Statistical Learning Engineers Final Project

Muchen Li and Julian Brown

This report contains three tasks. In Task 1 three different algorithms are applied to either regression or classification problem. Task 2 and Task 3 pertain to reinforcement learning algorithms.

Task 1 (done by Julian)

For the Machine Learning Task, a dataset including the 2020 offensive statistics of all 32 teams in the NFL was analyzed along with their win rate over the course of that season, in order to attempt to predict each teams win rate over the current season, 2021. Below are two tables with the data that were used.

				Tot Yds & TO						Passing							Rushing					Penalties				
Rk	Tm	G	PF	Yds	Ply	Y/P	TO	FL	1stD	Cmp	Att	Yds	TD	Int	NY/A	1stD	Att	Yds	TD	Y/A	1stD	Pen	Yds	1stPy	Sc%	TO%	EXP
1	Green Bay Packers	16	31.8	389.0	61.9	6.3	0.69	0.38	22.4	23.3	32.9	256.6	3.00	0.31	7.5	13.5	27.7	132.4	1.00	4.8	7.13	5.25	42.8	1.75	49.7	5.6	18.2
2	Buffalo Bills	16	31.3	396.4	64.6	6.1	1.38	0.69	24.8	25.6	37.3	288.8	2.50	0.69	7.4	15.0	25.7	107.7	1.00	4.2	7.44	6.38	58.8	2.38	49.4	11.8	14.3
3	Tampa Bay Buccaneers	16	30.8	384.1	63.6	6.0	1.06	0.31	22.8	25.6	39.1	289.1	2.63	0.75	7.1	14.9	23.1	94.9	1.00	4.1	5.13	5.25	44.7	2.75	47.8	8.9	15.4
4	Tennessee Titans	16	30.7	396.4	64.4	6.2	0.75	0.31	23.8	19.8	30.3	228.3	2.06	0.44	7.2	12.7	32.6	168.1	1.63	5.2	8.88	5.38	48.9	2.25	47.9	7.2	15.3
5	New Orleans Saints	16	30.1	376.4	65.3	5.8	1.06	0.56	22.9	23.1	32.6	234.9	1.75	0.50	6.8	12.4	30.9	141.6	1.88	4.6	9.19	6.13	62.8	1.31	45.5	9.0	11.1
6	Kansas City Chiefs	16	29.6	415.8	66.1	6.3	1.00	0.56	24.8	26.3	39.4	303.4	2.50	0.44	7.4	15.9	25.2	112.4	0.81	4.5	6.88	6.56	57.4	2.00	47.9	9.2	17.6
7	Baltimore Ravens	16	29.3	363.1	62.1	5.9	1.13	0.44	20.4	16.1	25.4	171.2	1.69	0.69	6.3	8.88	34.7	191.9	1.50	5.5	10.4	6.50	60.1	1.19	45.3	10.6	9.27
8	Seattle Seahawks	16	28.7	369.5	63.9	5.8	1.13	0.31	22.3	24.3	35.2	246.3	2.50	0.81	6.5	13.5	25.7	123.2	0.94	4.8	6.94	5.25	41.4	1.81	44.4	10.1	8.08
9	Indianapolis Colts	16	28.2	378.1	64.5	5.9	0.94	0.25	22.8	23.2	34.5	253.3	1.50	0.69	7.1	12.6	28.7	124.8	1.25	4.3	8.06	5.88	56.2	2.13	44.4	8.8	10.5
10	Las Vegas Raiders	16	27.1	383.3	64.8	5.9	1.63	1.00	22.4	23.1	34.4	263.6	1.75	0.63	7.3	12.9	28.6	119.8	1.25	4.2	7.56	6.13	53.5	2.00	47.9	14.8	9.23
11	Minnesota Vikings	16	26.9	393.3	63.9	6.2	1.44	0.63	23.9	21.8	32.3	250.6	2.19	0.81	7.2	13.3	29.3	142.7	1.25	4.9	8.69	5.13	40.6	2.00	39.8	12.5	9.86
12	Pittsburgh Steelers	16	26.0	334.6	65.2	5.1	1.13	0.44	20.1	26.8	41.0	250.2	2.19	0.69	6.0	12.9	23.3	84.4	0.75	3.6	5.06	5.06	43.2	2.19	37.2	8.5	4.89
13	Arizona Cardinals	16	25.6	384.6	67.7	5.7	1.31	0.50	23.8	24.2	35.9	244.8	1.69	0.81	6.5	13.2	29.9	139.8	1.38	4.7	8.50	7.06	54.3	2.13	40.2	11.7	6.48
14	Cleveland Browns	16	25.5	369.6	63.9	5.8	1.00	0.50	22.2	19.7	31.3	221.2	1.69	0.50	6.7	12.2	30.9	148.4	1.31	4.8	8.31	6.25	54.4	1.69	40.6	9.7	9.97
15	Miami Dolphins	16	25.3	339.0	63.8	5.3	1.25	0.44	21.6	23.1	34.9	233.5	1.50	0.81	6.3	12.9	26.8	105.5	0.94	3.9	6.25	4.63	39.7	2.38	41.4	11.0	3.87
16	Atlanta Falcons	16	24.8	368.4	67.4	5.5	1.13	0.44	22.9	25.5	39.3	272.7	1.69	0.69	6.5	15.2	25.6	95.8	0.81	3.7	5.38	5.19	46.0	2.31	44.8	10.3	8.11
17	Dallas Cowboys	16	24.7	371.8	69.6	5.3	1.63	0.81	23.2	25.8	39.9	260.1	1.56	0.81	6.1	14.4	26.9	111.8	0.88	4.2	7.19	6.00	53.1	1.56	40.6	14.4	3.69
18	Houston Texans	16	24.0	375.3	58.8	6.4	1.13	0.69	20.4	23.9	34.1	283.6	2.06	0.44	7.6	13.9	21.5	91.6	0.63	4.3	5.19	5.00	28.7	1.31	42.9	9.8	8.76
19	Los Angeles Chargers	16	24.0	382.1	70.4	5.4	1.00	0.38	23.3	25.8	39.2	270.6	1.94	0.63	6.5	14.1	29.1	111.5	0.75	3.8	6.94	5.31	44.4	2.25	38.3	8.6	8.10
20	Detroit Lions	16	23.6	350.2	61.9	5.7	1.31	0.50	21.9	23.4	36.4	256.5	1.69	0.81	6.6	13.8	22.9	93.7	1.06	4.1	5.81	5.94	53.8	2.31	38.5	11.2	5.92
21	San Francisco 49ers	16	23.5	370.1	65.4	5.7	1.94	0.88	21.9	23.2	35.6	252.1	1.56	1.06	6.6	13.6	27.3	118.1	1.19	4.3	6.31	5.31	45.7	2.00	36.3	14.8	3.48
22	Los Angeles Rams	16	23.3	377.0	68.0	5.5	1.56	0.69	22.0	24.5	36.9	250.9	1.25	0.88	6.5	12.9	29.6	126.1	1.19	4.3	7.69	4.44	40.9	1.44	34.2	12.5	3.58
23	Chicago Bears	16	23.3	331.4	65.2	5.1	1.38	0.38	20.9	25.1	38.4	228.4	1.63	1.00	5.6	13.6	24.6	102.9	0.75	4.2	5.81	5.50	48.6	1.50	38.9	12.0	2.18
24	Carolina Panthers	16	21.9	349.5	62.1	5.6	1.31	0.31	20.9	23.3	34.4	243.0	1.00	1.00	6.6	12.0	25.4	106.5	1.19	4.2	6.88	5.94	47.0	2.06	40.3	13.2	5.08
25	Washington Football Team	16	20.9	317.3	65.7	4.8	1.69	0.69	20.1	24.3	37.6	216.6	1.00	1.00	5.3	11.5	25.0	100.7	1.13	4.0	6.75	5.44	43.6	1.88	33.3	13.1	-0.65
26	Philadelphia Eagles	16	20.9	334.6	66.6	5.0	1.81	0.56	21.0	20.9	37.4	207.9	1.38	1.25	5.0	11.1	25.2	126.7	1.00	5.0	7.13	6.69	53.3	2.81	27.9	15.3	-0.85
27	New England Patriots	16	20.4	327.3	61.2	5.3	1.19	0.31	20.8	17.7	27.5	180.6	0.75	0.88	6.1	9.81	31.4	146.6	1.25	4.7	8.94	3.88	33.4	2.00	36.9	12.1	3.49
28	Denver Broncos	16	20.2	335.6	64.4	5.2	2.00	0.56	19.3	19.8	34.8	215.7	1.31	1.44	5.9	11.0	27.6	119.9	0.81	4.3	5.88	5.06	41.9	2.38	33.0	17.0	-1.36
29	Cincinnati Bengals	16	19.4	319.8	65.0	4.9	1.50	0.81	19.9	23.3	36.3	215.5	1.19	0.69	5.5	12.6	25.7	104.3	0.81	4.1	5.75	5.19	42.2	1.56	33.0	12.8	-0.32
30	Jacksonville Jaguars	16	19.1	326.1	62.3	5.2	1.56	0.56	19.4	24.2	38.5	231.2	1.56	1.00	5.6	12.8	21.1	94.9	0.56	4.5	5.00	6.69	66.9	1.56	30.4	14.0	-0.38
31	New York Giants	16	17.5	299.6	60.4	5.0	1.38	0.69	18.6	20.1	32.3	189.1	0.75	0.69	5.3	11.1	24.9	110.5	0.81	4.4	5.69	5.06	39.6	1.75	33.5	12.6	-1.67
32	New York Jets	16	15.2	279.9	59.3	4.7	1.19	0.31	16.8	18.3	31.2	174.8	1.00	0.88	5.2	9.13	25.4	105.2	0.56	4.1	5.88	6.31	59.5	1.81	26.3	10.9	-3.69

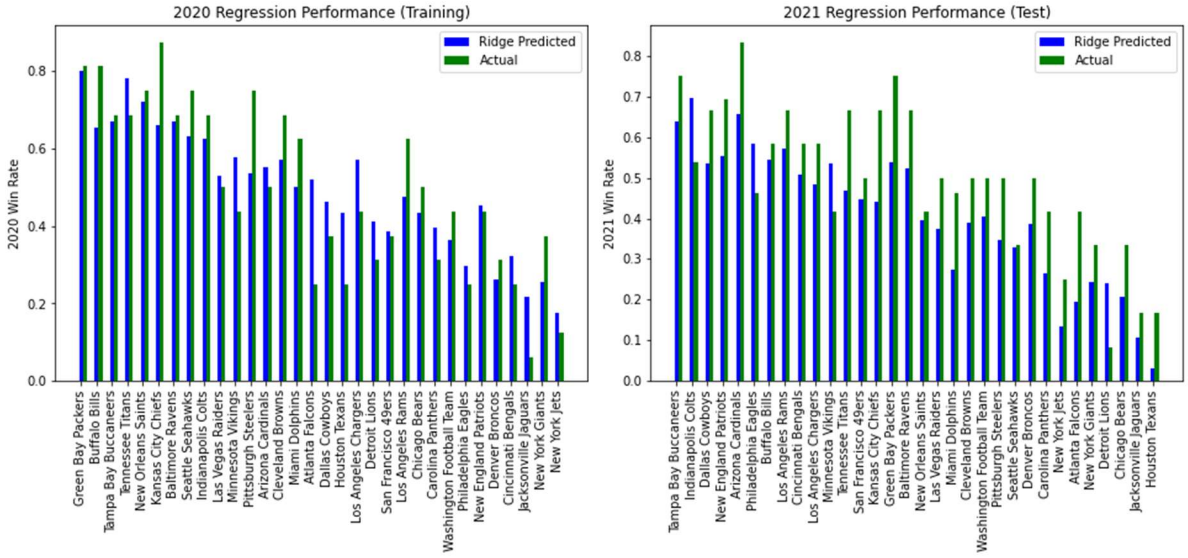
2020 NFL per-game Offensive Statistics – These data were used as training for each model.

				Tot Yds & TO						Passing							Rushing					Penalties				
Rk	Tm	G	PF	Yds	Ply	Y/P	TO	FL	1stD	Cmp	Att	Yds	TD	Int	NY/A	1stD	Att	Yds	TD	Y/A	1stD	Pen	Yds	1stPy	Sc%	TO%	EXP
1	Tampa Bay Buccaneers	13	31.5	410.2	67.1	6.1	1.23	0.46	24.5	29.6	43.3	314.2	2.77	0.77	7.0	16.3	22.5	96.0	1.08	4.3	6.54	6.31	57.3	1.69	45.0	10.7	13.6
2	Dallas Cowboys	13	29.2	409.1	68.6	6.0	1.38	0.54	22.9	26.1	38.5	280.6	2.00	0.85	6.9	14.3	28.2	128.5	0.85	4.6	6.46	7.92	70.1	2.15	41.1	11.3	6.57
3	Indianapolis Colts	13	28.5	368.1	63.8	5.8	1.23	0.77	22.3	20.6	32.7	216.4	1.69	0.46	6.3	10.7	29.5	151.7	1.54	5.1	9.54	4.77	42.9	2.08	45.1	9.2	9.96
4	Los Angeles Rams	13	28.2	384.5	62.0	6.2	1.08	0.31	21.1	24.4	36.5	287.3	2.54	0.77	7.5	14.2	23.9	97.2	0.62	4.1	5.46	4.92	42.2	1.38	48.6	9.4	10.1
5	Arizona Cardinals	13	28.2	374.8	65.0	5.8	1.00	0.23	21.6	23.4	32.3	252.2	1.69	0.77	7.3	12.6	30.3	122.5	1.62	4.0	7.31	6.38	56.2	1.69	45.3	8.6	9.29
6	Buffalo Bills	13	27.9	382.9	65.6	5.8	1.38	0.46	22.8	25.4	38.6	261.9	2.15	0.92	6.5	13.9	25.3	121.0	1.00	4.8	7.23	6.85	60.1	1.62	44.8	11.9	7.79
7	Cincinnati Bengals	13	27.2	358.8	62.4	5.8	1.62	0.54	19.8	22.1	32.2	250.1	2.00	1.08	7.1	12.4	27.1	108.7	1.15	4.0	5.85	4.00	34.1	1.62	41.1	12.6	4.57
8	Los Angeles Chargers	13	27.0	385.2	64.5	6.0	1.15	0.31	23.1	26.0	38.8	280.3	2.31	0.85	6.8	14.7	23.6	104.9	0.92	4.4	6.38	7.15	61.1	2.00	44.4	11.1	9.38
9	Kansas City Chiefs	13	27.0	389.6	66.7	5.8	1.77	0.85	24.4	26.2	39.9	277.9	2.08	0.92	6.7	15.5	24.9	111.7	0.92	4.5	6.85	6.69	55.4	2.08	45.4	16.2	9.89
10	New England Patriots	13	26.9	346.5	61.4	5.6	1.23	0.62	20.7	21.5	30.3	223.0	1.38	0.62	6.9	10.9	29.2	123.5	1.15	4.2	7.54	5.62	51.2	2.23	48.2	10.9	5.85
11	Minnesota Vikings	13	26.5	390.5	66.2	5.9	0.85	0.46	21.4	24.7	37.0	267.4	2.08	0.38	7.0	12.8	27.8	123.2	0.69	4.4	6.85	7.31	69.2	1.69	42.0	7.3	6.53
12	Philadelphia Eagles	13	25.9	356.2	62.6	5.7	1.00	0.38	20.9	17.9	29.2	196.2	1.15	0.62	6.3	9.69	31.7	160.0	1.46	5.0	9.85	6.38	49.5	1.38	42.1	9.8	6.58
13	San Francisco 49ers	13	25.3	364.3	61.8	5.9	1.38	0.69	21.0	19.9	30.7	240.4	1.54	0.69	7.3	11.5	29.1	123.9	1.31	4.3	7.54	6.08	63.3	2.00	40.7	11.4	6.80
14	Green Bay Packers	13	25.2	361.5	62.7	5.8	0.77	0.38	21.5	23.1	34.7	253.8	2.15	0.38	6.9	13.3	26.0	107.6	0.69	4.1	6.15	4.38	43.7	2.00	42.4	6.1	9.76
15	Tennessee Titans	13	24.9	347.2	67.2	5.2	1.62	0.62	21.8	21.4	32.5	209.4	1.15	1.00	5.9	11.7	31.8	137.8	1.46	4.3	7.92	6.08	56.6	2.23	38.2	13.9	4.89
16	Baltimore Ravens	13	23.4	388.0	70.8	5.5	1.54	0.46	23.8	23.4	35.9	243.2	1.31	1.08	6.1	12.6	31.3	144.8	1.08	4.6	9.46	6.38	56.9	1.77	39.0	13.0	3.70
17	New Orleans Saints	13	23.4	319.4	62.4	5.1	1.23	0.38	19.2	18.1	31.2	197.9	1.92	0.85	6.0	10.5	29.4	121.5	0.92	4.1	7.00	5.85	48.8	1.69	34.2	10.1	1.42
18	Las Vegas Raiders	13	21.8	372.5	62.7	5.9	1.31	0.54	20.4	26.1	38.2	287.8	1.38	0.77	7.1	13.0	22.2	84.6	0.85	3.8	5.08	8.00	71.1	2.31	39.0	11.6	4.38
19	Cleveland Browns	13	21.4	349.2	61.6	5.7	1.08	0.54	20.2	18.9	30.2	205.7	1.08	0.54	6.3	10.5	28.8	143.5	1.31	5.0	7.92	6.85	62.5	1.69	34.6	10.5	4.17
20	Denver Broncos	13	21.2	345.0	62.7	5.5	1.15	0.46	20.6	22.2	33.2	221.9	1.38	0.69	6.2	11.3	26.9	123.1	0.92	4.6	7.62	5.00	42.5	1.69	38.2	10.7	6.07
21	Seattle Seahawks	13	20.9	310.2	55.4	5.6	0.77	0.38	17.2	19.7	29.4	205.8	1.62	0.38	6.4	9.77	23.0	104.4	0.92	4.5	5.46	5.62	45.3	1.92	30.3	6.9	4.33
22	Pittsburgh Steelers	13	20.9	329.0	64.9	5.1	1.15	0.54	19.4	25.2	38.9	240.6	1.54	0.62	5.8	12.3	23.7	88.4	0.62	3.7	4.92	6.54	51.3	2.15	35.4	9.5	0.47
23	Washington Football Team	13	20.5	336.2	64.6	5.2	1.62	0.69	21.3	21.9	33.3	213.7	1.46	0.92	6.0	11.2	29.0	122.5	0.69	4.2	7.69	5.77	51.7	2.38	35.8	15.3	1.86
24	Carolina Panthers	13	19.8	310.6	64.5	4.8	1.77	0.46	19.5	20.2	34.4	200.9	0.85	1.31	5.5	10.6	27.6	109.7	1.15	4.0	7.08	7.23	58.6	1.77	31.8	15.2	-3.78
25	Miami Dolphins	13	19.5	309.8	64.8	4.8	1.54	0.77	19.0	25.8	38.5	230.5	1.31	0.77	5.6	12.9	23.9	79.2	0.69	3.3	4.46	6.38	48.2	1.62	30.3	12.0	-0.49
26	Atlanta Falcons	13	18.8	316.4	61.9	5.1	1.54	0.46	19.1	23.6	35.5	225.3	1.31	1.08	6.0	12.3	24.5	91.1	0.62	3.7	4.46	5.92	48.9	2.31	31.7	14.4	-0.50
27	New York Giants	13	17.8	312.2	61.7	5.1	1.31	0.46	18.9	22.3	36.2	217.9	1.00	0.85	5.7	11.3	23.3	94.2	0.62	4.0	5.38	5.38	42.5	2.23	32.2	11.9	-1.37
28	Chicago Bears	13	17.8	302.5	60.8	5.0	1.69	0.46	18.6	17.8	29.5	176.5	0.92	1.23	5.4	9.69	28.1	126.0	0.77	4.5	7.00	6.31	54.2	1.92	30.8	15.8	-1.85
29	New York Jets	13	17.4	321.5	62.8	5.1	1.92	0.38	19.2	23.4	38.8	237.4	1.31	1.54	5.7	12.5	21.3	84.2	0.69	3.9	4.69	6.69	54.3	2.00	29.0	17.2	-2.06
30	Detroit Lions	13	16.4	311.3	62.2	5.0	1.38	0.62	19.2	23.5	35.6	200.8	1.08	0.77	5.3	11.1	24.0	110.5	0.69	4.6	6.23	6.69	53.8	1.85	29.2	13.1	-2.30
31	Jacksonville Jaguars	13	13.8	303.1	60.1	5.0	1.92	0.85	17.4	21.0	36.0	200.4	0.69	1.08	5.3	10.1	22.2	102.7	0.85	4.6	5.38	6.85	56.3	1.92	21.7	17.4	-3.51
32	Houston Texans	13	13.6	264.3	59.3	4.5	1.54	0.54	14.9	20.8	32.6	186.8	1.00	1.00	5.3	9.46	23.8	77.5	0.46	3.3	4.00	6.62	54.7	1.46	23.1	12.9	-7.92

2021 NFL per-game Offensive Statistics – These data were used as a test for each model.

Three Regression techniques were performed: Ridge Regression, Logisitic Regression, and Support Vector Regression. Cross-Validation was performed for determine the best model for each type of regression. For Ridge Regression, alpha values of 0.01, 0.1, 1, 10, and 20 were attempted. The alpha value of 1 had the best R^2 score of 0.419. For Logistic Regression, cross-validation was attempted, but most of the parameter combinations used resulted in failures or warnings. The default configuration with a penalty of 'l2', a C of 1, and a solver of 'lbfgs' was the best of the warning-free models with an R^2 score of 0.281. For Support Vector Regression, a multitude of model configurations were attempted (see Python code for complete list). The SVR model with a C of 1, a gamma value of 'scale', and a kernel of 'rbf' had the best R^2 score of 0.417.

Comparing the scores of each of the three best models, the Ridge Regression was the best performing model for this application. The plot below shows the predicted and actual win rate for each NFL in the 2021 season as determined by the Ridge Regression.



Bar Charts comparing predicted win rate using the Ridge Regression model and the actual win rate for each NFL team in the 2020 and 2021 seasons.

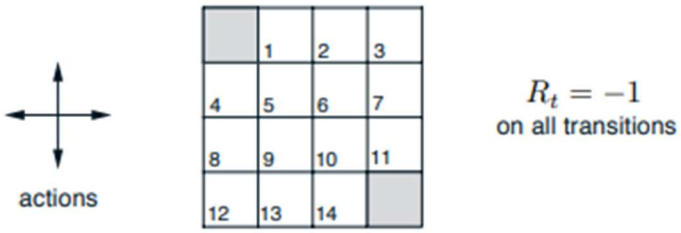
As expected, only using offensive statistics and one season of training data did not result in the most accurate of models for predicting win rate. Offense is, after all, only half of the game (less than half if you include Special Teams play). It is, however, interesting to see the predicted win rate was almost always lower than the actual win rate for the 2021 season. Seeing as the training was done on the 2020 data and that performance was much more evenly balanced in predicted and actual, this indicates that teams in 2021 are winning more with less offensive

production than in 2020. This could mean that better defense and special teams are being played this year, although a number of other factors could also be involved.

Data was accessed at https://www.pro-football-reference.com/ . Sklearn manuals were accessed online at https://scikit-learn.org/stable/index.html for this section.

Problem for DP (done by Julian)

In this task, we implemented a Policy Iteration loop for Example 4.1 in the RL textbook.



A screenshot from the RL textbook displaying the actions, states, and rewards.

As shown above this problem included two terminal states, 14 non-terminal states, 4 equiprobable actions (right, left, up and down), and a reward of –1 for all transitions. The resultant Final Value Function and Optimal Policy, after completing the Policy Iteration algorithm, are shown below.

```
Final Value Function:
[[ 0.   -1.   -1.9  -2.71]
 [-1.   -1.9  -2.71 -1.9 ]
 [-1.9  -2.71 -1.9  -1.  ]
 [-2.71 -1.9  -1.    0.  ]]
```

```
Optimal Policy (right, left, up, down):
[['O' '←' '←' '←']
 ['↑' '←' '→' '↓']
 ['↑' '→' '→' '↓']
 ['→' '→' '→' 'O']]
```

The Final Value Function and Optimal Policy after completion of Policy Iteration.

Both of these make sense in relation to the setup of this problem, indicating that the algorithm was successful. The policy indicates which way the optimal action is at each state. For states with more than one optimal action, the order of actions shown in the top right was used to determine which arrow to display. For example, the best action for state in position (1,1) would be to move either up or left, but because we chose left to come before up, left is displayed.

Task 2 Reinforcement Learning (done by Muchen)

In this task, we apply reinforcement learning techniques to a 2-D inverted pendulum. Figure 1 shows a schematic configuration. Specifically, we applied policy iteration, value iteration and q-learning. A complete description can be found in FinalProjectInstruction.pdf.

Policy Iteration and Value Iteration

This section describes the implementation of two dynamic programming algorithms: policy iteration and value iteration.

The boxes system for the state space is constructed by the following quantization thresholds:

$$x: \pm 0.8, \pm 2.4, \pm \inf\ m$$

$$x': \pm\ 0.5, \pm \inf\ \ m/s$$

$$\theta: 0, \pm 6, \pm 12, \pm \inf °$$

$$\theta': \pm\ 10, \pm 30, \pm 50, \pm inf\ °/s$$

Note that our boxes system is different from that in instruction. The angle velocity space is divided into more boxes because, as a result, the DP model learns faster. In other word, the policy converges with less epochs.

Physical model

For simplicity, let the cart mass be M, the pole (pendulum) mass be m with the half-pole length l. The model is written as the following nonlinear equations

$$\ddot{\theta} = \frac{g \sin\theta + \cos\theta \left[\frac{-F - ml\dot{\theta}^2 \sin\theta + \mu_c \mathrm{sgn}(\dot{x})}{M+m}\right] - \frac{\mu_p \dot{\theta}}{ml}}{l\left[\frac{4}{3} - \frac{m\cos^2\theta}{m+M}\right]}$$

$$\ddot{x} = \frac{F + ml\left[\dot{\theta}^2 \sin\theta - \ddot{\theta}\cos\theta\right] - \mu_c \mathrm{sgn}(\dot{x})}{M+m}$$

Figure 1 Non-linear differential equations to model the cartpole system

where sgn is the sign function [1]. In implementation, we add coefficients of friction μ_c and μ_p to the original equations in the CartPoleEnv class as the following [2]:

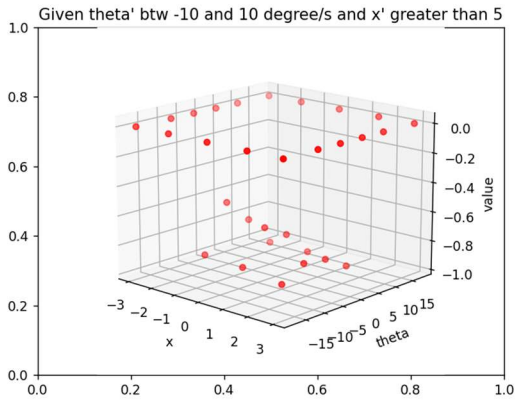
```
    temp = (
        force + self.polemass_length * theta_dot ** 2 * sintheta
    ) / self.total_mass
    temp2 = self.mu_p*theta_dot/self.polemass_length
    thetaacc = (self.gravity * sintheta - costheta * (temp - self.mu_c*sgn(x_dot)/self.total_mass) - temp2) / (
        self.length * (4.0 / 3.0 - self.masspole * costheta ** 2 / self.total_mass)
    )
    xacc = temp - self.polemass_length * thetaacc * costheta / self.total_mass - self.mu_c*sgn(x_dot) / self.total_mass
```

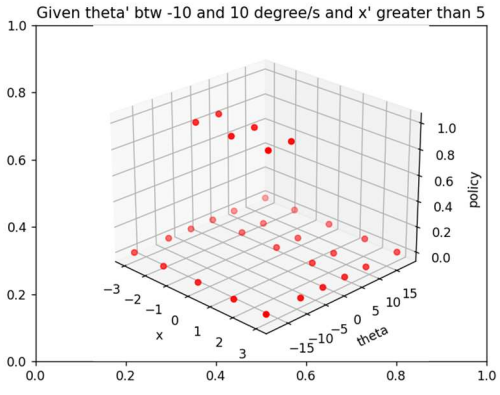
The rest of the implementation of the physical model are the same as that in the CartPoleEnv class.

Value Iteration Result

Since the value function is a function of four variables, we fix x' and θ' in a way described in the title of each plot, and then plot value function with respect to x and θ. Plots, policy and value function data are accessible in plot_task3_value_iteration.py.

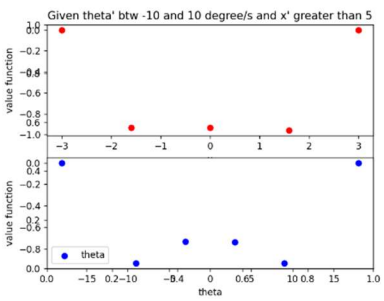
Below, the left figure is value function vs x and θ, and the right figure is policy vs x and θ, given $\theta' \in (-10, 10)°/s$ and $x' > 5\ m/s$. Noted that the dotted square at value = 0 corresponds to the 'terminal' states, i.e., states with pole angle more than 12 degrees or cart position is more than 2.4 meters. The value 0 is the initialized value of the value function at the start of the iteration. Since the only necessary action for the terminal states is 'exit', there is no need to update their value function. Also, in the right figure, except the terminal states, when $\theta < 0$, the policies are always going right and vice versa. The resulted policy from value iteration helps keep the pole straight and thus is correct.

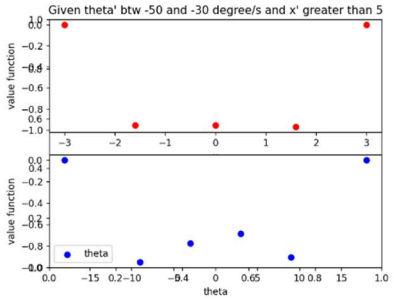


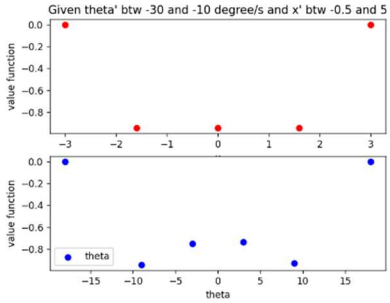


Policy Iteration Result

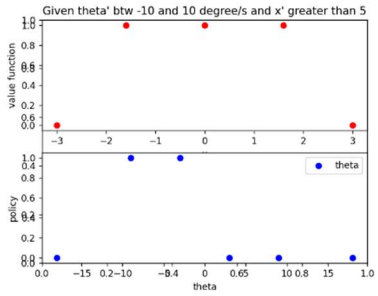
Since the value function is a function of four variables, we fix x' and θ' in a way described in the title of each plot, and then fix x to plot value function with respect to θ and vice versa. For example in the left figure, we first select the values corresponding to $\theta' \in (-10, 10)°/s$ and $x' > 5\ m/s$.

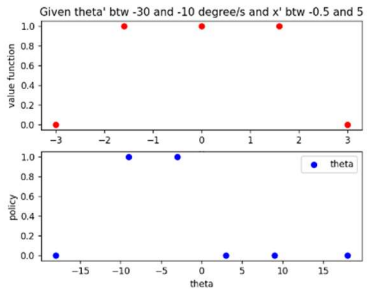


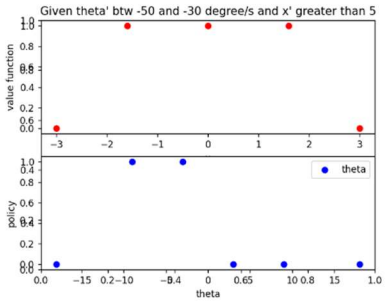




The following are the policy plots. Note that policy 1 is right direction and 0 left.







Q learning

The boxes system, learning rate and exploration rate ϵ

The boxes system for the state space is constructed differently than in the DP problem:

$$x: \pm 1.6, \pm \inf\ m$$

$$x': \pm 0.5, \pm \inf\ \ m/s$$

$$\theta: 0, \pm 8, \pm 16, \pm \inf °$$

$$\theta': \pm 10, \pm 30, \pm 50, \pm inf\ °/s$$

The learning rate and the exploration rate decreases logarithmically to 0.1 and 0.001 respectively [3]:

```
def select_learning_rate(x):
    return max(min_learning_rate, min(1.0, 1.0 - math.log10((x+1)/70)))


# change the exploration rate over time.
def select_explore_rate(x):
    return max(min_explore_rate, min(1.0, 1.0 - math.log10((x+1)/70)))
```

Result

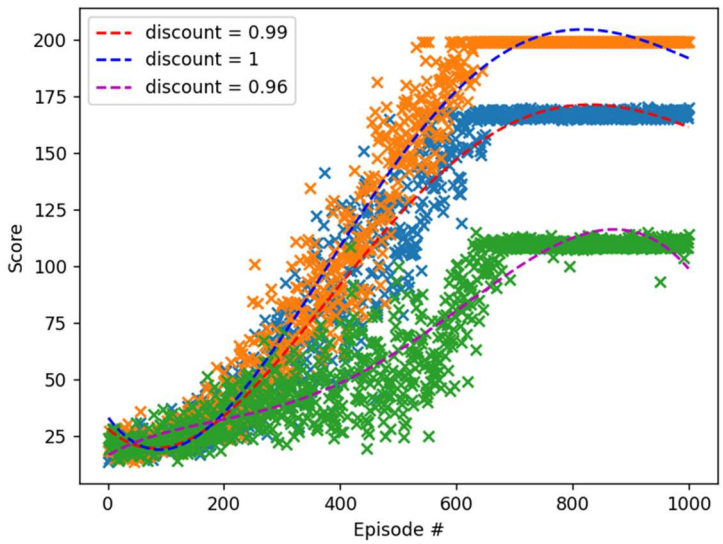


Figure 2 Plots of total reward received per episode (averaged over 10 trials) vs. number of episodes with three different discount rates.

Shown in Figure 2, as discount rate decreases, the average reward decreases. When discount = 1, 10 out of 10 trails succeed to solve the problem, whereas when discount = 0.96,

only 5 out of 10 trails solve the problem. Discount rates smaller than 0.96 hardly yield satisfactory results over 10 trails and therefore relative data are omitted here.

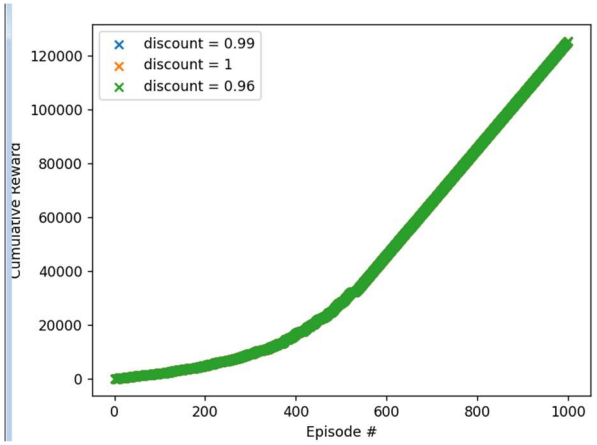


Figure 3 Plot of the cumulative reward (the sum of all rewards received so far) as a function of the number of episodes, with three different discount rates.

The overlapping results from seeding the random by 0 at the beginning of each trail. Seeding the pseudo-random number generators can help create reproductive result. Details are in task3Qlearning.py.

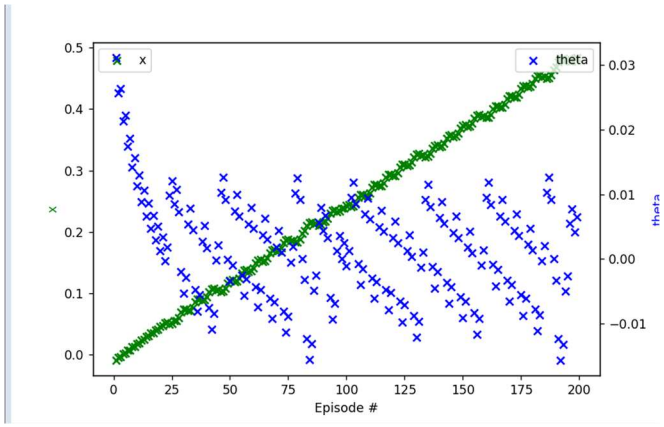


Figure 4 Plot of x and θ as functions of time for episodes in a successful trail together with number of time steps during which the pole does not fail

Figure 4 shows x and θ vs episodes in a successful trail. The cart is going towards one direction slowly but stay within the safe zone and the θ is within a tiny range from 0.

References

[1] A. G. Barto, R. S. Sutton, and C. W. Anderson, "Neuronlike adaptive elements that can solve difficult learning control problems. SMC-13(5)," *IEEE Transactions on Systems, Man, and Cybernetics,* p. 834–846, 1983.

[2] "Classic cart-pole system implemented by Rich Sutton et al.," [Online]. Available: https://github.com/openai/gym/blob/master/gym/envs/classic_control/cartpole.py.

[3] "Practical Work: Reinforcement Learning," [Online]. Available: https://scientific-python.readthedocs.io/ en/latest/notebooks_rst/6_Machine_Learning/ 04_Exercices/02_Practical_Work/ 02_RL_1_CartPole.html.

[4] Barto, Sutton, and Anderson, "CartPole v0," [Online]. Available: https://github.com/openai/gym/wiki/CartPole-v0.