15-889e Real Life Reinforcement Learning

Homework 1

Due October 5

Rules:

1. Homework is due October 5 at 1:30pm (start of class). Please see course website for policy on late submission.
2. We recommend that you typeset your homework using appropriate software such as LaTeX. If you are writing please make sure your homework is cleanly written and legible. The TAs will not invest undue effort to decrypt bad handwriting.
3. Please submit the written portion of the homework and the programming portion separately.
4. You may do the programming part of the homework in pairs. If you choose to do this, you should submit a single homework for that part, with both of your names and andrew ids.
5. You are allowed to discuss ideas about the written portion of the homework with others, but you must write up your own solution. If you do discuss the written portion with others, please indicate your collaborators in your submission. Also, if you use any outside sources (wikipedia, a technical paper or chapter, etc) to answer the questions, please list them.

---

1 Function Approximators for Fitted Value Iteration

We have learned that Fitted Value Iteration (FVI) converges, if the function approximator that we use to fit the Q-function is a non-expansion in the max-norm. A prominent class functions that satify this condition are *averagers*. Averagers are defined as function approximators that satify (1) linearity, (2) monotonicity and (3) nonexpansitivity. Let $x_1, \dots x_n \in \mathbb{R}^d$ be a set of inputs. It can be shown that the function approximator S is an averager if it can be written as

$$(Sf)(x_i) = k_i + \sum_{j=1}^{n} \theta_{ij} f(x_j) \tag{1}$$

for any input function f with $\theta_{ij} \geq 0$ and $\sum_j \theta_{ij} \leq 1$ for any $i = 1 \dots n$. The coefficients θ_{ij} and k_i may depend on $x_1, \dots, x_n$ but not on f. In class, we have seen that using a linear function approximator with a least-squares fit is not an averager.

1.1 Averagers or not

For each of the following function classes, decide whether they are averagers. Give an explanation for your answer. If you decide the function class is an averager, you should express it in the above form, or provide some other proof of why it is an averager. If you decide the function class is not an averager, you should provide an example or proof. For simplicity, you may assume one-dimensional inputs, i.e. $x_i \in \mathbb{R}$.

(a) k-nearest neighbor estimator

(b) Random forest regression with constant values in the leaf nodes

1.2 Properties of averagers

Is it always beneficial to have a function class be an averager? Give an example of a function class that is guaranteed to be a non-expansion in the max norm, yet we expect to perform very poorly.

1.3 Other Function Approximators

We know that FVI converges to a unique solution for non-expansive function approximators but not every class of approximators we would like to use is nonexpansive. We might wonder whether FVI still converges when we use them. The following statement shows that FVI with expansive function approximators that satify an additional technical assumption does not convege in general.

Let A be a function approximator and suppose there are value functions V_1 and V_2 that satisfy $\forall s V_1(s) \leq V_2(s)$ and

$$\|AV_1 - AV_2\|_\infty > \|V_1 - V_2\|_\infty. \tag{2}$$

Then there is a Markov decision process such that FVI does not converge to a unique solution.

Prove this statement.

2 Offline Batch RL: Customer Marketing

In the 1998 KDD Cup competition, the goal was to estimate the return for a consumer direct mailing task for solicting donations. We have built an approximate dynamics and reward model for this setting, and sampled trajectories given this model. Your task is to take this batch set of data, and use offline batch reinforcement learning to compute a good policy for future use. The dataset of 3000 episodes of length 24 is provided as a csv file with one episode per line. The state space is represented by 9 features

- customer age (`AGE_*`)
- customer income bracket, an integer in $[1, 7]$ (`INCOME_*`)
- the number of months between the first promotion the custom received and the first donation (`TIMELAG_*`)
- number of gifts to date (`NUM_GIFTS_*`)
- number of promotions to date (`NUM_PROM_*`)
- total amount of gifts to date (`AMOUNT_GIFTS_*`)
- the amount in dollars of the last gift (`AMOUNT_LASTGIFT_*`)
- number of months since the last gift (`GIFT_RECENCY_*`)
- number of months since the last promotion (`PROM_RECENCY_*`)

Missing values are marked by `NA`. The dataset was generated by a random policy that sends a promotion with probability 0.6 (available in `ACTION_*`). The reward for each timestep is the gift received minus the costs of sending the promotion (`REWARD_*`).

Please implement FVI and LSPI to compute policies. You should try FVI with a linear function approximator which is the same as a linear function approximator you use for LSPI, and also try FVI with another approximator. You are allowed to use libraries for supervised machine learning algorithms.

A crucial second issue is what features should be used. Select and implement at least one method for feature selection.

Submit a write up (for 2-4 pages) discussing your choice of approximators, feature selection method(s), and which approach yielded the highest estimated policy value, as well as which approach you expect to generalize the best.

You must also turn in your code as a zip file. We will evaluate your best learned policy using the model we used to generate the data: for this we required you to provide us with a function that takes a state description (quantities listed above) as inputs and returns the action as a boolean. Please

- provide sufficient documentation for your function. Clearly state what inputs your function expects, especially how missing values are represented;
- make sure your function is sufficiently fast. On modern hardware, we should be able to evaluate your policy at least 10-100 times per second;
- make sure that your function can be called from `Julia`, `Python`, `Matlab` or `C`. If you want to use another language, please ask us first.