CMPUT 655
Assignment 7

October 9, 2024

Dikshant
1877098

---

1. **How is RL different from SL regression?**
   (a) Why does regression can be seen as a subproblem of RL? i.e., where does SL appear as a problem in RL? In SL, we have $\{x_i, y_i\}_{i=1...N}$ samples. In RL, what are our x_i and y_i?
   (b) What is i.i.d. data?
   (c) What does it mean for data or a loss function to be stationary?
   (d) In what way does RL violate such assumptions?

Ans: In SL regression, we have fixed input-output pairs and try to predict outputs based on inputs, assuming the data is i.i.d. and stationary. In RL, data is sequential and depends on the agent's interactions with the environment, making it non-i.i.d. and often non-stationary.

(a) Regression is a subproblem of RL because RL often involves fitting value functions, where the state-action pairs (s, a) are inputs and the expected return $r + \gamma V(s')$ is the output, similar to regression in supervised learning. Thus, in RL, the agent essentially performs regression to minimize the prediction error of future rewards based on past experiences.

(b) i.i.d.(independent and identically distributed) data means each sample has the same probability distribution and is independent of other samples. This assumption helps ensure unbiased and consistent estimations in learning algorithms.

(c) Stationary means that data distribution ($P(x, y)$) or a loss function remains constant over time, i.e. the statistical properties such as mean and variance do not change over time.

(d) RL violates i.i.d. and stationarity because the agent's actions affect the state transitions, creating sequential dependencies or trajectories. Additionally, the policy changes over time as the agent learns, leading to non-stationary state distributions and TD targets.

2. *FQI tries to adhere to regression's i.i.d. and stationary assumptions. What are the advantages and disadvantages over tabular methods like Q-Learning?*

> **Ans: Advantages of FQI**: FQI fits the Q-function using fixed data and ensures i.i.d. sampling by randomizing the dataset. This leads to more stable learning and avoids the non-stationarity seen in online updates.
> **Disadvantages of FQI**: FQI requires large batches of data and relies on off-policy data, which can lead to mismatches between the learned policy and the actual state distribution. It also assumes stationarity within each batch, which can be unrealistic in dynamic environments.

3. **Discuss the algorithm:**

   (a) *Fixing the target before fitting the weights guarantees one condition needed for regression. Which one?*

   (b) *FQI was originally developed as offline batch algorithm. That is, updates are not performed online as data is collected, but instead after some a large (ideally) amount of data is collected and we sample from it.*
   • *This guarantees one condition needed for regression. Which one?*

   (c) *In-class, we presented the pseudocode with $K = 1$: after we collect data, we perform one fitting and the immediately collect new data again. The algorithm you implemented, instead, performs multiple fits before collecting new data when $K > 1$.*
   While we still adhere to the i.i.d. and stationary assumptions, we try to minimize the TD error of a Q-function (newly updated) with data collected following the previous Q-funciton.
   • *What could we do to correct our estimates?*

   (d) *The algorithm you implemented keeps updating the weights at every "update routine". This introduces bias (often called "primacy bias") before each update.*
   i.e., the fitting step (in our case, gradient descent) starts from a biased point (the current weights).
   • *However, this does not matter in our case. Why? (hint: think about what kind of FA we use, what kind of loss function we minimize, and how many optima there are)*

   (e) *The setting $N = 1$, $K = 1$, $D = 1$, is a special case as it implements an algorithm we have already seen.*
   • *Which one?*

Ans:

(a) Fixing the target ensures stationarity of the targets during the regression process

(b) This guarantees i.i.d. data. By collecting a large batch and sampling randomly from it, FQI ensures that the data points used for updates are independent of each other.

(c) We can use importance sampling. This adjusts for the difference in the state distribution between the current and previous policies by weighting samples according to their likelihood under the current policy.

(d) This does not matter in our case because we are using a linear function approximator and minimizing MSE, which is a convex loss function . Since it's convex, there is a unique global optimum, so the gradient descent algorithm will converge to the same solution regardless of the starting point, eliminating the effect of primacy bias over time

(e) Q-learning

4. **Discuss the results**:

   (a) There is an hyperparameter (between N, K, and D) that matters the most. Which one and why?
   - *If we increase the other two but not this one, the agent does not learn.*
   - *Think about the environment and its stochasticity.*
   - *Assume the environment is now deterministic. Which hyperparameter can we significantly decrease?*
   - *Would we need fewer updates? Less samples?*

   (b) The TD error is not a good indicator of whether the agent is learning or not, i.e., in many cases, the TD error goes quickly to 0 but the agent does not learn.
   - *This denotes that a well-known problem of regression is happening. Which one?*
   - *Think about the dataset size D and the number of total updates (which depends on N and K).*
   - *Can we do "too many" updates?*
   - *To mitigate this issue, do you think it's more important to have fewer fits (K) or fewer updates per fit (N)?*
   - *Think about the breaking condition of the loop: if N is low, do you expect the outer loop (K) to break sooner or later?*

   (c) Consider the algorithm implemented for this Assignment. Do you think results would be different with a non-linear FA, e.g., neural networks? Why?
   - *This relates to a question asked in the previous set of bullet points.*

Ans:

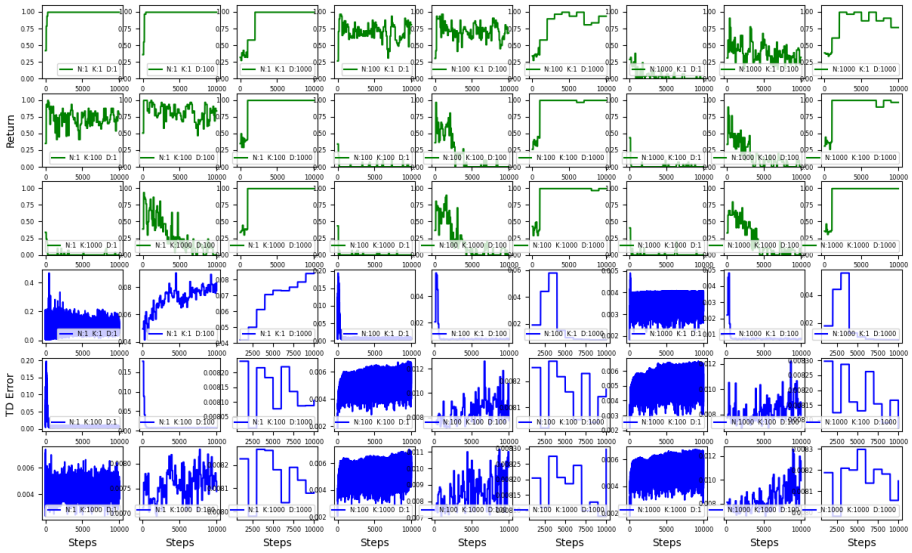

Figure 1: FQI

(a) The most important hyperparameter is D (dataset size), especially with transition and reward noise. With noise in the environment, having enough data becomes critical to approximate the Q-function reliably. Increasing only N or K without enough data will not help due to the randomness in the environment.

In a deterministic environment, D can be reduced significantly, as fewer samples would capture the environment dynamics almost perfectly. Fewer updates and less data would be required because the agent's transitions and rewards would be consistent, reducing the need for extensive exploration.

(b) The TD error going to 0 despite the agent not learning indicates overfitting to the noisy data.

Too many updates (N or K) can cause overfitting to the noise in the data. If the fitting process runs too many times on noisy data, it will try to fit onto the fluctuations caused by the noise, rather than the true value function.

To mitigate overfitting issue, fewer fits (K) would be better, as repeating fits with noisy data would only reinforce the bias from the noise.

If N is low, meaning fewer updates are done in each fit, the outer loop (K) is likely to run longer since the Q-function will require more iterations to converge.

(c) Results will differ for a non-linear FA like neural networks because they are non-convex functions, making the optimization sensitive to initialization. Poor starting points may lead to local minima, impacting learning in noisy environments.
Also, in this noisy gridworld non-linear models can lead to worse overfitting because are more flexible and can fit the noise in the data too well, making proper initialization and careful updates essential.

Just for fun, I also ran my code for 100,000 steps with $\alpha = 0.005$ (it took roughly 12 hours) and I got these plots:

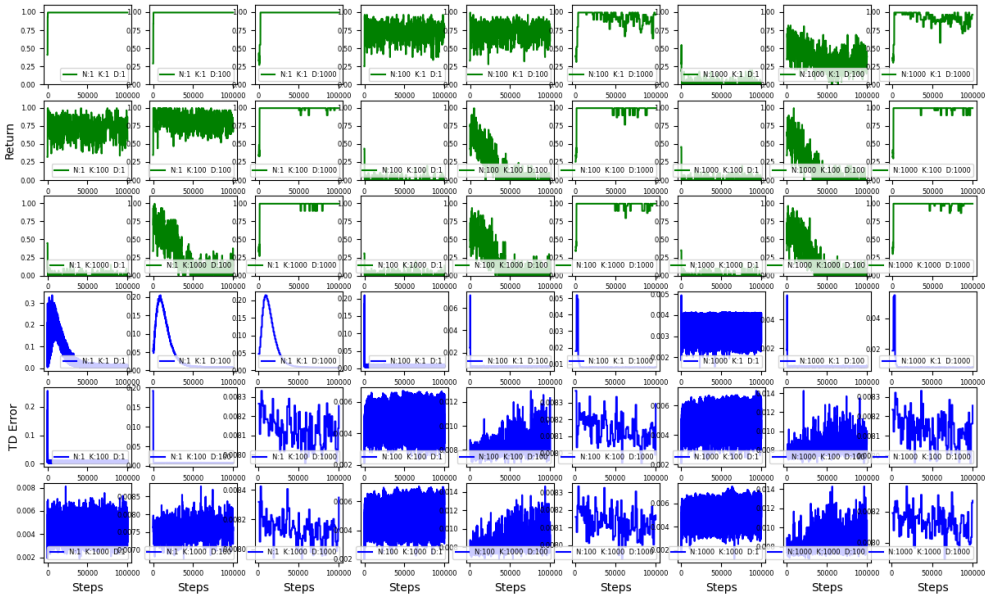

Figure 2: FQI with 100000 steps

Code can be accessed on github from here: FQI