Learning to Optimize as Policy Learning

Yisong Yue
Policy Learning (Reinforcement & Imitation)

**Goal:** Find “Optimal” Policy

**Imitation Learning:**
Optimize imitation loss

**Reinforcement Learning:**
Optimize environmental reward

**Learning-based Approach for Sequential Decision Making**
Basic Formulation

(Typically a Neural Net)

- Policy: \( \pi(s) \rightarrow P(a) \)

- Roll-out: \( \tau = \langle s_0, a_0, s_1, a_1, s_2, \ldots \rangle \) (aka trace or trajectory)

- Objective: \( \sum_i r(s_i, a_i) \)
Optimization as Sequential Decision Making

- Many Solvers are Sequential
  - Tree-Search
  - Greedy
  - Gradient Descent

- Can view solver as “agent” or “policy”
  - State = intermediate solution
  - Find a state with high reward (solution)
  - Learn better local decision making
Example #1: Learning to Search (Discrete)

**Integer Program**

\[
\text{max } - \sum_{i=1}^{5} x_i, \\
\text{subject to:} \\
x_1 + x_2 \geq 1, \\
x_2 + x_3 \geq 1, \\
x_3 + x_4 \geq 1, \\
x_3 + x_5 \geq 1, \\
x_4 + x_5 \geq 1, \\
x_i \in \{0, 1\}, \forall i \in \{1, \ldots, 5\}
\]

[He et al., 2014][Khalil et al., 2016] [Song et al., arXiv]
Example #1: Learning to Search (Discrete)

**Integer Program**

\[
\max \ - \sum_{i=1}^{5} x_i, \\
\text{subject to:} \\
x_1 + x_2 \geq 1, \\
x_2 + x_3 \geq 1, \\
x_3 + x_4 \geq 1,
\]

- Deterministic State Transitions
- Massive State Space
- Sparse Rewards

[He et al., 2014] [Khalil et al., 2016] [Song et al., arXiv]
Example #2: Learning Greedy Algorithms (discrete)

**Contextual Submodular Maximization:**

- Greedy Sequential Selection:
  - \( \Psi \leftarrow \Psi \oplus \arg\max_a F_x(\Psi \oplus a) \)

- Train policy to mimic greedy:
  - \( \pi(s) \rightarrow a \)

**Dictionary of Trajectories**

**Select Diverse Set**

\[ \arg\max_{\Psi:|\Psi| \leq B} F_x(\Psi) \]

State \( s = (\Psi, x) \)

Learning Policies for Contextual Submodular Prediction S. Ross, R. Zhou, Y. Yue, D. Dey, J.A. Bagnell. ICML 2013
Example #2: Learning Greedy Algorithms (discrete)

Contextual Submodular Maximization:

$\arg\max_{\Psi:|\Psi| \leq B} F_x(\Psi)$

- Greedy Sequential Selection:
  - $\Psi \leftarrow \Psi \oplus \arg\max_a F_x(\Psi \oplus a)$

- Train policy to mimic greedy:
  - $\pi(s) \rightarrow a$

State $s = (\Psi, x)$

- Deterministic State Transitions
- Massive State Space
- Dense Rewards
- Note: Not Learning Submodular Function

Learning Policies for Contextual Submodular Prediction S. Ross, R. Zhou, Y. Yue, D. Dey, J.A. Bagnell. ICML 2013
Example #3: Iterative Amortized Inference (continued)

Gradient Descent Style Updates:
- State = description of problem & current point
- Action = next point

Useful for Accelerating Variational Inference

Iterative Amortized Inference, Joe Marino, Yisong Yue, Stephan Mandt. ICML 2018
Example #3: Iterative Amortized Inference (continued)

Gradient Descent Style Updates:

• State = description of problem & current point
• Action = next point

- ( Mostly) Deterministic State Transitions
- Continuous State Space
- Dense Rewards
- Simplest Case: One-Shot Inference
  • "Variational Autoencoders" [Kingma & Welling, ICLR 2014]

Useful for Accelerating Variational Inference

Iterative Amortized Inference, Joe Marino, Yisong Yue, Stephan Mandt. ICML 2018
Optimization as Sequential Decision Making

Learning to Search
- Discrete Optimization (Tree Search), Sparse Rewards
- Learning to Search via Retrospective Imitation [arXiv]
- Co-training for Policy Learning [UAI 2019]

Contextual Submodular Maximization
- Discrete Optimization (Greedy), Dense Rewards
- Learning Policies for Contextual Submodular Prediction [ICML 2013]

Learning to Infer
- Continuous Optimization (Gradient-style), Dense Rewards
- Iterative Amortized Inference [ICML 2018]
- A General Method for Amortizing Variational Filtering [NeurIPS 2018]
Optimization as Sequential Decision Making

Learning to Search
- Discrete Optimization (Tree Search), Sparse Rewards
- Learning to Search via Retrospective Imitation [arXiv]
- Co-training for Policy Learning [UAI 2019]

Contextual Submodular Maximization
- Discrete Optimization (Greedy), Dense Rewards
- Learning Policies for Contextual Submodular Prediction [ICML 2013]

Learning to Infer
- Continuous Optimization (Gradient-style), Dense Rewards
- Iterative Amortized Inference [ICML 2018]
- A General Method for Amortizing Variational Filtering [NeurIPS 2018]
Learning to Optimize for Tree Search

• Idea #1: Treat as Standard RL

• Randomly explore for high rewards
  • Very hard exploration problem!

• Issues: massive state space & sparse rewards
Learning to Optimize for Tree Search

• Idea #2: Treat as Standard IL

• Convert to Supervised Learning
  • Assume access to solved instances

  "Demonstration Data"

• Training Data: $D_0 = \{(\text{\text{tree}}, \text{\text{target}})\}$

• Basic IL: $\arg\min_{\pi \in \Pi} L_{D_0}(\pi) \equiv E_{(s,a) \sim D_0} [\ell(a, \pi(s))]$

Behavioral Cloning
Challenges w/ Imitation Learning

• Issues with Behavioral Cloning
  • Minimize $L_{D_0}$ ... implications?
  • If $\pi$ makes a mistake early, subsequent state distribution $\approx D_0$ ??
  • Some extensions to Interactive IL [He et al., NeurIPS 2014]

  Our Approach is also Interactive IL

• Demonstrations not Available on Large Problems
  • How to (formally) bootstrap from smaller problems?
  • Bridging the gap between IL & RL

  Our Approach gives one solution
Retrospective Imitation

• Given:
  • Family of Distributions of Search problems
    • Family is parameterized by size/difficulty
  • Solved Instances on the Smallest/Easiest Instances
    • “Demonstrations”

• Goal:
  • Interactive IL approach
  • Can Scale up from Smallest/Easiest Instances
  • Formal Guarantees

Learning to Search via Retrospective Imitation, Jialin Song, Ravi Lanka, et al., arXiv
Retrospective Imitation

- Two-Stage Algorithm

- Core Algorithm
  - Fixed problem difficulty
  - Reductions to Supervised Learning

- Full Algorithm w/ Scaling Up
  - Uses Core Algorithm as Subroutine

Learning to Search via Retrospective Imitation, Jialin Song, Ravi Lanka, et al., arXiv
Retrospective Imitation (Core Algorithm)

Supervised Learning Reduction

1. Initial Learning

2. Policy Roll-out (optional exploration)

3. Retrospective Oracle (Algorithm 2)

4. Policy Update with Further Learning

Retrospective Oracle Feedback

Region A

Region B

Learning to Search via Retrospective Imitation, Jialin Song, Ravi Lanka, et al., arXiv
Retrospective Imitation (Full Algorithm)

Learning to Search via Retrospective Imitation, Jialin Song, Ravi Lanka, et al., arXiv

- Initialize $k=1$
- Initialize Gurobi/SCIP/Cplex
- Problem Difficulty $k$
- Base Solver
- Instances & Demonstrations
- $k=k+1$
- Use trained $\pi$
Core Algorithm
• Does this converge?
• Converges to what?

Learning to Search via Retrospective Imitation, Jialin Song, Ravi Lanka, et al., arXiv
Imitation Learning Tutorial (ICML 2018)

https://sites.google.com/view/icml2018-imitation-learning/

Yisong Yue

Hoang M. Le

yyue@caltech.edu

hmle@caltech.edu

@YisongYue

@HoangMinhLe

yisongyue.com

hoangle.info
Issues w/ Distribution Drift & Imitation Signal

• Demonstrations from initial Solver: \( D_0 = \left\{ \left(\begin{array}{c} \text{tree} \\ \text{tree} \end{array}\right) \right\} \)

  "correct" decision in this state
  Which input states?
  Correct relative to what?

• Supervised learning: \( \arg\min_{\pi \in \Pi} L_{D_0}(\pi) \equiv E_{(s,a) \sim D_0}[\ell(a, \pi(s))] \)

  Oracle call to TensorFlow/PyTorch/etc...
  If \( \pi \) achieves low error on \( D_0 \), so what?
Interactive Imitation Learning (Core Alg)

• First popularized by [Daume et al., 2009] [Ross et al., 2011]

• Basic idea:
  • Train $\pi_{i-1} = \arg\min_{\pi \in \Pi} L_{D_{i-1}}(\pi)$  \hspace{1cm} \textit{Supervised Learning}
  • Roll-out $\pi_{i-1}$, collect traces $\{\tau\}$  \hspace{1cm} \textit{Run on instances}
  • Demonstrator converts $\{\tau\}$ into per-state feedback: $\hat{D}_i$  \hspace{1cm} \textit{Depends on}
  • $D_i = \hat{D}_i \cup D_{i-1}$  \hspace{1cm} \textit{Data aggregation}

Search-based Structured Prediction, Daume, Langford, Marcu, Machine Learning Journal 2009
A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning, Ross, Gordon, Bagn...
Interactive Imitation

• First popularized by [Daume et al.]

• Basic idea:
  • Train $\pi_{i-1} = \arg\min_{\pi \in \Pi} L_{D_{i-1}}(\pi)$
  • Roll-out $\pi_{i-1}$, collect traces $\{\tau\}$
  • Demonstrator converts $\{\tau\}$ into per-state feedback: $\hat{D}_i$
  • $D_i = \hat{D}_i \cup D_{i-1}$

Convergence Guarantees:
• $\sum_{i=0}^{M} L_{D_i}(\pi_i) \to \min_{\pi \in \Pi} \sum_{i=0}^{M} L_{D_i}(\pi)$
• Follow-the-Leader argument
• Also studied in [He et al., NeurIPS 2014]

Requires defining “correct”
• Retrospective Oracle

Search-based Structured Prediction, Daume, Langford, Marcu, Machine Learning Journal 2009
A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning, Ross, Gordon, Bagnell
$\pi_1$ Policy Rollout

★: best solution found by $\pi_1$
Retrospective Oracle Feedback

Feedback: (red > white)
for all (red, white) pairs
in the trajectory
\( \pi_2 \) Policy Rollout
Retrospective Oracle Feedback

Feedback: (red > white) for all (red, white) pairs in the trajectory
$\pi_3$ Policy Rollout
Core Algorithm Summary

- Sequence of Learning Reductions
- Leverages Retrospective Oracle to Define “Correct”
  - Relies on sparse environmental rewards
- Converges to near-optimal policy in class
  - Offloads computational challenges to Supervised Learning Oracle
- For supervised learning error $\varepsilon$:

  $$\text{Expected Search Length} = \frac{H^*}{1 - 2\varepsilon}$$

Learning to Search via Retrospective Imitation, Jialin Song, Ravi Lanka, et al., arXiv
Guarantees for Full Algorithm

• Run $\pi^k$ on problems of difficulty $k+1$
  • Initial demonstrations for the harder problem instances

• Suppose: we could have run external solver on harder instances
  Gurobi/SCIP/Cplex/Etc…

• Suppose: search trace includes feasible solution of external solver

• Then $\pi^k$ is as good as using original external solver!
  • (might take longer to converge)

_Learning to Search via Retrospective Imitation_, Jialin Song, Ravi Lanka, et al., arXiv
Empirically Validating Theoretical Results. Finally, we evaluate how well our theoretical results in Section 5 characterize experimental results. Figure 4b and 4c presents the optimal move error rates for the maze experiment, which validates Proposition 1 that retrospective imitation is guaranteed to result in a policy that has lower error rates than imitation learning. The benefit of having a lower error rate is explained by Theorem 2, which informally states that a lower error rate leads to shorter search time. This result is also verified by Figure 2a and 2d, where Retrospective DAgger/SMILe, having the lowest error rates, explores the fewest number of squares at each problem scale.

Learning to Search via Retrospective Imitation, Jialin Song, Ravi Lanka, et al., arXiv
Comparisons w/ Conventional IL

Learning to Search via Retrospective Imitation, Jialin Song, Ravi Lanka, et al., arXiv
Retrospective Imitation

• Two-Stage Algorithm
  • Leverages Supervised Learning Oracle

• Initial demonstrations on small problems

• Exploits sparse environmental reward
  • “Retrospective Oracle”

• Iteratively scale up to harder problems
Co-Training for Policy Learning
(Multiple Views)

Example: Minimum Vertex Cover

\[
\text{max} - \sum_{i=1}^{5} x_i, \\
\text{subject to:} \\
x_1 + x_2 \geq 1, \\
x_2 + x_3 \geq 1, \\
x_3 + x_4 \geq 1, \\
x_3 + x_5 \geq 1, \\
x_4 + x_5 \geq 1, \\
x_i \in \{0, 1\}, \forall i \in \{1, \cdots, 5\}
\]

Graph View

Integer Program View
(Branch & Bound View)

[He et al., 2014]

[Khalil et al., 2017]
Co-Training for Policy Learning
(Multiple Views)

Example: Different Types of Integer Programs

ILP

QCQP
Co-Training [Blum & Mitchell, 1998]

- Many learning problems have different sources of information.
- Webpage Classification: Words vs Hyperlinks

(Taken from Nina Balcan’s slides)
What’s Different about Policy Co-Training?

• Sequential Decisions vs 1-Shot Decisions

• (Sparse) Environmental Feedback
  • Can collect more “labels”

• Different Action Spaces
  • Graph vs Branch-and-Bound

Co-training for Policy Learning, Jialin Song, Ravi Lanka, et al., UAI 2019
**Intuition**

Example 1: [1] “Learning combinatorial optimization algorithms over graphs” [Khalil et al., 2017]


---

**MVC Instance**

E.g., [1]

E.g., [2,3]

---

Maximize $-\sum_{i=1}^{5} x_i$

Subject to:

- $x_1 + x_2 \geq 1$
- $x_2 + x_3 \geq 1$
- $x_3 + x_4 \geq 1$
- $x_3 + x_5 \geq 1$
- $x_4 + x_5 \geq 1$

$x_i \in \{0, 1\}, \forall i \in \{1, \ldots, 5\}$
Intuition

E.g., [1]

MVC Instance

E.g., [2,3]

[1] “Learning combinatorial optimization algorithms over graphs” [Khalil et al., 2017]
Intuition

E.g., [1]

[1] “Learning combinatorial optimization algorithms over graphs” [Khalil et al., 2017]

E.g., [2,3]


Theoretical Insight

• Different representations differ in hardness
• Goal: quantify improvement

\(\Omega_1: \text{representation 1 easier}\)

\(\Omega_2: \text{representation 2 easier}\)

\(\Omega: \text{all problems}\)

*Co-training for Policy Learning*, Jialin Song, Ravi Lanka, et al., UAI 2019
(Towards) a Theory of Policy Co-Training

- Two MDP “views”: $M^1$ & $M^2$
  - $f^{1\rightarrow 2}(\tau^1) \implies \tau^2$ (and vice versa)
  - “Trajectory” / “Rollout”
  - Realizing $\tau^1$ on $M^1 \iff$ realizing $\tau^2$ on $M^2$

**Question:** when does having two views/policies help?

- Policy Improvement (next slide)
  - Builds upon [Kang et al., ICML 2018]
- Optimality Gap for Shared Action Spaces (in paper)
  - Builds upon [DasGupta et al., NeurIPS 2002]
Policy Improvement Bound

Standard for Policy Gradient

1-step suboptimality of $\pi^1$ on $\Omega$

JS Divergence of $\pi^2$ vs $\pi^1$ on $\Omega_2$

KL Divergence of $\pi^1$ vs $\pi^1'$ on $\Omega$

1-step suboptimality of $\pi^1$ on $\Omega$

$J(\pi^1') \geq J_{\pi^1}(\pi^1') - \frac{2\gamma(\alpha_{\pi^1}^1\epsilon_{\pi^1}^1 + 4\beta_{\pi^2}^2\epsilon_{\pi^2}^2)}{(1 - \gamma)^2} + \delta_{\Omega_2}^2$

Performance of new policy (either RL or IL)

Approximation by sampling from $\pi^1$

Discount

Performance Gap of $\pi^2$ over:

$J(\pi^2 | M \sim \Omega_2) - J(\pi^1 | M \sim \Omega_1)$

Want to Maximize

$\Omega_2: \pi_2$

$\Omega_1: \pi_1$ better

$\Omega$: all instances

Builds upon theoretical results from [Kang et al., ICML 2018]
Policy Improvement Bound (Summary)

\[ J(\pi'1) \geq J_{\pi1}(\pi'1) - \frac{2\gamma(\alpha^1_\Omega \epsilon^1_\Omega + 4\beta^2_{\Omega_2} \epsilon^2_{\Omega_2})}{(1 - \gamma)^2} + \delta^2_{\Omega_2} \]

• Minimizing \( \beta^2_{\Omega_2} \) → low disagreement between \( \pi^2 \) vs \( \pi^1 \)

• Maximizing \( \delta^2_{\Omega_2} \) → high performance gap \( \pi^2 \) over \( \pi^1 \) on some
CoPiEr Algorithm (Co-training for Policy Learning)

**Update** (only showing 1 view)

Augmented Obj: \( \tilde{J}(\pi') = J_\pi(\pi') - \lambda L(\pi', \tau') \)

Take gradient step

**Exchange** (only showing 1 version)

If \( \pi^1 \) better: \( \tau'^2 = f^{1\to2}(\tau^1), \tau'^1 = \emptyset \)

If \( \pi^2 \) better: \( \tau'^1 = f^{2\to1}(\tau^2), \tau'^2 = \emptyset \)

---

**Rollout**

Run \( \pi^1 \to \tau^1 \)

Run \( \pi^2 \to \tau^2 \)

---

**Augmented Obj:**

\[
\max \sum \limits_i \left( \sum \limits_m \log P \left( \tau'_m^{1\to2} \mid \pi^1, \tau^1 \right) \right) - \lambda \left( \sum \limits_m \log P \left( \tau'_m^{2\to1} \mid \pi^2, \tau^2 \right) \right)
\]

subject to

\[
\begin{align*}
x_1 + x_2 & \geq  \sum \limits_m \log P \left( \tau'_m^{1\to2} \mid \pi^1, \tau^1 \right) \\
x_2 + x_3 & \geq  \sum \limits_m \log P \left( \tau'_m^{2\to1} \mid \pi^2, \tau^2 \right) \\
x_3 + x_4 & \geq  \sum \limits_m \log P \left( \tau'_m^{1\to2} \mid \pi^1, \tau^1 \right) \\
x_4 + x_5 & \geq  \sum \limits_m \log P \left( \tau'_m^{2\to1} \mid \pi^2, \tau^2 \right) \\
x_i & \in \{0, 1\}
\end{align*}
\]

---

Co-training for Policy Learning, Jialin Song, Ravi Lanka, et al., UAI 2019
Performance comparison for Minimum Vertex Cover

- **Strong vs Baselines** (w/o Co-Training)
- **CoPiEr Final Outperforms Individual Views**
- **Strong vs Gurobi**

### Results Summary

- **Strong vs Baselines (w/o Co-Training)**: CoPiEr Final outperforms individual views.
- **Strong vs Gurobi**: CoPiEr Final outperforms Gurobi.
- **CoPiEr Final Outperforms Individual Views**

#### Graphs
- **Graph (CoPiEr)**
- **Graph (RL)**
- **ILP (Retrospective imitation)**
- **ILP (CoPiEr)**
- **ILP (DAgger)**

#### #vertices in the graph

- **100-200**
- **200-300**
- **300-400**
- **400-500**
Ongoing: Integration with ENav
Ongoing: Additive Manufacturing

- Planning for 3D Inkjet Droplet Printing
Iterative Amortized Inference (for Deep Probabilistic Models)

Related to “Learning to Learn” [Andychowicz et al., 2016]

Iterative Amortized Inference, Joe Marino et al., ICML 2018
A General Framework for Amortizing Variational Filtering, Joe Marino et al, NeurIPS 2018
Ongoing: Amortized Planning

Learning dynamics:

\[(a_t, s_t) \rightarrow f_{\text{reward}}(a_t, s_t) \rightarrow \hat{r}_t \rightarrow f_{\text{state}}(a_t, s_t) \rightarrow \hat{s}_{t+1}\]

Planning:

\[\ldots \rightarrow a_1, a_2, \ldots, a_T\]

Optimize:

\[
\max_{a_1, \ldots, a_T} \sum_{t=1}^{T} f_{\text{reward}}(f_{\text{state}}(\hat{s}_{t-1}, a_{t-1}), a_t)
\]

Baseline: Gradient-based Planning

Can use (offline) training to amortize.
Learning to Optimize as Policy Learning

- Optimization as Sequential Decision Making
- Formulate New Learning Problems
  - Builds upon RL/IL
- Interesting Algorithms
  - Theoretical Analysis/Guidance
  - Good Empirical Performance
Learning to Search via Retrospective Imitation, Jialin Song, Ravi Lanka, et al., arXiv
Co-Training for Policy Learning, Jialin Song, Ravi Lanka, et al., UAI 2019
Learning Policies for Contextual Submodular Optimization, Stephane Ross et al., ICML
Iterative Amortized Inference, Joe Marino et al., ICML 2018
A General Framework for Amortizing Variational Filtering, Joe Marino et al, NeurIPS 2018

https://github.com/ravi-lanka-4/CoPiEr
https://github.com/joelouismarino/iterative_inference