End-to-End Learning for Information Gathering

Rares Cristian
Meta (previously MIT)
raresc2105@gmail.com

Pavithra Harsha
IBM Research
pharsha@us.ibm.com

Georgia Perakis
MIT Sloan Business School
georgiap@mit.edu

Brian Quanz
IBM Research
blquanz@us.ibm.com

Abstract

This paper introduces an end-to-end, or joint prediction and optimization, framework for the class of two-stage contextual optimization problems with information-gathering. We showcase the approach on a dynamic electricity-scheduling problem on real data. We show that the adaptiveness of the end-to-end approach indeed provides benefits over other methods which train their forecasting method independently of the first information-gathering stage.

1 Introduction

We consider the problem of two-stage contextual stochastic optimization problems where the first stage consists of data-gathering or uncertainty quantification. In this class of problems, there is an initial stage before the decision-making step where one is allowed to gather information about some of the random variables ahead of time. For example, one can send out a survey or set up a poll to better understand demand at a particular location out of an array of warehouses. Given this new information, one can then make more informed predictions about the rest of the random variables (for e.g. demand at the remaining locations), and subsequently make a more informed decision. If demand across locations is correlated, one can gain significant information from polling a single location. As a first stage, one must decide which location to poll, observe information about this location. As the second stage, one then makes a new prediction and a decision for all locations conditioned on this new observation in the first stage. So, we propose to answer three questions (1) which random variable should we "poll" (2) what predictions to make conditioned on observing this chosen variable and (3) what decision to make based off of these predictions.

As another example which we also use to apply our method computationally, consider an electricity-scheduling problem. The goal is to make a generation schedule and decide on the amount of electricity to generate hour per hour, over the next 24 hours. We consider the problem in two stages. First, we make an initial forecast for the 24 hours dependent only on feature information for that day. Then we decide on a time t to update the schedule. Up to time t, we use the initial forecast to generate electricity, then given the new observations of true demand up to time t we regenerate the forecast and electricity schedule for the rest of the day. There is now a balancing act in deciding what hour t to change the schedule. If we wait longer, we gain a better estimate of future demand, however we also use a worse forecast up to the waiting time t. In this scenario, polling is equivalent to waiting until time t and observing the exact demand until that time.

One drawback of existing methods is the need to learn the forecasting model independently of the polling problem. Consider again the electricity scheduling problem. First, we must make an initial forecast for the full 24 hours. This initial forecast will affect the decision we make for t, how long

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: MLxOR: Mathematical Foundations and Operational Integration of Machine Learning for Uncertainty-Aware Decision-Making.

to wait to update the forecast. Hence, we propose an *end-to-end* approach to this problem where the forecasting models and the decision-making model (of what variables to poll/how long to wait) are trained jointly as part of a single optimization problem. We also show computational results of our method applied to an *electricity generation and scheduling problem*. Here we make an initial forecast, and must decide on the optimal time to update the forecast. This involves learning how to balance the benefits of waiting for more accurate information against the costs of delaying decisions.

2 Information gathering

We first formally describe the problem. First we consider the setting without the initial information gathering stage. One wishes to makes decisions $\mathbf{v} \in \mathcal{V}$ in feasible region $\mathcal{V} \subset \mathbb{R}^d$. Associated with this decision is a cost function $c(\mathbf{v}, \mathbf{z})$ dependent on a random variable $\mathbf{z}$. In turn, the distribution of $\mathbf{z}$ also depends on the decision contextual information $\mathbf{x}$. We say $\mathbf{z}$ is distributed according to some unknown distribution $\mathbf{Z}|\mathbf{x}$. As an example, $\mathbf{z} = (z_1, \dots \mathbf{z}_d)$ corresponds to demand for an item at d different locations and v is the corresponding inventory decision.

In the first stage, the decision space will consist of choosing some index $w \in \mathcal{W} = \{1, \dots, d\}$ to survey, or gain more information about, the w^{th} entry of $\mathbf{z}$, namely z_w. This could be more general beyond observing a single value, for example observing multiple values. We will see this in an experiment in section 3. But we will keep this simple here for the sake of notation. In the second stage, we are given some auxiliary decision variables $\mathbf{v} \in \mathcal{V}$ with objective function $c(\mathbf{v}, \mathbf{z})$. In short, the entire process is as follows:

1. For an out-of-sample point $\mathbf{x}$, we make a decision w to observe $z_w \sim Z_w|\mathbf{x}$.
2. Given the observation z_w, the full vector $\mathbf{z}$ is distributed according to $\mathbf{Z}|\mathbf{x}, Z_w = z_w$.
3. We are now given some second-stage decision-making problem with variables $\mathbf{v} \in \mathcal{V}$ with objective $c(\mathbf{v}, \mathbf{z})$ and we wish to make decision $\mathbf{v}$ minimizing expected cost:
$$\min_{\mathbf{v} \in \mathcal{V}} \mathbb{E}_{\mathbf{z} \sim Z|\mathbf{x}, Z_w = z_w}[c(\mathbf{v}, \mathbf{z})]. \tag{1}$$
4. Ultimately, we wish to know which observational decision w will minimize overall loss:
$$\min_{w \in \mathcal{W}} \mathbb{E}_{z_w \sim Z_w|\mathbf{x}} \left[\min_{\mathbf{v} \in \mathcal{V}} \mathbb{E}_{\mathbf{z} \sim Z|\mathbf{x}, Z_w = z_w}[c(\mathbf{v}, \mathbf{z})] \right]. \tag{2}$$

In terms of data, we observe N points $(\mathbf{x}^n, w^n, \mathbf{z}^n), n = 1 \dots, N$ where $\mathbf{z}^n$ is distributed according to an (unknown) distribution $Z|\mathbf{x}^n$. Given decision w^n, we observe the realization of $\mathbf{z}^n_{w^n}$ before making the second-stage decision.

2.1 End-to-end information-gathering

At a high level, our goal is to remove the expectations in (2) by point forecasts / deterministic functions we can optimize over. To illustrate, first consider the inner minimization problem (1). After making a decision w of observing z_w, we can make a forecast for some statistic of the distribution $Z|\mathbf{x}, Z_w = z_w$. Let $p(\mathbf{x}, z_w)$ denote this point statistic. Ideally, optimizing $\mathbf{v}$ with respect to the predictions $p(\mathbf{x}, z_w)$ would give the same decision as optimizing with respect to the true distribution of $\mathbf{z}$, namely $\mathbf{Z}|\mathbf{x}, Z_w = z_w$:
$$\arg\min_{\mathbf{v} \in \mathcal{V}} c(\mathbf{v}, p(\mathbf{x}, z_w)) \approx \arg\min_{\mathbf{v} \in \mathcal{V}} \mathbb{E}_{\mathbf{z} \sim \mathbf{Z}|\mathbf{x}, Z_w = z_w}[c(\mathbf{v}, \mathbf{z})]. \tag{3}$$

The problem above in (3) can be solved by a more classical end-to-end approach (see for example [3, 4]). We learn such a p by solving the following empirical risk minimization problem,
$$\min_p \sum_{n=1}^N c(v^*(p(\mathbf{x}^n, z^n_{w^n})), \mathbf{z}^n); \quad v^*(\mathbf{z}) = \arg\min_{\mathbf{v} \in \mathcal{V}} c(\mathbf{v}, \mathbf{z}) \tag{4}$$

Now, the information-gathering problem (2) reduces to
$$\min_w \mathbb{E}_{z_w \sim \mathbf{Z}_w|\mathbf{x}} \left[\mathbb{E}_{\mathbf{z} \sim \mathbf{Z}|\mathbf{x}, Z_w = z_w}[c(v^*(p(\mathbf{x}, z_w)), \mathbf{z})] \right]. \tag{5}$$

Algorithm 1 End-to-end information-gathering

Input: Training data $\{(\mathbf{x}^n, \mathbf{z}^n, \mathbf{v}^n)\}_{n=1}^N$, out-of-sample $\mathbf{x}$
Output: First-stage decision w and second-stage decision $\mathbf{v}$.
Learn model $p(\mathbf{x}, z_w)$ to predict $\mathbf{z}$ conditioned on observing z_w. Learn p by solving (4) with gradient descent. Compute gradients $\partial v^*(\mathbf{z})/\partial \mathbf{z}$ using prior work such as [3, 2, 1].
Learn point forecast $f(w, \mathbf{x})$ by solving (7) by gradient descent.
For out-of-sample $\mathbf{x}$, choose decision w by solving (8).
For out-of-sample $\mathbf{x}$ and decision w, observe z_w. And take second-stage decision $v^*(p(\mathbf{x}, z_w))$.

In a similar spirit as above, we wish to learn a single point statistic $f(w, \mathbf{x})$ to replace the expectation over $\mathbf{z}$. That is, our goal is to learn a function f so that

$$c\Big(v^*\big(p(\mathbf{x}, f_w(w, \mathbf{x}))\big), f(w, \mathbf{x})\Big) \approx \mathbb{E}_{\mathbf{z}\sim\mathbf{Z}|\mathbf{x}}[c(v^*(p(\mathbf{x}, z_w)), \mathbf{z})]. \tag{6}$$

We replace $\mathbf{z}$ with a point forecast $f(w, \mathbf{x})$. We propose to do this by the following empirical minimization problem:

$$\min_f \sum_{i=1}^n \Big(c\Big(v^*\big(p(\mathbf{x}^i, f_{w^i}(w^i, \mathbf{x}^i))\big), f(w^i, \mathbf{x}^i)\Big) - c\Big(v^*\big(p(\mathbf{x}^i, z^i_{w^i})\big)\big), \mathbf{z}^i\Big)\Big)^2. \tag{7}$$

Finally, for an out-of-sample $\mathbf{x}$, we make decisions by solving

$$\arg\min_w c\Big(v^*\big(p(\mathbf{x}, f_w(w, \mathbf{x}))\big), f(w, \mathbf{x})\Big). \tag{8}$$

In practice, we cannot observe z_w before making decision w, so in (7) and (8) we "observe" the w^{th} entry of the predicted $f(w, \mathbf{x})$ instead. Algorithm 1 provides a concise description.

Proposition 1 (Learning guarantees) *For continuous functions $c(\mathbf{v}, \mathbf{z})$, and continuous $v^*(\mathbf{z})$ both with respect to $\mathbf{z}$, there exists a point forecast $f(w, \mathbf{x})$ in the convex hull of the support of $\mathbf{Z}|\mathbf{x}$ such that the learning goal in* (6) *can be achieved.*

We can simplify our reasoning by viewing the objective/cost function $c(v^*(p(\mathbf{x}, z_w)), \mathbf{z})$ as a function $g(\mathbf{z})$, ignoring the special structure. This function is also continuous in $\mathbf{z}$. Our goal is to show that there exists some point forecast $f(w, \mathbf{x})$ such that $g(f(w, \mathbf{x})) = \mathbb{E}_{\mathbf{z}\sim\mathbf{Z}|\mathbf{x}}[g(\mathbf{z})]$. Take any two points $\mathbf{z}^1, \mathbf{z}^2$ such that $g(\mathbf{z}^1) < \mathbb{E}_{\mathbf{z}\sim\mathbf{Z}|\mathbf{x}}[g(\mathbf{z})] < g(\mathbf{z}^2)$. Since g is a continuous function with respect to $\mathbf{z}$, there must exist a convex combination $\hat{\mathbf{z}}$ of $\mathbf{z}^1, \mathbf{z}^2$ so that $g(\hat{\mathbf{z}}) = \mathbb{E}_{\mathbf{z}\sim\mathbf{Z}|\mathbf{x}}[g(\mathbf{z})]$. We set $f(w, \mathbf{x}) = \hat{\mathbf{z}}$.

3 Computational results

We consider an electricity generation scheduling problem using data from PJM, an electricity routing company coordinating the movement of electricity throughout 13 states. The goal is to make a generation schedule and decide on the amount of electricity to generate hour per hour, over the next 24 hours. We consider the problem in two stages. First, we make an initial forecast for the 24 hours dependent only on feature information for that day. Then we decide on a time w to update the schedule. Up to time w, we use the initial forecast to generate electricity, then given the new observations of true demand up to time w we regenerate this forecast and generation schedule for the rest of the day. There is now a balancing act in deciding what hour w to change the schedule. If we wait longer, we gain a better estimate of future demand, however we also use a worse forecast up to the waiting time w. Finally, we define the objective function. The operator incurs a unit cost γ_e for excess generation and a cost γ_s for shortages. The cost of generating $v_1, \ldots, v_{24}$ while true demand is $z_1, \ldots, z_{24}$ is given by $c(\mathbf{v}, \mathbf{z}) = \sum_{i=1}^{24} \gamma_s \max\{z_i - v_i, 0\} + \gamma_e \max\{v_i - z_i, 0\}$.

Methods: There are three components for each model: (1) how to make the initial forecast, (2) which time w to choose, (3) how to update the forecast given w. We first introduce two baselines which always choose $w = 0$, so they never observe any of the day's demand, and only use their initial forecast. (1) We consider a predict-*then*-optimize approach, where we learn a demand function independent of the decision-making step. We refer to this as "Predict then optimize." (2) We learn an end-to-end model which aims to directly minimize decision cost c (for ex., as in [3]). We refer to this as "Vanilla E2E" (vanilla end-to-end).

We then introduce baselines that choose w in different ways, including our proposed approach. Each of these methods use the same model for the initial forecast, and for making the updated forecast (components (1) and (3) above). We choose the vanilla end-to-end method for this initial forecast since it performs significantly better than the 2-stage method. We only vary the method to decide w. The goal is to highlight the differences in objective cost resulting from various methods of choosing w. Here we train a model p to predict future demand given observations $z_1, \ldots, z_w$, as well as features $\mathbf{x}$. Note that here we observe all variables up to time w. Up to time w, the baseline decisions are made by the vanilla end-to-end approach. After time w, the schedule is made according to p, based on *true* demand up to time w.

We have three baselines for methods on choosing w. (1) choosing a (uniformly) random action w. We refer to this as "Random." (2) We fix a single w for all data points (choosing this best w from training data). We refer to this as "Single action." Finally, (3) the optimal in-hindsight decision w which may change for every out-of-sample data point $\mathbf{x}$. We refer to this as "Optimal." We will denote our proposed method to decide w as "Endogenous E2E" (endogenous end-to-end). This entails solving eq. (7) by gradient descent for f and choosing decision w by solving eq. (8). As a final baseline, we also compare against a more traditional approach: for each decision w and features $\mathbf{x}$, predict directly the loss/cost of this decision. This does not take into account the structure of the problem and simply minimizes mean-squared error between predicted cost and observed cost of each action on the training data. We refer to this as "Cost Learner."

Results: In table 1, we report the average difference between decision cost of our method and the other methods for each datapoint. Our approach is 7.5% better than the cost-learning method and less than 8% worse than optimal on average. We also measure the median cost of each method, as well as the percentage of test datapoints on which our approach performs better than every other approach. For example, our method only outperforms the predict-then-optimize method 66% of the time, indicating this method does well on some data, but on also performs significantly more poorly on others, likely where it proposes a shortage (not knowing that a shortage is significantly worse than excess, since mean-squared error loss is unaware of this). In addition, we also plot the cost distribution of each method on the test data in Figure 1 alongside the optimal cost distribution. We observe that our proposed method most closely aligns with the optimal cost everywhere. Knowing the additional structure of the problem, our approach can better learn it.

Method	Average difference	Median cost	% Endo. E2E Wins
Predict then optimize	710%	0.588	66%
Vanilla E2E	55%	0.339	90%
Cost Learner	7.5%	0.220	71%
Endogenous E2E	0%	**0.204**	100%
Random	21%	0.264	88%
Single action	20%	0.261	81%
Optimal	-7.9%	0.187	—

Table 1: Electricity scheduling: cost comparison across methods.

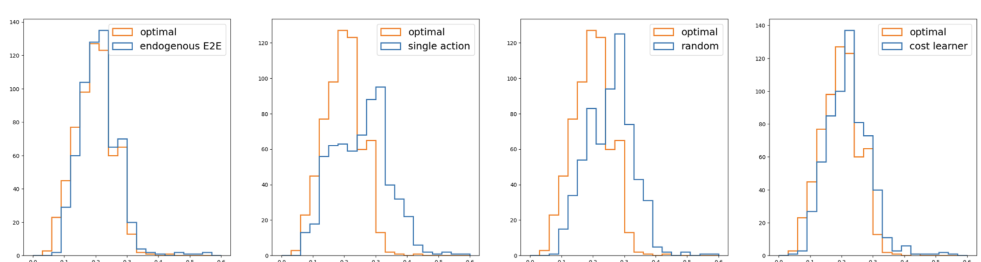

Figure 1: Cost distribution of each method. From left to right: endogenous E2E, single action, random, cost-learner.

References

[1] Brandon Amos and J Zico Kolter. Optnet: Differentiable optimization as a layer in neural networks. In *International conference on machine learning*, pages 136–145. PMLR, 2017.

[2] Rares Cristian, Pavithra Harsha, Georgia Perakis, Brian L Quanz, and Ioannis Spantidakis. End-to-end learning for optimization via constraint-enforcing approximators. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 7253–7260, 2023.

[3] Priya Donti, Brandon Amos, and J Zico Kolter. Task-based end-to-end model learning in stochastic optimization. *Advances in neural information processing systems*, 30, 2017.

[4] Adam N Elmachtoub and Paul Grigas. Smart "predict, then optimize". *Management Science*, 68(1):9–26, 2022.