Kernel Mean Embedding for Stochastic Optimization and Control

Based on joint work with Moritz Diehl and Bernhard Schölkopf

Jia-Jie Zhu
Max Planck Institute for Intelligent Systems
Tübingen, Germany

2020/6/10

Optimization under uncertainty

- Chance constraint [Charnes et al. 50s] (downside: intractable)

$$\min_x \; l(x) \quad \text{subject to} \quad \mathrm{P}\, C(x, \xi) \leq 0 \geq 1 - \alpha$$

- Robust optimization [Ben-Tal et al. 90s] (downside: conservative)

$$\min_x \; l(x) \quad \text{subject to} \quad f(x, \xi) \leq 0, \; \forall \xi \in \mathcal{U}$$

Scenario approach to chance constraint

$$\min_x \ l(x), \quad \text{subject to} \ \ C(x, \xi_i) \leq 0 \text{ for } i = 1, \ldots, N.$$

- This is a convex approximation to the channce-constrained problem

$$\min_x \ l(x), \quad \text{subject to} \ \ \mathrm{P}(C(x, \xi) \leq 0) \geq 1 - \alpha.$$

- If $N \to \infty$, chance constraint is satisfied at level almost $= 1$. i.e. conservative
- If N is small, we may be too optimistic

Reducing conservativeness

$$\min_x \; l(x), \quad \text{subject to} \;\; C(x, \xi_i) \leq 0 \text{ for } i = 1, \ldots, N.$$

How about we pick a subset of scenarios $\{\xi_i\}_{i=1}^N$ to discard?

Kernel mean embedding

- Recall a kernel is a symmetric, positive semi-definite bivariate function, e.g., $k(x, x') = \exp\left(-\frac{1}{2\sigma^2}\|x - x'\|_2^2\right)$.

- Kernel mean embedding (KME) maps probability distributions to functions in a Hilbert space.

$$\mu : P \mapsto \int k(x, \cdot)\, dP(x), \quad \hat{\mu} : P \mapsto \sum_{i=1}^{N} \alpha_i k(x_i, \cdot),\ x_i \sim P$$

- μ can be thought of a generalized moment vector

Why use kernel mean embedding for optimization

- It allows us to perform optimization problem in the space of probability distribution.

$$
\begin{aligned}
\min_{P,\mu} \quad & \int l \, dP \\
\text{subject to} \quad & \|\mu - \mu_{\hat{P}}\|_{\mathcal{H}} \leq \epsilon. \\
& \int \phi(x)\, dP(x) = \mu_p.
\end{aligned}
$$

- By virtue of the RKHS tools, optimization is often tractable.
- It induces a metric on the space of distributions, which can be used for distributionally robust optimization (DRO; cf. Z. 2020).

Illustration of our idea

Use L-1 penalty to discard scenarios while staying close to the original distribution.

$$\min_{\alpha} \ \|\sum_{i=1}^{N} \alpha_i \phi(\xi_i) - \hat{\mu}_{\xi}\|_{\mathcal{H}}^2 + \lambda \|w^\top \alpha\|_1.$$

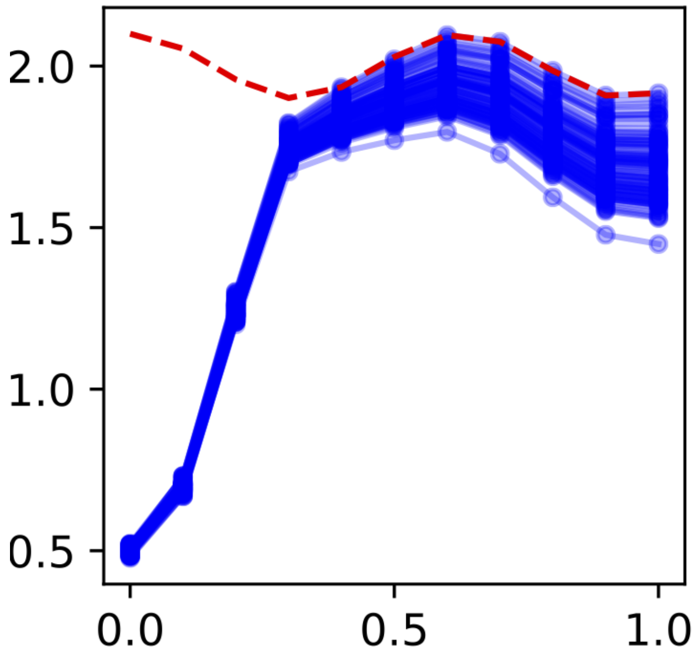
Scenario approach with discarding

- Discard the scenarios ξ_i with the index set $\mathcal{I} = \{i \mid \alpha_i = 0, i = 1, \ldots, n\}$ by solving

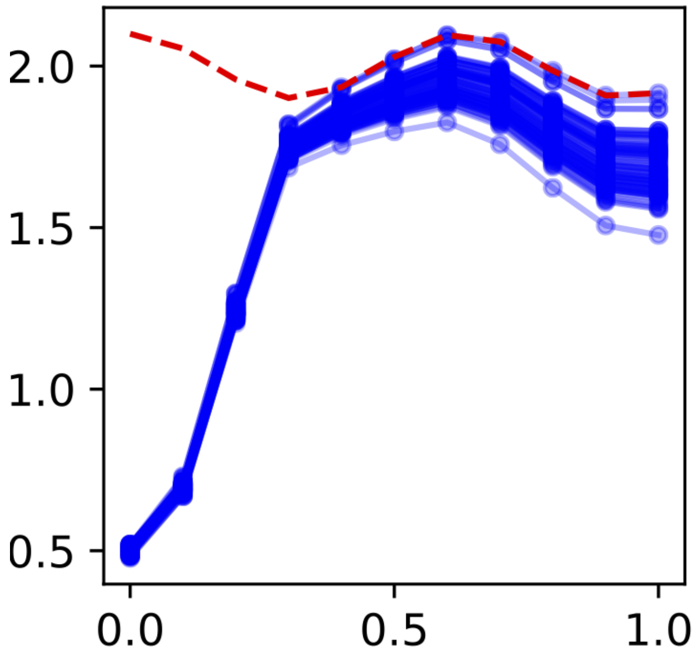
$$\min_{\alpha} \ \| \sum_{i=1}^{N} \alpha_i \phi(\xi_i) - \hat{\mu}_{\xi} \|_{\mathcal{H}}^2 + \lambda \| w^\top \alpha \|_1.$$

- Then, we re-solve the stochastic programming problem with the reduced-set scenarios $\mathcal{R} := \{1, \ldots, n\} \setminus \mathcal{I}$.

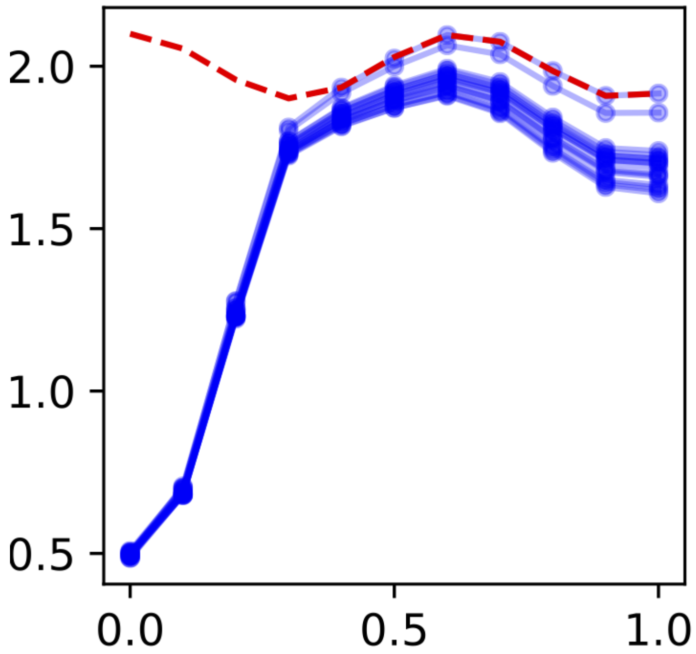
Stochastic control example



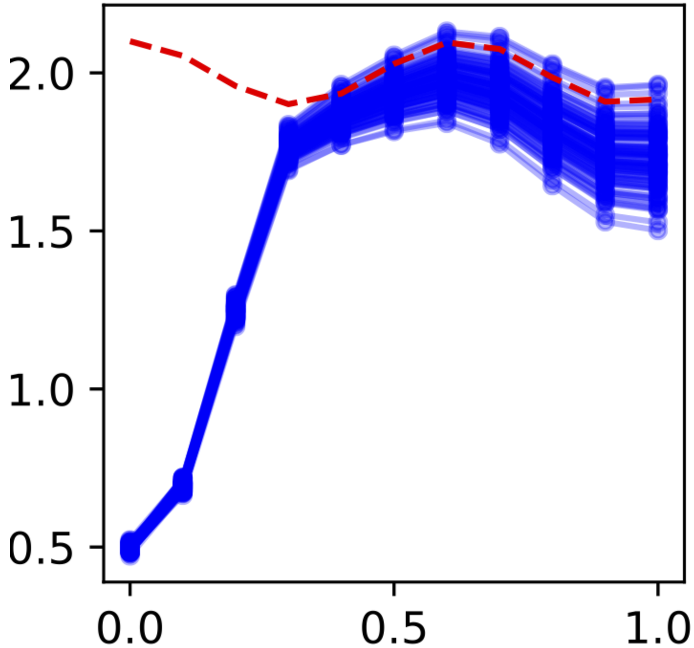
Discard scenarios



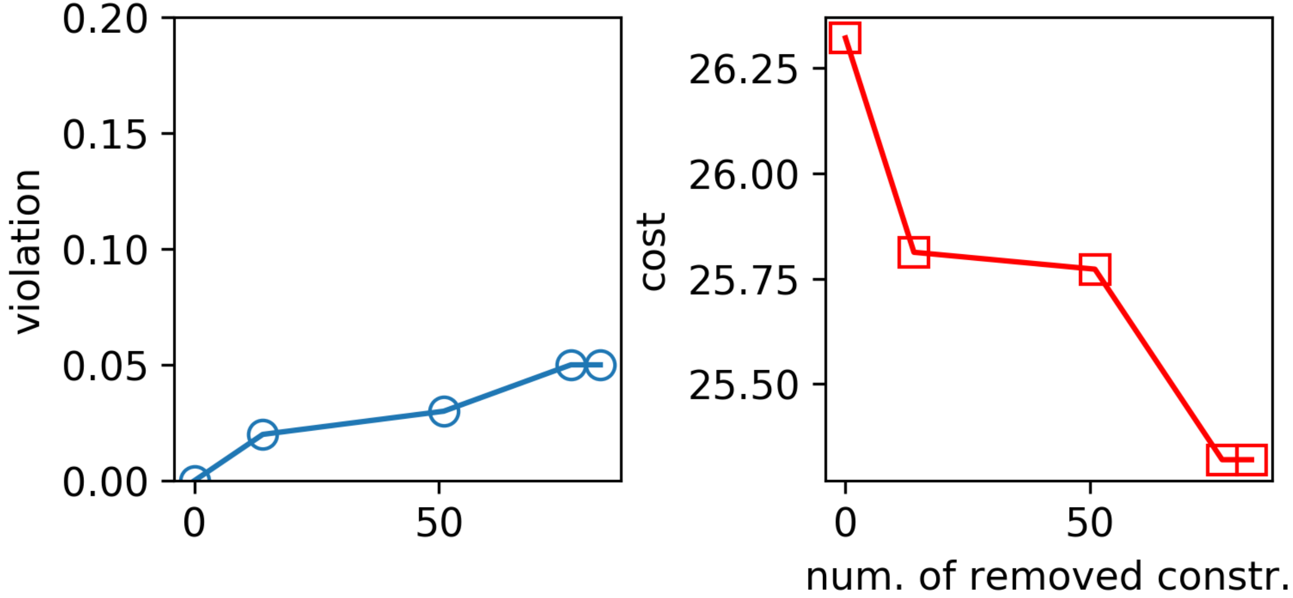
Discard more...



Result: optimistic controller



Result: reduced conservatism



Takeaway

- Kernel mean embedding in RKHS allows efficient optimization in distribution space

$$\min_{P} f(P) \text{ becomes } \min_{\mu_p} f(P).$$

  - It can be combined with other methods like DL, e.g., MMD-GAN, adversarial training, blah blah blah.
- This paper focus on reducing conservativeness of the scenario approach to stochastic programming and control. What if we wish to be more robust?
  - Kernel Distributionally robust optimizaiton (K-DRO. See the next paper.)

Thank you! This talk is based on

- **Z, Diehl, Schölkopf, 2020. A Kernel Mean Embedding Approach to Reducing Conservativeness in Stochastic Programming and Control. L4DC**
- **Z, Jitkrittum, Diehl, Schölkopf, 2020. Kernel Distributionally Robust Optimization. Preprint**
- Z, Jitkrittum, Diehl, Schölkopf, 2020. Worst-Case Risk Quantification under Distributional Ambiguity using Kernel Mean Embedding in Moment Problem. Preprint
- Z, Muandet, Diehl, Schölkopf, 2019. A New Distribution-Free Concept for Representing, Comparing and Propagating Uncertainty in Dynamical Systems with Kernel Probabilistic Programming. IFAC 2020