McGill

Towards Decomposable Rewards with Generative Adversarial Inverse Reinforcement Learning

Peter Henderson | Wei-Di Chang

Background

- Direct vs Indirect Inverse Reinforcement Learning
- Want to learn a reward function and policy without knowing expert's actions
- Apprenticeship Learning via Reinforcement Learning

  - Pieter Abbeel and Andrew Y. Ng. "Apprenticeship learning via inverse reinforcement learning." *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004.

- Generative Adversarial Imitation Learning

  - Jonathan Ho and Stefano Ermon. "Generative adversarial imitation learning." *Advances in Neural Information Processing Systems*. 2016.

- Guided Cost Learning

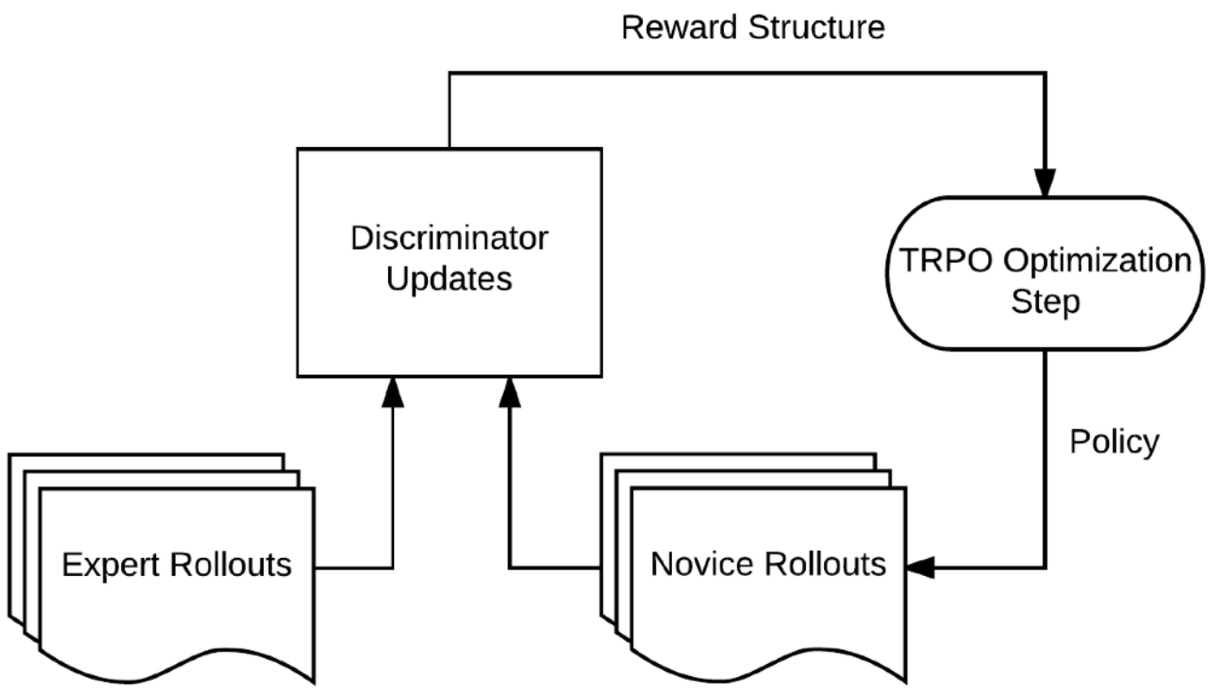
  - Chelsea Finn, Sergey Levine, and Pieter Abbeel. "Guided cost learning: Deep inverse optimal control via policy optimization." *Proceedings of the 33rd International Conference on Machine Learning*. Vol. 48. 2016.
  - Chelsea Finn, et al. "A Connection between Generative Adversarial Networks, Inverse Reinforcement Learning, and Energy-Based Models." *arXiv preprint arXiv:1611.03852* (2016).

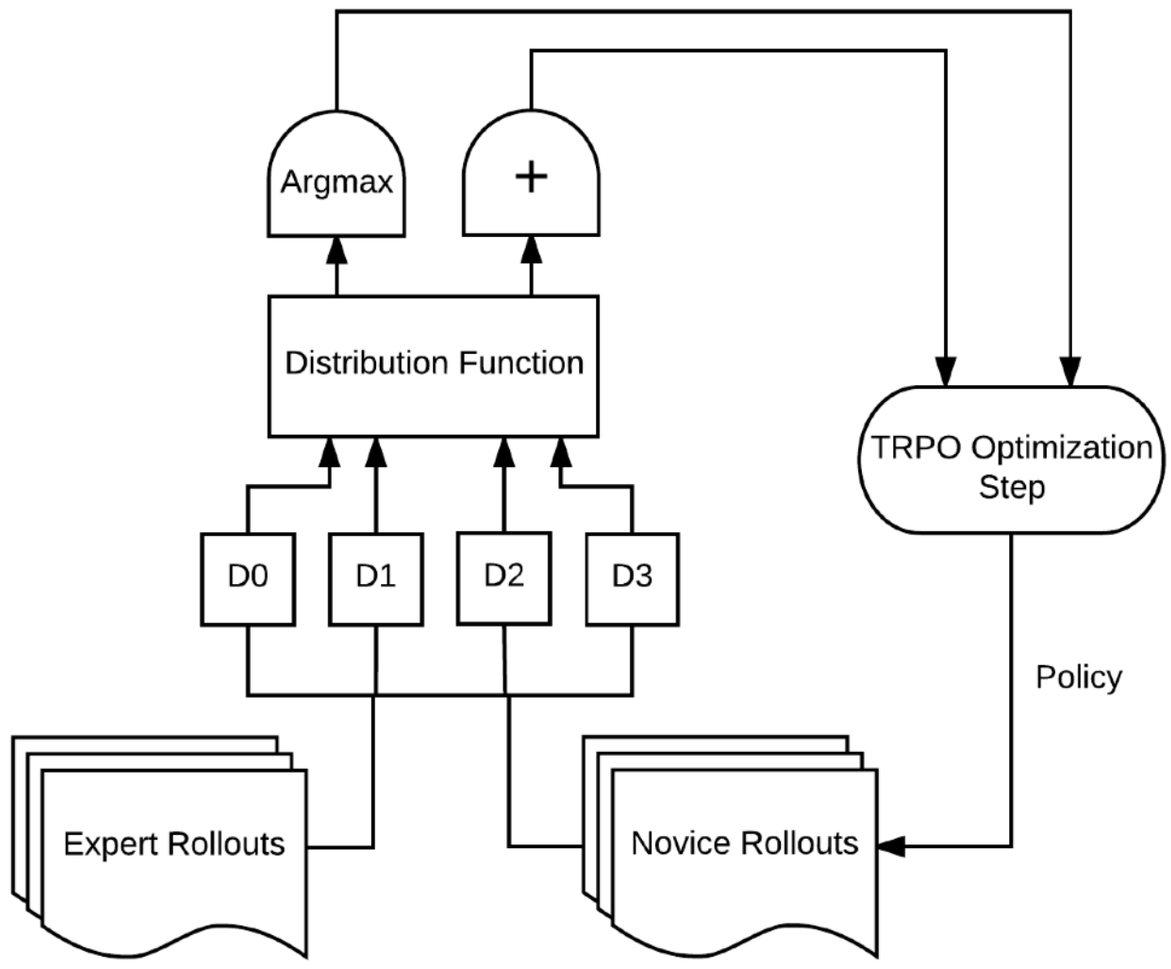
Generative Adversarial Inverse Reinforcement Learning

- We propose a framework similar to Ho and Ermon's Generative Adversarial Imitation Learning
  - We do not know actions in advance (Indirect IRL)
  - Similar to Third Person Imitation Learning
  - We use states instead of pixels for simplicity
  - Third Person Imitation Learning paper not very reproducible.
  - *"extremely high variance, and some of the experiments are averaged over 10-1000 trials to achieve stable learning"* - From the authors
- As part of this framework we take a step toward formulating "decomposable" rewards (optioned-rewards or mixtures-of-rewards)
  - Inspired by learning rewards for multi-step actions, the options framework, and mixtures of experts
  - How can we learn different reward functions that decompose (as in multi-step tasks) as part of the generative adversarial IRL?

- *Sermanet, Pierre, Kelvin Xu, and Sergey Levine. "Unsupervised Perceptual Rewards for Imitation Learning." arXiv preprint arXiv:1612.06699 (2016).*
- *Sutton, Richard S., Doina Precup, and Satinder Singh. "Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning." Artificial intelligence 112.1-2 (1999): 181-211.*
- *Bacon, Pierre-Luc, Jean Harb, and Doina Precup. "The option-critic architecture." arXiv preprint arXiv:1609.05140 (2016).*
- *Jacobs, Robert A., et al. "Adaptive mixtures of local experts." Neural computation 3.1 (1991): 79-87.*
- *Shazeer, Noam, et al. "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer." arXiv preprint arXiv:1701.06538 (2017).*
- *Stadie, Bradly C., Pieter Abbeel, and Ilya Sutskever. "Third-Person Imitation Learning." arXiv preprint arXiv:1703.01703 (2017).*

Generative Adversarial Inverse Reinforcement Learning





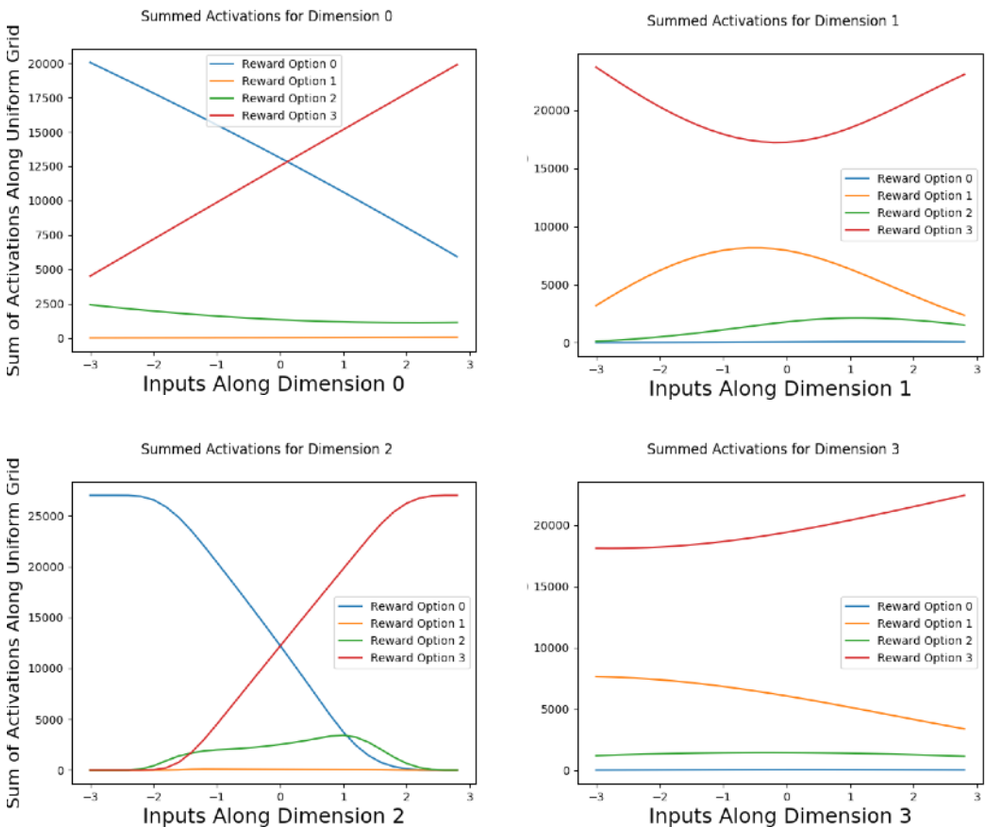
$$L(R_\Omega) = \sum_\omega \zeta_\omega L(R_\omega) + w_{importance} \cdot CV(\sum_n D(S_n))^2$$

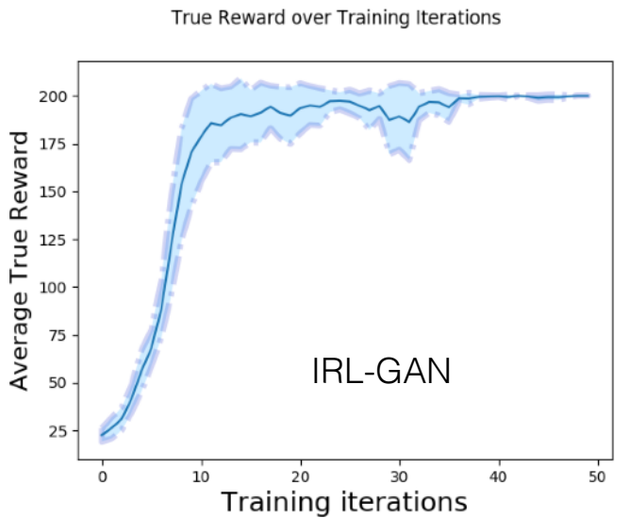
$$L(R_\omega) = \frac{-1}{n} \sum_{n=1}^{N} [p_n log\sigma(D_\omega(s_n)) + (1-p_n) log(1-\sigma(D_\omega(s_n)))]$$

Initial Results

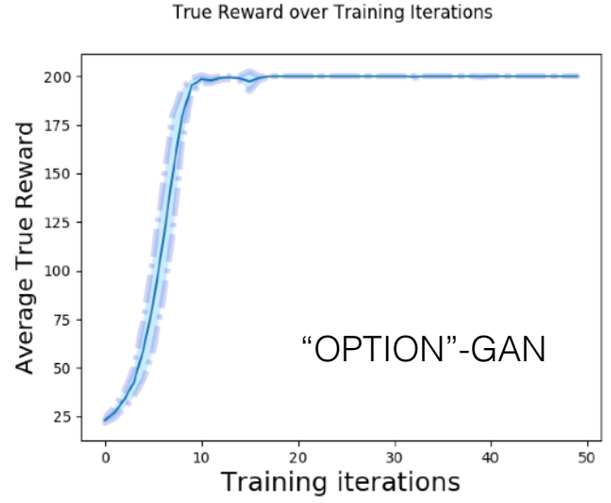
Task	Expert	Behavioral Cloning	IRL-GAN	MixGan	OptionGan
CartPole	200 ± 0.0	200 ± 0.0	200 ± 0.0	200 ± 0.0	200 ± 0.0
MountainCar	-100.5 ± 11.6	-111.95 ± 9.9	-121.68 ± 11.1	-118.72 ±12.8	still running

Final Learned Policy Across Rollouts (20 rollouts CartPole, 200 rollouts MountainCar)
Parameters CartPole (10 experts, 20 sample rollouts per iteration, 30 iterations, importance coefficient .25)
Parameters MountainCar (50 expert rollouts, 100 sample rollouts, 200 iterations, importance coefficient .25)
Note: MountainCar results are much more unstable and may vary across experiments. Number of Iterations until convergence may also vary, graphs here represent average across 5 experiments. Importance Weighting .25.

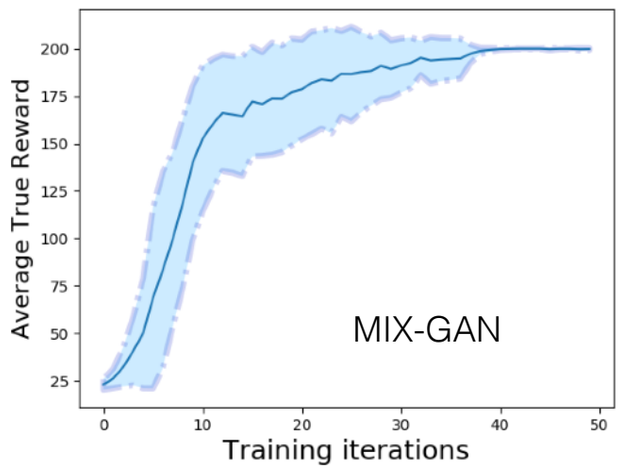




True Reward over Training Iterations



True Reward over Training Iterations



What's Next

- Formalize the decomposable rewards framework
  - Combine with policy options to have policy-reward options
- Try to address stability and variance issues with the GAN-style framework
  - Apply parameter variation that we found to help qualitatively in a more formalized way
- Benchmark on more complex environments
  - Mujoco tasks
  - Tasks with clear subtasks