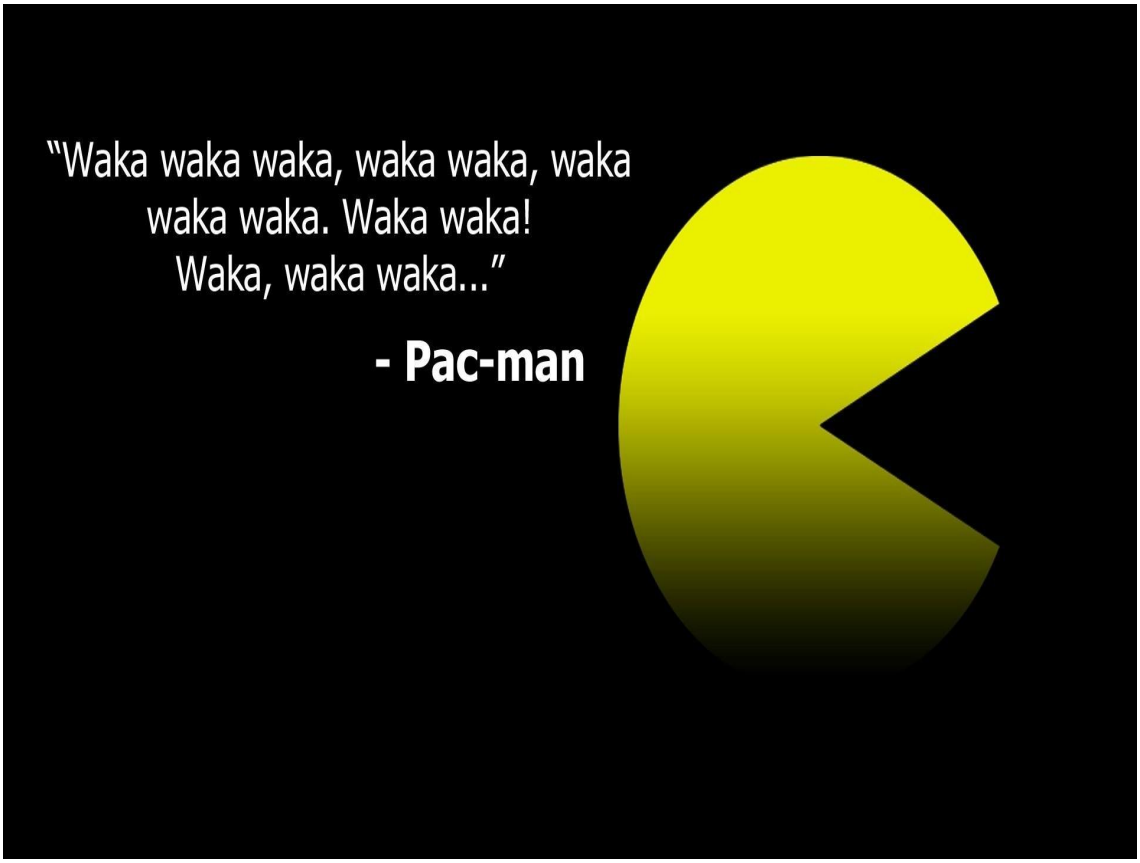
MACHINE LEARNING TUTORIAL I

CARLOS III UNIVERSITY

DATA SCIENCE AND ENGINEERING BACHELOR DEGREE



ALEJANDRO PÉREZ, 100429952
CHRISTOPHER MANZANO, 100429927

APRIL 9TH, 2021

Introduction

Our goal is to create a better agent than the one we created in the first tutorial. To achieve these goals, we will follow the recommendations and instructions given for this project.

This project is divided into three phases. The first one involves translating the environment into states, actions, and rewards. The second phase describes how to implement phase 1 ideas in the *QLearningAgent*. Finally, the third phase contains information related to the performance of our model. We were told that the agents we would train wouldn't be so good, so it's safe to say that this project involved quite a lot of testing to get decent results.

In order to follow the phases and include our insights into the most relevant models, we are going to include subchapters at each phase. Each subchapter at all the phases will refer to a model in specific. They will be of the form 'our n-th approach'.

If you want to read this report in the way we wrote it. You should jump at the end of each phase's subchapter to the next phase's subchapter.

Remark: We have implemented some stuff in many files. So we recommend you to execute the program with our files in order to avoid compatibility issues.
Remark: You can play with our plots at the excel file

Phase 1: Selection of reward and states

We are going to create a QState class at qState.py which will be a data structure for the states. Then we are going to give some methods our QState instances instead of the gameState instances. That way we consider it will be easier for us to implement changes in the future.

Our first approach

As our first approach, we do not want to generate too many states for our q-table since we consider that it will be better to add complexity in advance. We do not have any knowledge about how the agent will perform, thus if we select complex models we could be losing our time because the states may not represent the environment correctly.

Taking this into account, we have considered including the tutorial's one programmed agent. We will introduce the move it would execute in a given moment as a recommended direction. We also are going to include the number of ghosts that are remaining in the labyrinth. Thus we (0-5, where all the states which have 0 will be contained in one final state). Finally have 4x4 + 1 = 17 possible states.

In addition, we have excluded the pac-dots stuff in favor of simplicity. We will add them later in another approach.

Regarding the reward function, we have thought about giving a positive value if it gets closer to the ghost, giving a negative value if it gets far from the ghost, if its distance with the ghost does not change we give 0, and if we eat a ghost we give 5.

Go to phase 2

Our second approach

The first approach was not a waste of time, to be honest. However, it needed to improve.

In this approach, we are going to remove the number of remaining ghosts as an attribute of our q-state. We have realized that it does not make much difference. We have thought it would be a way of telling our agent when it eats a ghost. But this can be done (and was done) by the reward function.

We also are going to add a new attribute: recommended_dir2 (0:8) which will consider in which direction the agent should go to reach the zone with the most density of phantoms and ghosts (for instance 0:'Already-Here' 1:'North', 2:'North-West', etc).

As a result of this, we will have 4*9 =36 states. A q-table is considerably big. But we think it will be worth it.

About the reward function… We consider it is doing fine for now. So we just will add a reward for eating a pacdot.

Go to phase 2

Our third approach

For our third approach, we wanted to try using two programmed agents (from past tutorials) which will give us their recommendations on where Pacman should go. Besides, we need to take into account the pacdots… for this purpose, we will use a recommended zone without the diagonal directions.
Then we should have 4x4x5 states plus a final state which will be when all the ghosts get eaten. These states will be called be:
recommended_dir1: {'North', 'South', 'East', 'West'}
recommended_dir2:{'North', 'South', 'East', 'West'}
recommended_zone:{'North', 'South', 'East', 'West', 'Stop'}

We are a bit worried about the number of states since we were recommended less than 40. However, we think that the agent will perform well with these states.

Go to phase 2

Phase 2: Agent

Our first approach

First of all, we are going to create our *QState class*. For now, it will contain, as we discussed earlier, the recommended direction by the tutorial1 agent and the number of remaining ghosts. Which are defined in the constructor.

To define these attributes we created the *behavior1(self, gameState)* and the *countGhosts(self, gameState)* methods respectively.

In addition, we included *self.__legalActions, getPacmanLegalActions(self)* because the QLearningAgent needs it. Also, we included *__getId(self)* which will be the one in charge of assigning a number from 1 to 17 at each state according to its attributes. This id will be stored at *self.__id* and will be acceded by the getter *id(self)*.

Next, we are going to focus on our QLearningAgent.
In the computePosition method, we just made it return the q-state id - 1. At the constructor, we specified the size of the qtable(17). Finally, we copy-pasted our tutorial's 4 update method in this update method.

Respect with to the reward function, we have added as arguments our q-states. This method just contains some conditionals which return the before-mentioned rewards in the given case.

The behavior of our agent is okay with the first two new labyrinths. Also, it does good in open Hunt. It did not perform that well at first, so we started to play with alpha y epsilon. We have found out that an alpha of about 0.5 is good for our agent. Also, a small epsilon was needed. We also needed to increase the rewards by eating a ghost to 100 so they are significant enough.

Also, we have to remark that in the q-table a lot of states are still 0. We consider that the number of ghosts is not that relevant and adds noise to the model. We need to look for other attributes.

Go to phase 3

Our second approach

Regarding the QLearningAgent we have not made any important changes. We have implemented most of the changes in the QState class. Now It will have the attributes recommended_dir and recommended_zone.

Recommended dir stills being the move our tutorial 1 agent would do. In contrast, the recommended zone contains 9 possible outputs which are the directions {‘Stop’, ‘North’, ‘NorthWest’} Pacman should take to go to the zone which has the most populated zone.

For this, we divided all the maps into 4 zones. We will compute the zone in which Pacman is, and the most populated zone by selecting the zone which contains more units. The ghosts are worth 2 units, and the pacdots 1 unit. If Pacman is already in the most populated zone, it will return ‘Stop’. If Pacman needs to go in diagonal, it will return a composed direction i.e. ‘NorthWest’.

Regarding the __getId method. We modified it so that it could output rows from 0 to 35 since we had 36 states.

At the training phase, it did not seem to improve quite a lot. Even it became worse than our first try. We considered making some changes to the reward function so that it takes into account when the pac dots are eaten. It did not have any effect. The result was the same: Pacman ends up following the recommended direction in all the cases.

We were not able to remove this bias with this approach. Worse than it. It seems we just have increased the noise in our model since it takes longer to learn and ends up doing almost the same thing as our first model.

Go to phase 3

Our third approach

At the QLearningAgent we have added the method final which is called when the game ends. This method updates the qtable at the final state with the *score x 0.01* as a reward.

Regarding the QSate we had to do some extra work. First of all, we had a lot of problems trying to correctly implement the behavior2 (our second programmed Pacman) since it requires the use of a stack of target directions. If we implemented this method at the QState class we would have an empty stack of target directions at each time we want to get a recommendation since the QStates are created and destroyed at each iteration.

Thus, we implemented a new class called Advisor (at advisor.py) which contains the methods behavior1 and behavior2 (both of our programmed agents). This advisor is created in the constructor of the Class Game to avoid that it gets deleted at iterations. And it is called at every Pacman iteration.

Finally, at the game loop, we assign the attributes recommended_dir1 and recommended_dir2 to the gameState class, making it accessible by the QState constructor.

In addition, we have discovered that we could avoid updating the display at the game speeding up the game flow. Thanks to this we were able to train all our agents a lot. And we were able to gather all the statistics we got.

Go to phase 3

Phase 3: Evaluation

Our first approach

We consider that the agent performs relatively well in open maps such as open Hunt and labAA1, 2, 4, 5. But it gets stacked in more complex maps such as classic. We have seen that in classic Pacman gets stacked if the ghosts are static. When the ghosts move randomly it catches them.

State	Frequency	North	East	South	West
State 01: <recomended:North, ghosts:1>	4481	-21.07851475	-11.27557643	-19.80176286	-12.40559511
State 02: <recomended:North, ghosts:2>	3259	1.553142259	-13.16883582	-13.55708259	-12.52358659
State 03: <recomended:North, ghosts:3>	2254	-6.959284004	-10.40402928	-9.324537743	-1.535447344
State 04: <recomended:North, ghosts:4>	1511	7.451807403	30.62528921	35.95977528	13.02437368
State 05: <recomended:East, ghosts:1>	327	0.357499969	1.900485933	1.709838102	-2.945860827
State 06: <recomended:East, ghosts:2>	297	9.624875502	9.78922394	8.506023444	1.419341626
State 07: <recomended:East, ghosts:3>	181	11.04611671	42.60138284	17.87056841	3.160247662
State 08: <recomended:East, ghosts:4>	120	19.83769566	43.92494547	21.21197394	24.39569139
State 09: <recomended:South, ghosts:1>	176	0.84975125	1.944056658	12.89025325	0.88552036
State 10: <recomended:South, ghosts:2>	83	-0.20603361	4.654420647	23.52001003	1.121130075
State 11: <recomended:South, ghosts:3>	78	8.206511254	8.730882039	57.12904373	11.40310211
State 12: <recomended:South, ghosts:4>	93	22.59742517	16.55286181	69.87195794	27.7266924
State 13: <recomended:West, ghosts:1>	644	1.905593191	-0.577943484	2.098601882	3.145789746
State 14: <recomended:West, ghosts:2>	348	15.59148592	4.51907847	14.31210804	26.68318782
State 15: <recomended:West, ghosts:3>	265	20.42580173	8.665771359	12.2253014	22.81223161
State 16: <recomended:West, ghosts:4>	117	17.5551253	17.96297657	13.01805921	59.8916467
Statistics					
Standard deviation	1324.532817	11.4617858	17.23354768	23.6093508	18.44104153
Max	4481	22.59742517	43.92494547	69.87195794	59.8916467
Min	78	-21.07851475	-13.16883582	-19.80176286	-12.52358659
Total	14234	108.758999	156.4449899	247.6401315	166.2584653
Average	889.625	6.797437435	9.777811871	15.47750822	10.39115408

We can see that there is a clear inclination in favor of following the recommended direction. So we can say that the agent has learned a policy. We can see also that the most likely decision it will take is south, and the less likely is north.

We think that Pacman is just following the recommended direction. It seems we are biasing our agent, but we think it is a good first approach, we think we could keep the recommended direction by adding a different attribute to our states so it could unbias the model a bit.

[Go to the second approach]

Our second approach

We have tested our agent in the new maps and some of the old maps. It performs worse than our first model. It ended up getting stuck within many maps. It seems it does not learn a clear policy as we can see in the Q Table.

State	Frequency	North	East	South	West
State 00: <recomended_dir:North, recommended_zone_dir:North>	11394	-1.3474702	-9.3750523	-7.9498	-8.7894
State 01: <recomended_dir:North, recommended_zone_dir:South>	9626	-22.595366	-18.125992	-20.406	-24.308
State 02: <recomended_dir:North, recommended_zone_dir:East>	9539	0.0427613	-6.6731551	2.1325	-3.3479
State 03: <recomended_dir:North, recommended_zone_dir:West>	8525	-8.4585342	-9.2544411	-8.5613	-5.787
State 04: <recomended_dir:North, recommended_zone_dir:NorthEast>	156	2.8752918	-7.0086333	-10.506	-9.9499
State 05: <recomended_dir:North, recommended_zone_dir:NorthWest>	45526	-22.083745	-24.36261	-25.956	-26.253
State 06: <recomended_dir:North, recommended_zone_dir:SouthEast>	5088	-6.5074629	-9.2208652	-4.7828	-8.2301
State 07: <recomended_dir:North, recommended_zone_dir:SouthWest>	13467	-21.628661	-11.298596	-11.008	-21.832
State 08: <recomended_dir:North, recommended_zone_dir:Stop>	180757	1.9505613	-4.5175237	-7.3692	-2.1145
State 09: <recomended_dir:South, recommended_zone_dir:North>	712	-0.8863237	-0.4774432	-0.4778	-0.4944
State 10: <recomended_dir:South, recomended_zone_dir:South>	49	2.5983722	2.5923897	2.6146	2.6802
State 11: <recomended_dir:South, recomended_zone_dir:East>	112	-3.2926382	-2.1942417	5.614	-0.6949
State 12: <recomended_dir:South, recomended_zone_dir:West>	310	3.1562318	3.3130782	3.3723	3.0415
State 14: <recomended_dir:South, recomended_zone_dir:NorthWest>	237	-1.3531351	-2.655056	-0.7069	-0.553
State 15: <recomended_dir:South, recomended_zone_dir:SouthEast>	25	-1.5690496	-0.2937329	-0.2979	-0.4865
State 16: <recomended_dir:South, recomended_zone_dir:SouthWest>	120	-1.9815876	-5.474874	12.868	-1.8494
State 17: <recomended_dir:South, recomended_zone_dir:Stop>	123	2.6163734	11.26401	15.806	5.4761
State 18: <recomended_dir:East, recomended_zone_dir:North>	1602	6.7706193	7.2715372	7.3118	7.432
State 19: <recomended_dir:East, recomended_zone_dir:South>	49	-0.6273737	4.3648689	0.8659	-3.7705
State 20: <recomended_dir:East, recomended_zone_dir:East>	1584	2.1441325	2.8630083	2.8121	1.7169
State 21: <recomended_dir:East, recomended_zone_dir:West>	142	0.3930977	6.2287425	-3.2185	-1.3582
State 22: <recomended_dir:East, recomended_zone_dir:NorthEast>	678	3.2026025	3.4149243	3.3087	3.1617
State 23: <recomended_dir:East, recomended_zone_dir:NorthWest>	63	0.8495165	8.5357292	5.5393	-2.666
State 24: <recomended_dir:East, recomended_zone_dir:SouthEast>	48	4.0881915	6.330787	4.5478	3.1137
State 25: <recomended_dir:East, recomended_zone_dir:SouthWest>	107	8.3968941	26.918736	-5.875	-1.4395
State 26: <recomended_dir:East, recomended_zone_dir:Stop>	1701	4.7972779	4.8835192	4.8862	-0.4141
State 27: <recomended_dir:West, recomended_zone_dir:North>	1127	-0.7138179	-0.2432013	-2.3249	-3.0215
State 28: <recomended_dir:West, recomended_zone_dir:South>	124	2.2015626	-0.0003845	1.1115	3.3381
State 29: <recomended_dir:West, recomended_zone_dir:East>	41	-3.2220529	-3.1130156	-3.3417	-3.1848
State 30: <recomended_dir:West, recomended_zone_dir:West>	1108	4.4780661	-4.7659414	-0.9629	2.7869
State 31: <recomended_dir:West, recomended_zone_dir:NorthEast>	18	1.2674107	-6.6560009	2.2085	3.2764
State 32: <recomended_dir:West, recomended_zone_dir:NorthWest>	335	0.7147368	6.1972484	2.8576	6.2349
State 33: <recomended_dir:West, recomended_zone_dir:SouthEast>	8	5.8182707	-5.0880986	1.8645	8.8301
State 34: <recomended_dir:West, recomended_zone_dir:SouthWest>	213	-2.1189891	-0.7152455	-0.9568	6.7452
State 35: <recomended_dir:West, recomended_zone_dir:Stop>	2086	0.1375519	-2.2567816	0.2148	0.1748
Final State	Not records	2.0562778	-8.5164821	0.954	4.9149

Statistics	Frequency	North	East	South	West
Standard deviation	31077.922	7.3569577	8.916213	8.0453	8.1624
Max	180757	8.3968941	26.918736	15.806	8.8301
Min	8	-22.595366	-24.36261	-25.956	-26.253
Total	296800	-39.886685	-39.592307	-34.766	-72.536
Average	8480	-1.1396196	-1.1312088	-0.9933	-2.0725

These results show that the recommended_zone works fine. At the frequency table, we can see that the most frequent state is the one in which the agent is at the recommended zone.

This is reflected in the behavior of the agent which seems to go first to the recommended zone, and then to take into account the recommended direction. However, we think that the racommended_zone could have fewer states. We could remove the diagonal directions in favor of simplicity.

Go to our third approach

Our third approach

Another of the drawbacks of having a huge qtable is that it reduces interpretability considerably.

State	Frequency	North	East	South	West
State 00: <recommended_dir1:North, recommended_dir2:North, recommended_zone:North>	2078	5.636781113	0.7573672250	4.432723353	0.592453003
State 01: <recommended_dir1:North, recommended_dir2:North, recommended_zone:South>	44	-15.18831003	-25.91130350	-19.82650244	-11.24614144
State 02: <recommended_dir1:North, recommended_dir2:North, recommended_zone:East>	18	-23.80064196	-12.13173372	-30.91280178	-25.39277535
State 03: <recommended_dir1:North, recommended_dir2:North, recommended_zone:West>	114	-18.25309822	-28.92391595	-24.09380171	-15.75101934
State 04: <recommended_dir1:North, recommended_dir2:North, recommended_zone:Stop>	4347	14.67871239	0.475837857	1.5899469	-2.783754463
State 05: <recommended_dir1:North, recommended_dir2:South, recommended_zone:North>	2225	12.61316267	-0.7502396627	-5.073585361	1.948025801
State 06: <recommended_dir1:North, recommended_dir2:South, recommended_zone:South>	333	-8.537815211	-8.656236478	-13.38847832	-5.549234716
State 07: <recommended_dir1:North, recommended_dir2:South, recommended_zone:East>	82	-14.24079856	-9.439364265	-24.63502135	-22.27954704
State 08: <recommended_dir1:North, recommended_dir2:South, recommended_zone:West>	653	-11.91994318	-17.28970585	-17.43429847	-11.8962421
State 09: <recommended_dir1:North, recommended_dir2:South, recommended_zone:Stop>	12793	23.98536404	-1.398110183	-1.913018703	-3.176026932
State 10: <recommended_dir1:North, recommended_dir2:East, recommended_zone:North>	2022	11.22801427	0.962903517	7.39813691	3.956967328
State 11: <recommended_dir1:North, recommended_dir2:East, recommended_zone:South>	72	6.040339432	-11.55063381	-5.456879526	-9.389520416
State 12: <recommended_dir1:North, recommended_dir2:East, recommended_zone:East>	63	20.79483405	1.972257614	-0.923344128	2.39727247
State 13: <recommended_dir1:North, recommended_dir2:East, recommended_zone:West>	161	-0.209875504	-9.983493881	-11.98152103	4.476781568
State 14: <recommended_dir1:North, recommended_dir2:East, recommended_zone:Stop>	4475	11.81921419	0.470656316	3.427123899	-2.393844958
State 15: <recommended_dir1:North, recommended_dir2:West, recommended_zone:North>	3606	21.99487748	12.8000933	10.92133184	11.37684825
State 16: <recommended_dir1:North, recommended_dir2:West, recommended_zone:South>	139	-3.285711514	-17.08773681	-9.973925418	-10.48103239
State 17: <recommended_dir1:North, recommended_dir2:West, recommended_zone:East>	41	-8.109760511	4.053088413	-24.3884806	-10.72336846
State 18: <recommended_dir1:North, recommended_dir2:West, recommended_zone:West>	405	9.335887609	2.932934213	1.742093065	3.31246405
State 19: <recommended_dir1:North, recommended_dir2:West, recommended_zone:Stop>	9249	9.845765266	4.600596391	5.4900033024	4.19054441
State 20: <recommended_dir1:South, recommended_dir2:North, recommended_zone:North>	2360	-14.33455235	-18.97787191	-17.50335803	-13.40177237
State 21: <recommended_dir1:South, recommended_dir2:North, recommended_zone:South>	73	-9.685925908	-12.76101706	-10.69735711	-5.906785015
State 22: <recommended_dir1:South, recommended_dir2:North, recommended_zone:East>	21	-28.72093062	-4.779388301	-12.92209206	-24.91912818
State 23: <recommended_dir1:South, recommended_dir2:North, recommended_zone:West>	191	-26.8515275	-20.70959775	-22.52871901	-14.13544198
State 24: <recommended_dir1:South, recommended_dir2:North, recommended_zone:Stop>	6821	-8.734405619	-13.94209461	1.560139814	-12.22911007
State 25: <recommended_dir1:South, recommended_dir2:South, recommended_zone:North>	2298	-1.913709336	2.723393206	1.753082247	5.018919583
State 26: <recommended_dir1:South, recommended_dir2:South, recommended_zone:South>	309	-1.333635155	9.670605114	3.826242263	8.581426123
State 27: <recommended_dir1:South, recommended_dir2:South, recommended_zone:East>	52	-13.65304387	1.350575681	0.1123397911	1.793921751
State 28: <recommended_dir1:South, recommended_dir2:South, recommended_zone:West>	358	-23.32713806	-15.53930744	-20.92312179	-11.66878025
State 29: <recommended_dir1:South, recommended_dir2:South, recommended_zone:Stop>	7959	5.574577267	13.52724881	37.16176887	17.02382668
State 30: <recommended_dir1:South, recommended_dir2:East, recommended_zone:North>	2169	-18.2889641	-22.39673453	-17.83984781	-20.86581522
State 31: <recommended_dir1:South, recommended_dir2:East, recommended_zone:South>	85	-2.350226364	-0.9008596362	28.6791054	-4.872762239
State 32: <recommended_dir1:South, recommended_dir2:East, recommended_zone:East>	51	-0.3417089952	-6.746396096	11.66023243	-3.295624981
State 33: <recommended_dir1:South, recommended_dir2:East, recommended_zone:West>	241	-24.77308508	-10.35583786	-18.41637547	-18.65844845
State 34: <recommended_dir1:South, recommended_dir2:East, recommended_zone:Stop>	6897	-2.447401385	-5.801734516	19.76479689	-8.693329456
State 35: <recommended_dir1:South, recommended_dir2:West, recommended_zone:North>	505	-3.304987445	4.961507258	3.89292103	3.572629355
State 36: <recommended_dir1:South, recommended_dir2:West, recommended_zone:South>	132	1.120222639	0.6022631	12.22124839	3.713669216
State 37: <recommended_dir1:South, recommended_dir2:West, recommended_zone:East>	67	-28.6303232	0.563769303	6.395965792	-12.55801014
State 38: <recommended_dir1:South, recommended_dir2:West, recommended_zone:West>	564	0.200494026	1.631721951	8.828347054	7.507043294
State 39: <recommended_dir1:South, recommended_dir2:West, recommended_zone:Stop>	22416	-0.3533339023	-4.002371702	7.278416316	-0.550966079
State 40: <recommended_dir1:East, recommended_dir2:North, recommended_zone:North>	413	-10.35219847	-19.99076875	-31.56897	-37.20442172
State 41: <recommended_dir1:East, recommended_dir2:North, recommended_zone:South>	46	-32.47994709	-35.78511912	-29.69037894	-66.94202039
State 42: <recommended_dir1:East, recommended_dir2:North, recommended_zone:East>	18	-9.426128412	2.693097808	-14.02322093	-5.877052758
State 43: <recommended_dir1:East, recommended_dir2:North, recommended_zone:West>	226	-28.11714279	-7.257375997	-31.11909457	-18.42210324
State 44: <recommended_dir1:East, recommended_dir2:North, recommended_zone:Stop>	9510	-15.64102032	-9.39329943	-15.47355392	-14.20529578
State 45: <recommended_dir1:East, recommended_dir2:South, recommended_zone:North>	5	26.47769458	16.92216877	19.48437186	23.96841104
State 46: <recommended_dir1:East, recommended_dir2:South, recommended_zone:South>	7	36.69513702	42.69644935	27.89627951	17.69639336
State 47: <recommended_dir1:East, recommended_dir2:South, recommended_zone:East>	4	7.641398569	20.24894143	6.836103086	3.107770842
State 48: <recommended_dir1:East, recommended_dir2:South, recommended_zone:West>	5	33.03726744	41.16526797	28.53422033	3.907171236
State 49: <recommended_dir1:East, recommended_dir2:South, recommended_zone:Stop>	15	13.64239046	22.75440781	15.74839002	13.93659623
State 50: <recommended_dir1:East, recommended_dir2:West, recommended_zone:North>	445	-2.229855308	31.95734055	-7.835749583	-0.970939083
State 51: <recommended_dir1:East, recommended_dir2:West, recommended_zone:South>	134	8.001402475	35.95194285	13.94562787	8.340477399
State 52: <recommended_dir1:East, recommended_dir2:West, recommended_zone:East>	87	9.336983258	26.57837902	0.463312601	11.23525791
State 53: <recommended_dir1:East, recommended_dir2:West, recommended_zone:West>	209	3.372847645	20.71554339	-10.85872034	2.926990026
State 54: <recommended_dir1:East, recommended_dir2:West, recommended_zone:Stop>	9812	1.400185528	37.58303051	5.861209435	14.65437198
State 55: <recommended_dir1:East, recommended_dir2:West, recommended_zone:North>	875	-7.010425837	-16.04638771	2.184938555	-16.37938772
State 56: <recommended_dir1:East, recommended_dir2:West, recommended_zone:South>	267	-24.26156422	-18.19689916	-17.1239555	-27.17032128
State 57: <recommended_dir1:East, recommended_dir2:West, recommended_zone:East>	119	1.121223049	7.684432675	7.5181611152	-3.083379082
State 58: <recommended_dir1:East, recommended_dir2:West, recommended_zone:West>	708	-5.070193291	-2.274815499	-5.787677749	-5.555289913
State 59: <recommended_dir1:East, recommended_dir2:West, recommended_zone:Stop>	14644	-9.213518402	-2.297116776	-4.206593172	-4.938754674
State 60: <recommended_dir1:West, recommended_dir2:North, recommended_zone:North>	88	-4.903467141	-38.58151746	-23.02414562	-31.30395247
State 61: <recommended_dir1:West, recommended_dir2:North, recommended_zone:South>	57	-20.23987977	-37.38242572	-13.16259583	-31.4498574
State 62: <recommended_dir1:West, recommended_dir2:North, recommended_zone:East>	28	-29.83188018	-16.72150747	-27.86305453	-19.84384892
State 63: <recommended_dir1:West, recommended_dir2:North, recommended_zone:West>	196	-16.97633637	-13.1004376	-18.93897597	8.129492662
State 64: <recommended_dir1:West, recommended_dir2:North, recommended_zone:Stop>	157	-8.258878273	-1.409497269	-17.88401949	9.515936223
State 65: <recommended_dir1:West, recommended_dir2:South, recommended_zone:North>	5	4.885563032	9.1974953325	-1.011620167	21.41122656
State 66: <recommended_dir1:West, recommended_dir2:South, recommended_zone:South>	4	-8.079091413	-3.969302252	-13.56634272	-0.275874743
State 67: <recommended_dir1:West, recommended_dir2:South, recommended_zone:East>	7	3.705524779	-11.73022764	-25.04044665	-10.47350494
State 68: <recommended_dir1:West, recommended_dir2:South, recommended_zone:West>	18	6.417205604	-2.483562933	2.62315422	11.7348656
State 69: <recommended_dir1:West, recommended_dir2:South, recommended_zone:Stop>	3	13.69279979	17.79508613	10.64632446	23.61348717
State 70: <recommended_dir1:West, recommended_dir2:East, recommended_zone:North>	24	15.15345575	7.811959596	9.921573821	20.37173593
State 71: <recommended_dir1:West, recommended_dir2:East, recommended_zone:South>	20	12.03246701	3.7937577562	12.93222064	24.6557017
State 72: <recommended_dir1:West, recommended_dir2:East, recommended_zone:East>	7	2.117101669	-3.333975724	-0.274727706	8.023944293
State 73: <recommended_dir1:West, recommended_dir2:East, recommended_zone:West>	61	14.70295067	7.848543099	11.19126738	17.35247407
State 74: <recommended_dir1:West, recommended_dir2:East, recommended_zone:Stop>	82	11.22332964	9.1641001193	14.75572098	16.49555523
State 75: <recommended_dir1:West, recommended_dir2:West, recommended_zone:North>	5677	9.342291913	-15.57505377	-5.993864553	9.810021003
State 76: <recommended_dir1:West, recommended_dir2:West, recommended_zone:South>	291	-18.77503853	-21.03573843	3.208579178	-10.84214819
State 77: <recommended_dir1:West, recommended_dir2:West, recommended_zone:East>	93	-7.70199301	-7.695683306	-6.396476231	10.97355007
State 78: <recommended_dir1:West, recommended_dir2:West, recommended_zone:West>	953	9.021142726	2.911998529	9.45542536	18.73342937
State 79: <recommended_dir1:West, recommended_dir2:West, recommended_zone:Stop>	21104	7.692506442	8.529766441	13.74276096	37.80113607

We still can see that recommended_zone is a very good attribute… The most frequent states are the ones that have stopped as recommended_zone. And once again we see that the agent tends to go to the recommended zone before taking into account the recommended directions. Even though we can see a possible policy… We do not think we ended up doing any better than in our second approach.
The agent gives more importance to the recommended_dir1 than the recommended_dir2 in general. This fact makes us think that recommended_dir2 is adding noise.

Investigating a little deep in this result we concluded that even though recommended_dir2 comes from our best-programmed agent it is not a good idea to implement it in the reinforcement learning approach. This agent keeps staking target positions. The agent will recommend the direction to the last target position.

The problem is that in this model, the QLearningAgent will not necessarily go in the target direction. Therefore, the recommended_dir2 will always point to a target position that nothing has to do with the ghosts or the dots.

We had tried to fix this problem… but the results did not get better.

From the Qtable we get that the most valued action was going east and the less valued is south. (Which makes a lot of sense since in most of the maps Pacman is at the inferior right corner).

Also, we can see that there are states that only got explored 3 times in all the processes. This is a clear indication that we are adding unnecessary complexity to our model.

Statistics	Frequency	North	East	South	West
Standard deviation	4414.81145	15.30804262	16.66235161	15.57736902	16.43264146
Max	22416	38.08726744	42.69644805	37.16176987	37.80113609
Min	3	-32.47994709	-38.58151746	-31.566976	-66.94920208
Total	162696	-163.5776081	-115.0891058	-236.6495214	-165.5033424
Average	2033.7	-2.044720101	-1.438613822	-2.958119018	-2.06879178

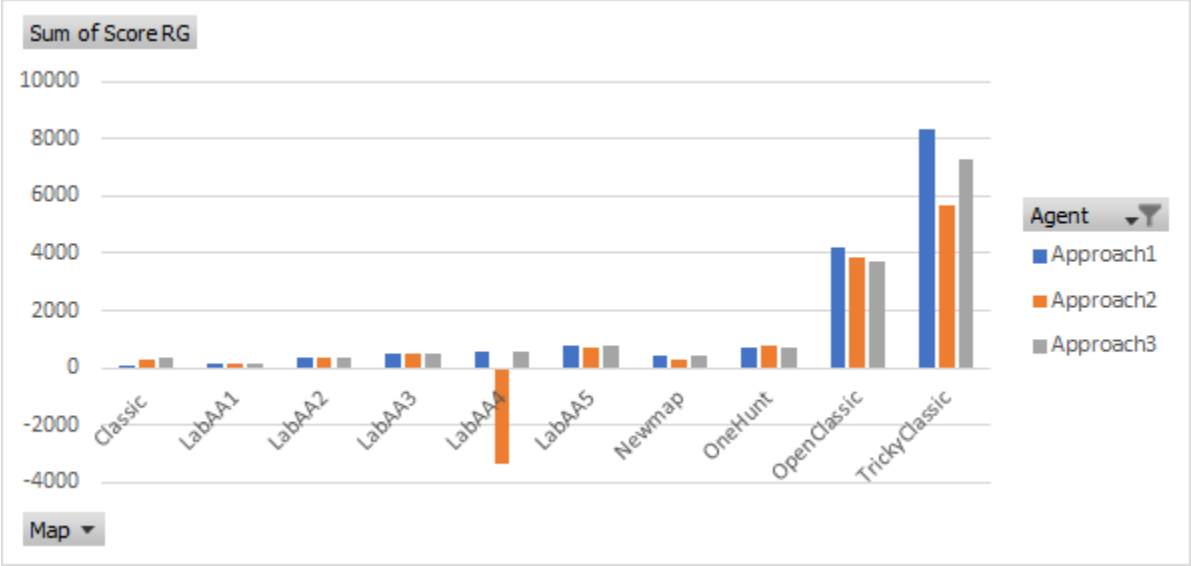
Our conclusions

Finally, we are going to compare all the average scores (20 samples) gotten by the three models we have discussed in 10 different maps. We also are going to compare them with two control models. The behavior1-BasicAgentAA (Our first programmed agent) and the RandomPAgent as control metrics.

Our best model is gotten from approach1. So we are going to compare it against the control models. And the other models. If you want to learn more about these results. You can go to the excel file and keep filtering the plots so you can see the information you need. (We have made more diagnostics than these… but we could not put them all together)

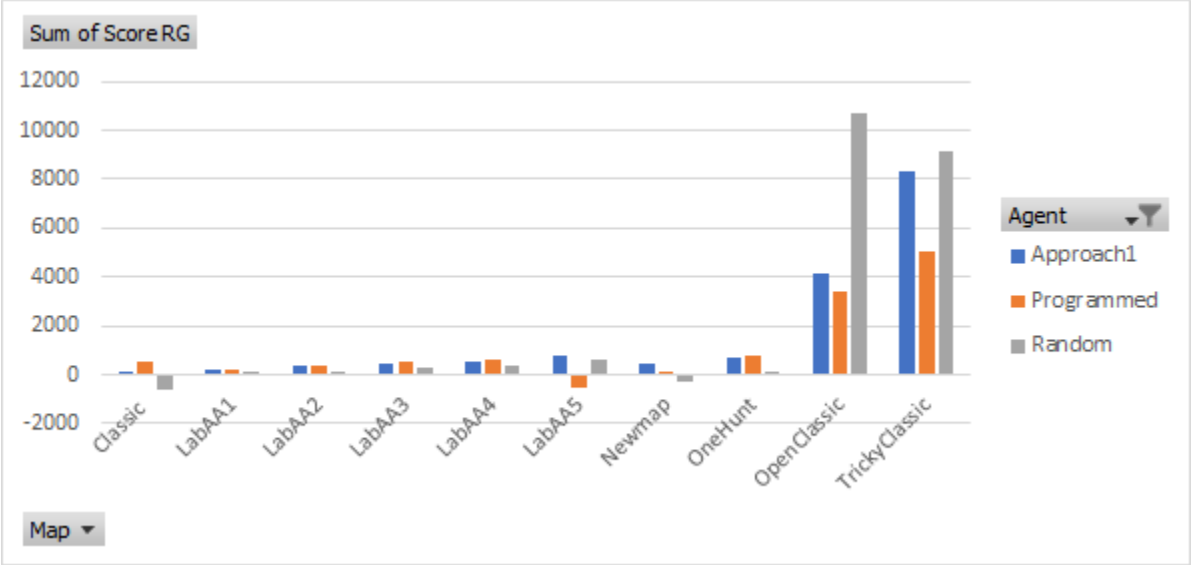
We expect that we could do better than the random model, and we hope we get better results than the behavior1 model.

With Ghost moving randomly:
Against other approaches



We see that Approach one performs better in the layouts with more pacdots, and similarly in the layouts with few pacdots.

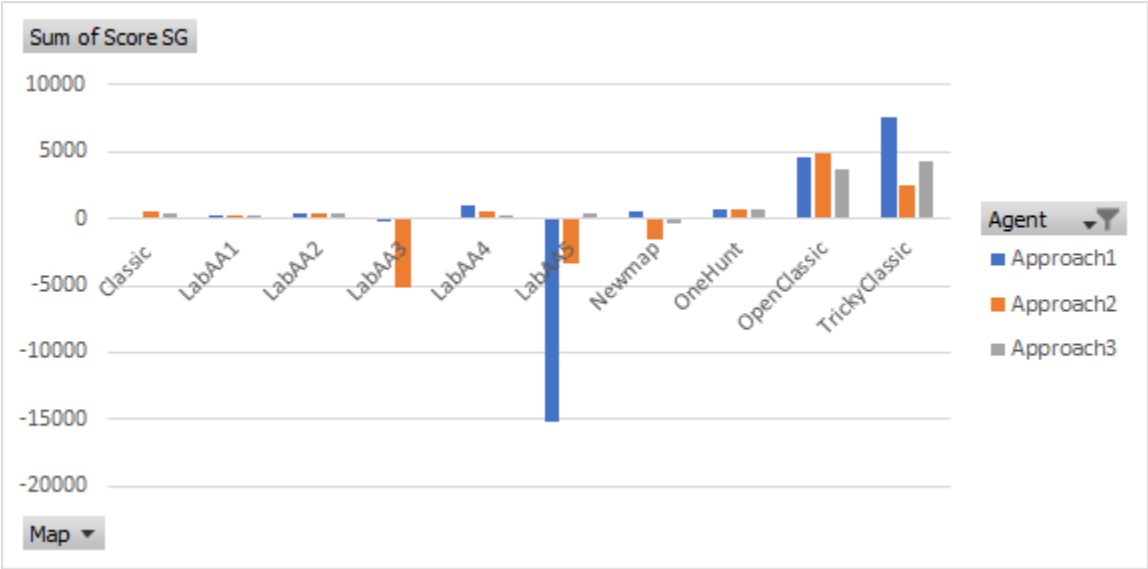
Against the control models



Here We can see that the Approach1 model does better than the programmed model by a lot in most of the layouts. Also it is curious that the Random Agent does better than approach 1 agent at the layouts with more pacdots.

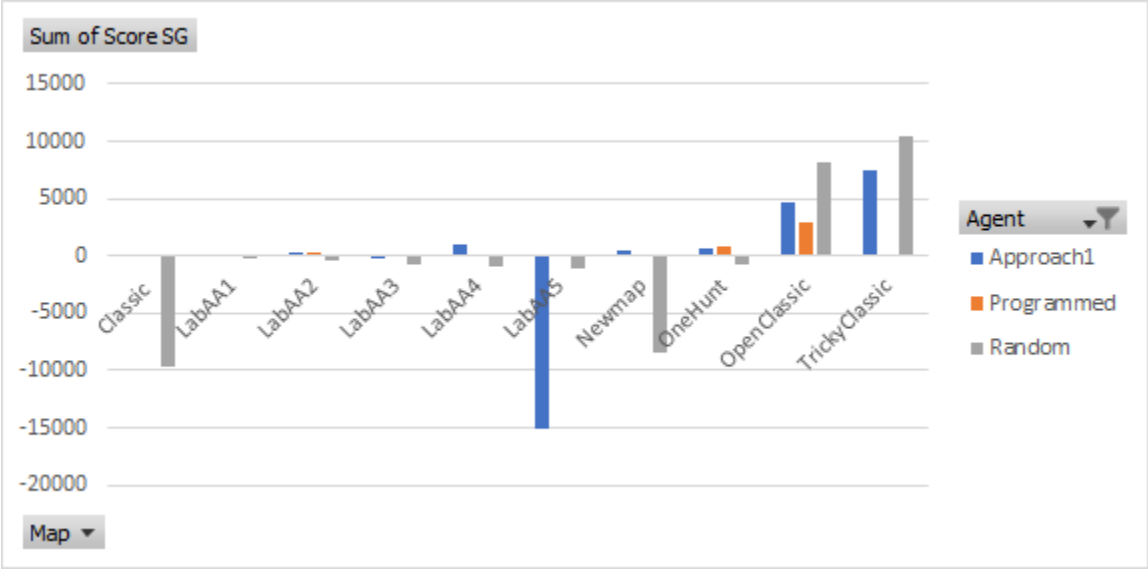
With Ghost static:

Against other models



Approach1 does relatively well in most of the layout but in labaa5. On the others, it performs similarly than the other models.

Against control models



We can see that there are layouts that the programmed agent could not even finish. Fact, that makes our approach1 model better. Even better than our weka model. The random agent only performs well if there are a lot of pacdots. In other cases it is completely defeated by the approach1 agent.

It is a bit ironic that our best model is the first one we tried. We may have beginner's luck. We were not really sure if the agent improved the performance until we saw these statistics. Approach1 model is clearly better than our programmed agent since it can solve all the maps behavior 1 could not (We remark that bigHunt was not included, that was another difficult map. We do not know whether our agent would perform better on it).

This is a confusing ending, because we improved our programmed agent performance… but we are far away from getting the ideal model (Random Pacman performs better in many cases).

Conclusions

After all our comparisons, we made tests with every approach to figure out which one performed better. To our surprise, our initial approach turned out to be the best one. Approaches 2 and 3 were not bad but far worse than the first one, but had more abstract attributes, like the recommended zone. In a way, our first approach was guilty of overfitting, since it followed our recommended direction for the most part.

General insights

Throughout the project, we checked how reinforcement learning is not as straightforward as other alternatives, like classification. It is quite clear that the selected attributes are more abstract. Nevertheless, we achieved our goal which was to improve the performance of our first behavior. It performs better in quite a few maps.

Problems encountered

Our biggest challenge was finding the proper attributes to use in our qtable and defining the best reward function. Sometimes a slight change in the reward function or the attributes made the entire behavior much worse. Furthermore, finding the optimal values for the alpha and epsilon was also quite tricky. After some trial and error, we found the optimal values for these variables, which gave pretty decent results, so that Pac-Man would explore and learn as expected.

Personal comments

All in all, we must say that these projects tested on the classic Pacman game have been of great help, and have been very fun to play around with. With this second practice, we've learned how to implement qlearning to make Pacman teach itself new and even creative ways to win the game. It was a bummer that we couldn't get a cool-looking result, but it was educational nonetheless. For further years, we would strongly suggest that git is used as a way to manage the different assignments and keep track of the changes. We used it and must say it was a life saver :)