RoboStats Homework 4
Luka eerens

Q2.1.1
Answer:

Let us begin by denoting the trajectory likelihood as $P(\tau;\pi)$, where $P(\tau;\pi)_1$ denotes the first trajectory likelihood.

We know that:

$$P(\tau;\pi) = P(\tau;\pi)_H$$

Where H is the planning horizon, and we know that:

$$P(\tau;\pi)_H = P(\tau;\pi)_1 * P(\tau;\pi)_2 * \dots * P(\tau;\pi)_H \quad (1)$$

We therefore need to find what these values ($P(\tau;\pi)_1$ and $P(\tau;\pi)_2$ …) are in order to notice a relationship which can be expressed as the variables requested. Now $P(\tau;\pi)_1$ and $P(\tau;\pi)_2$ can both be expressed as:

$$P(\tau;\pi)_1 = p(s_1) * p(a_1|s_1) * P(s_2|s_1,a_1)$$

$$P(\tau;\pi)_2 = P(\tau;\pi)_1 * p(a_2|s_2) * P(s_3|s_2,a_2)$$

Now expressing them in terms of policy π we get:

$$P(\tau;\pi)_1 = p\pi(a_1|s_1) * P(s_2|s_1,a_1)$$

$$P(\tau;\pi)_2 = p\pi(a_1|s_1) * P(s_2|s_1,a_1) * p\pi(a_2|s_2) * P(s_3|s_2,a_2)$$

Now going back to (1) we find that:

$P(\tau;\pi) = p\pi(a_1|s_1) * P(s_2|s_1,a_1) * p\pi(a_2|s_2) * P(s_3|s_2,a_2) *$ … until H

$$\therefore P(\tau;\pi) = p\prod_{i=1}^{H}\pi(a_i|s_i)P(s_{i+1}|s_i,a_i)$$

Q2.1.2
Answer:

Starting with the equation from the previous question:

$$P(\tau;\pi) = p\prod_{i=1}^{H}\pi(a_i|s_i)P(s_{i+1}|s_i,a_i)$$

The red sub-expression within this expression above is the only part that is differentiable wrt θ. This differentiating wrt to theta gives:

$$\nabla_\theta P(\tau;\theta) = \nabla_\theta \ln\big(P(\tau;\theta)\big)\,P(\tau;\theta)$$

Here we can find $\nabla_\theta J(\theta)$:

$$\nabla_\theta J(\theta) = \sum_\tau \nabla_\theta P(\tau;\theta)R(\tau)$$

Which is

$$\nabla_\theta J(\theta) = \sum_\tau \nabla_\theta \ln\big(P(\tau;\theta)\big)\,P(\tau;\theta)\,R(\tau)$$

And can be re-expressed as:

$$\nabla_\theta J(\theta) = E_{\tau\sim P(\tau;\theta)}\nabla_\theta \ln\big(P(\tau;\theta)\big)\;R(\tau)$$

Now from earlier:

$$\nabla_\theta P(\tau;\theta) = \prod_{i=1}^{H}\nabla_\theta\pi(a_i|s_i\;;\theta)$$

$$\therefore \nabla_\theta J(\theta) = E_{\tau\sim P(\tau;\theta)}\left[\sum_{i=1}^{H}\nabla_\theta \ln\left(\pi(a_i|s_i\;;\theta)R(\tau)\right)\right]$$

Q2.1.3
Answer:

Start with

$$\widehat{\nabla_\theta J(\theta)} = \frac{1}{K}\sum_{i=1}^{K}\left[\sum_{t=1}^{H}\nabla_\theta \ln\left(\pi(a_t^i|s_t^i;\theta)\right)R(\tau_i)\right]$$

Plug into the equation we need to prove:

$$Var\left(\widehat{\nabla_\theta J(\theta)}\right) = Var\left(\frac{1}{K}\sum_{i=1}^{K}\left[\sum_{t=1}^{H}\nabla_\theta \ln\left(\pi\left(a_t^i|s_t^i;\theta\right)\right)R(\tau_i)\right]\right)$$

Now:

$$Var\left(\widehat{\nabla_\theta J(\theta)}\right) = Var\left(\frac{1}{K}\sum_{i=1}^{K}\left[\sum_{t=1}^{H}\nabla_\theta \ln\left(\pi\left(a_t^i|s_t^i;\theta\right)\right)\sum_{i=1}^{H}c\right]\right)$$

Now given that variance grows in a quadratic fashion with changes in input:

$$Var\left(\widehat{\nabla_\theta J(\theta)}\right) = c^2 Var\left(\frac{1}{K}\sum_{i=1}^{K}\left[\sum_{t=1}^{H}\nabla_\theta \ln\left(\pi\left(a_t^i|s_t^i;\theta\right)\right)\right]\right)$$

$$\therefore Var\left(\widehat{\nabla_\theta J(\theta)}\right) = O(c^2)$$

Q2.1.5
Answer:

From 2.1.2 we have:

$$\nabla_\theta J(\theta) = E_{\tau \sim P(\tau;\theta)} \left[\sum_{i=1}^{H} \nabla_\theta (\mathrm{V}(\mathrm{s_i})) \ln (\pi(a_i|s_i \ ;\theta) \right]$$

We can put the sum notation outside, to nest the whole expression within the sum expression to give:

$$\nabla_\theta J(\theta) = \left[\sum_{i=1}^{H} E_{s_i \sim P_i(s_i;\theta)} [E_{a_i \sim \pi(a_i \mid s_i;\theta)} \nabla_\theta \big(\mathrm{V}(\mathrm{s_t})\big) \ln(\pi(a_i|s_i \ ;\theta)] \right]$$

Now the next sub-expression $E_{a_i \sim \pi(a_i \mid s_i;\theta)} \nabla_\theta (\mathrm{V}(\mathrm{s_i})) \ln (\pi(a_i|s_i \ ;\theta)$ needs to be proven to equal to 0.

So the following expression:

$$E_{a_i \sim \pi(a_i \mid s_i;\theta)} \nabla_\theta (\mathrm{V}(\mathrm{s_i})) \ln (\pi(a_i|s_i \ ;\theta)$$

Can be expressed as:

$$\sum_{i=1}^{H} \pi(a_i|s_i \ ;\theta) \nabla_\theta (\mathrm{V}(\mathrm{s_i})) \ln (\pi(a_i|s_i \ ;\theta)$$

Which a bit of re-arranging of terms is equal to:

$$(\mathrm{V}(\mathrm{s_i})) \nabla_\theta \sum_{i=1}^{H} \pi(a_i|s_i \ ;\theta)$$

And since $\sum_{i=1}^{H} \pi(a_i|s_i \ ;\theta) = 1$

We are looking at the gradient of a constant, which should be 0, therefore the whole expression is equal to 0, which completes the proof.

Q2.1.6
Answer:

Q2.2.1
Answer:

We start with the definition of the value function:

$$V^{\pi}(s) = E\left[\sum_{t=1}^{\infty} \gamma^{t-1} r(s_t, a_t) | s_1 = s, a_i \sim \pi(a|s_i) \forall i \right]$$

Isolate the reward from being in the state 1 with action 1, and multiply the summation with the initial discount factor from being in state 1:

$$V^{\pi}(s) = E\left[r(s_t, a_t) + \gamma \sum_{t=1}^{\infty} \gamma^{t-2} r(s_t, a_t) | s_1 = s, a_i \sim \pi(a|s_i) \forall i \right]$$

In the context of reinforcement learning, the apostrophe ' is used to denote the variable at the previous timestep. Also the expected trajectories are the weighted mean of actions coming from a distribution of policies:

$$V^{\pi}(s) = E_{a \sim \pi(.|s)}\left[r(s_t, a_t) + \gamma \sum_{t=1}^{\infty} \gamma^{t'-1} \sum_{s'}^{S} r(s'_t, a'_t)\, P(s'|s, a) \right]$$

Now:

$$\sum_{t=1}^{\infty} \gamma^{t'-1} \sum_{s'}^{S} r(s'_t, a'_t)\, P(s'|s, a) \equiv E_{S' \sim P(.|s,a)}[V^{\pi}(s')]$$

As you have the weighted sum of rewards given state and action all discounted by some discount factor. Therefore:

$$V^{\pi}(s) = E_{a \sim \pi(.|s)}\left[r(s_t, a_t) + \gamma E_{S' \sim P(.|s,a)}[V^{\pi}(s')]\right]$$

Q2.2.2
Answer:

Let's start off by more simply defining ugly expressions:

$$X = E_{a\sim\pi(.|s)}\left[r(s,a) + \gamma E_{s'\sim P(.|s,a)} f(s';\theta_n)\right]$$

$$Y = E_{a\sim\pi(.|s)}\left[r(s_1,a_1) + \gamma E_{s'\sim P(.|s,a)}[V^{\pi}(s')]\right]$$

We know that:

$$|f(s;\theta^*) - V^{\pi}(s)| = |f(s;\theta^*) - X| + |X - Y|$$

If in terms of Bellman Equation we can assume that it is smaller than a small positive number such as c, then error bound should be:

$$|f(s;\theta^*) - V^{\pi}(s)| = c + X - Y$$

Now doing this again but this time in the future, where we swap old states with new upcoming ones we get:

$$|f(s;\theta^*) - V^{\pi}(s)| = c + \frac{\gamma}{1-\gamma}[\ E_{a\sim\pi(.|s)}\left[r(s_t,a_t) + \gamma E_{s\sim P(.|s,a)} f(s;\theta_n)\right]$$

$$-E_{a\sim\pi(.|s)}\left[r(s_t,a_t) + \gamma E_{s'\sim P(.|s,a)}[V^{\pi}(s)]\right]\ ]$$

Solving algebraically we get:

$$|f(s;\theta^*) - V^{\pi}(s)| \leq c + \frac{\gamma}{1-\gamma}[c]$$

$$\therefore |f(s;\theta^*) - V^{\pi}(s)| \leq \frac{c}{1-\gamma}$$

Q2.2.3
a)
Answer:

Starting with the horrifying loss function expression from the documents:

$$l_n(\theta) = E_{s\sim v}\left[\left(f(s;\theta) - E_{a\sim\pi(.|s)}\left[r(s,a) + \gamma E_{s'\sim P(.|s,a)} f(s';\theta_n)\right]\right)^2\right]$$

Let's denote the waffle in blue as X for simplicity sake as the only term that is differentiable with a gradient in that loss expression is $f(s;\theta)$.

The gradient of the simplified equation now becomes:

$$\nabla_\theta l_n(\theta) = E_{s\sim v}[\nabla_\theta (f(s;\theta) - X)^2]$$

$$\nabla_\theta l_n(\theta) = E_{s\sim v}[2(f(s;\theta) - X)^1 \nabla_\theta f(s;\theta)]$$

$$= E_{s\sim v}\left[2\left(f(s;\theta) - \left(E_{a\sim\pi(.|s)}\left[r(s,a) + \gamma E_{s'\sim P(.|s,a)} f(s';\theta_n)\right]\right)\right)^1 \nabla_\theta f(s;\theta)\right]$$

b)
Answer:

From the previous question, if we run $\nabla_\theta l_n(\theta) = E_{s\sim v}[2(f(s;\theta) - X)^1 \nabla_\theta f(s;\theta)]$ for T timesteps we get:

$$\nabla_\theta l_n(\theta) = \frac{1}{T}\sum_{t=1}^{T}[2(f(s;\theta) - X)^1 \nabla_\theta f(s;\theta)]$$

In the case of the Bellman equation:

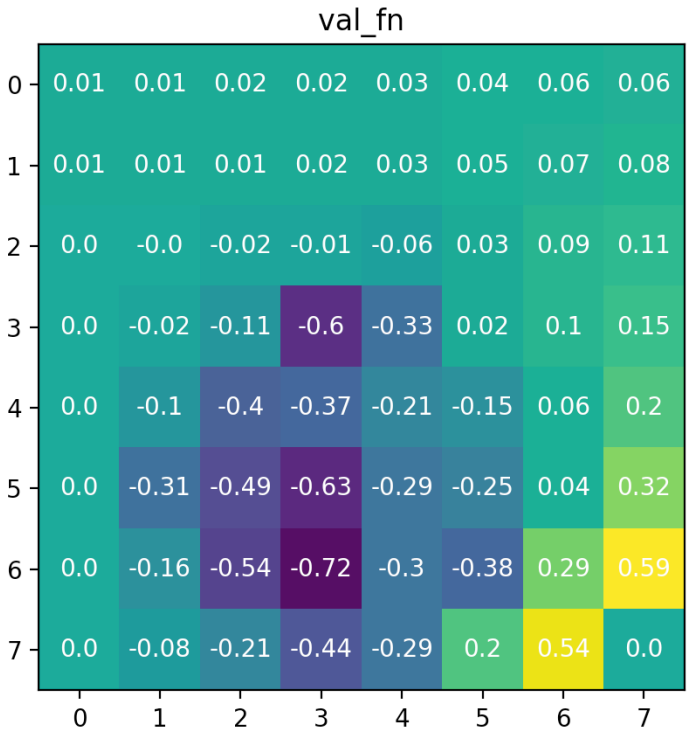
$$\nabla_\theta l_n(\theta) = \frac{1}{T}\sum_{t=1}^{T}\left[2\left(\frac{c}{1-\gamma}\right)\nabla_\theta f(s;\theta)\right]$$

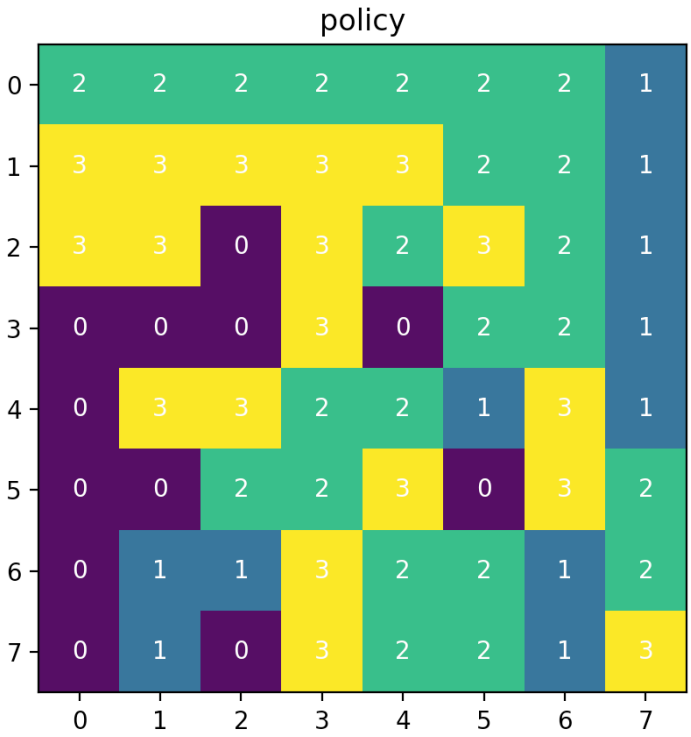
And since from:

$$E_{s\sim v}\left[\left(f(s;\theta) - E_{a\sim\pi(.|s)}\left[r(s,a) + \gamma E_{s'\sim P(.|s,a)} f(s';\theta_n)\right]\right)^2\right]$$

The subcomponent $E_{s\sim v}[\nabla_\theta f(s;\theta)]$ has the same expected value as $\frac{1}{T}\sum_{t=1}^{T}[\nabla_\theta f(s;\theta)]$ then the estimate is not biased.

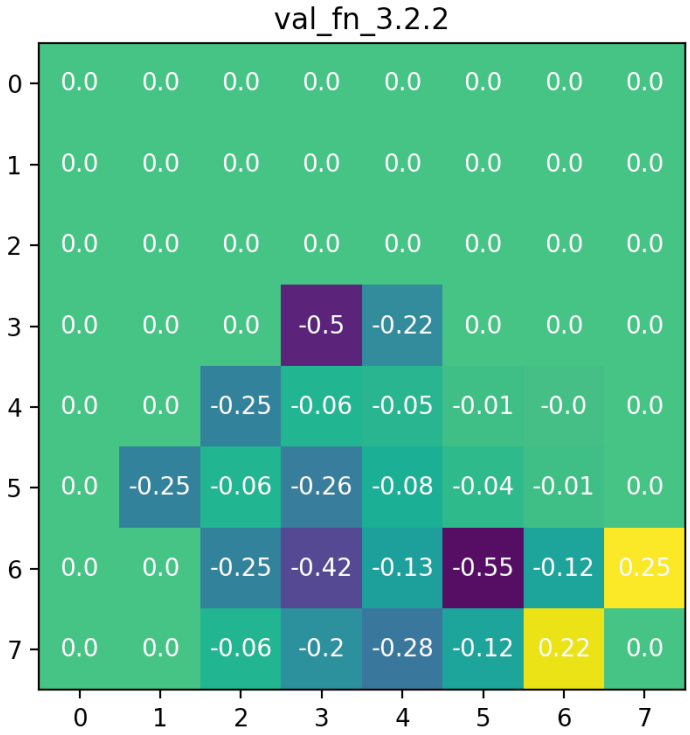
Q3.2.1
Answer:

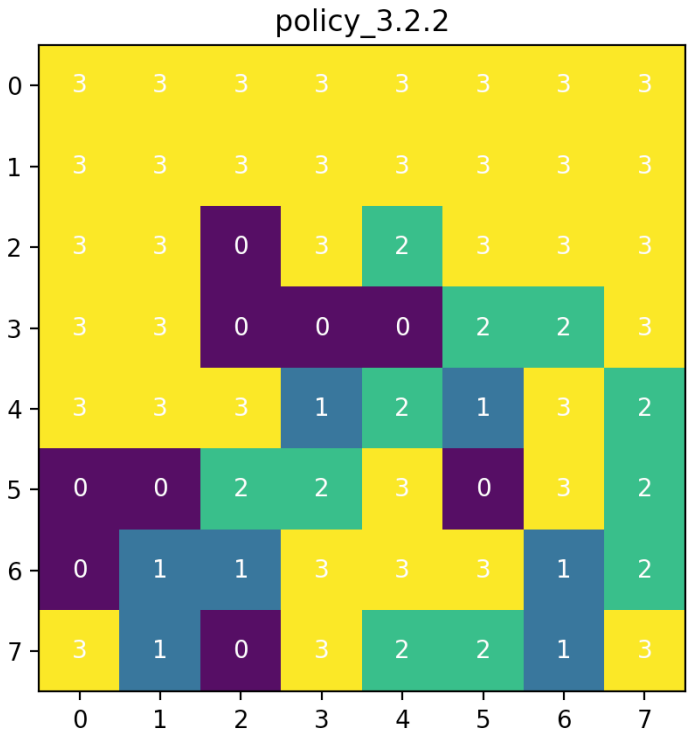




Number of iterations: 29

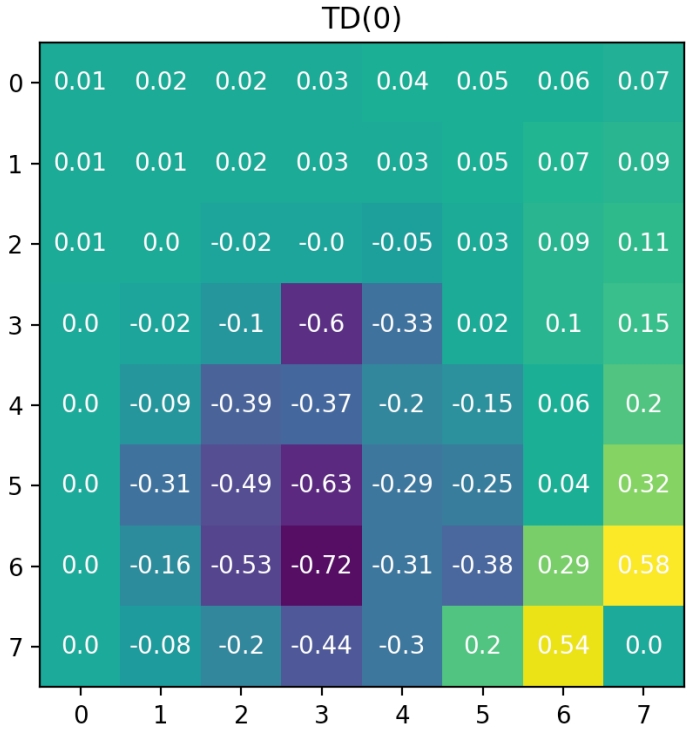
Q3.2.2
Answer:





Convergence is achieved at 10 iterations

Q3.2.3
Answer:



The value of alpha that was used: 0.05 (default) and yields the same value function as value iteration.

Q3.2.4
Answer:

Q3.3.1
Answer:

L1 norm expressed as:

$$|x|_l = \sum_{i=0}^{n} ||x_i||$$

Where n is the subscript of the last variable in a vector of x

We now introduce the need for an additional variable called e. Now, if:

$$s.t. \quad e_i \geq |x_i|$$

Then minimizing e achieves the same as maximizing the absolute value of x:

$$s.t. \quad -e_i - reward_i \leq 0$$

So this becomes:

$$reward_i - e_i \leq 0$$

Q3.3.2
Answer:

From the expression:

$$\hat{R} = arg\max_{R}\left(\sum_{s}\min_{a}\{(P_{a^*}(s) - P_a(s))(I - \gamma P_{a^*})^{-1}R\}\right) - \lambda||R||_1$$

We can find that part of the constraints is this term in blue above, namely:

$$\min_{a}\{(P_{a^*}(s) - P_a(s))(I - \gamma P_{a^*})^{-1}R\}$$

Which really means:

$$M - (P_{a^*}(s) - P_a(s))(I - \gamma P_{a^*})^{-1} \leq 0$$

Now another part of the constraint is actually given to us directly in the preamble to the question:

$$s.t.\ (P_a - P_{a^*})(I - \gamma P_{a^*})^{-1}R \leq 0$$

Combining them together:

$$-(P_{a^*}(s) - P_a(s))(I - \gamma P_{a^*})^{-1}R + M \leq 0$$

So this means that since we are trying to minimize:

$$s.t.\ \left(-(P_{a^*}(s) - P_a(s))(I - \gamma P_{a^*})^{-1}I\right)\begin{pmatrix}R\\M\end{pmatrix} \leq 0$$

$$\therefore \left((P_a(s) - P_{a^*}(s))(I - \gamma P_{a^*})^{-1}\ \ 0\right)\begin{pmatrix}R\\M\end{pmatrix} \leq 0$$

This represents the objective in standard form.

Q3.3.3
Answer:

From the previous question, we already have this constrain:

$$\therefore \left(\left(P_a(s) - P_{a^*}(s) \right)(I - \gamma P_{a^*})^{-1} \;\; 0 \right) \begin{pmatrix} R \\ M \end{pmatrix} \leq 0$$

Now on top of this, from question 1 we have: $reward_i - e_i \leq 0$

In matrix form it takes on:

$$s.t. (-I - I) \begin{pmatrix} R \\ E \end{pmatrix} \leq 0$$

Where represents the rewards and E represents e_i

Therefore this is equal to:

$$(I - I) \begin{pmatrix} R \\ E \end{pmatrix} \leq 0$$

Q3.3.4
Answer:

From question 3.3.2, you need to start with minimizing the expression $arg\min_a\{\big(P_{a^*}(s) - P_a(s)\big)(I - \gamma P_{a^*})^{-1}R\}$ which gives:

$$s.t. \quad \Big(-\big(P_{a^*}(s) - P_a(s)\big)(I - \gamma P_{a^*})^{-1} \;\; I\Big)\begin{pmatrix} R \\ M \end{pmatrix} \leq 0$$

$$\Big(\big(-P_{a^*}(s) + P_a(s)\big)(I - \gamma P_{a^*})^{-1} \;\; 0\Big)\begin{pmatrix} R \\ M \end{pmatrix} \leq 0$$

Next, from the previous question we need to minimize e. So just used use what is available from the previous question:

$$s.t. \; (-I - I)\begin{pmatrix} R \\ U \end{pmatrix} \leq 0$$

$$(I - I)\begin{pmatrix} R \\ U \end{pmatrix} \leq 0$$

And combining everything together we get the minimization of $\begin{pmatrix} 0 & -I & -\gamma I \end{pmatrix}\begin{pmatrix} R \\ M \\ U \end{pmatrix}$:

$$s.t. \quad \begin{pmatrix} -\big(P_{a^*}(s) - P_a(s)\big)(I - \gamma P_{a^*})^{-1} & I & 0 \\ -\big(P_{a^*}(s) - P_a(s)\big)(I - \gamma P_{a^*})^{-1} & 0 & 0 \\ -I & 0 & -I \\ I & 0 & -I \end{pmatrix}\begin{pmatrix} R \\ M \\ U \end{pmatrix} \leq 0$$

This is the final formulation in standard form for the linear program.

Q3.4
Answer:

Not attempted

Q4
Answer:

Drew who is the Chief Technology Officer of Aurora gave a talk about Aurora's strategy to get to level 5 automation in self-driving cars. Their inspiration for crafting level 5 automation systems is to rely on the principal of imitation learning, where a computer can leverage the human's remarkable driving acumen, and learn to imitate how humans drive.

Drew's presentation revolved around 2 strategies, which Aurora has attempted towards the problem of imitation learning. The first was to construct a policy, the other was to construct a cost map both of which were done through imitation learning. Drew gave a bit of a background on the strengths and weaknesses of both approaches.

Here is a bit of a background on what he mentioned about each of them:

Constructing a policy:
Drew showed us an experiment using imitation learning on a car driving video game. Clearly the system performed poorly and this largely boils down to a bias that was the human player was really good and therefore only showcased data that was heavily skewed towards fluid, optimal driving while having minimal corrective or evasive strategies by the driver. The system thus learned only from good driving and thus did not learn how to craft a policy to address states that were not nominal for good driving. The way to deal with this according to Drew was to prod the human player to override faulty actions and let the machine directly learn from this override.

Constructing a cost-map
Drew mentioned the alternative, which according to him was the easier and more appropriate approach of the two. The system would imitate the driver by first painting a picture of the navigable landscape in terms of the cost of traversing over that terrain. It would do this by learning what is preferable by the human driver, and associate avoided zones as high cost zones. And then on top of this motion planners are overlaid to navigate this cost map. The benefits of this method according to Drew are that it is more generalizable than directly constructing the policy because it is not bespoke to the vehicle the human is driving in, it can be more easily transferred from vehicle to vehicle as the cost map is vehicle invariant, however the policy is as one evasive manoeuver may be harder to execute in one vehicle than the other.

Drew mentioned how figuring out the right way to do this was crucial to the development of level 5 automation in self driving cars. He believes that imitation learning is the way to go, as it is almost intractable to go down the route that other companies are going down. This is not a solved problem yet but they prioritize and align their development roadmap to mature this method, which they are betting on.