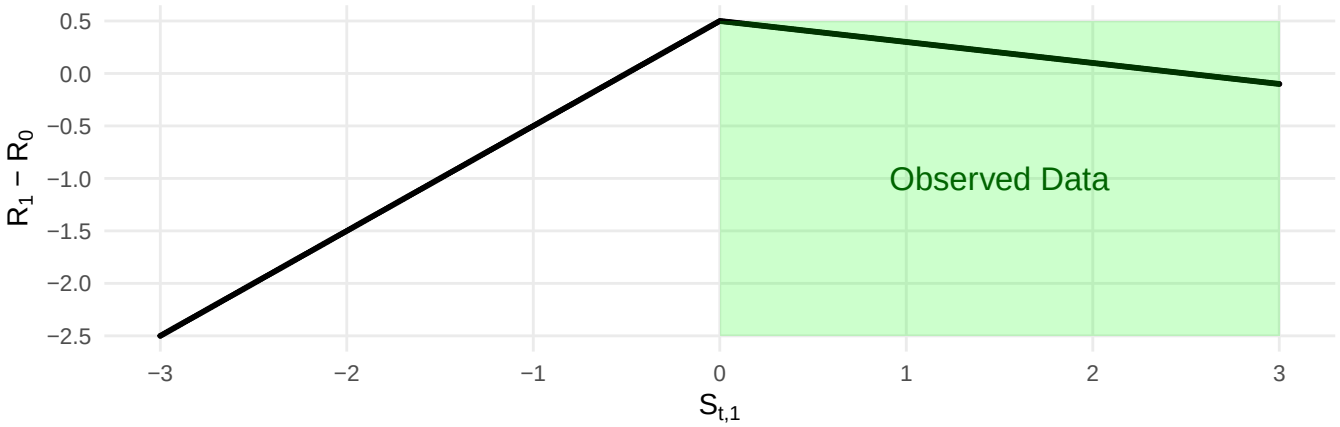
Statistical Inference Experiments

A Toy Problem

Suppose $S_t \in \mathbb{R}^2, A_t \in \{0,1\}, R_t = -S_{t+1,2}$ where S_t are covariates of a patient, A_t is a binary indicator of whether or not a treatment is given and R_t is a health outcome that is better when higher. We assume the data follows a standard Markov Decision Process. The state transition probabilities are $S_{t+1,1} = \sqrt{0.8}S_{t,1} + N(0, \sqrt{0.2}^2)$ and $S_{t+1,2} = A_1(0.2^{I(S_{t,1}>0)}|S_{t,1}| + 0.5) + N(0, 0.1^2)$. The behavioral policy $\mu(a|s) = I(s_1 > 0)\text{Bernoulli}(p = 0.5)$ never treats ($A_t = 0$) whenever $S_{t,1} < 0$ and assigns treatment with probability 0.5 otherwise. Shown below is the difference between $R_1(S_{t,1}) = \mathbb{E}[R_t|S_{t,1}, A_t = 1]$ and $R_0(S_{t,1}) = \mathbb{E}[R_t|S_{t,1}, A_t = 0]$ for varying values of $S_{t,1}$.



We can see that the treatment's effect on R_t increases as $S_{t,1}$ decreases when $S_{t,1} > 0$, and decreases as $S_{t,1}$ decreases when $S_{t,1} < 0$. However, because the behavioral policy never assigns treatments when $S_{t,1} < 0$, we only have treatment data for observations with $S_{t,1} > 0$. Thus, from the observed data it looks like the treatment effect **always** improves when $S_{t,1}$ decreases. This will cause estimates from the observed data to be biased for observations where $S_{t,1} < 0$, estimating a much more positive treatment effect than reality. This kind of data bias, or *distribution shift*, occurs frequently in offline RL. Because of this bias, applying fitted Q-iteration (FQI) yields an estimated policy that almost always recommends treatment and yields an average return that is worse than the behavior policy. *Conservative Q-learning (CQL)* avoids taking *out-of-distribution (OOD) actions*, or actions outside the support of the data, and applying it here yields a better policy than the behavior policy.

Estimated treatment effects are also inflated for $S_{t,1} < 0$ when estimating the *transition kernel* $\Pr(S_{t+1} = s'|S_t = s, A_t = a) = T(s'|s, a)$ from the observed data via maximum likelihood and performing *model-based evaluation*. Specifically, we fit a multivariate normal model with constant diagonal covariance via MLE, which corresponds to the true distributional family of the state transitions, where the conditional mean was approximated using polynomial models. The model still works well when we evaluate policies that avoid OOD actions, as it is only these actions where the fitted transition model is inaccurate. For example, this model accurately estimates the expected return for the behavioral policy and CQL policy. However, when evaluating the FQI policy, the model estimates it has a mean return 5x as high as reality, incorrectly concluding that FQI is the best-performing policy when it is actually the worst-performing. Similar issues arose when using *fitted Q-evaluation (FQE)* to evaluate policies.

Accounting for Distribution Shift

We need an *offline policy selection* procedure that chooses CQL over the behavioral policy and the behavioral policy over FQI. For problems where the behavioral policy is bounded away from zero, the standard procedure is to use log-likelihood on a held-out validation set to tune the hyperparameters of the transition model (e.g. the functional approximator, distributional family and training algorithm), and then use the transition model with the highest

validation log-likelihood to evaluate and select policies. This would not solve the problem though, as the validation set would have the same coverage and bias issues as the training set. Validation log-likelihood can still be used to tune the transition model's hyperparameters, as it still ensures accurate predictions for in-distribution state-action pairs, but we need a way to deal with the OOD actions. One idea based on discussions w/ Dr. Laber is to use estimation uncertainty in the state transition probabilities to calculate a 95% lower confidence bound of the expected return. Such a procedure should choose policies only if its estimated return is both large *and* accurate.

A straightforward way to adjust for estimation uncertainty is with a *bootstrap distribution* of estimated returns. Each estimate in this distribution comes from generating an artificial dataset by sampling trajectories from the observed data with replacement, estimating the transition kernel via MLE on this bootstrapped dataset and using the estimated transitions to estimate expected returns for a policy. However, in this case, the bootstrapped datasets are all biased similarly to the real dataset, causing the bootstrap estimates to be all similarly optimistic for observations with $S_{t,1} < 0$. As a result, a bootstrap distribution of 100 return estimates for the FQI policy still has a minimum estimate 5x as large as reality. The *accelerated bias-corrected percentile confidence interval*, which was designed to correct the standard bootstrap percentile interval for bias, would also generate very large lower confidence bounds as it is based on quantiles from this bootstrap distribution.

We could also use the data log-likelihood $\log \Pr(\mathcal{D}|\theta) = \sum_{(s,a,s')\sim\mathcal{D}} \log T_\theta(s'|s,a)$ and a noninformative prior $\Pr(\theta)$ over θ, the parameter vector of the transition model T_θ, to derive a *parametric posterior* distribution $\Pr(\theta|\mathcal{D})$. In this case, with the right conjugate prior, $\Pr(\theta|\mathcal{D})$ is multivariate normal and can be written in closed form. We can then generate a posterior over returns by sampling θ from its posterior and using each sampled parameter vector to generate an expected return estimate. This adjusts for estimation variance, but not bias: as the sample size approaches infinity, the estimated posterior $\Pr(\theta|\mathcal{D})$ will converge to a point mass at the MLE, which will still be biased. We note that this is also mathematically equivalent to using the asymptotic normal distribution of maximum likelihood estimators to generate a 95% lower *parametric confidence interval* of the expected return.

There are also many model-free approaches to estimate uncertainty in an approximated value function, as discussed in works such as Luckett et. al. 2019 (*V-learning* algorithm) and Shi, Zhang, Lu and Song 2021 (*Sequential Value Estimation* algorithm). However, these works assume that the behavioral policy is bounded away from zero. Moreover, these works assume correctly specified linear models for the value function, and tuning the number and kind of basis functions to make accurate predictions is not straightforward even for in-distribution state-action pairs. This contrasts to transition models, whose prediction accuracy for in-distribution actions can be evaluated and optimized via validation log-likelihood. Model selection methods based on Bellman errors, such as *Batch Value Function Tournament (BVFT)* and *Supervised Bellman Validation (SBV)*, can tune their own hyperparms offline. However, the estimated quantities are often too loosely associated with online returns to be practical, and more importantly, such methods assume the behavior policy is bounded away from zero as well.

Alternative Strategies

For this strategy to work well, I believe that any proposed posterior or confidence bound will need to give high uncertainty when making inference on state-action pairs that have low coverage. In this case, this means high uncertainty for estimated returns associated with policies that recommend $A_t = 1$ when $S_{t,1} < 0$. However, I'm not aware off-hand of any standard posterior distribution or confidence bound estimator that would do this, including the methods mentioned above.

To then end, we could quantify uncertainty of our model at a given state-action pair as $U(s,a) = \frac{1}{|\mathcal{D}|}\sum_{(s',a')\in\mathcal{D}} k((s,a),(s',a'))d((s,a),(s',a'))$ where k is a *kernel* function and d is a *distance* metric. An example kernel function is the *k-Nearest Neighbor* kernel, equal to one only for the few state-action pairs (s',a') closest to the target state-action pair (s,a). We can then modify the original reward function $r(s,a)$ with the penalized reward $\tilde{r}(s,a) = r(s,a) - \lambda U(s,a)$ so as to penalize transitions into states with high model error. Previous work has evaluated policies using an estimated transition model and reward functions w/ similar penalties to yield an empirical lower bound on the true expected return. The biggest problem with this approach is that $U(s,a)$ does not quantify transition model error in absolute terms: rather, it only tells us how concentrated the observations are at one data point compared to another, and thus how big the **relative** error is at one state-action pair compared to another. Thus it is not clear how to use this approach to derive a confidence bound of the expected return at a desired probability level. Instead we must rely on proper tuning of a hyperparameter λ, and it is not clear how to do this (all previous work tuned this hyperparameter using online interactions).