PAC-Bayes Control: Learning Policies that Provably Generalize to Novel Environments

Anirudha Majumdar, Alec Farid, and Anoopkumar Sonar

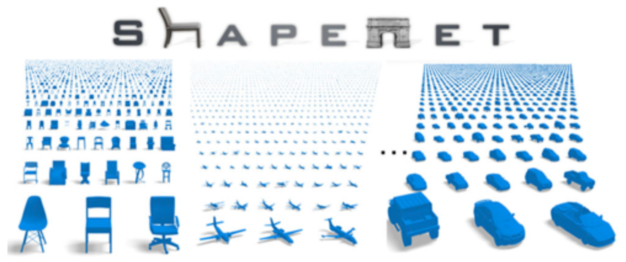
Motivation

Goal: Synthesize controllers that provably generalize well to novel environments

Target applications:

Navigation

Grasping



Related Work

- Real-time planning + assumptions on environment [Schouwenaars '04, Fraichard '07, Althoff '15, Majumdar '17]
- POMDP: environment is part of state [Richter '15, '17]
- PAC-MDP and PAC-Bayes MDP bounds [Fard '10 '12, Kearns '02, Brafman '02, Fu '14]
- Deep learning-based control [Lenz '15, Levine '16, Agrawal '16, Mahler '17, Tobin ' 17, Gupta '17, many others ...]

PAC-Bayes theory

- **Generalization bound** for **supervised learning** [McAllester '99, Seeger '02, Langford '03]
- Recently used for explaining and promoting generalization in deep learning [Dziugaite '17 '18, Neyshabur '17 '18]

Problem Setup

- **System dynamics:**
  $$x(t+1) = f(x(t), u(t); E)$$
  State Control Environment
- **Environments** E are drawn i.i.d. from unknown distribution $\mathcal{D}$
- **Dataset:** $S = \{E_1, \dots, E_N\}$
- **Sensor:** $y = g(x; E)$ (e.g., depth)
- **Policy:** $\pi : y(t) \mapsto u(t)$
- **Rollout function:**
  $$r_\pi : E \mapsto ((x_1, u_1), \dots, (x_T, u_T))$$
- **Cost:** $C(r_\pi; E) \in [0, 1]$
- **Goal:**
  $$\min_P \; C_{\mathcal{D}}(P) := \mathop{\mathbb{E}}_{E\sim\mathcal{D},\pi\sim P} C(r_\pi; E)$$
- **Challenge:** $\mathcal{D}$ is **unknown!**

PAC-Bayes for Control

Key idea: Translate generalization bounds from supervised learning to control setting via reduction

Supervised Learning	Control Synthesis
Input: z	Environment: E
Hypothesis: h_w	Rollout: r_π
Loss: $l(h_w; z)$	Cost: $C(r_\pi; E)$

PAC-Bayes for Control

Theorem 1: With probability $1-\delta$ over sampled environments

$$C_{\mathcal{D}}(P) \leq C_{\mathrm{PAC}}(P)$$
$$:= \frac{1}{N}\sum_{i=1}^{N} \mathop{\mathbb{E}}_{\pi\sim P} C(r_\pi; E_i) + \sqrt{\frac{\mathrm{KL}(P\|P_0) + \log(\frac{2\sqrt{N}}{\delta})}{2N}}$$

Training cost + "Regularizer"

Here P_0 is a prior over controllers.

- If train $D \neq$ test D', obtain upper bound with help from DV inequality [Donsker and Varadhan '75]

Theorem 2: With probability $1-\delta$

$$C_{\mathcal{D}'}(P) \leq$$
$$\mathcal{B} + \log\left(\frac{1}{N}\sum_{i=1}^{N} \mathop{\mathbb{E}}_{\pi\sim P} e^{C(r_\pi, E_i)} + (e-1)\mathrm{Regularizer}\right)$$

Requiring $\mathrm{KL}(\mathcal{D}'||\mathcal{D}) \leq \mathcal{B}$

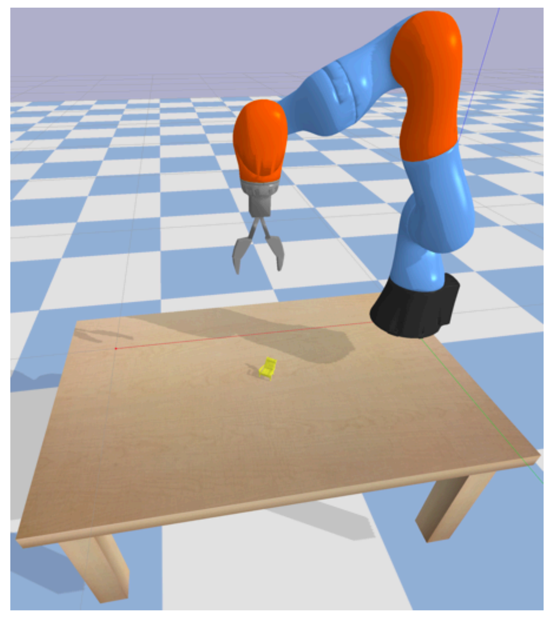
Algorithmic approach: Minimize RHS of bound, i.e., $C_{\mathrm{PAC}}(P)$ via

- Convex optimization (Relative Entropy Programming) for finite policy spaces
- SGD for continuously parameterized policies

Results

Grasping:

- Training data from ShapeNet
- 2000 training objects
- Deep neural net policy
- Bound: ~71%

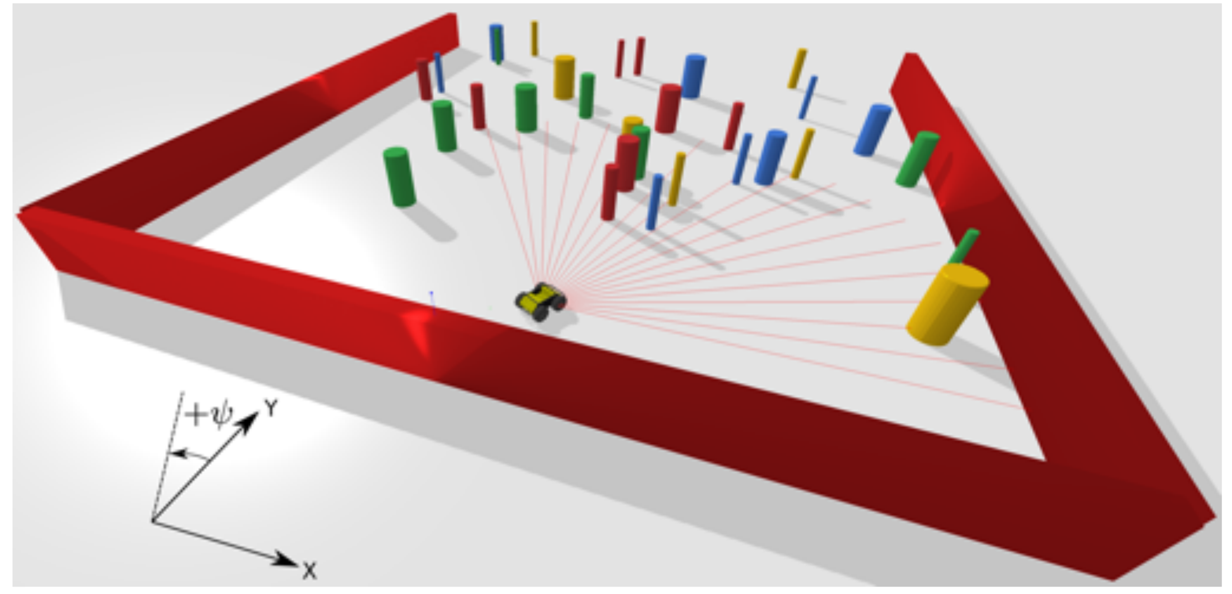


Results Cont.

Obstacle avoidance:

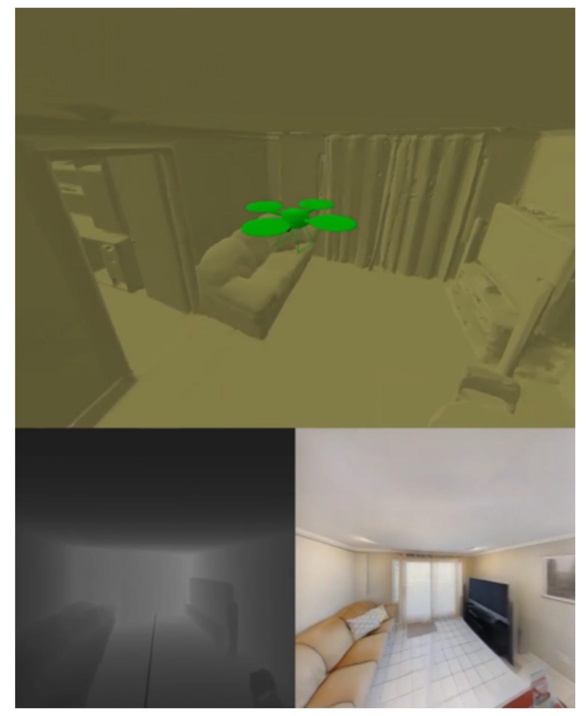
N	100	500	1000	10^4
Bound on success %	73.8	83.0	86.2	90.4
True success % (estimate)	91.9	92.0	91.9	91.7
Robust bound on success %	54.7	72.4	76.2	81.5

- Robust PAC-Bayes controllers trained on environments with ~2.5% higher success rate, $\mathcal{B} = 0.0819$



Navigation:

- Training data from Stanford 2D-3D-S [Armeni '17]
- 1000 training environments
- Gibson environment for simulation
- Bound: ~74%



Current/Future Work

- Relax modeling assumptions (e.g., i.i.d. data)
- Combine with metalearning (e.g., MAML [Finn '17])