1707.05300 - Reverse Curriculum Generation for Reinforcement Learning

- **Yunqiu Xu**

- 在Deep RL Bootcamp上看到的一篇文章, 为了解决sparse reward问题, 使用反向课程学习, 从目标开始倒推

---

1. Introduction

- Challenging:
  - Sparse reward $\rightarrow$ hard for learning-based approaches and non-sparse reward functions
  - The use of demonstrations $\rightarrow$ requires an expert intervention
- Key insights:
  - It's easier to reach the goal from states nearby the goal
  - Applying random actions from such states makes the agent go to new feasible nearby states, thus easier to reach the goal
- Our method:
  - **Do not use reward engineering and demonstrations**
  - Requires no prior knowledge of the task, only need to provide the final state (target position)
  - Train the robot to reach the goal which the start position is nearby the goal
  - Then train it from further start position
  - How to choose start position:
    - Perform random walk from previous start states
    - You can get reward by starting from these states: can reach final state
    - But these are not best start states: require more training

2. Related Work on Curriculum Learning

- Reject examples which is too hard currently:
  - Applied in SL and RL with pre-specified task sequences
  - Few implementations, only preliminary tasks
- Intrinsic motivation based on learning progress:
  - Obtain "developmental trajectories"
  - Requires iteratively partitioning the full task space
- Use baseline performance of easy tasks to gauge hard tasks
  - Can only handle finite sets of tasks
  - Requires each task to be learnable on its own
- Our method:
  - Train a policy that can generalize to continuousl parameterized tasks
  - Perform well under sparse rewards, do not allocate training effore to tasks

3. Problem Definition

- Learn a policy that leads a system into a specified goal space, from any start state sampled from a given distribution.
  - A large set of start states S^0 : more robust than using only one start state, avoid undesired deviations from intended trajectory
  - A small set of goals S^g : goal, as well as its nearby states

$$R(\pi, s_0) = \mathbb{E}_{\pi(\cdot|s_t)} \mathbb{1}\left\{ \bigcup_{t=0}^{T} s_t \in S^g | s_0 \right\} = \mathbb{P}\left(\bigcup_{t=0}^{T} s_t \in S^g \,\middle|\, \pi, s_0 \right)$$

- Assumptions for reverse curriculum generation
  - We can arbitrarily reset the agent into any start state $s^0 \in S$ at the beginning of all trajectories.
  - S^g is not empty $\rightarrow$ at least one goal state
  - The Markov Chain induced by taking uniformly sampled random actions has a communicating class including all start states S^0 and the given goal state s^g

4. Methodology

Algorithm 1: Policy Training

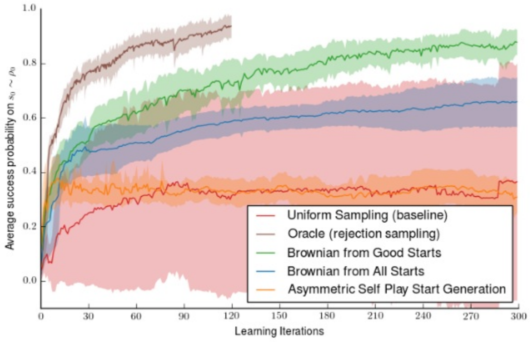
Input : $\pi_0, s^g, \rho_0, N_{\text{new}}, N_{\text{old}}, R_{\min}, R_{\max}, Iter$
Output: Policy π_N
$starts_{\text{old}} \leftarrow [s^g]$;
$starts,\ rews \leftarrow [s^g],\ [1]$;
for $i \leftarrow\ 1$ **to** $Iter$ **do**
  $starts \leftarrow \texttt{SampleNearby}(starts,\ N_{\text{new}})$;
  $starts$.append[sample$(starts_{\text{old}}, N_{\text{old}})$];
  $\rho_i \leftarrow \text{Unif}(starts)$;
  $\pi_i,\ rews \leftarrow \texttt{train\_pol}(\rho_i,\ \pi_{i-1})$;
  $starts \leftarrow \texttt{select}(starts,\ rews, R_{\min}, R_{\max})$;
  $starts_{\text{old}}$.append[$starts$];
  $\texttt{evaluate}(\pi_i, \rho_0)$;
end

Procedure 2: `SampleNearby`

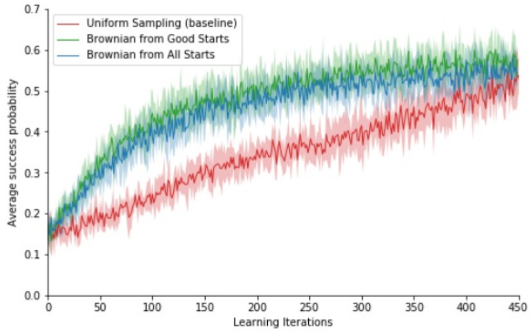
Input : $starts, N_{\text{new}}, \Sigma, T_B, M$
Output: $starts_{\text{new}}$
while len$(starts) < M$ **do**
  $s_0 \sim \text{Unif}(starts)$;
  for $t \leftarrow 1$ **to** T_B **do**
    $a_t = \epsilon_t,\ \ \epsilon_t \sim \mathcal{N}(0, \Sigma)$;
    $s_t \sim \mathcal{P}(s_t | s_{t-1}, a_t)$;
    $starts$.append(s_t);
  end
end
$starts_{\text{new}} \leftarrow$ sample$(starts, N_{\text{new}})$

5. Experimental Results

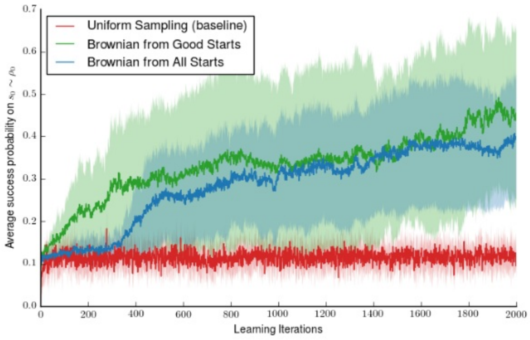
- Questions:
  - Does the performance of the policy on the target start state distribution ρ_0 improve by training on distributions ρ_i growing from the goal?
  - Does focusing the training on "good starts" speed up learning?
  - Is Brownian motion a good way to generate "good starts" from previous "good starts"?
- Tasks: http://bit.ly/reversecurriculum
  - Point-mass maze
  - Ant maze
  - Ring on Peg task
  - Key insertion task
- Results:



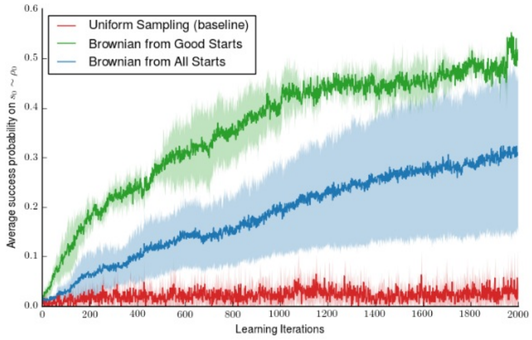
(a) Point-mass Maze task



(b) Ant Maze task

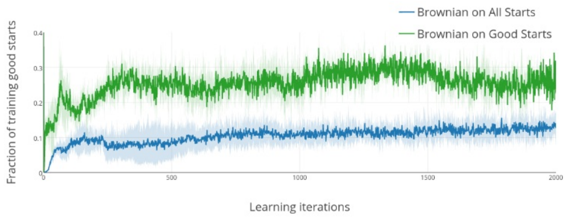


(c) Ring on Peg task

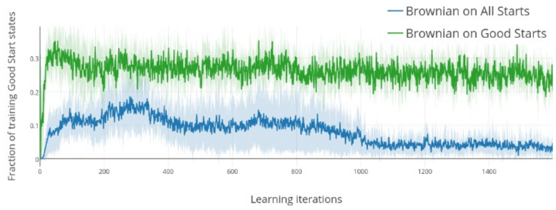


(d) Key insertion task

Figure 2: Learning curves for goal-oriented tasks (mean and variance over 5 random seeds).



(a) Key insertion task



(b) Ring on Peg task

Figure 3: Fraction of Good Starts generated during training for the robotic manipulation tasks

6. Conclusion and Future Work

- Conclusion:
  - Propose a method to automatically adapt the start state distribution on which an agent is trained
  - If 3 assumptions are satisfied, hard goal-oriented problems can be tackled
- Future work:
  - Combine curriculum-generation with goal generation (1705.06366 - Automatic goal generation for reinforcement learning agents)

- Combine curriculum-generation with domain randomization (1703.06907 - Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World) → policy can be transferred to real world