Integrating Learning and Optimization in Data-Driven Problems

Tito Homem-de-Mello

Universidad Adolfo Ibañez, Chile

Supported by Grant FONDECYT 1221770

PUC RIO

LAMPS Laboratory of Applied Mathematical Programming and Statistics

UNIVERSIDAD ADOLFO IBÁÑEZ

Joint work with

Davi Valladão (PUC-Rio, Brazil)

Thuener Silva (PUC-Rio, Brazil)

Joaquim Dias Garcia (PSR & PUC-Rio, Brazil)

Alexandre Street (PUC-Rio, Brazil)

Francisco Muñoz (Chilean Association of Generators)

PUC RIO

LAMPS Laboratory of Applied Mathematical Programming and Statistics

UNIVERSIDAD ADOLFO IBÁÑEZ

Data-driven problems

- In many stochastic optimization problems of the form

$$\min_{x \in X} \mathbb{E}_P[G(x, \xi)]$$

all we have available as input are observations $\xi_1, \ldots, \xi_n$.

- How to make a decision using this input?

Data-driven problems

Some typical approaches:

- Use directly the empirical distribution from the data in place of *P.*
- Estimate a distribution from the data, use it in place of *P*.
- Solve a distributionally robust problem with ambiguity set centered on empirical distribution.

But…what if the data has some structure and we want to learn it ?

Learning and optimizing

Yesterday Jim Luedtke gave an excellent review of approaches to do learning+optimizing:

- As separated tasks

- As integrated tasks
  - Modify the optimization step → 1st example
  - Modify the learning step → 2nd example
  - Direct solution learning

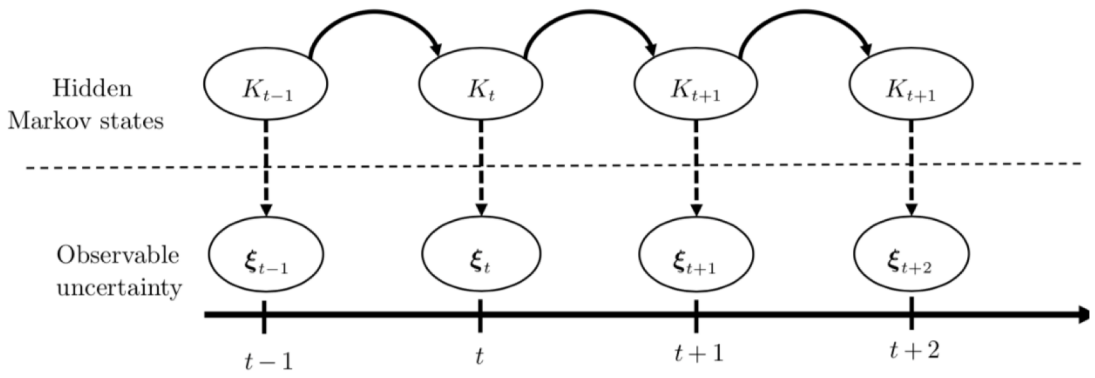
First example: dynamic asset allocation

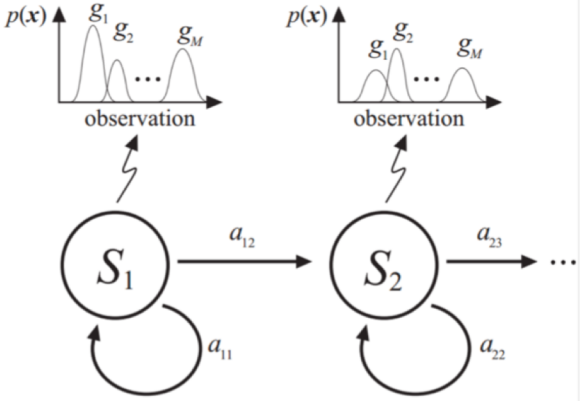
- The goal to find the asset allocations in each period so that the expected final wealth is maximized, taking into account the transaction costs.

- Here $\xi_1, \ldots, \xi_n$ are the returns, which can be modeled using Hidden Markov Models.

- Conditionally to each state, the log-returns are iid with multivariate Gaussian distribution estimated from data (Rydén et al. [1998], Hassan and Nath [2005]).

HMM – training





Parameters estimated by the learning algorithm:

- $\hat{p}_j(k) := P\left(K_{t+1} = k \mid K_t = j\right), \forall t = 0, \ldots, T-1.$
- The coefficients Θ_t associated with the conditional density function $p(\xi_t \mid K_t = k\,; \Theta_t)$ (e.g., mixture of normals).

HMM – prediction

With those parameters at hand, we can compute

$$P\left(K_t = k \mid \xi_t, \xi_{t-1}, \ldots, \xi_1\right), \forall k \in \mathcal{K}.$$

Now we can predict the values of ξ_{t+1} on out-of-sample data, i.e.

$$p(\xi_{t+1} \mid \xi_t, \ldots, \xi_1 ; \Theta_{t+1}) \propto$$
$$\sum_{j \in \mathcal{K}} \sum_{k \in \mathcal{K}} p(K_t = j, \xi_t, \ldots, \xi_1) \times \hat{p}_j(k) \times p(\xi_{t+1} \mid K_{t+1} = k ; \Theta_{t+1}).$$

Estimation errors

- One issue with the procedure is that the parameters estimated from HMM are a function of the observed data.

- Thus, this is likely to result in poor out-of-sample performance! (Mohajerin Esfahani and Kuhn, 2018)

- In order to account for estimation errors, we shall use a Distributionally Robust Stochastic Optimization (DSRO) model for the HMM probabilities.

The DRSO model

IDEA:

- Construct an ambiguity set $\mathcal{P}$ of distributions for the HMM transition matrix.
- Hedge against the worst probability distribution in the ambiguity set.

- In our case, we solve, in each period,

$$Q_t^j(x_{t-1}, \xi_t) := \max_{x_t \in \mathbb{X}_t(x_{t-1}, \xi_t)} \left\{ f_t(x_t, \xi_t) + \min_{p_j \in \mathcal{P}_j} \sum_{k \in \mathcal{K}} \mathbb{E}\left[Q_{t+1}^k(x_t, \xi_{t+1}) \middle| K_{t+1} = k \right] p_j(k) \right\}.$$

Choosing the ambiguity set

Several approaches have been proposed in the literature to form the ambiguity set.

In our case, since the support is fixed for the HMM, we use the set

$$\mathcal{P}_j = \left\{ p_j \in \mathbb{R}^{|\mathcal{K}|} \;\middle|\; \begin{array}{r} \sum_{k \in \mathcal{K}} p_j(k) = 1 \\ d(p_j, \hat{p}_j) \leq \Delta \\ p_j \geq 0 \end{array} \right\},$$

where $d(p_j, \hat{p}_j)$ measures the total variation distance between p_j and $\hat{p}_j$ (recall that $\hat{p}_j$ is the vector of state-j probabilities estimated for the HMM), i.e.,

$$d(p_j, \hat{p}_j) := (1/2) \sum_{k \in \mathcal{K}} |p_j(k) - \hat{p}_j(k)|.$$

Note that $\Delta \in [0, 1]$.

Reformulating the DRSO model

- Using duality, we can re-write the problem as

$$
\begin{aligned}
Q_t^j(x_{t-1},\xi_t) \quad = \quad & \max_{x_t,\theta^-,\theta^+,\lambda,\eta} \quad f_t(x_t,\xi_t) + \sum_{k\in\mathcal{K}} \hat{p}_j(k)(\theta_k^+ - \theta_k^-) - \eta - 2\Delta\lambda \\
\text{s.t.} \quad & -\theta_k^- + \theta_k^+ - \eta - \mathbb{E}\left[Q_{t+1}^k(x_t,\xi_{t+1}) \mid K_{t+1} = k\right] \le 0, \quad \forall k \in \mathcal{K} \\
& \theta_k^- + \theta_k^+ - \lambda = 0, \qquad \forall k \in \mathcal{K} \\
& \theta^-,\theta^+,\lambda \ge 0 \\
& x_t \in \mathbb{X}_t(x_{t-1},\xi_t).
\end{aligned}
$$

Solution method

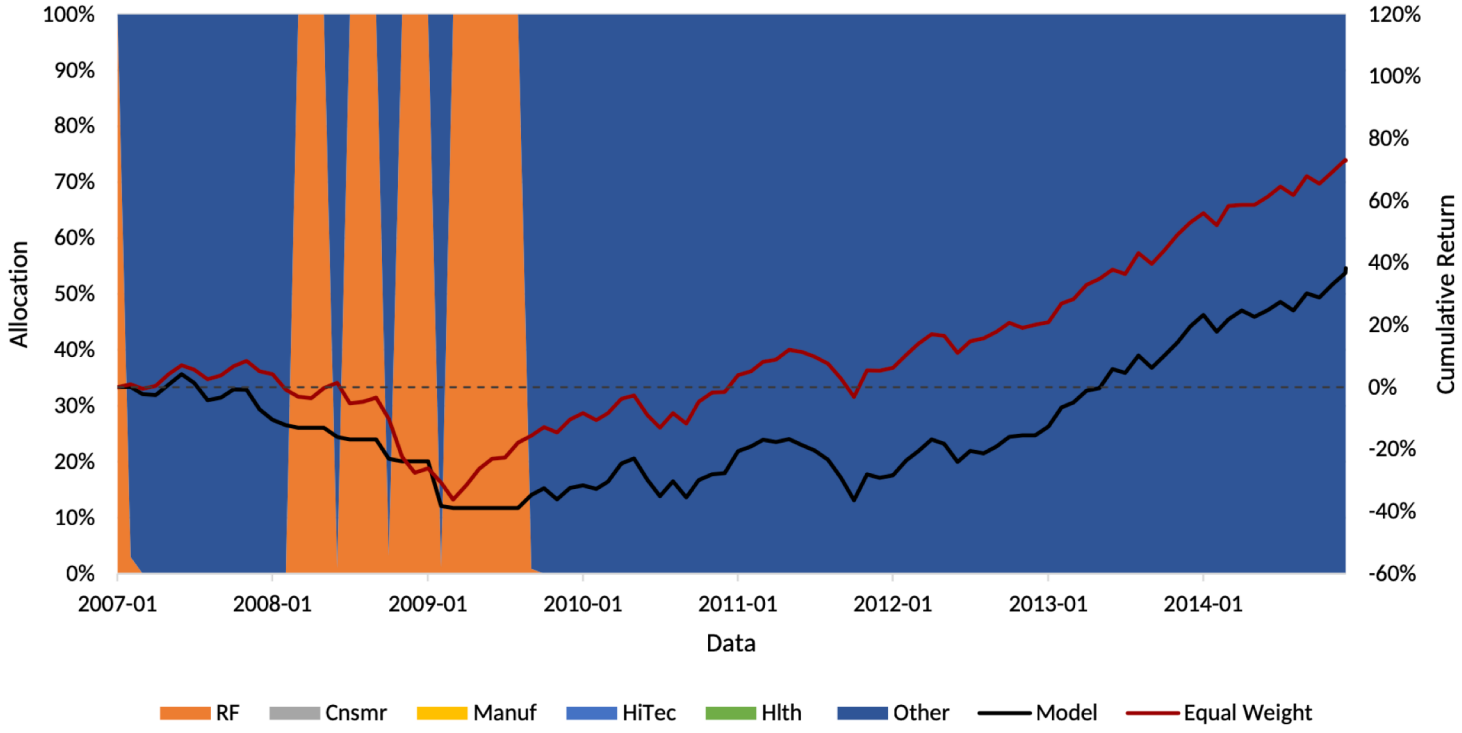
- Assuming that the reward function $f_t(\cdot, \xi_t)$ is linear and that the feasibility set $\mathbb{X}\_t$ is polyhedral, we can solve the problem using a modified version of the SDDP algorithm of Pereira and Pinto (1991).

- Moreover, we provide deterministic upper and lower bounds for the overall objective function (via outer and inner approximations of the value function, respectively), and prove that the gap becomes zero after finitely many iterations w.p.1.

- We use validation data to estimate the hyper-parameter Δ.

Results without ambiguity

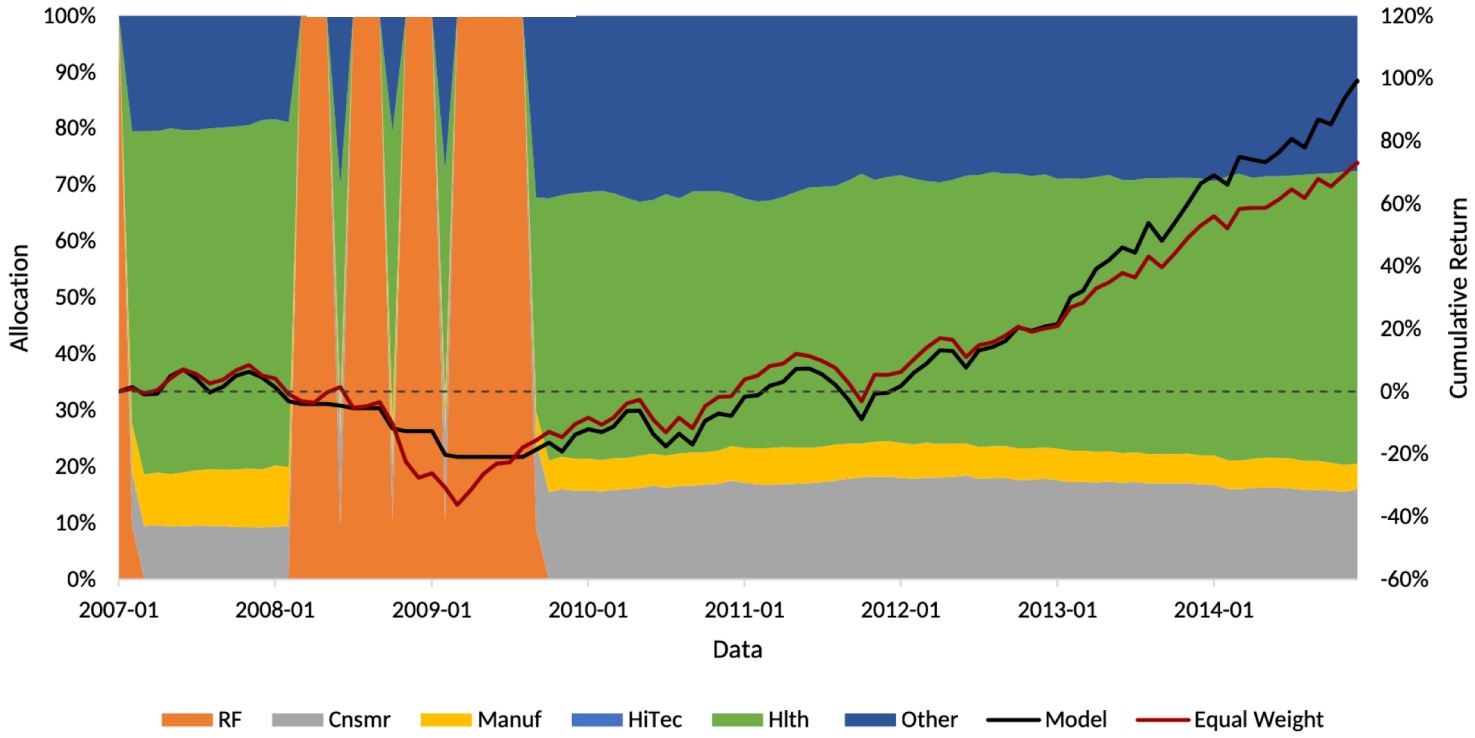
The figure shows the optimal allocation on a monthly basis over an 8-year period



$\Delta = 0$

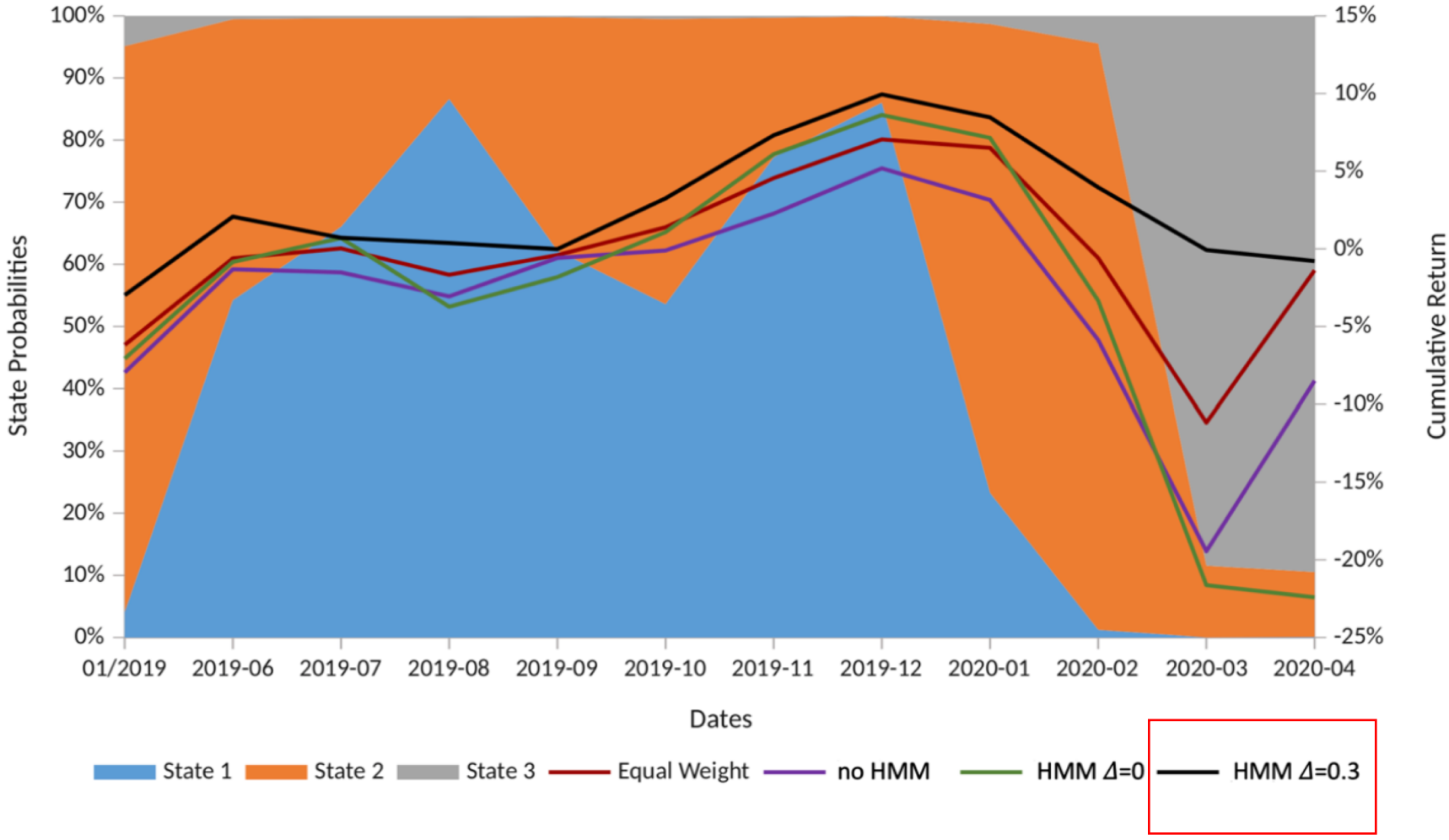
Results WITH ambiguity

The figure shows the optimal allocation on a monthly basis over an 8-year period



$\Delta = 0.2$

Testing data (May 2019 to April 2020)



2nd example: Forecasting loads and reserves

- A classical problem in the electricity sector is that of determining day-ahead energy reserves in the context of uncertain loads and/or generation.
- A common approach goes as follows:
  - Forecasts of loads are made, based on standard statistical techniques.
  - Reserve requirements are defined by applying simple rules to the forecasts (e.g., using the standard deviation).
  - A decision is made to allocate generation resources following an energy and reserve scheduling program.
  - In real-time, reserves are deployed to ensure that power is balanced at every node, compensating for forecast errors.

Forecasting loads

- Ideally, we would like to forecast the loads using probability distributions, and then calculate reserves using for example stochastic programming methods.
  - Indeed, several works in the literature do that.

- Unfortunately, however, oftentimes such an approach is not used by the operators, due to
  - Difficulty to model a proper distribution (e.g., presence of features)
  - Computational costs

    (⇒ small sample sizes ⇒ sample dependency ⇒

    ⇒ compromises market transparency (Wang and Hobbs 2015))

One-scenario forecasts

- As a result, operators often apply some heuristic bias to a point forecast in order to determine the reserves (California ISO 2020).

- Can we do better than that?....

One-scenario forecasts (cont.)

The energy-reserve example illustrates a particular instance of the following general problem:

- Consider a two-stage stochastic optimization problem of the form

$$\min_{x \in X} \underbrace{c^T x + \mathbb{E}[Q(x,\xi)]}_{G(x,\xi)}$$

where $Q(x,\xi) = \min_y \{q^T y \mid Wy + Tx \geq b + H\xi\}$.

- Suppose we can only choose one scenario of ξ to approximate the expectation. Which scenario should we choose?

Two initial ideas

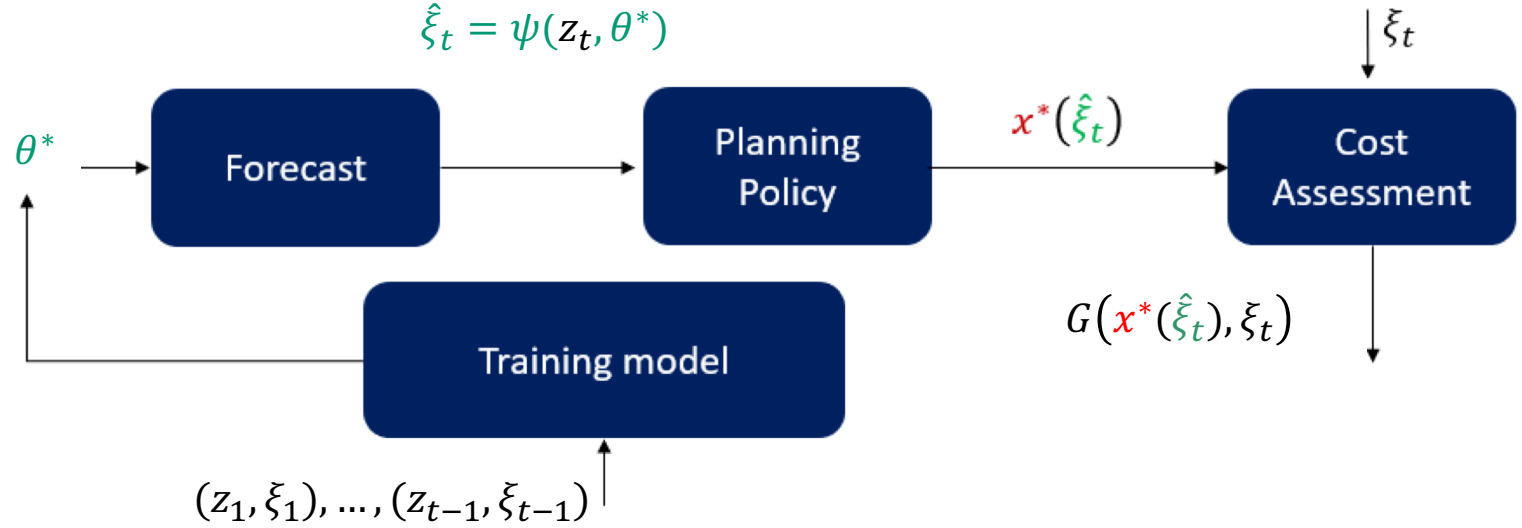
1. Collect data $\xi_1, \ldots, \xi_n$, choose the scenario based on some statistics (mean, median, barycenter, etc.).

2. When the data is accompanied by features $z_1, \ldots, z_n$ (extra attributes of each data point), one can use machine learning to learn the dependence of ξ on z.

   - Let us represent the dependence as $\xi = \psi(z, \theta)$, where θ is a vector of parameters.
   - For example, z can represent demographic or geographic information.
   - Then, given a value of $z = \bar{z}$, we forecast ξ as $\psi(\bar{z}, \theta)$.

A classical open-loop model

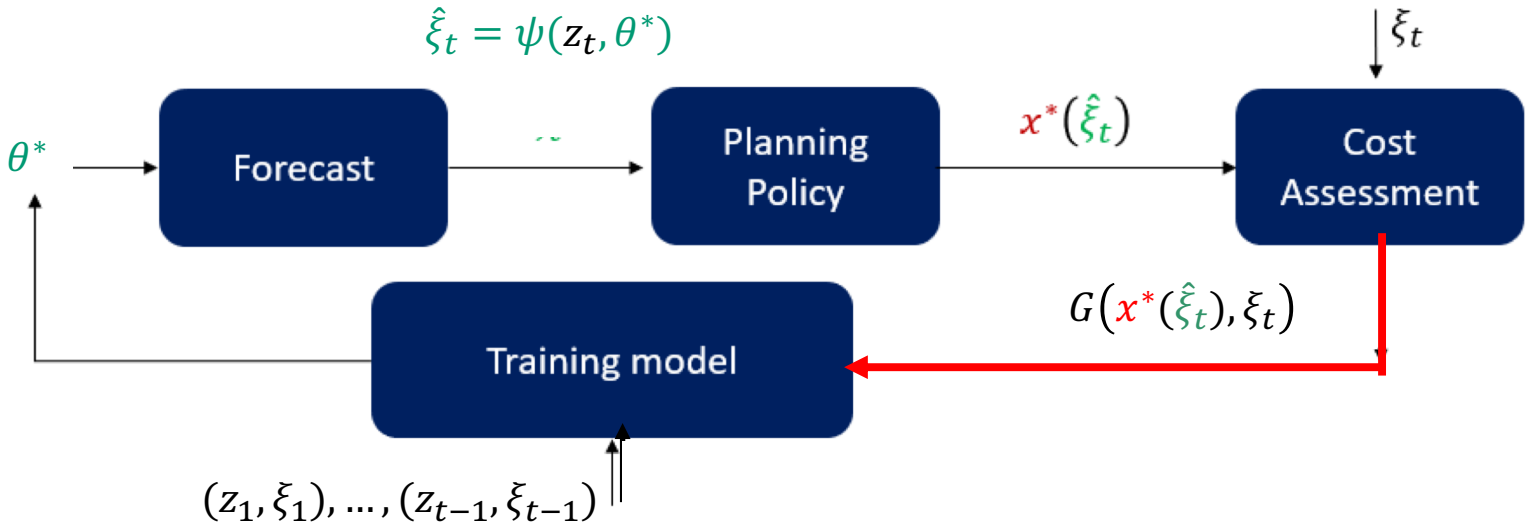
$$\theta^* := \text{argmin} \frac{1}{t-1} \sum_{s=1}^{t-1} \text{loss}(\psi(z_s, \theta), \xi_s)$$



- The key question: Can we improve the forecasting model by incorporating the problem information into it?

A closed-loop predictive model

$$\theta^* := \operatorname{argmin} \frac{1}{t-1} \sum_{s=1}^{t-1} \text{loss}^{\text{P}}(\psi(z_s, \theta), \xi_s)$$



- The function loss^{P} measures the impact of using the estimate $\psi(z_s, \theta)$ of ξ_s in the objective function of the problem!

Comments

- The above idea can be viewed as a way of biasing the point forecast in an optimal fashion, using the structure of the problem.

- The idea of using problem information to bias the forecasts was first proposed by Bengio (1997) and has recently been revisited and enhanced in a number of publications.
  - Donti, Amos, and Kolter (2017)
  - Elmachtoub and Grigas (2022)
  - Muñoz, Pineda, and Morales (2022)

Comments

- Some recent works have also studied the idea of doing problem-based scenario generation for stochastic programming.
  - Arpón, HdM and Pagnoncelli (2018)
  - Fairbrother, Turner and Wallace (2022)

  } Risk-averse setting

  - Bertsimas and Mundru (2022)
  - Henrion and Römisch (2022)
  - Keutchayan, Ortmann, and Rei (2021)

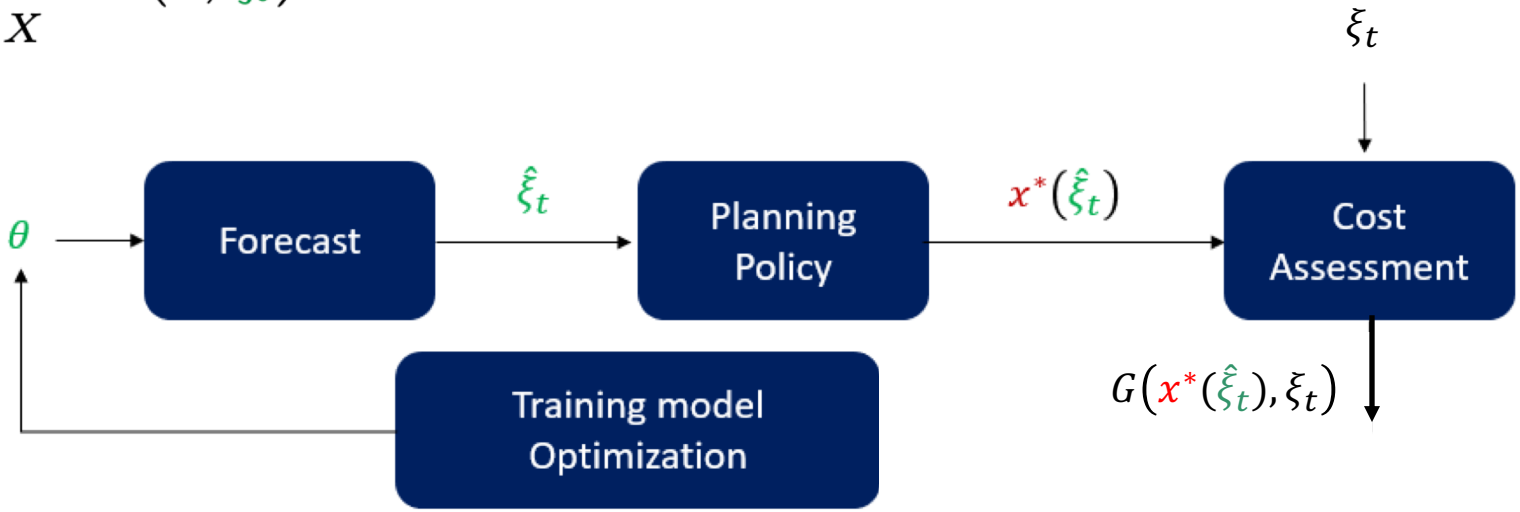
- How to model the closed-loop problem?

Formulating the open-loop model

$$\theta_T = \arg\min_{\theta \in \Theta} \quad \frac{1}{T} \sum_{t \in \mathbb{T}} \text{loss}\left(\hat{\xi}_t, \xi_t\right),$$

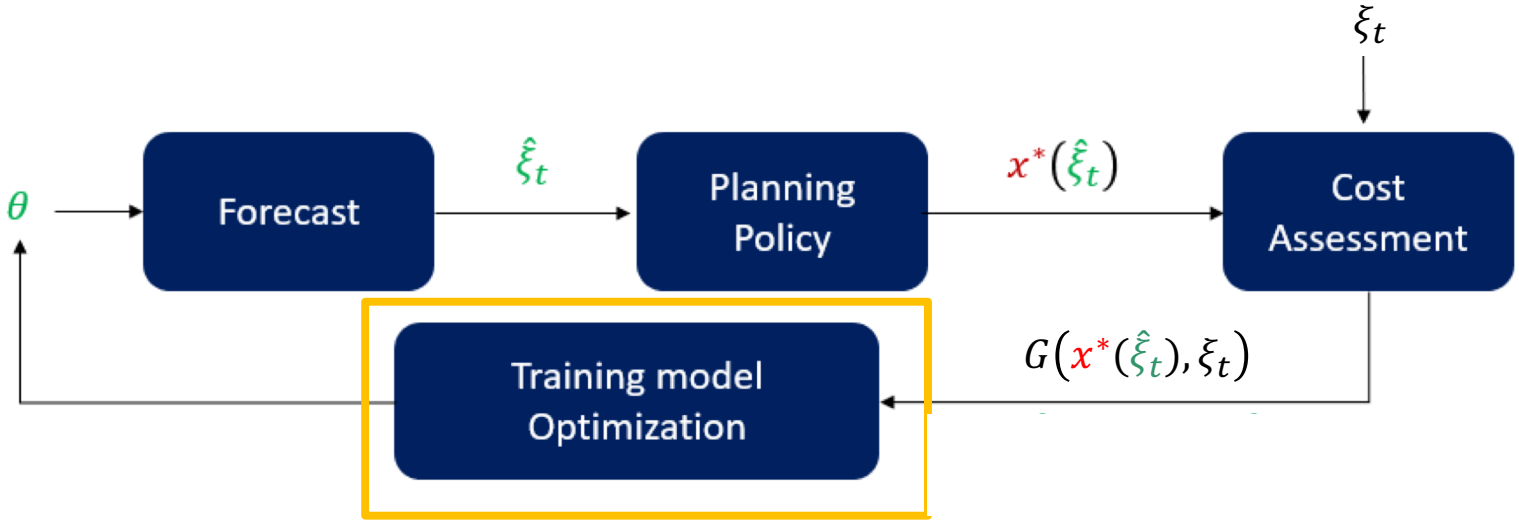
$$s.t. \quad \hat{\xi}_t = \Psi(z_t, \theta) \; \forall t \in \mathbb{T}$$

$$x_t^* \in \arg\min_{x \in X} G(x, \hat{\xi}_t) \; \forall t \in \mathbb{T}$$



Formulating the closed-loop model

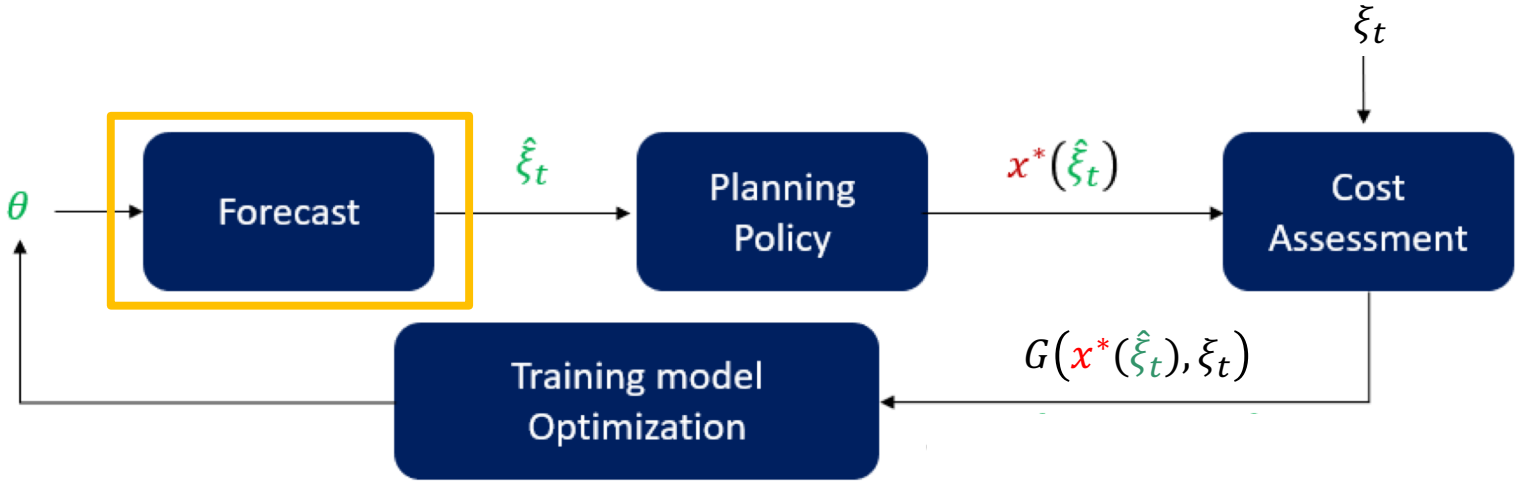
$$\theta_T = \underset{\theta \in \Theta, \hat{\xi}_t, x_t^*}{\arg\min} \quad s.t.$$



Formulating the closed-loop model

$$\theta_T = \underset{\theta \in \Theta, \hat{\xi}_t, x_t^*}{\arg\min}$$

$$s.t. \quad \hat{\xi}_t = \Psi(z_t, \theta) \ \forall t \in \mathbb{T}$$



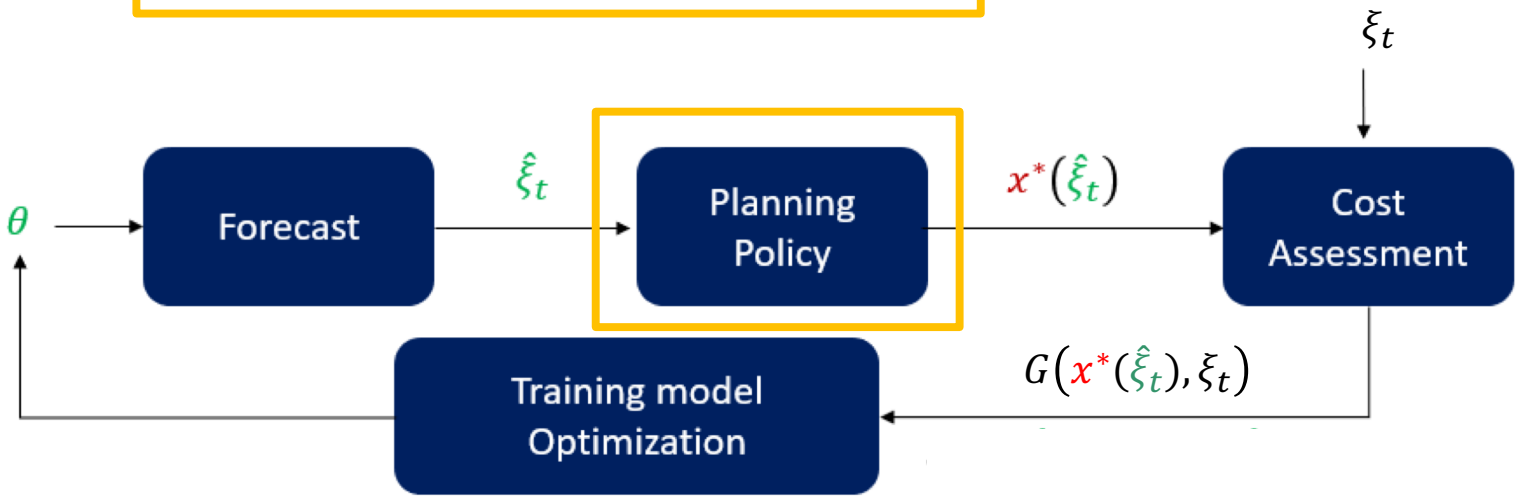
Formulating the closed-loop model

$$\theta_T = \underset{\theta \in \Theta, \hat{\xi}_t, x_t^*}{\arg\min}$$

$$s.t. \quad \hat{\xi}_t = \Psi(z_t, \theta) \; \forall t \in \mathbb{T}$$

$$x_t^* \in \underset{x \in X}{\arg\min} \, G(x, \hat{\xi}_t) \; \forall t \in \mathbb{T} \qquad (*)$$

Optimal solution corresponding to θ

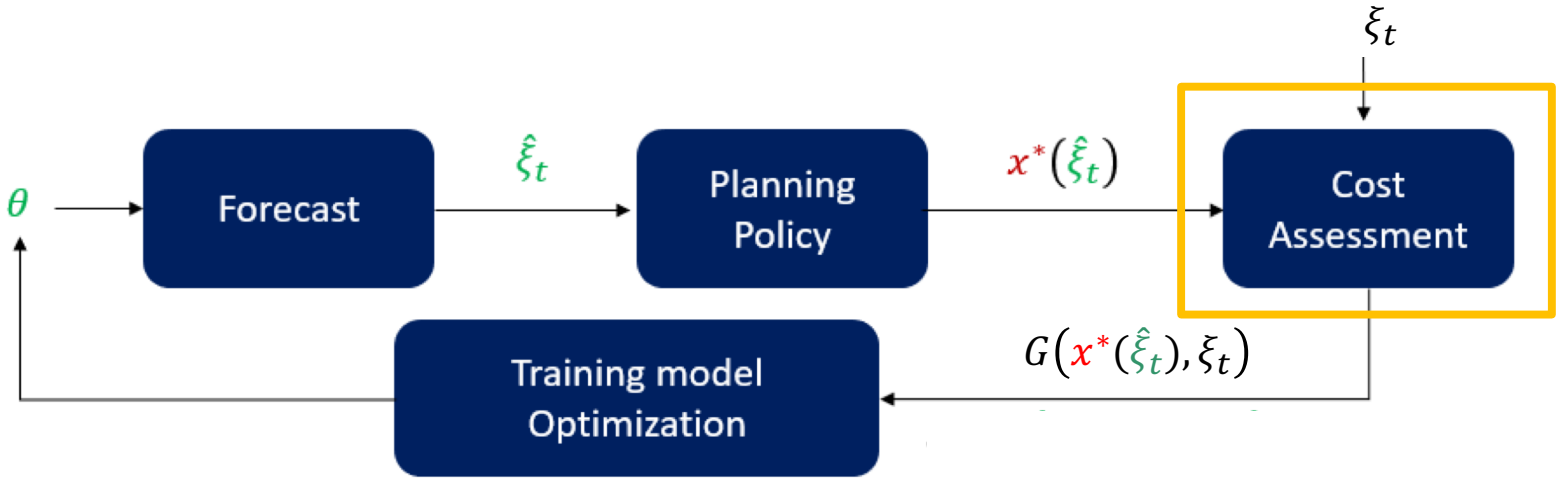


Formulating the closed-loop model

$$\theta_T = \underset{\theta \in \Theta, \hat{\xi}_t, x_t^*}{\arg\min} \; \frac{1}{T} \sum_{t \in \mathbb{T}} G(x_t^*, \xi_t)$$

$$s.t. \quad \hat{\xi}_t = \Psi(z_t, \theta) \; \forall t \in \mathbb{T}$$

$$x_t^* \in \underset{x \in X}{\arg\min} \, G(x, \hat{\xi}_t) \; \forall t \in \mathbb{T} \qquad (*)$$



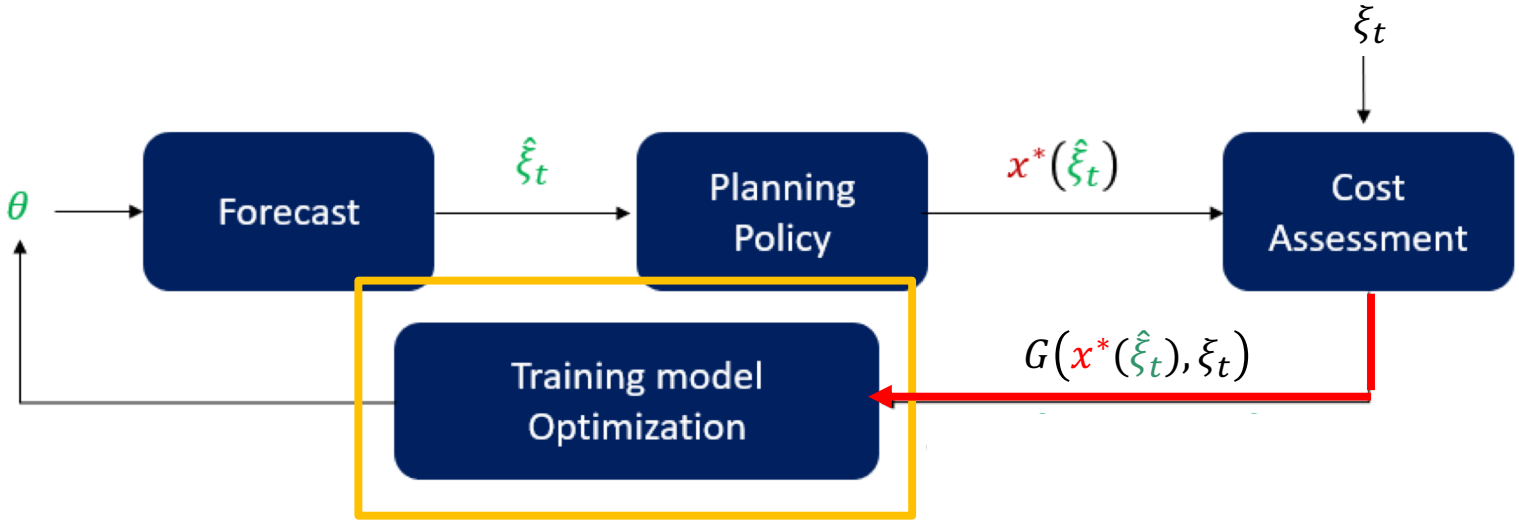
Formulating the closed-loop model

$$\theta_T = \underset{\theta \in \Theta, \hat{\xi}_t, x_t^*}{\arg\min} \quad \frac{1}{T} \sum_{t \in \mathbb{T}} G(x_t^*, \xi_t)$$

$$s.t. \quad \hat{\xi}_t = \Psi(z_t, \theta) \; \forall t \in \mathbb{T}$$

$$x_t^* \in \underset{x \in X}{\arg\min} \, G(x, \hat{\xi}_t) \; \forall t \in \mathbb{T} \qquad (*)$$

A bi-level problem!



Does the method converge?

Not trivial because of arg min function. We need some assumptions:

Assumption 1: The solution set of the optimization problem in (*) is a singleton for all possible values of $\hat{\xi}_t$.

- This assumption holds, for example, if a lexicographic method is used to solve (*).
- Then, we can define the vector-valued mapping

$$h(\xi) := \arg\min_{x \in X} G(x, \xi)$$

which is continuous and piecewise linear (Böhm 1975, Borelli et al. 2003).

Convergence

Assumption 2: The feasibility set X that appears in $(*)$ is a non-empty and bounded polyhedron.

- Easily enforceable.

Assumption 3: The feasibility set D of the dual of the problem that defines $Q(x, \xi)$ is non-empty and bounded.

- Note that D does not depend on x or ξ.

Convergence

THEOREM: Consider the training process defined above and its output θ_T. Suppose that

(i) Assumptions 1-3 hold,

(ii) the forecasting function $\Psi(\cdot,\cdot)$ is continuous in both arguments,

(iii) the data process $(Z_1, \xi_1), \ldots, (Z_T, \xi_T)$ is independent and identically distributed (with (Z, ξ) denoting a generic element),

(iv) the random variable ξ is integrable, and

(v) the set Θ is compact.

Then,

$$\lim_{T\to\infty} d(\theta_T, S^*) = 0$$

where $d(\cdot,\cdot)$ is the Euclidean distance from a point to a set and S^* is defined as

$$S^* = \arg\min_{\theta\in\Theta} \mathbb{E}\left[G\big(h(\Psi(Z,\theta)),\xi\big)\right]$$

with $h(\xi) := \arg\min_{x\in X} G(x,\xi)$

Comments

- Once θ_T is obtained, given a new observation z_{new} we compute $\xi^* := \Psi(z_{new}, \theta_T)$ (optimal forecast) and

$$x^* = \arg\min_{x \in X} G(x, \xi^*)$$

- The iid assumption can be relaxed to the case where (Z_n, ξ_n) is an ergodic process.
  - This holds for instance when $\{\xi_n\}$ is a stationary AR(1) process and $Z_n = \xi_{n-1}$.

- The approach can be easily extended to the case where the G function in the upper level (e.g. planning) is different from the one in the lower level (e.g. operation).

Solving the bi-level problem

- Bi-level problems that are relatively small can be solved to global optimality with MIP solvers and KKT reformulations.
  - One example is the BilevelJuMP.jl library (Dias-Garcia et al. 2021).

- We have also devised a heuristic method for our case for larger problems.

Pseudo-algorithm for heuristic

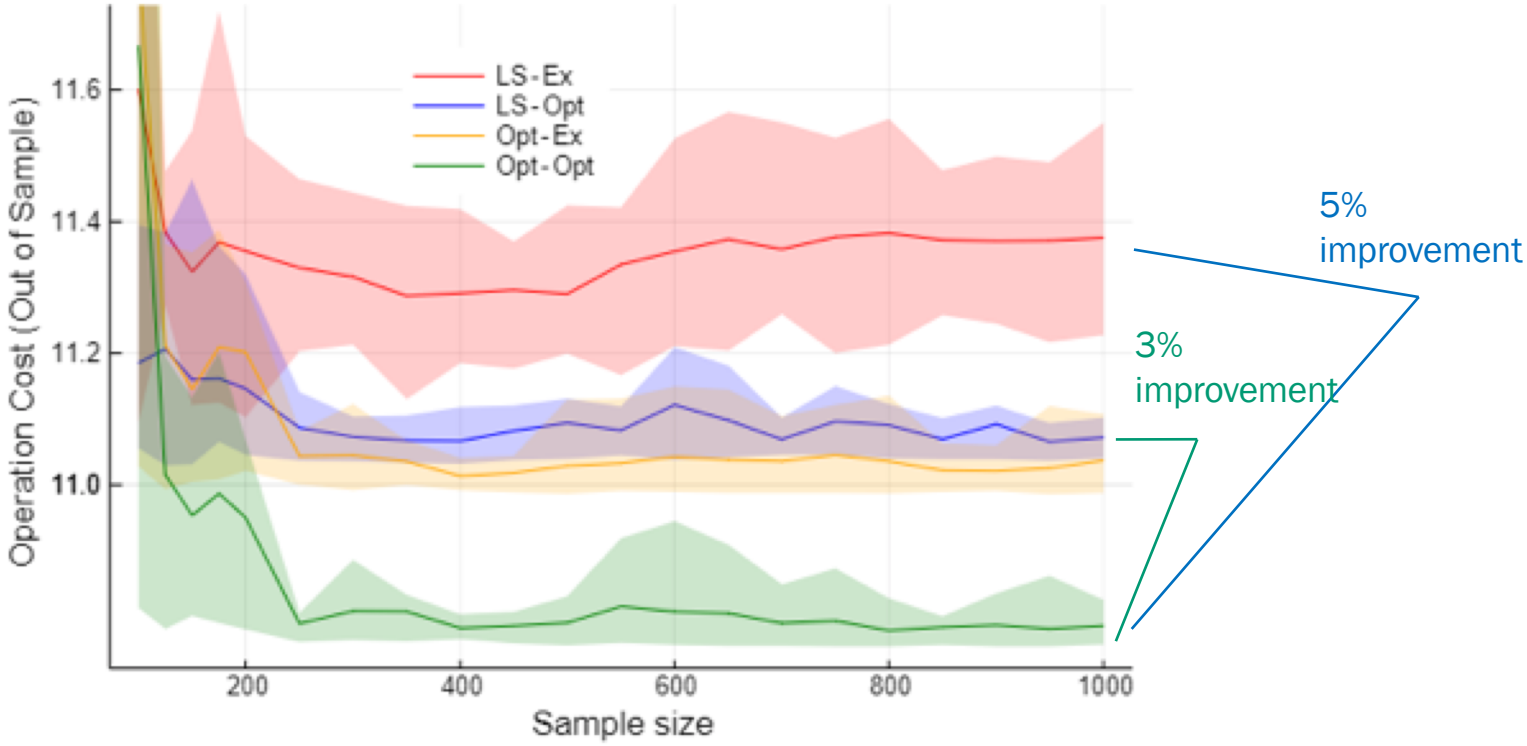
Initialize θ;
while (Not converged) do
    Update θ;
    for $t \in \mathbb{T}$ do
        Forecast: $\hat{\xi}_t := \Psi(z_t, \theta)$
        Plan Policy: $x_t^* := \arg\min_{x \in X} G(x, \hat{\xi}_t)$
        Cost Assessment: $\text{cost}_t := G(x_t^*, \xi_t)$
    end
    Compute cost: $\text{cost}(\theta) := \sum_{t \in \mathbb{T}} \text{cost}_t$
end

Numerical results

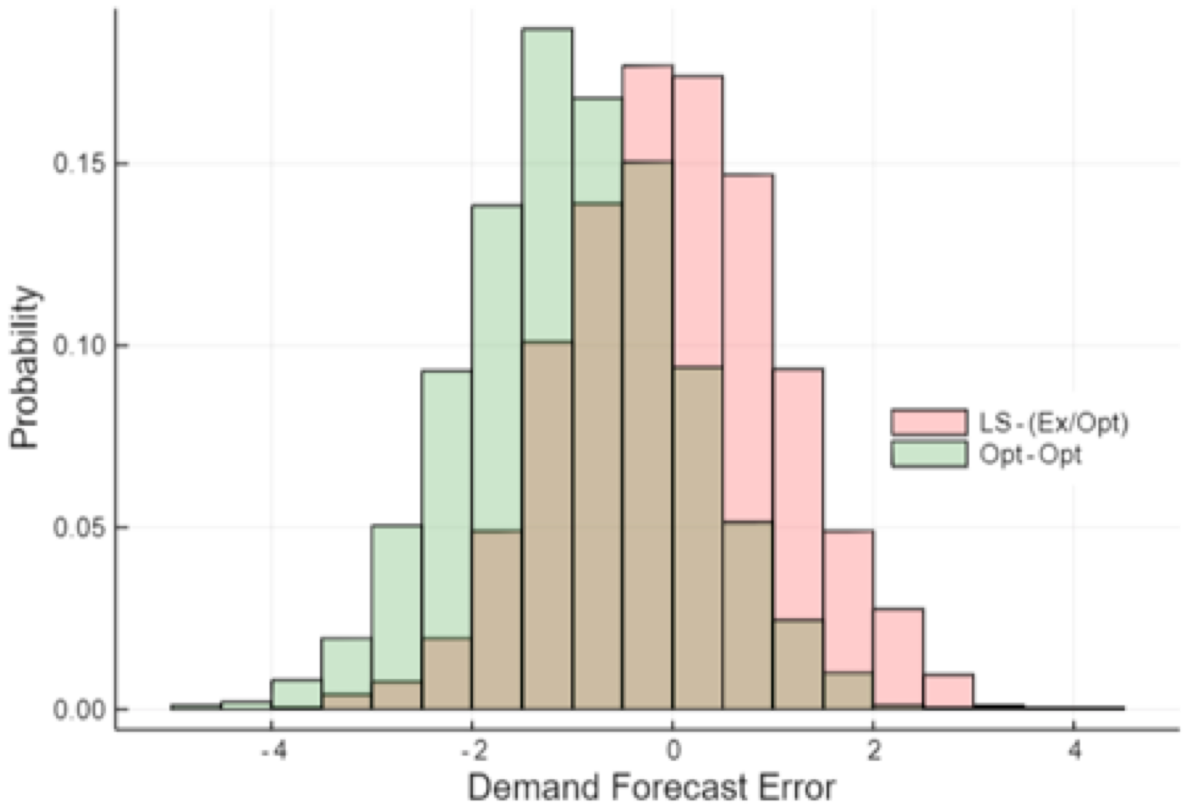
- The key to analyze the performance of the proposed approach is to evaluate it with an out-of-sample procedure using the optimal values of θ found by the algorithm.
- We compared four methods:
  - LS-Ex (red)
    Least-Squares load, Exogenous reserves (base case)
  - LS-Opt (blue)
    Least-Squares load, Optimized reserves
  - Opt-Ex (orange)
    Optimized load, Exogenous reserves
  - Opt-Opt (green)
    Optimized load, Optimized reserves

Single-bus example

- 100 trials, all trials are evaluated on a single out-of-sample dataset with size 10, 000 observations.
- Shaded areas represent the 10% and 90% quantiles.



Forecast bias (single-bus example)



Larger problems

System	Model	Test Cost($)		Train Cost ($)		Train Time (s)	
		Mean	Std	Mean	Std	Mean	Std
24	LS-Ex	414.70	1.14	397.01	4.39	0.00	0.00
	LS-Opt	398.48	0.69	378.49	93.00	450.81	90.35
	Opt-Opt	398.20	1.04	376.35	14.04	643.89	11.16
118	LS-Ex	2956.01	11.33	3163.74	3.78	0.00	0.00
	LS-Opt	2829.86	4.20	3041.28	4.86	639.87	4.97
	Opt-Opt	2815.85	4.57	3029.93	13.20	911.43	12.37
300	LS-Ex	7697.25	40.78	7646.84	26.09	0.00	0.00
	LS-Opt	7329.44	36.62	7278.88	18.75	803.15	34.08
	Opt-Opt	6820.47	37.67	6787.65	294.53	2748.17	303.56

0.5% improvement over LS-Opt

7% improvement over LS-Opt

Large-scale problems

Buses	Time	Test				
		Opt-Opt		LS-Opt		LS-Ex
	(h)	($)	(%)	($)	(%)	($)
600	4	15256	8.87	16741	13.87	19437
600	12	15226	9.04	16739	13.88	19437
1200	4	35261	12.07	40103	16.28	47899
1200	12	35158	9.49	38843	18.91	47899
1800	4	60043	13.43	69355	5.98	73770
1800	12	53378	14.98	62786	14.89	73770
2400	4	81018	14.43	94675	6.78	101561
2400	12	80330	14.30	93739	7.70	101561
3000	4	120389	4.05	125465	1.19	126972
3000	12	110302	10.00	122560	3.47	126972
3600	4	149141	2.83	153484	0.69	154558
3600	12	136479	9.20	150303	2.75	154558
4200	4	177451	1.30	179785	0.43	180555
4200	12	165963	6.47	177444	1.72	180555
4800	4	206358	0.93	208300	0.29	208910
4800	12	197707	4.28	206539	1.13	208910
5400	4	232071	0.82	233992	0.23	234524
5400	12	225203	3.20	232658	0.80	234524
6000	4	260427	0.79	262493	0.18	262963
6000	12	255384	2.34	261500	0.56	262963

Improvement of Opt-Opt over LS-Opt

Improvement of LS-Opt over LS-Ex

Conclusions

- The integration of learning from data and optimization can be highly beneficial!
- There are multiple ways to do the integration; the "optimal" one might be problem-dependent.
- Whatever stochastic optimization problem we solve, it is crucial to do out-of-sample experiments to check the quality of the obtained solutions.

THANK YOU !!!!

Main references:

Dias-Garcia J, Street A, Homem-de-Mello T and Muñoz FD (2021) Application-driven learning via joint prediction and optimization of demand and reserves requirement. Preprint available at *Optimization Online*,

http://www.optimization-online.org/DB_FILE/2021/03/8291.pdf.

Silva, T, Valladão D, and Homem-de-Mello, T (2021) A data-driven approach for a class of stochastic dynamic optimization problems, *Computational Optimization and Applications*, pp. 1–43.