Imperial College London

Robust Data-Driven Dynamic Programming

Grani A. Hanasusanto[1] and Daniel Kuhn[2]

[1]Imperial College London, [2]École Polytechnique Fédérale de Lausanne

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Introduction

Motivation: Multi-stage decision making under uncertainty:

- Utilize historical observations directly in the solution procedure
- Handle data sparsity
- Two-stage setting: Exploit side information in decision making
- Dynamic setting: Mitigate the three curses of dimensionality

Our solution:

- Use **Nadaraya-Watson (NW) kernel regression** to estimate conditional expectations
- Use **robust optimization** to reduce the effects of estimation errors

Stochastic Programming (SP)

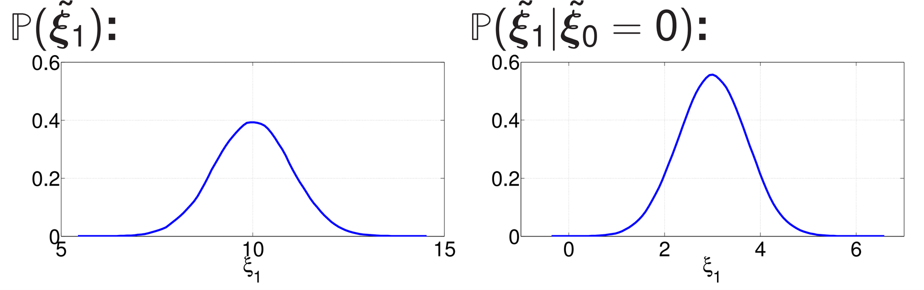
Stochastic program with side information:

$$\text{SP:} \quad \min_{\boldsymbol{x}\in\mathcal{X}} \mathbb{E}[c(\boldsymbol{x},\tilde{\xi}_1)|\tilde{\xi}_0=\xi_0]$$

Example: Newsvendor problem

- The current demand $\tilde{\xi}_1$ is uncertain but depends on the demand $\tilde{\xi}_0$ of the previous period
- The conditional distribution of $\tilde{\xi}_1$ given $\tilde{\xi}_0$ is less dispersed than the marginal distribution of $\tilde{\xi}_1$

$\mathbb{P}(\tilde{\xi}_1)$: $\quad$ $\mathbb{P}(\tilde{\xi}_1|\tilde{\xi}_0=0)$:



Statistics:

$\mathbb{E}[\tilde{\xi}_1]=10, \text{var}(\tilde{\xi}_1)=1$

$\mathbb{E}[\tilde{\xi}_1|\tilde{\xi}_0=0]=3, \text{var}(\tilde{\xi}_1|\tilde{\xi}_0=0)=\frac{1}{2}$

Data-Driven SP

Motivation:

- The joint distribution of $\tilde{\xi}_0$ and $\tilde{\xi}_1$ is unknown
- Only few historical observations $(\xi_0^i,\xi_1^i)_{i=1}^N$ are available

Data-driven stochastic program:

- Estimate the conditional expectation using **NW kernel regression**:

$$\text{DSP:} \quad \min_{\boldsymbol{x}\in\mathcal{X}} \mathbb{E}[c(\boldsymbol{x},\tilde{\xi}_1)|\tilde{\xi}_0=\xi_0] \approx \min_{\boldsymbol{x}\in\mathcal{X}} \sum_{i=1}^N q_i(\xi_0)c(\boldsymbol{x},\xi_1^i)$$

- $q_i(\xi_0)=\frac{K(\xi_0-\xi_0^i)}{\sum_{j=1}^N K(\xi_0-\xi_0^j)}$ are normalized weights constructed using kernel $K(\cdot)$

Shortcomings:

- When only few observations are available the NW estimate of the conditional expectation exhibits a high variability
- Estimation errors lead to an optimistic downward bias in DSP

$$\mathbb{E}[\min_{\boldsymbol{x}\in\mathcal{X}} \sum_{i=1}^N q_i(\tilde{\xi}_0)c(\boldsymbol{x},\tilde{\xi}_1^i)|\tilde{\xi}_0] \le \min_{\boldsymbol{x}\in\mathcal{X}} \mathbb{E}[c(\boldsymbol{x},\tilde{\xi}_1)|\tilde{\xi}_0]$$

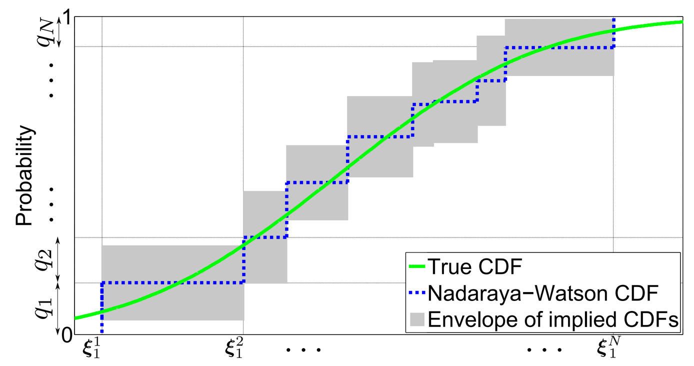
Robust Data-Driven SP

Robust data-driven stochastic program: Optimize worst-case conditional expectation:

$$\text{RDSP:} \quad \min_{\boldsymbol{x}\in\mathcal{X}} \max_{\boldsymbol{p}\in\Delta(\boldsymbol{q}(\xi_0))} \sum_{i=1}^N p_i c(\boldsymbol{x},\xi_1^i)$$

- Use a χ^2-distance uncertainty set

$$\Delta(\boldsymbol{q})=\{\boldsymbol{p}\in\Delta : \sum_{i=1}^N (p_i-q_i)^2/p_i \le \gamma\}$$

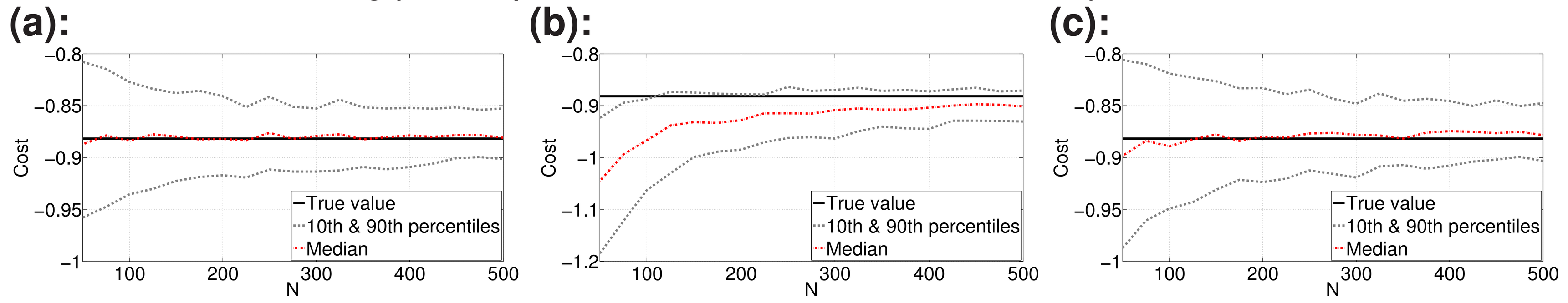


Example: Error maximization in mean-variance optimization

$$\min_{\boldsymbol{x}\in\mathbb{R}^5, \boldsymbol{e}^\intercal\boldsymbol{x}=1} \mathbb{E}[0.1(\tilde{\xi}_1^\intercal\boldsymbol{x})^2-\tilde{\xi}_1^\intercal\boldsymbol{x}|\tilde{\xi}_0=\boldsymbol{e}] \quad \text{with} \quad \begin{bmatrix}\tilde{\xi}_0\\ \tilde{\xi}_1\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\boldsymbol{e}\\ \boldsymbol{e}\end{bmatrix}, \begin{bmatrix}\mathbb{1} & 0.3\mathbb{1}\\ 0.3\mathbb{1} & \mathbb{1}\end{bmatrix}\right)$$

- **(a):** By symmetry, the solution of SP is $\boldsymbol{x}^*=\frac{1}{5}\boldsymbol{e}$; the NW estimate of the cost of $\boldsymbol{x}^*$ is unbiased but exhibits large fluctuations
- **(b):** The optimal value of DSP is even more noisy, displays a significant downward bias and converges slowly with N
- **(c):** Interestingly the optimal value of RDSP is less noisy and unbiased

(a): (b): (c):



Dynamic Programming (DP)

Stochastic dynamic program:

$$\text{DP:} \quad V_t(\boldsymbol{s}_t,\xi_t)=\min \quad c_t(\boldsymbol{s}_t,\xi_t,\boldsymbol{x}_t)+\mathbb{E}[V_{t+1}(\tilde{\boldsymbol{s}}_{t+1},\tilde{\xi}_{t+1})|\tilde{\xi}_t=\xi_t]$$
$$\text{s.t.} \quad \boldsymbol{x}_t\in\mathcal{X}_t, \quad \boldsymbol{s}_{t+1}=g_t(\boldsymbol{s}_t,\boldsymbol{x}_t,\xi_{t+1})$$

- **Endogenous state**: $\boldsymbol{s}_t$ is decision-dependent
- **Exogenous state**: ξ_t is decision-*in*dependent

Data-driven dynamic program:

$$\text{DDP:} \quad \hat{V}_t^{\text{d}}(\boldsymbol{s}_t,\xi_t)=\min \quad c_t(\boldsymbol{s}_t,\xi_t,\boldsymbol{x}_t)+\sum_{i=1}^N q_{ti}(\xi_t)\hat{V}_{t+1}^{\text{d}}(\boldsymbol{s}_{t+1}^i,\xi_{t+1}^i)$$
$$\text{s.t.} \quad \boldsymbol{x}_t\in\mathcal{X}_t \quad \boldsymbol{s}_{t+1}^i=g_t(\boldsymbol{s}_t,\boldsymbol{x}_t,\xi_{t+1}^i) \quad \forall i=1,\dots,N$$

Robust data-driven dynamic program:

$$\text{RDDP:} \quad \hat{V}_t^{\text{r}}(\boldsymbol{s}_t,\xi_t)=\min \quad c_t(\boldsymbol{s}_t,\xi_t,\boldsymbol{x}_t)+\max_{\boldsymbol{p}\in\Delta(\boldsymbol{q}_t(\xi_t))}\sum_{i=1}^N p_i\hat{V}_{t+1}^{\text{r}}(\boldsymbol{s}_{t+1}^i,\xi_{t+1}^i)$$
$$\text{s.t.} \quad \boldsymbol{x}_t\in\mathcal{X}_t \quad \boldsymbol{s}_{t+1}^i=g_t(\boldsymbol{s}_t,\boldsymbol{x}_t,\xi_{t+1}^i) \quad \forall i=1,\dots,N$$

Separate architectures for endogenous and exogenous states:

- **Endogenous**: conic representable parametric approximation
- **Exogenous**: NW kernel regression to estimate the conditional expectation

Complexity of RDDP:

- If ⏩ c_t is convex quadratic ⏩ g_t is affine ⏩ $\mathcal{X}_t$ is second-order conic representable, then RDDP reduces to a second-order cone program (SOCP) that can be solved in $\mathcal{O}(\sqrt{N}m^3)$ iterations, where $m=\#$decision variables

Algorithm

Inputs: State trajectories $\{\boldsymbol{s}_t^k\}_{t=1}^T, 1\le k\le K$, observation histories $\{\xi_t^i\}_{t=1}^{T+1}, 1\le i\le N$

For t=T,...,1 ⏩
  For i=1,...,N ⏩
    For k=1,...,K ⏩
      Evaluate RDDP with inputs $\hat{V}_{t+1}^{\text{r}}$, $\boldsymbol{s}_t^k$ and ξ_t^i ⏪
    Construct parametric approximation $\hat{V}_t^{\text{r}}(\cdot,\xi_t^i)$ in the endogenous state ⏪ ⏪

Outputs: Approximate cost-to-go functions $\hat{V}_t^{\text{r}}(\cdot,\xi_t^i)$ for $i=1,\dots,N$ and $t=1,\dots,T$

Main features of algorithm:

- Number of RDDP problems to solve is $\mathcal{O}(NKT)$
- The NK optimization problems are independent (**parallelizable**)

Results

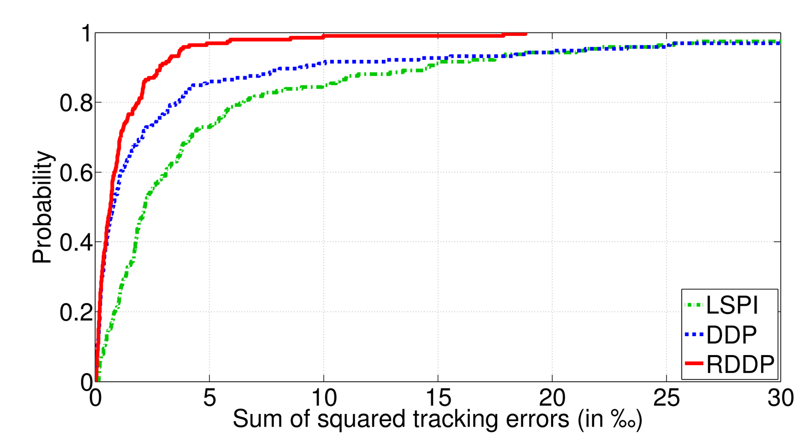
Index tracking:

- Track S&P 500 index over 20 trading days using a portfolio of indices: NASDAQ Composite, Russell 2000, S&P MidCap 400, and AMEX Major Market
- Training data: 26.8.1991 - 13.3.1998 (80 months)
- Test data: 16.3.1998 - 3.8.2013 (192 months)
- RDDP outperforms DDP and LSPI substantially
- The distribution of tracking errors generated by RDDP stochastically dominates those generated by DDP and LSPI

Statistics of tracking errrors (in ‰):

Statistic	LSPI	DDP	RDDP
Mean	5.69	4.70	1.28
Std. dev.	11.70	15.07	2.23
90th prct.	14.60	9.05	2.85
Worst case	126.71	157.20	18.83

CDF of tracking errors:



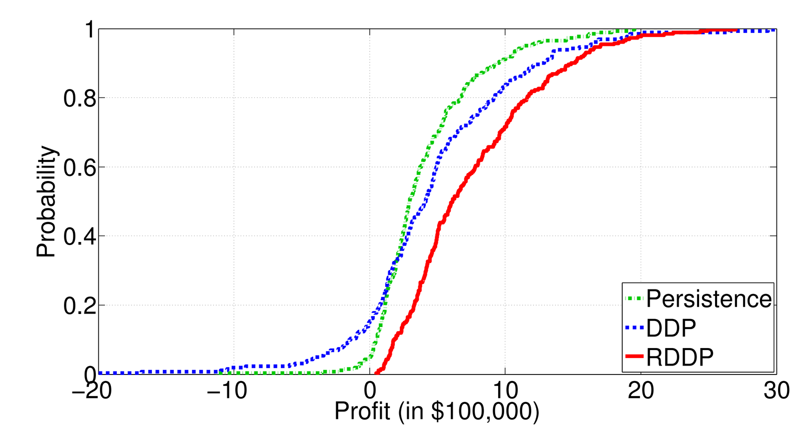
Wind energy commitment:

- Wind energy producer chooses day-ahead energy commitments over a planning horizon of $T=7$ days
- There are three (inefficient) storage facilities and penalties for unmet commitments
- Training data: 1.1.2002 - 26.12.2006 (260 weeks)
- Test data: 27.12.2006 - 21.12.2012 (260 weeks)
- The profit distribution generated by RDDP stochastically dominates those generated by DDP and a persistence heuristic
- RDDP cuts off any losses while all other algorithms bear a significant risk of incurring a loss

Statistics of profit (in $100,000):

Statistic	Persistence	DDP	RDDP
Mean	4.04	4.70	7.55
Std. dev.	3.97	6.34	5.13
10th prct.	0.52	-1.46	1.81
Worst case	-11.22	-22.67	0.48

Profit distribution:



Neural Information Processing Systems Foundation

computational optimisation group

EPSRC
Engineering and Physical Sciences Research Council