Learning the Reward Function for a Misspecified Model: Supplemental Material

Erik Talvitie [1]

A. Hallucinated DAgger-MC Details

Hallucinated DAgger-MC, like earlier variations on DAgger, requires the ability to reset to the initial state distribution μ and also the ability to reset to an "exploration distribution" ν. The exploration distribution ideally ensures that the agent will encounter states that would be visited by a good policy. The performance bound for H-DAgger-MC depends in part on the quality of the selected ν.

In addition to assuming a particular form for the planner (one-ply MC with a blind rollout policy), H-DAgger-MC requires the dynamics model to be "unrolled". Rather than learning a single $\hat{P}$, H-DAgger-MC learns a set $\{\hat{P}^1, \ldots, \hat{P}^{T-1}\} \subseteq \mathcal{P}$, where model $\hat{P}^i$ is responsible for predicting the outcome of step i of a rollout, given the state sampled from $\hat{P}^{i-1}$. While this impractical condition is important theoretically, Talvitie (2017) showed that in practice a single $\mathcal{P}$ can be used for all steps; the experiments in Section 5 make use of this practical alteration.

Algorithm 1 augments H-DAgger-MC to learn a reward model as well as a dynamics model. In particular, H-DAgger-MC proceeds in iterations, each iteration producing a new plan, which is turn used to collect data to train a new model. In each iteration state-action pairs are sampled using the current plan and the exploration distribution (lines 7-13), and then the world and model are rolled out in parallel to generate hallucinated training examples (lines 14-21). The resulting data is used to update the model. We simply add a reward model learning process, and collect training examples along with the state transition examples during the rollout. After both parts of the model have been updated, a new plan is generated for the subsequent iteration. Note that while the dynamics model is "unrolled", there is only a single reward model that is responsible for predicting the reward at every step of the rollout. We assume that the reward learning algorithm is performing a weighted regression (where each training example is weighted by γ^{t-1} for the rollout step t in which it occurred).

Algorithm 1 Hallucinated DAgger-MC (+ reward learning)

Require: LEARN-DYNAMICS, LEARN-REWARD, exploration distr. ν, MC-PLANNER(blind rollout policy ρ, depth T), # iterations N, # rollouts per iteration K.

1: Get initial datasets $\mathcal{D}_1^{1:T-1}$ and $\mathcal{E}_1$ (maybe using ν)
2: Initialize $\hat{P}_1^{1:T-1} \leftarrow$ LEARN-DYNAMICS($\mathcal{D}_1^{1:T-1}$).
3: Initialize $\hat{R}_1 \leftarrow$ LEARN-REWARD($\mathcal{E}_1$).
4: Initialize $\hat{\pi}_1 \leftarrow$ MC-PLANNER($\hat{P}_1^{1:T-1}$, $\hat{R}_1$).
5: **for** $n \leftarrow 2 \ldots N$ **do**
6: **for** $k \leftarrow 1 \ldots K$ **do**
7: With probability... ▷ First sample from ξ
8: $1/2$: Sample $(x, b) \sim D_\mu^{\hat{\pi}_n}$
9: $1/4$: Reset to $(x, b) \sim \nu$.
10: $(1-\gamma)/4$: Sample $x \sim \mu$, $b \sim \hat{\pi}_n(\cdot \mid x)$.
11: $\gamma/4$: Reset to $(y, c) \sim \nu$
12: Sample $x \sim P(\cdot \mid y, c)$, $b \sim \hat{\pi}_n(\cdot \mid x)$
13: Let $s \leftarrow x$, $z \leftarrow x$, $a \leftarrow b$.
14: **for** $t \leftarrow 1 \ldots T-1$ **do** ▷ Parallel rollouts...
15: Sample $s' \sim P(\cdot \mid s, a)$.
16: Add $\langle z, a, s' \rangle$ to $\mathcal{D}_n^t$.
▷ (DAgger-MC adds $\langle s, a, s' \rangle$)
17: Add $\langle z, a, R_s^a, \gamma^{t-1} \rangle$ to $\mathcal{E}_n$.
▷ (Standard approach adds $\langle s, a, R_s^a, \gamma^{t-1} \rangle$)
18: Sample $z' \sim \hat{P}_{n-1}^t(\cdot \mid z, a)$.
19: Let $s \leftarrow s'$, $z \leftarrow z'$, and sample $a \sim \rho$.
20: **end for**
21: Add $\langle z, a, R_s^a, \gamma^{T-1} \rangle$ to $\mathcal{E}_n$.
▷ (Standard approach adds $\langle s, a, R_s^a, \gamma^{T-1} \rangle$)
22: **end for**
23: $\hat{P}_n^{1:T-1} \leftarrow$ LEARN-DYNAMICS($\hat{P}_{n-1}^{1:T-1}$, $\mathcal{D}_n^{1:T-1}$)
24: $\hat{R}_n \leftarrow$ LEARN-REWARD($\hat{R}_{n-1}$, $\mathcal{E}_n$)
25: $\hat{\pi}_n \leftarrow$ MC-PLANNER($\hat{P}_n^{1:T-1}$, $\hat{R}_n$).
26: **end for**
27: **return** the sequence $\hat{\pi}_{1:N}$

A.1. Analysis of H-DAgger-MC

We now derive theoretical guarantees for this new version of H-DAgger-MC. The analysis is similar to that of existing

[1]Department of Computer Science, Franklin & Marshall College, Lancaster, Pennsylvania, USA. Correspondence to: Erik Talvitie <erik.talvitie@fandm.edu>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

DAgger variants (Ross & Bagnell, 2012; Talvitie, 2015; 2017), but the proof is included for completeness. Let H_n^t be the distribution from which H-DAgger-MC samples a training example at depth t (lines 7-13 to pick an initial state-action pair, lines 14-21 to roll out). Define the average error of the dynamics model at depth t to be

$$\bar{\epsilon}_{prd}^t = \frac{1}{N}\sum_{n=1}^{N} \mathbf{E}_{(s,z,a)\sim H_n^t}[1 - \hat{P}_n^t(\sigma_s^a \mid z, a)].$$

Let $\epsilon_{\hat{R}_n}(s,z,a) = |R(s,a) - \hat{R}_n(z,a)|$ and let

$$\bar{\epsilon}_{hrwd} = \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T} \gamma^{t-1}\, \mathbf{E}_{(s,z,a)\sim H_n^t}[\epsilon_{\hat{R}_n}(s,z,a)|]$$

be the average reward model error. Finally, let D_n^t be the distribution from which H-DAgger-MC samples s and a during the rollout in lines 14-21. The error of the reward model with respect to these environment states is

$$\bar{\epsilon}_{erwd} = \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T} \gamma^{t-1}\, \mathbf{E}_{(s,a)\sim D_n^t}[|R(s,a) - \hat{R}(s,a)|].$$

For a policy π, let $c_\nu^\pi = \sup_{s,a} \frac{D_{\mu,\pi}(s,a)}{\nu(s,a)}$ represent the mismatch between the discounted state-action distribution under π and the exploration distribution ν. Now, consider the sequence of policies $\hat{\pi}_{1:N}$ generated by H-DAgger-MC. Let $\bar{\pi}$ be the uniform mixture over all policies in the sequence. Let $\bar{\epsilon}_{mc} = \frac{1}{N}\frac{4}{1-\gamma}\sum_{n=1}^{N}\|\bar{Q}_n - \hat{Q}_{T,n}^\rho\|_\infty + \frac{2}{1-\gamma}\|BV_T^\rho - V_T^\rho\|_\infty$ be the error induced by the choice of planning algorithm, averaged over all iterations.

Lemma 7. *In H-DAgger-MC, the policies $\hat{\pi}_{1:N}$ are such that for any policy π,*

$$\begin{aligned}\mathop{\mathbf{E}}_{s\sim\mu}\big[V^\pi(s) - V^{\bar{\pi}}(s)\big] &\le \frac{4}{1-\gamma}c_\nu^\pi \bar{\epsilon}_{hrwd} + \bar{\epsilon}_{mc} \\ &\le \frac{4}{1-\gamma}c_\nu^\pi\Big(\bar{\epsilon}_{erwd} + 2M\sum_{t=1}^{T-1}\gamma^{t-1}\bar{\epsilon}_{prd}^t\Big) + \bar{\epsilon}_{mc}.\end{aligned}$$

Proof. Recall that

$$\mathop{\mathbf{E}}_{s\sim\mu}\big[V^\pi(s) - V^{\bar{\pi}}(s)\big] = \frac{1}{N}\sum_{n=1}^{N}\mathop{\mathbf{E}}_{s\sim\mu}\big[V^\pi(s) - V^{\hat{\pi}_n}(s)\big].$$

and by Lemma 1 for any $n \ge 1$,

$$\mathop{\mathbf{E}}_{s\sim\mu}\big[V^\pi(s) - V^{\hat{\pi}_n}(s)\big] \le \frac{4}{1-\gamma}\mathop{\mathbf{E}}_{(s,a)\sim\xi_\mu^{\pi,\hat{\pi}_n}}[|\hat{Q}_{T,n}^\rho(s,a) - Q_T^\rho(s,a)|] + \bar{\epsilon}_{mc},$$

where

$$\begin{aligned}\xi_\mu^{\pi,\hat{\pi}_n}(s,a) = \frac{1}{2}D_{\mu,\hat{\pi}_n}(s,a) + \frac{1}{4}D_{\mu,\pi}(s,a) \\ + \frac{1}{4}\Big((1-\gamma)\mu(s)\hat{\pi}_n(a\mid s) \\ + \gamma\sum_{z,b} D_{\mu,\pi}(z,b)P_z^b(s)\hat{\pi}_n(a\mid s)\Big).\end{aligned}$$

Then, combining the above with Theorem 5,

$$\begin{aligned}&\frac{1}{N}\sum_{n=1}^{N}\frac{4}{1-\gamma}\mathop{\mathbf{E}}_{(s,a)\sim\xi_\mu^{\pi,\hat{\pi}_n}}[|\hat{Q}_{T,n}^\rho(s,a) - Q_T^\rho(s,a)|] + \bar{\epsilon}_{mc} \\ &\le \frac{1}{N}\sum_{n=1}^{N}\frac{4}{1-\gamma}\sum_{t=1}^{T}\gamma^{t-1}\mathop{\mathbf{E}}_{\substack{(s,\,z,\,a)\\ \sim H^{t,n}_{\xi_\mu^{\pi,\hat{\pi}_n},\rho}}}[\epsilon_{\hat{R}_n}(s,z,a)] + \bar{\epsilon}_{mc}\end{aligned}$$

Now note that for any t and any n,

$$\begin{aligned}&\mathop{\mathbf{E}}_{(s,z,a)\sim H^{t,n}_{\xi_\mu^{\pi,\hat{\pi}_n},\rho}}\big[\epsilon_{\hat{R}_n}(s,z,a)\big] \\ &= \frac{1}{2}\sum_{s',a'}D_{\mu,\hat{\pi}_n}(s',a')\mathop{\mathbf{E}}_{(s,z,a)\sim H^{t,n}_{s',a',\rho}}\big[\epsilon_{\hat{R}_n}(s,z,a)\big] \\ &+ \frac{1}{4}\sum_{s',a'}D_{\mu,\pi}(s',a')\mathop{\mathbf{E}}_{(s,z,a)\sim H^{t,n}_{s',a',\rho}}\big[\epsilon_{\hat{R}_n}(s,z,a)\big] \\ &+ \frac{\gamma}{4}\sum_{s',a'}\sum_{s'',a''}D_{\mu,\pi}(s'',a'')P_{s''}^{a''}(s')\hat{\pi}_n(a'\mid s') \\ &\qquad\mathop{\mathbf{E}}_{(s,z,a)\sim H^{t,n}_{s',a',\rho}}\big[\epsilon_{\hat{R}_n}(s,z,a)\big] \\ &+ \frac{1-\gamma}{4}\sum_{s',a'}\mu(s')\hat{\pi}_n(a'\mid s') \\ &\qquad\mathop{\mathbf{E}}_{(s,z,a)\sim H^{t,n}_{s',a',\rho}}\big[\epsilon_{\hat{R}_n}(s,z,a)\big] \\ &\le \frac{1}{2}\sum_{s',a'}D_{\mu,\hat{\pi}_n}(s',a')\mathop{\mathbf{E}}_{(s,z,a)\sim H^{t,n}_{s',a',\rho}}\big[\epsilon_{\hat{R}_n}(s,z,a)\big] \\ &+ \frac{1}{4}c_\nu^\pi\sum_{s',a'}\nu(s',a')\mathop{\mathbf{E}}_{(s,z,a)\sim H^{t,n}_{s',a',\rho}}\big[\epsilon_{\hat{R}_n}(s,z,a)\big] \\ &+ \frac{\gamma}{4}c_\nu^\pi\sum_{s',a'}\sum_{s'',a''}\nu(s'',a'')P_{s''}^{a''}(s')\hat{\pi}_n(a'\mid s') \\ &\qquad\mathop{\mathbf{E}}_{(s,z,a)\sim H^{t,n}_{s',a',\rho}}\big[\epsilon_{\hat{R}_n}(s,z,a)\big] \\ &+ \frac{1-\gamma}{4}\sum_{s',a'}\mu(s')\hat{\pi}_n(a'\mid s') \\ &\qquad\mathop{\mathbf{E}}_{(s,z,a)\sim H^{t,n}_{s',a',\rho}}\big[\epsilon_{\hat{R}_n}(s,z,a)\big]\end{aligned}$$

$$
\begin{aligned}
&\le c_\nu^\pi \Bigg(\frac{1}{2} \sum_{s',a'} D_{\mu,\hat{\pi}_n}(s',a') \\
&\qquad \mathop{\mathbf{E}}_{(s,z,a)\sim H^{t,n}_{s',a',\rho}} \left[\epsilon_{\hat{R}_n}(s,z,a)\right] \\
&\quad + \frac{1}{4} \sum_{s',a'} \nu(s',a') \mathop{\mathbf{E}}_{(s,z,a)\sim H^{t,n}_{s',a',\rho}} \left[\epsilon_{\hat{R}_n}(s,z,a)\right] \\
&\quad + \frac{\gamma}{4} \sum_{s',a'} \sum_{s'',a''} \nu(s'',a'') P^{a''}_{s''}(s') \hat{\pi}_n(a' \mid s') \\
&\qquad \mathop{\mathbf{E}}_{(s,z,a)\sim H^{t,n}_{s',a',\rho}} \left[\epsilon_{\hat{R}_n}(s,z,a)\right] \\
&\quad + \frac{1-\gamma}{4} \sum_{s',a'} \mu(s') \hat{\pi}_n(a' \mid s') \\
&\qquad \mathop{\mathbf{E}}_{(s,z,a)\sim H^{t,n}_{s',a',\rho}} \left[\epsilon_{\hat{R}_n}(s,z,a)\right] \\
&= c_\nu^\pi \mathop{\mathbf{E}}_{(s,z,a)\sim H^{t,n}_{\xi_n,\rho}} \left[\epsilon_{\hat{R}_n}(s,z,a)\right].
\end{aligned}
$$

When $t = 1$,

$$
\mathop{\mathbf{E}}_{(s,z,a)\sim H^{t,n}_{\xi_n,\rho}} \left[\epsilon_{\hat{R}_n}(s,z,a)\right] = \mathop{\mathbf{E}}_{(s,a)\sim \xi_n(s,a)} \left[\epsilon_{\hat{R}_n}(s,z,a)\right].
$$

When $t > 1$,

$$
\begin{aligned}
&\mathop{\mathbf{E}}_{(s,z,a)\sim H^{t,n}_{\xi_n,\rho}} \left[\epsilon_{\hat{R}_n}(s,z,a)\right] \\
&= \sum_{s_t,z_t,a_t} \mathop{\mathbf{E}}_{(s_1,a_1)\sim\xi_n} \Bigg[\sum_{a_{1:t-1}} \rho(a_{2:t} \mid a_1) \\
&\qquad P^{a_{0:t-1}}_{s_1}(s_t \mid s_1, a_{0:t-1}) \hat{P}^{1:t-1}_n(z_t \mid s_1, a_{0:t-1}) \\
&\qquad\qquad \epsilon_{\hat{R}_n}(s_t,z_t,a_t) \Bigg] \\
&= \mathop{\mathbf{E}}_{(s,z,a)\sim H^t_n} \left[\epsilon_{\hat{R}_n}(s,z,a)\right].
\end{aligned}
$$

Thus, putting it all together, we have shown that

$$
\begin{aligned}
&\mathop{\mathbf{E}}_{s\sim\mu} \left[V^\pi(s) - V^{\bar{\pi}}(s)\right] \\
&\le \frac{4}{1-\gamma} c_\nu^\pi \frac{1}{N} \sum_{n=1}^N \sum_{t=1}^T \gamma^{t-1} \mathop{\mathbf{E}}_{(s,z,a)\sim H^t_n} \left[\epsilon_{\hat{R}_n}(s,z,a)\right] \\
&\qquad\qquad + \bar{\epsilon}_{mc} \\
&= \frac{4}{1-\gamma} c_\nu^\pi \bar{\epsilon}_{hrwd} + \bar{\epsilon}_{mc}.
\end{aligned}
$$

Thus we have proven the first inequality. Furthermore, by Theorem 6,

$$
\begin{aligned}
\bar{\epsilon}_{hrwd} &= \frac{1}{N} \sum_{n=1}^N \sum_{t=1}^T \mathop{\mathbf{E}}_{(s,z,a)\sim H^t_n} \left[\hat{R}_n(s,z,a)\right] \\
&\le \frac{1}{N} \sum_{n=1}^N \Bigg(\sum_{t=1}^T \gamma^{t-1} \mathop{\mathbf{E}}_{(s,a)\sim D^t_n} \left[|R(s,a) - \hat{R}_n(s,a)|\right] \\
&\qquad + 2M \sum_{t=1}^{T-1} \gamma^{t-1} \mathop{\mathbf{E}}_{(s,z,a)\sim H^t_n} \left[1 - \hat{P}^t_n(\sigma^a_s \mid z,a)|\right] \Bigg) \\
&= \frac{1}{N} \sum_{n=1}^N \sum_{t=1}^T \gamma^{t-1} \mathop{\mathbf{E}}_{(s,a)\sim D^t_n} \left[|R(s,a) - \hat{R}_n(s,a)|\right] \\
&\quad + 2M \sum_{t=1}^{T-1} \gamma^{t-1} \frac{1}{N} \sum_{n=1}^N \mathop{\mathbf{E}}_{(s,z,a)\sim H^t_n} \left[1 - \hat{P}^t_n(\sigma^a_s \mid z,a)|\right] \\
&\le \bar{\epsilon}_{erwd} + 2M \sum_{t=1}^{T-1} \gamma^{t-1} \bar{\epsilon}^t_{prd}.
\end{aligned}
$$

This gives the second inequality. □

Note that this result holds for *any* comparison policy π. Thus, if $\bar{\epsilon}_{mc}$ is small and the learned models have low error, then if ν is similar to the state-action distribution under *some* good policy, $\bar{\pi}$ will compare favorably to it. That said, Lemma 7 shares the limitations of the comparable results for the other DAgger algorithms. It focuses on the L1 loss, which is not always a practical learning objective. It also assumes that the expected loss at each iteration can be computed exactly (i.e. that there are infinitely many samples per iteration). It also applies to the average policy $\bar{\pi}$, rather than $\hat{\pi}_N$. Ross & Bagnell (2012) discuss extensions that address more practical loss functions, finite sample bounds, and results for $\hat{\pi}_N$.

Lemma 7 effectively says that *if* the models have low training error, the resulting policy will be good. It does not promise that the models will have low training error. Following Ross & Bagnell (2012) note that $\bar{\epsilon}^t_{prd}$ and $\bar{\epsilon}_{hrwd}$ can each be interpreted as the average loss of an online learner on the problem defined by the aggregated datasets. Then for each horizon depth t let $\bar{\epsilon}^t_{\mathcal{P}}$ be the error of the best dynamics model in $\mathcal{P}$ under the training distribution at that depth, in retrospect. Specifically,

$$
\bar{\epsilon}^t_{\mathcal{P}} = \inf_{P'\in\mathcal{P}} \frac{1}{N} \sum_{n=1}^N \mathop{\mathbf{E}}_{(s,z,a)\sim H^t_n} [1 - P'(\sigma^a_s \mid z,a)].
$$

Similarly, let

$$
\bar{\epsilon}_{\mathcal{R}} = \inf_{R'\in\mathcal{R}} \frac{1}{N} \sum_{n=1}^N \sum_{t=1}^T \gamma^{t-1} \mathop{\mathbf{E}}_{(s,z,a)\sim H^t_n} [\epsilon_{R'}(s,z,a)]
$$

be the error of the best reward model in $\mathcal{R}$ in retrospect.

The average regret for the dynamics model at depth t is $\bar{\epsilon}^t_{prgt} = \bar{\epsilon}^t_{prd} - \bar{\epsilon}^t_{\mathcal{P}}$. For the reward model it is $\bar{\epsilon}_{rrgt} = \bar{\epsilon}_{hrwd} - \bar{\epsilon}_{\mathcal{R}}$. For a no-regret online learning algorithm, average regret approaches 0 as $N \to \infty$. This gives the following bound on H-DAgger-MC's performance in terms of model regret.

Theorem 8. *In H-DAgger-MC, the policies $\hat{\pi}_{1:N}$ are such that for any policy π,*

$$\begin{aligned}
&\mathop{\mathbf{E}}_{s\sim\mu}\left[V^{\pi}(s) - V^{\bar{\pi}}(s)\right] \\
&\leq \frac{4}{1-\gamma} c^{\pi}_{\nu}(\bar{\epsilon}_{\mathcal{R}} + \bar{\epsilon}_{rrgt}) + \bar{\epsilon}_{mc} \\
&\leq \frac{4}{1-\gamma} c^{\pi}_{\nu}\left(\bar{\epsilon}_{erwd} + 2M \sum_{t=1}^{T-1} \gamma^{t-1}(\bar{\epsilon}^t_{\mathcal{P}} + \bar{\epsilon}^t_{prgt})\right) + \bar{\epsilon}_{mc}
\end{aligned}$$

and if the learning algorithms are no-regret then as $N \to \infty$, $\bar{\epsilon}_{rrgt} \to 0$ and $\bar{\epsilon}^t_{prgt} \to 0$ for each $1 \leq t \leq T-1$.

Theorem 8 says that if $\mathcal{R}$ contains a low-error reward model relative to the learned dynamics models then, as discussed above, if $\bar{\epsilon}_{mc}$ is small and ν visits important states, the resulting policy will yield good performance. If $\mathcal{P}$ and $\mathcal{R}$ contain perfect models, $\bar{\pi}$ will be comparable to the plan generated by the perfect model.

As noted by Talvitie (2017), this result does *not* promise that H-DAgger-MC will eventually achieve the performance of the best available set of dynamics models. The model at each rollout depth is trained to minimize prediction error given the input distribution provided by the shallower models without regard for the effect on deeper models. It is possible that better overall error could be achieved by *increasing* the prediction error at one depth in exchange for a favorable state distribution for deeper models. Similarly, as discussed in Section 4, H-DAgger-MC will not necessarily achieve the performance of the best available combination of dynamics and reward models. The dynamics model is trained without regard for the impact on the reward model. It could be that a dynamics model with higher prediction error would allow for lower hallucinated reward error. H-DAgger-MC does not take this possibility into account.

References

Ross, S. and Bagnell, D. Agnostic system identification for model-based reinforcement learning. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, pp. 1703–1710, 2012.

Talvitie, E. Agnostic system identification for monte carlo planning. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI)*, pp. 2986–2992, 2015.

Talvitie, E. Self-correcting models for model-based reinforcement learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI)*, pp. 2597–2603, 2017.