of:
N:  0.0
S:  0.0
E:  0.0
W:  0.0
```

Part 6

Contributions:

Going Shopping

Tyler Gillette -- Eric Smrkovsky -- Vincent Weinberger

Tyler Gillette

- Python Coding
    - Implementing 2d grid worlds
    - Structuring/Formatting
    - Peer programming
    - Implementing algorithms
    - Refactoring

    - Setup
    - Cleanup
- ReadMe
    - Implemented
    - Maintained
- Final Report
    - Setup
    - Structure
    - Main Chunk

Eric Smrkovsky

- Python Coding
    - Found and added required support files
    - Implementing 2d grid worlds
    - Peer programming
    - Presentations
    - Debugging
    - Q-Learning function

Vincent Weinberger

- Python Coding
    - Implementing 2d grid worlds
    - Debugging

- Final Report
    - Experiments and Conclusion sections of Final Report

Michael Novella

- DID NOT PARTICIPATE.

Part 7

Conclusion:

Started out with 5 members among which to divide work and ended up with only 3 members giving meaningful contributions; making the project a bit more difficult for us. We managed to implement Value iteration, Policy Extraction, and Q-learning for our tests on the Midterm world. Value iteration and Policy Extraction also worked for our variations of our shopping world, but we got stuck as far as Q-learning applied to the shopping worlds. We were able to apply a more stochastic approach to our shopping world by randomizing our rewards within a certain range to emulate real world randomness when shopping.Overall I think we all learned a lot from this project and the experiments that we performed. I believe we all have a greater understanding and appreciation for the field of Machine Learning and will be able to use these tools we have acquired down the line in our careers.

Thank you Dr. Ruby!

Part 8

Links:

GitHub Project

GitHub Book Code

W3 School