ACL for DRL - Review IJCAI 2020

Saturday, 10 October 2020 13:50

- Automatic Curriculum Learning (ACL):
  used with DRL to shape the learning trajectories of agents by challenging them with tasks adapted to their capacities.
- ACL can be used to:
  1. improve sample efficiency and asymptotic performance.
  2. organize exploration.
  3. encourage generalization.
  4. solve sparse reward problems.
- Introduction (History of Curriculum learning):
  Huma learning → learning scheme by Selfridge '85
  → using artificial curiosity Schmidhuber '91
  → in supervised learning → organize the presentation of data. → organize the growth in model cap,
  → in robotic → learning progress
  (self organize open-ended development trajectories of learning agents)
- ACL for DRL:
  It is a family of mechanisms that automatically adapt the distributions of training data by learning to adjust the selection of learning situations to the capabilities of DRL agents.
- Connections with other fields:
  - ACL can be used in the context of Transfer learning
  - ACL ≠ Continual learning (trains to be robust to unforseen changes)
    ↳ controls the learning scenario.
  - ACL + Policy Distillation for multi-task RL settings.
- Multi-task DRL problems:
  agents are trained on tasks sampled from a task space. $(T \sim \mathcal{T})$
- Multi-goal DRL:
  policies and reward functions are conditioned on goals.
- ACL proposes → learn a task selection function
  $\mathcal{D}: \mathcal{H} \to \mathcal{T}$ (uses information to select from tasks)

  with an objective:

$$Obj: \max_{\mathcal{D}} \int_{T \sim \mathcal{T}_{target}} P_T^N \, dT$$

  - tune the parameters of the task selection function to maximize the sum (integral) over all tasks sampled from a target space $(T \sim \mathcal{T}_{target})$ of a performance quantifies the agent's behaviour on each task T after N training step.
  - the metric P could be cumulative reward; exploration score.
- Classifican of ACL:

  ACL
  - by purpose
  - Learning problem
    - for Data Collection
    - for Data Exploration
  - optimization

- Purposes:
  1. Improving performance on a restricted data-set.
     - Prioritized experience replay [Schaul '15]
     - Distributed " [Horgan '18]
     - Samples are not useful: Denoising PG [Flet '19]
  2. Solving hard tasks:
     - hard task → + auxiliary tasks.
     - Scheudle DRL agents from simple to hard
       Teacher-student curriculum learning [Matiisen '17]
     - from close-to-success to challenging
       Reverse curriculum generation for RL [Florensa '17]
     - organize the exploration of the state space so as to solve sparse reward problems.
     - performance reward is augmented with a intrinsic reward guiding the agent towards uncertain areas of the state space.
       Unifying count-based exploration [Bellemare '16]
       Model-based active exploration [Shyam '18]
  3. Training generalist agents:
     - Solve tasks haven't been encountered during training.
       by ① shaping learning trajectories
       Teacher algorithms for CL of DRL [Portelas '19]
       ② Sim2Real
       Solving rubic's cube [Open AI '19]
       ③ Self-play in multi-agent settings
       Mastering the game of Go [Silver '17]
  4. Training multi-goal agents:
     - learn a behavioral repertoire through one or several goal conditioned policies.
       Hindsight Experience Replay [Andrychowicz '17]
       CURIOUS [Colas '19]
  5. Organizing open-ended exploration:
     - ACL is used to organize the discovery and acquisition of repertoires of robust and diverse behaviors; from ① visual observations
       Diversity is all you need [Eysenbash '18]
       or ② natural language interactions
       Language grounding through social... [Liar '19]

- Learning problem:
  1. for Data Collection:

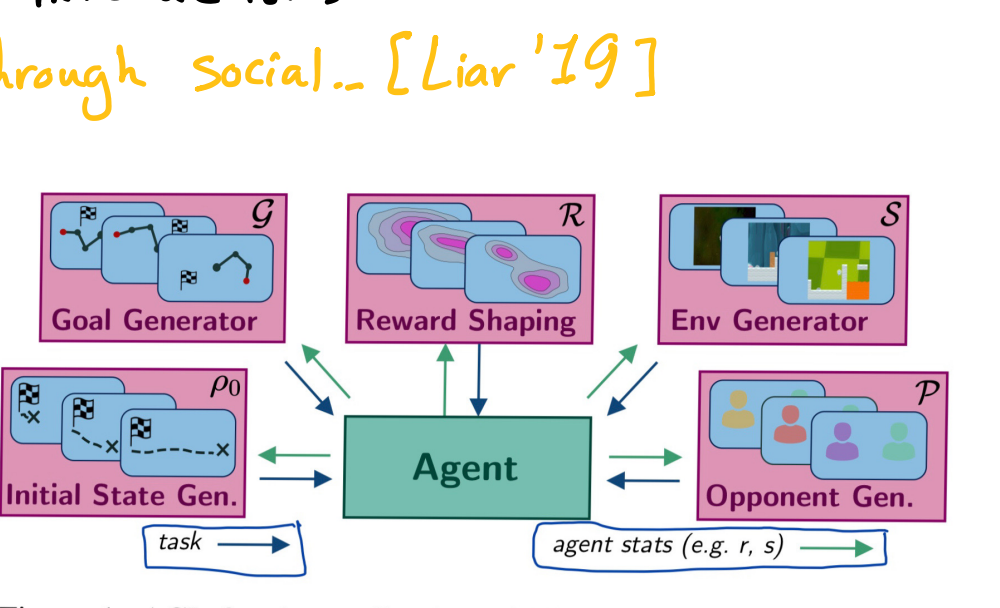

Figure 1: ACL for data collection. ACL can control each elements of task MDPs to shape the learning trajectories of agents. Given metrics of the agent's behavior like performance or visited states, ACL methods generate new tasks adapted to the agent's abilities.

     a. Initial state (ρ_0):
        Florens '19
     b. reward function (R):
        ① curiosity-based:
        (go to high-uncertainty)
        Bellemare '16
        ② accuracy-based (low accuracy → high)
        Accuracy-based CL in DRL [Fournier '18]
        ③ diversity-based (learn skill space in unsupervised setting)
        Eysenbash '18
     c. Goals (G) multi-goal RL;
        discrete, continuous or mix goal spaces.
     d. Environments (S, P):
        ① organize the selection of environments from a discrete set
        Reward guided curriculum for robust RL [Mysore '18]
        ② generate rich task spaces
        Procedural content generation [Risi '19]
     e. Opponents (S, P): self-play algorithms
        Robust adversarial RL [Pinto '17]
  2. for Data Exploration:
     - Acting on training data previously collected and stored in replay memory Experience replay [Lin '92]
     - two types of control on the distribution of training data:
       a. Transition selection $(S \times A)$:
          Some transitions might be more informative than others.
          Prioritized sweeping [Moore '93]
          Prioritized experience replay [Scaul '15]
          can be used for on-policy algorithms to filter batches
          Samples are not all useful [Flet '19]
       b. Transition modification (G)
          HER [Andryehowicz '17]
          shifts the data distribution from simple goals toward more complex goals as the agent makes progress.

- Optimization:
  the objective is hard to optimize directly
  → ACL methods uses surrogate objectives
  1. Rewards:
     - rewarded transitions are usually considered as more informative than others; especially in sparse reward scenarios
       ① transition selection → increase the ratio of high rewards
       RL with unsupervised auxiliary tasks [Jaderberg '16]
       ② multi-goal → balance the proportion of positive rewards for each of the goals. CURIOUS
       ③ transition modification → substituting goals to increase the probability of observing rewards. HER
       ④ data collection → adapt training distribution toward more rewarded experience ✗
       focus on solved tasks so we use other objectives
  2. Intermediate difficulty:
     - agent should target tasks that are neither too easy nor too difficult to maximize learning progress.
     - GoalGAN [Florensa '18]
     - Judge network AC through setter-solver [Racaniere '19]
     - Intrinsic motivation and AC via self-play [Sukhbaatar '17]
     - Sequence domain randomization [Open AI '19]
  3. Learning progress (LP)
     - the difference between the final score and the initial score.
     - LP ⇔ intermediat difficulty.
       ① usually formed as a multi-arm bandit (MAB).
       assumption of concave learning profiles
       The strategic student approach [Lopes '12]
       ② LP as estimated derivative of the performance
       Teache-student CL [Matiisen '17]
       ③ absolute LP (redirects learning to forgotten tasks)
       CURIOUS [Colas '19]
  4. Diversity:
     - maximize measures of diversity (novelty or low density)
     - multi-goal → goals from low density areas.
     - single-task → count-based (visitation count)
     - update latent space representation using VEM
       Unsupervised C for visual meta-learning [Jabri '19]
  5. Surprise:
     - models tend to give bad predictions for states rarely visited, inducing a bias toward less visited states.
     - compute intrinsic rewards based on:
       ① prediction error Curiosity-driven exploration [Pathak '17]
       ② disagreement (variance) between several models from an ensemble Model-based active exploration [Shyam '18]
     - bias the sampling of transition for policy update depending on their TD-error PER [Shaul '15]
     - ACL mechanisms favor states related to maximum surprise; i.e. a maximal difference between the expected and truth.
  6. Energy:
     priorities transitions from high-energy trajectories or those where the goal moved.
  7. Adversarial reward maximization (ARM)
     ① self-play with current or past versions
     A generalized framework for self-play [Hernandez '19]
     ② with copies Go [Silver '17]
     ③ train in parallel
     Emergent complexity via multi-agent competition [Bansal '17]