17. Introduction

Batch learning is concerned with problems when a learning algorithm must work with data collected in some manner that is not under the control of the learning algorithm: on a batch of data. In batch RL the data is given in the form of a sequence of trajectories of varying length, where each trajectory is of the form $\tau = (S_0, A_0, R_0, S_1, A_1, R_1, \dots, S_t, A_t, R_t, S_{t+1})$, where A_i is chosen in a causal fashion (based on "past" data), $(R_t, S_{t+1}) \sim Q_{A_t}(S_t)$, where $Q = (Q_a(s))_{s,a}$ is a collection of probability distributions over pairs of reals and states, as usual (when we want to allow stochastic rewards).

Batch RL problems fall into two basic categories:

1. **Value prediction**: Predict the value μv^π of using a policy π from the initial distribution μ, where both μ and v^π are given in an explicit form.
2. **Policy optimization**: Find a good (ideally, near optimal) policy given the batch of data from an MDP.

These two problems are intimately related. On the one hand, a good value predictor can potentially be used to find good policies. On the other hand, a good policy optimizer can also be used to decide about whether the value of some policy is above or below some fixed threshold by appropriately manipulating the data fed to the policy optimizers. One can then put a binary search procedure around this decision routine to find out the value of some policy.

Value prediction problems have some common variations. In policy evaluation, rather than evaluating a policy for some fixed initial distribution, the goal is to estimate the entire value function of the policy. Of course, this is at least as hard as the simpler, initial value estimation problem. However, much of the hardness of the problem is already captured by the initial value estimation problem. In initial value prediction, oftentimes the goal is to predict an interval that contains the true unknown value with a prescribed probability, rather than just producing a "point estimate". In the case of policy evaluation, the analogue is to predict a set that contains the true unknown value function with a prescribed probability. Here, a simpler goal is to estimate confidence intervals for each potential input (state), which when "pasted together" can be visualized as forming a confidence band.

There is also the question of how to collect data. In statistics, the problem of designing a "good way" of collecting the data is called the experimental design problem. The best is of course, if data can be collected in an **active manner**: This is when the data collection strategy changes in response to what data has been collected so far.

The problem of designing good active data collection strategies belongs to the bigger group of designing **online learning** algorithms. These are defined exactly based on that the data is collected in a way that depends on what data has been previously collected. The last segment of the part will be solely devoted to these online learning strategies.

In many applications, active data collection is not an option. There can be many reasons for this: active data collection may be deemed to be risky, expensive, or just technically challenging. When data is collected in a passive fashion, it may simply miss key information that would allow for good solutions. Still, in this case, there may be better and worse ways collecting data. Optimizing **experimental designs** is the problem of choosing good passive data collection strategies that lead to good learning outcomes. This topic came up in the context planning algorithms as they also need to create value function estimates and for this the data collection is better to be planned so that learning can succeed.

Oftentimes though, there is no control over how data is collected. Even worse, the method that was used to collect data may be unknown. When this is the case, not much can be done, as the following example shows:

Consider a bandit problem with two actions, denoted by $0, 1$ and a Bernoulli reward. Assume that the reward distribution is Bernoulli with parameter 0.1 when $a = 1$ and Bernoulli with parameter 0.9 when $a = 0$. Let Z be a random variable, which is normally unavailable, but which, together with the action a taken completely determines the reward. For example, Z could have a Bernoulli distribution with parameter $p = 0.1$, and if action a is chosen, the reward $R(a)$ obtained is

$$R(a) = aZ + (1-a)(1-Z)\,.$$

This is indeed consistent with that $R(a)$ has Bernoulli 0.1 distribution when $a = 1$ and has Bernoulli 0.9 distribution when $a = 0$. Assume now that during data collection the actions are chosen based on Z: $A = \pi(Z)$ with some π. For concreteness, assume that during data collection $A = Z$. Then, the action is random, yet, if the data is composed of pairs that have the distribution shared by $(A, R(A))$, or $(Z, 1)$, clearly no method will be able to properly estimate the mean of $R(0)$ or $R(1)$, let alone choosing the action that leads to a higher reward. It is not hard to construct examples when the conditional mean of the observed data makes an optimal action look worse than a suboptimal action.

This is an example where the correct model cannot be estimated because of the way data is collected: The presence of the spurious correlation between a variable that controls outcomes but is not recorded can easily make the data collected useless, regardless of quantities. This is an instance when the model is unidentifiable even with an infinite amount of data.

When data collection is as arbitrary as in the above example, only a very careful study of the **domain** can tell us whether the model is identifiable or not from the data. Note that this is an activity that involves thinking about the structure of the problem at hand. The best is of course if data collection can be influenced to avoid building up spurious correlations. When data is collected in a causal way (following a policy, while recording both the decisions made and the data is used to make those decisions), spurious correlations are avoided and the remaining problem is to guarantee sufficient "coverage" to achieve statistical efficiency.

How good is the plug-in method?

The **plug-in method** estimates a model and uses the estimated model in place of the real one to solve the problem at hand. Let $M = (\mathcal{S}, \mathcal{A}, P, r)$ be a finite MDP, $\hat{M} = (\mathcal{S}, \mathcal{A}, \hat{P}, \hat{r})$ be an estimate. The estimate can be produced in a number of ways, but from the perspective of the result that comes, how the estimate is produced does not matter.

We consider the discounted case with a discount factor $0 \le \gamma < 1$. We will use $\hat{v}^\pi$ to denote the value function of a policy π in $\hat{M}$ (as opposed to v^π, which is the value function of policy in M), and similarly, we will use $\hat{v}^*$ to denote the optimal value function in $\hat{M}$. We analogously use $\hat{q}^\pi$ and $\hat{q}^*$. Every other quantity that is usually associated with an MDP but which now is associated with $\hat{M}$ receives a "hat". For example, we use $\hat{T}_\pi$ for the policy evaluation operator of memoryless policy π in $\hat{M}$ (either for the state values, or the action-values), while we use $\hat{T}$ to denote the Bellman optimality operator underlying $\hat{M}$ (again, both for the state and action-values).

We start with a generic result about contraction mappings:

Proposition (residual bound): Let $F : V \to V$ be a γ-contraction over a normed vector space V and let $x \in V$ be a fixed-point of F. Then for any $y \in V$,

$$\|x - y\| \le \frac{\|Fy - y\|}{1 - \gamma} \,. \tag{1}$$

Proof: By the triangle inequality,

$$\|x-y\| \le \|Fx-Fy\| + \|Fy-y\| \le \gamma\|x-y\| + \|Fy-y\|\,.$$

Reordering and solving for $|x-y|$ gives the result. ■

An immediate implication is that good model estimates are guaranteed to give rise to (relatively) good value estimates.

Proposition (value estimation error): Let $H_\gamma = 1/(1-\gamma)$ and assume that the rewards in M are in the $[0,1]$ interval. For any policy π, the following holds:

$$\begin{aligned}\|v^\pi - \hat{v}^\pi\|_\infty &\le H_\gamma\left(\|r_\pi - \hat{r}_\pi\|_\infty + \gamma\|(P_\pi - \hat{P}_\pi)v^\pi\|_\infty\right) \quad (2)\\ &\le H_\gamma\left(\|r-\hat{r}\|_\infty + \gamma H_\gamma\|P-\hat{P}\|_\infty\right). \quad (3)\end{aligned}$$

Also,

$$\begin{aligned}\|v^* - \hat{v}^*\|_\infty &\le H_\gamma\left(\|r - \hat{r}\|_\infty + \gamma\|(P - \hat{P})v^*\|_\infty\right) \quad (4)\\ &\le H_\gamma\left(\|r-\hat{r}\|_\infty + \gamma H_\gamma\|P-\hat{P}\|_\infty\right). \quad (5)\end{aligned}$$

Similarly,

$$\begin{aligned}\|q^\pi - \hat{q}^\pi\|_\infty &\le H_\gamma\left(\|r - \hat{r}\|_\infty + \gamma\|(P - \hat{P})v^\pi\|_\infty\right) \quad (6)\\ &\le H_\gamma\left(\|r-\hat{r}\|_\infty + \gamma H_\gamma\|P-\hat{P}\|_\infty\right). \quad (7)\end{aligned}$$

and

$$\begin{aligned}\|q^* - \hat{q}^*\|_\infty &\le H_\gamma\left(\|r - \hat{r}\|_\infty + \gamma\|(P - \hat{P})v^*\|_\infty\right) \quad (8)\\ &\le H_\gamma\left(\|r-\hat{r}\|_\infty + \gamma H_\gamma\|P-\hat{P}\|_\infty\right). \quad (9)\end{aligned}$$

Note that in general the value estimates are more sensitive to errors in the transition probabilities then in the rewards. In particular, the transition errors can be magnified by a factor as large as H_γ, while the reward errors are magnified by at most H_γ. Also note that sometimes one can obtain tighter estimates with stopping earlier in the derivations of these bounds. We will see some examples of how this can help later.

Proof: To reduce clutter, we write $\|\cdot\|$ for $\|\cdot\|_\infty$. Let $F = \hat{T}_\pi$, where $\hat{T}_\pi$ is defined via $\hat{T}_\pi v = \hat{r}_\pi + \gamma \hat{P}_\pi v$. By the residual bound (1),

$$\|\hat{v}^\pi - v^\pi\| \le H_\gamma \|\hat{T}_\pi v^\pi - v^\pi\| = H_\gamma \|\hat{T}_\pi v^\pi - T_\pi v^\pi\| \le H_\gamma \left(\|r_\pi - \hat{r}_\pi\| + \gamma \|(P_\pi - \hat{P}_\pi) v^\pi\| \right).$$

The second inequality follows from separating v^π from the second term and bounding it using $\|v^\pi\| \le H_\gamma$ and also using that $r_\pi = M_\pi r$, $\hat{r}_\pi = M_\pi \hat{r}$, $P_\pi = M_\pi P$ and $\hat{P}_\pi = M_\pi \hat{P}$ and finally using that M_π is a nonexpansion. The remaining inequalities can be obtained in an entirely analogous manner and hence their proof is omitted. ■

The result just shown suffices to quantify the size of the value errors. For quantifying the **policy optimization error** that results from finding an optimal (or near optimal) policy for $\hat{M}$, recall the Policy Error Bound from Lecture 6:

Lemma (Policy error bound - I.): Let π be a memoryless policy and choose a function $q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ and $\epsilon \ge 0$. Then, the following hold:

1. If π is ϵ-**optimizing** in the sense that $\sum_a \pi(a|s) q^*(s,a) \ge v^*(s) - \epsilon$ holds for every state $s \in \mathcal{S}$ then π is $\epsilon/(1-\gamma)$ suboptimal: $v^\pi \ge v^* - \frac{\epsilon}{1-\gamma} \boldsymbol{1}$.
2. If π is greedy with respect to q then π is 2ϵ-optimizing with $\epsilon = \|q - q^*\|_\infty$ and thus

$$v^\pi \ge v^* - \frac{2\|q - q^*\|_\infty}{1-\gamma} \boldsymbol{1} \,.$$

This leads to the following result:

Theorem (bound on policy optimization error): Assume that the rewards both in M and $\hat{M}$ belong to the $[0,1]$ interval. Take any $\varepsilon > 0$ and ε-optimal policy π in $\hat{M}$: $\hat{v}^\pi \ge \hat{v}^* - \varepsilon \boldsymbol{1}$. Then, π is δ-optimal in M with δ satisfying

$$\delta \le (1 + 2\gamma) H_\gamma \varepsilon + 2 H_\gamma^2 \left\{ \|r - \hat{r}\|_\infty + \gamma \|(P - \hat{P}) v^*\|_\infty \right\} .$$

Note that, up to a small constant factor, the optimization error is magnified by a factor of H_γ, the reward errors are magnified by a factor of H_γ^2, while the transition errors can get magnified by a factor of up to H_γ^3, depending on the magnitude of v^*.

Proof: Let π be a policy as in the theorem statement. Our goal now is to use the first part of the "Policy error bound", i.e., that π is ε'-optimizing with some $\varepsilon' > 0$.

On the one hand, we have

$$M_\pi \hat{q}^\pi = \hat{v}^\pi \ge \hat{v}^* - \varepsilon \mathbf{1} = M\hat{q}^* - \varepsilon \mathbf{1} \ge M\hat{q}^\pi - \varepsilon \mathbf{1}\,.$$

Let z be defined by $M_\pi \hat{q}^\pi = M\hat{q}^\pi + z$. From the previous inequality, we know that $\|z\|_\infty \le \varepsilon$. We also have

$$\begin{aligned}
M_\pi q^* &= M_\pi \hat{q}^\pi + M_\pi(q^* - \hat{q}^\pi) \\
&= M\hat{q}^\pi + M_\pi(q^* - \hat{q}^\pi) + z \\
&= Mq^* + M\hat{q}^\pi - Mq^* + M_\pi(q^* - \hat{q}^\pi) + z \\
&\ge Mq^* - (2\|\hat{q}^\pi - q^*\| + \varepsilon)\mathbf{1} \\
&= v^* - (2\|\hat{q}^\pi - q^*\| + \varepsilon)\mathbf{1}\,.
\end{aligned}$$

Hence, by Part 1. of the "Policy Error Bound I." lemma from above,

$$v^\pi \ge v^* - H_\gamma(2\|\hat{q}^\pi - q^*\| + \varepsilon)\mathbf{1}\,.$$

By the triangle inequality and the assumption on π,

$$\|\hat{q}^\pi - q^*\|_\infty \le \|\hat{q}^\pi - \hat{q}^*\|_\infty + \|\hat{q}^* - q^*\|_\infty \le \gamma\varepsilon + \|\hat{q}^* - q^*\|_\infty\,.$$

By Eq. (8),

$$\|q^* - \hat{q}^*\|_\infty \le H_\gamma\left(\|r - \hat{r}\|_\infty + \gamma\|(P - \hat{P})v^*\|_\infty\right)\,.$$

The result is obtained by chaining the inequalities:

$$\begin{aligned}
\|v^* - v^\pi\|_\infty &\le H_\gamma(2\|\hat{q}^\pi - q^*\| + \varepsilon) \\
&\le H_\gamma\left\{2\gamma\varepsilon + 2H_\gamma\left(\|r - \hat{r}\|_\infty + \gamma\|(P - \hat{P})v^*\|_\infty\right) + \varepsilon\right\}\,.
\end{aligned}$$

■

Model estimation error: Tabular case

As usual, it is worthwhile to clean up the foundations by considering the tabular case. In this case, the model can be estimared by using sample means. To allow for a unified presentation, let the data available be given in the form of triplets of the form $E_i = (S_i, A_i, R_i, S_{i+1})$ where $i = 1, \dots, n$ and $S_{i+1} \sim P_{A_i}(S_i)$ given $E_1, \dots, E_{i-1}, S_i, A_i$ and $\mathbb{E}[R_i|S_i, A_i, E_1, \dots, E_{i-1}] = r_{A_i}(S_i)$. Introducing the visit counts

$$N(s,a,s') = \sum_{i=1}^{n} \mathbb{I}(S_i = s, A_i = a, S_{i+1} = s')$$

and $N(s,a) = \sum_{s'} N(s,a,s')$, provided that the visit count for (s,a) is positive, for the transition probability estimates we have

$$\hat{P}_a(s,s') = \frac{N(s,a,s')}{N(s,a)}$$

and for the reward estimate we have

$$\hat{r}_a(s) = \frac{1}{N(s,a)} \sum_{i=1}^{n} \mathbb{I}(S_i = s, A_i = a) R_i\,.$$

For ensuring that these are always defined, let $\hat{P}_a(s)$ be the uniform distribution over the states and let $\hat{r}_a(s) = 0$ when $N(s,a) = 0$. From the perspective of the results to be presented, the particular values chosen here do not matter.

Consider now the simple case when the above triplets are so that for each state-action pair (s,a), $N(s,a) = n(s,a)$ for some deterministic counts $(n(s,a))_{s,a}$. Say, one has access to a generative model (simulator) and for each state-action pair the model is used to generate a fixed number of independent transitions. In this case, one can use Hoeffding's inequality.

In particular, defining

$$\beta(n,\zeta) = \sqrt{\frac{\log\left(\frac{\mathrm{SA}}{\zeta}\right)}{2n}}$$

provided that $R_i \in [0,1]$, Hoeffding's inequality gives that with probability $1-2\zeta$, for any s,a,

$$\begin{aligned} |\hat{r}_a(s) - r_a(s)| &\le \beta(n(s,a),\zeta)\,, \\ |\langle \hat{P}_a(s) - P_a(s), v^* \rangle| &\le H_\gamma \beta(n(s,a),\zeta)\,, \end{aligned}$$

from which it follows that with probability $1-2\zeta$,

$$\begin{aligned} \|\hat{r} - r\|_\infty &\le \beta(n_{\min},\zeta)\,, \\ \|(\hat{P} - P)v^*\|_\infty &\le H_\gamma \beta(n_{\min},\zeta)\,, \end{aligned}$$

where $n_{\min} = \min_{s,a} n(s,a)$. Plugging the obtained deviation bound into our policy suboptimality bound, we get that with probability $1-\zeta$,

$$\delta \le (1+2\gamma)H_\gamma \varepsilon + 2H_\gamma^2(1+\gamma H_\gamma)\beta(n_{\min},\zeta)\,.$$

One can alternatively write this in terms of the total number of observations, n. The best case is when $n(s,a)=n_{\min}$ for all (s,a) pairs, in which case $n=\mathrm{SA}n_{\min}$ and the above bound gives

$$\delta \le (1+2\gamma)H_\gamma \varepsilon + 2H_\gamma^2(1+\gamma H_\gamma)\sqrt{\mathrm{SA}\frac{\log\left(\frac{\mathrm{SA}}{\zeta}\right)}{2n}}\,.$$

It follows that for any target suboptimality δ_{trg}, as long as n, the number of observations satisfies

$$n \ge \frac{8H_\gamma^6 SA\log\left(\frac{\mathrm{SA}}{\zeta}\right)}{\delta_{\text{trg}}^2}\,,$$

we are guaranteed that the optimal policy of the estimated model is at most δ_{trg} suboptimal. As we shall see soon, the optimal dependence on the horizon H_γ is cubic, unlike the dependence shown here.

Notes

Between batch and online learning

In applications it may happen that one can change the data collection strategy a limited number of times. This creates a scenario that is in between batch and online learning. This setting can be thought to be between batch and online learning. From the perspective of online learning, this is learning in the presence of constraints on the data collection strategy. One such widely studied constraint is the number of switches of the data collection strategy. As it happens, only very few switches are necessary to get the full power of online learning and this is not really specific to reinforcement learning but follows because the empirical distribution converges are a slow rate to the true distribution. For parametric problems, the rate is $O(1/\sqrt{n})$ where n is the number of observations. Thus, to change "accuracy" of the estimates of any quantity in a significant fashion, the sample size should increase by much, which means, few changes to the data collection are sufficient. In other words, there is no reason to change the data collection strategy before one obtains sufficient new evidence that can help with deciding in what way the data collection strategy should be changed. This usually means that with only logarithmically many changes in the total sample size, one gets the full power of online methods.

Batch RL with no access to state information

For simplicity, we stated the batch learning problem in a way that assumes that the states in the transitions are observed. This may be seen as problematic. One "escape" is to treat the whole history as the state: Indeed, in a causal, controlled stochastic process, the history can always be used as a Markov state. Because of this, the assumption that the state is observed is not restrictive, though the state space becomes exponential in the length of the trajectories. This reduces to the problem to learning in large state-space MDPs. Of course, even lower bounds for planning tell us that in lack of extra structure, all algorithms need a sample size proportional to the size of the state-action space, hence, one needs to add extra structure to deal with this case, such as function approximation. It also holds that if one uses, say, linear function approximation, then only the features of the states (or state-action pairs) need to be recorded in the data.

Causal reasoning and batch RL

Whether a causal effect can be learned from a batch of data (to be more precise, from data drawn from a specific distribution) is the topic of **causal reasoning**. In batch RL, the "effect" is the value of a policy, which, in the language of causal reasoning, would be called a multistage treatment. As the example in the text shows, in batch RL, just because of our assumptions on how the data is collected, the identifiability problem is just "assumed away". When the assumption on how the data is generated/collected is not met, the tools of causal reasoning can potentially be still used. It is important to emphasize though that there is no causality without assuming causality. The statements that causal reasoning can make are conditional on the data sampling assumptions met. Even "causal discovery" is contingent on these assumptions. However, with care, oftentimes it is possible to argue for that some suitable assumptions are met (e.g., arguing based on what information is available at what time in a process), in which case, the nontrivial tools of causal reasoning may be very useful.

Nevertheless, especially in engineered systems, our standard data collection assumptions are reasonable and can be arranged for, though in large engineered systems, mistakes, such as not logging critical quantities may happen. One example of this is an action to be taken is overriden by some part of a system, which will, say, later be turned off. Clearly, if no one logs the actual actions taken, the effects of actions become unidentifiable. As we shall see later, batch RL and the causality literature share some of their vocabulary, such as "instrumental variables", "propensity scores", etc.

Plug-in or certainty equivalence

Plug-in generally means that a model is estimated and then is used as if it was the "true" model. In control, when a controller (policy) is derived with this approach, this is known as the "certainty equivalence" controller. The "certainty equivalence principle" states that the "random" errors can be neglected. The principle originates from the observation that in various scenarios, the optimal controller (optimal policy) has a special form that confirms this principle. In particular, this was first observed in the control of linear quadratic Gaussian control, where the optimal controller can be obtained by solving for the optimal control under perfect state information then substituting optimal state prediction for the the perfect state information. This strict optimality result is quite brittle. As we shall see soon, from the perspective of minimax optimality, certainty equivalent policies are not a bad choice.

Bibliographic remarks

In the early RL literature, online learning was dominant. When people tried to apply RL to various "industrial"/"applied" settings, they were forced to think about how to learn from data collected before learning starts. One of the first papers to push this agenda is the following one:

- Tree-Based Batch Mode Reinforcement Learning Damien Ernst, Pierre Geurts, Louis Wehenkel; 6(18):503−556, 2005.

Earlier mentions of "batch-mode RL" include

- Efficient Value Function Approximation Using Regression Trees (1999) by Xin Wang , Thomas G. Dietterich, Proceedings of the IJCAI Workshop on Statistical Machine Learning for Large-Scale Optimization. pdf

Even in online learning, efficient learning may force one to save all the data to be used for learning. The so-called LSTD algorithm, and later the LSPI algorithm, were explicitly proposed to address this challenge:

- J. A. Boyan. Technical update: least-squares temporal difference learning. Machine Learning, 49 (2-3):233–246, 2002.
- M. G. Lagoudakis and R. Parr. Least-squares policy iteration. Journal of Machine Learning Research, 4:1107–1149, 2003a.

Off-policy learning refers to the case when an algorithm needs to produce value function (or action-value function) estimates for some policy and the data available is **not** generated by the policy to be evaluated. In all the above examples, we are thus in the setting of off-policy learning. The policy evaluation problem, accordingly, is often called the **off-policy policy evaluation** (OPPE) problem, while the problem of finding a good policy is called the **off-policy policy optimization** (OPPO) problem.

For a review of the literature of around 2012, consult the following paper:

- S. Lange, T. Gabel, M. Riedmiller (2012) Batch Reinforcement Learning. In: M. Wiering, M. van Otterlo (eds) Reinforcement Learning. Adaptation, Learning, and Optimization, vol 12. Springer, Berlin, Heidelberg pdf