Decision Focused Scenario Generation for Contextual Two-Stage Stochastic Linear Programming

Jonathan Hornewall
ENPC, Institut Polytechnique Paris
jonathan.hornewall@enpc.fr

Solène Delannoy-Pavy
RTE, ENPC, Institut Polytechnique Paris
solene.delannoy-pavy@enpc.fr

Tito Homem-de-Mello
Universidad Adolfo Ibáñez
tito.hmello@uai.cl

Vincent Leclere
ENPC, Institut Polytechnique Paris
vincent.leclere@enpc.fr

Abstract

We introduce a decision-focused scenario generation framework for contextual two-stage stochastic linear programs that bypasses explicit conditional distribution modeling. A neural generator maps a context x to a fixed-size set of scenarios $\{\xi_s(x)\}_{s=1}^{S}$. For each generated collection we compute a first-stage decision by solving a single log-barrier regularized deterministic equivalent whose KKT system yields closed-form, efficiently computable derivatives via implicit differentiation. The network is trained end-to-end to minimize the true (unregularized) downstream cost evaluated on observed data, avoiding auxiliary value-function surrogates, bi-level heuristics, or differentiation through generic LP solvers. Unlike single-scenario methods, our approach natively learns multi-scenario representations; unlike distribution-learning pipelines, it scales without requiring density estimation in high dimension. We detail the barrier formulation, the analytic gradient structure with respect to second-stage data, and the resulting computational trade-offs.

Preliminary experiments on contextual synthetic instances illustrate that the method can rival current state-of-the-art methods, even when trained on small amounts of training data.

Keywords: contextual stochastic programming; decision-focused learning; differentiable optimization; log-barrier methods; scenario generation.

1 Introduction

Contextual stochastic programming studies decision problems under uncertainty when the distribution of the uncertain parameters depends on an observed context x. This setting has received considerable attention in the literature in recent years, as evidenced by the survey paper by Sadana et al. [2024], and is common in applications such as energy systems, supply chains, and finance, where forecasts and exogenous signals materially affect optimal decisions. Practitioners typically face limited historical data and high-dimensional uncertainties, which makes accurate estimation of conditional distributions a challenging and often unnecessary task if the ultimate goal is to make good decisions.

The standard "predict-then-optimize" pipeline (see *e.g.,* Bertsimas and Kallus [2020], Deng and Sen [2022], Kannan et al. [2025], Tian et al. [2024]) first estimates the conditional law $\mathcal{L}(\xi \,|\, x)$ and then solves the induced stochastic program. Although conceptually clean, this two-stage approach does not take advantage of the structure of the underlying optimization problem. Decision-focused learning offers an alternative by training predictive models end-to-end with respect to downstream

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

decision quality rather than likelihood or moment matching. Some recent works have explored this avenue; we refer to Mandi et al. [2024] for a comprehensive review. Out of these works, the majority aim at developing *pointwise* forecasts. In some cases this is enough—for instance, Homem-de Mello et al. [2024] show that, for a certain class of two-stage stochastic programs, single scenario forecasts are sufficient to obtain an optimal decision on the downstream problem. In general, however, it is well known from the stochastic programming literature that a collection of scenarios is required in order to reliably yield optimality (see, e.g., Wallace [2000]). But the task of learning a deterministic map from contexts to finite collections of scenarios is less well explored. Some works in that direction include Islip et al. [2025], who rely on neural surrogates to estimate recourse functions, and Grigas et al. [2021], who fix the set of scenarios and determine the probability of each scenario in a decision-focused fashion.

In this paper we propose a neural decision-focused scenario generation framework for contextual two-stage stochastic linear programs. A neural generator maps a context x to a fixed-size collection of representative scenarios. For each generated collection we compute a first-stage decision by solving a log-barrier regularized version of its associated deterministic two-stage problem. By leveraging KKT optimality conditions of this smooth surrogate, we obtain closed-form expressions for all required gradients via implicit differentiation, allowing neural net training via backpropagation.

- A decision-focused pipeline that trains a neural scenario generator end-to-end, extending beyond pointwise forecasts to learning full collections of scenarios.
- A smoothing procedure to enable back-propagation involving log-barrier surrogates, which have the benefit of being more naturally adjusted to the underlying geometry of linear programs than current alternatives in the literature (such as quadratic regularization and neural surrogates).
- Preliminary empirical evidence on a small synthetic contextual instance showing that the method outperforms benchmarks.

2 Methodological approach

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, and (x, ξ) a random vector on $\mathbb{R}^d \times \mathbb{R}^k$. We denote by $\mathcal{L}(\xi \mid x)$ the conditional distribution of ξ given x.

2.1 Problem setting

We study a contextual two-stage stochastic program of the form

$$\min_{z \in Z, z \geq 0} \; c^\top z + Q(z, x), \qquad Z = \{z \in \mathbb{R}^{n_1} : Az = b\}, \tag{1}$$

where $A \in \mathbb{R}^{m_1 \times n_1}$, $b \in \mathbb{R}^{m_1}$, $c \in \mathbb{R}^{n_1}$. The second stage cost function $Q(z, x)$ is defined as:

$$Q(z, x) \;=\; \mathbb{E}_{\xi|x}\left[\min_{u \geq 0} \; q(\xi)^\top u \quad \text{s.t.} \quad W(\xi)u = h(\xi) - T(\xi)z\right], \tag{2}$$

here $u \in \mathbb{R}^{n_2}$, $W(\xi) \in \mathbb{R}^{m_2 \times n_2}$, $T(\xi) \in \mathbb{R}^{m_2 \times n_1}$, $h(\xi) \in \mathbb{R}^{m_2}$, and $q(\xi) \in \mathbb{R}^{n_2}$. We assume that we have relatively complete recourse, *i.e.,* the second-stage problem is feasible for all $z \in Z$ and all $\xi \in \mathbb{R}^k$. We also make the standard assumption that A is full row-rank, and that the same holds for $W(\xi)$ and $T(\xi)$ almost surely.

2.2 Learning a mapping from contexts to scenarios

Our aim is to learn a mapping $\phi_w : x \mapsto \{\hat{\xi}_s(x)\}_{s=1}^S$, parametrized by w, that outputs a fixed-size collection of equally-likely representative scenarios for each context variable x. For a given $\phi_w(x)$, its associated first-stage decision is obtained by solving a scenario-based, log-barrier regularized surrogate problem. More precisely we define

$$z_\mu^* : (x, w) \mapsto \arg\min_{z \in Z} \; \left\{ c^\top z - \mu \sum_i \log(z_i) + \tfrac{1}{S} \sum_{s=1}^S \dot{Q}_\mu(z, [\phi_w(x)]_s) \right\}, \tag{3}$$

where $\dot{Q}_\mu : (z,\xi) \mapsto \min_{u\in\mathbb{R}^{n_2}} \{q(\xi)^\top u - \mu \sum_i \log(u_i) \mid W(\xi)u = h(\xi) - T(\xi)z\}$ is the log-barrier regularized recourse function.

Given training data consisting of a collection of scenario-observation pairs $\{(x_i, \xi_i)\}_{i=1}^N$, we define the training loss as the average (non-regularized) cost of decisions $z_\mu^*(x_i)$ on the observed scenarios ξ_i:

$$R_\mu(w) := \frac{1}{N}\sum_{i=1}^N G(z_\mu^*(x_i, w), \xi_i), \tag{4}$$

where $G(z,\xi) := c^\top z + \dot{Q}_0(z,\xi)$ is the non-regularized objective function for a fixed scenario ξ.

3 Explicit expression for sensitivities

The map ϕ_w is typically implemented as a neural network with weight w that are optimized through some variant of stochastic gradient descent. To this end, we need to compute the gradient of the loss $R_\mu(w)$ with respect to w. Chain rules yields

$$\nabla_w R_\mu(w) = \frac{1}{N}\sum_{i=1}^N \left[\frac{(\partial z_\mu^*(x_i,w))_k}{\partial w_j}\right]^\top_{\substack{j\,=\,1,\ldots,d\\ k\,=\,1,\ldots,n.}} \nabla_z G(z,\xi_i)\Big|_{z=z_\mu^*(x_i,w)}. \tag{5}$$

To obtain an expression for $\nabla_z G(z,\xi_i)$, we first note that evaluating the recourse function $\dot{Q}_0(z,\xi)$ is equivalent to solving the second stage linear program. Thus, Danskin's theorem provide subgradients of the recourse cost in terms of its optimal dual variables $\lambda^\star(\xi)$ as $\partial_z \dot{Q}_0(z,\xi) \ni -T(\xi)^\top \lambda^\star(\xi)$. We hence obtain the desired gradient as

$$\nabla_z G(z,\xi_i) = c - T(\xi_i)^\top \lambda^\star(\xi_i). \tag{6}$$

To compute the first term inside the sum in (5), we leverage the implicit function theorem applied to the KKT conditions of the barrier-regularized surrogate problem. The surrogate problem associated with scenario collection $\phi_w(x_i) = \{\xi_k\}_{k=1}^S$ and regularization parameter μ is a log-barrier regularized linear program in canonical form, as follows:

$$\min_{\hat{z}>0}\ \hat{c}^\top \hat{z} - \sum_i \hat{\mu}_i \log \hat{z}_i \quad \text{s.t.} \quad \hat{A}\,\hat{z} = \hat{b}. \tag{7}$$

where $\hat{z} := (z, u_1, \ldots, u_S)$, $\hat{c} := (c, \frac{1}{S}q_1, \ldots, \frac{1}{S}q_S)$, $\hat{b} := (b, h_1, \ldots, h_S)$, $\hat{\mu} := (\mu, \frac{1}{S}\mu, \ldots, \frac{1}{S}\mu)$, and

$$\hat{A} := \begin{bmatrix} A & 0 & \cdots & 0 \\ W_1 & T_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ W_S & 0 & \cdots & T_S \end{bmatrix}. \tag{8}$$

The KKT condition for optimality for optimality for problem 7 with respect to a decision $\hat{z}$ is that there should exist a dual variable $\lambda \in \mathbb{R}$ such that the pair $Y = (\hat{z}, \lambda)$ satisfies

$$F(Y; \hat{A}, \hat{b}, \hat{c}) \;=\; \begin{bmatrix} \hat{c} - \hat{A}^\top \lambda - \mathrm{Diag}(\mu)\,\hat{z}^{-1} \\ \hat{A}\hat{z} - \hat{b} \end{bmatrix} = 0, \tag{9}$$

where $\mathrm{Diag}(\mu)$ denotes a matrix with $(\mu, \ldots, \mu)$ in the diagonal and 0 otherwise, and $\hat{z}^{-1} := (1/\hat{z}_1, \ldots, 1/\hat{z}_n)^\top$. The Jacobian of the optimality condition F is given as the following symmetric matrix

$$\nabla_Y F(Y) \;=\; \begin{bmatrix} \mathrm{Diag}(\mu)\,\mathrm{Diag}(\hat{z}^{-2}) & \hat{A}^\top \\ \hat{A} & 0 \end{bmatrix}, \tag{10}$$

. Note that $\nabla_Y F(Y)$ is nonsingular under full row-rank assumption on A, W_i, T_i. By applying the implicit function theorem, the derivatives of the optimal solution $Y^\star(\hat{A}, \hat{b}, \hat{c})$ with respect to the scenario collection $\phi_w(x_i) = \{\xi_k\}_{k=1}^S = (\hat{A}, \hat{b}, \hat{c})$ are given by:

$$\nabla_{\hat{A},\hat{b},\hat{c}} Y^* = -\left(\nabla_Y F(Y^*; \hat{A}, \hat{b}, \hat{c})\right)^{-1} \nabla_{\hat{A},\hat{b},\hat{c}} F(Y^*; \hat{A}, \hat{b}, \hat{c}) \tag{11}$$

with

$$\frac{\partial F}{\partial \hat{A}_{jk}} = \begin{bmatrix} \lambda_j \mathbf{e}_k \\ z_k \mathbf{e}_j \end{bmatrix}, \quad \frac{\partial F}{\partial \hat{b}_j} = \begin{bmatrix} 0 \\ -\mathbf{e}_j \end{bmatrix}, \quad \frac{\partial F}{\partial \hat{c}_k} = \begin{bmatrix} \mathbf{e}_k \\ 0 \end{bmatrix}. \tag{12}$$

In the above equations, $\mathbf{e}_j$ represents a vector of appropriate dimension with 1 in the jth component and zeros in the remaining ones. Combining (8), (11) and (12), we obtain the desired derivatives $\frac{\partial \hat{z}^*}{\partial \phi_w(x_i)}$ by extracting the relevant components of ∇Y^*. Finally, to compute the derivative with respect to w we are only required to compute $\nabla_w \phi_w(x_i)$, which can be accomplished via back-propagation.

4 Experiments

We implement the resource allocation problem first introduced in Kannan et al. [2025]. The neural network is trained using 100 context-scenario pairs and outputs a single scenario. We compare our algorithm against the methods tested in Homem-de Mello et al. [2024], and clearly outperform them. We emphasize that we achieve this performance even when generating only a single scenario, showing the clear power of the log-barrier regularization.

We test the performance of each method by approximating its corresponding optimality gap using the estimation procedure described in Mak et al. [1999].

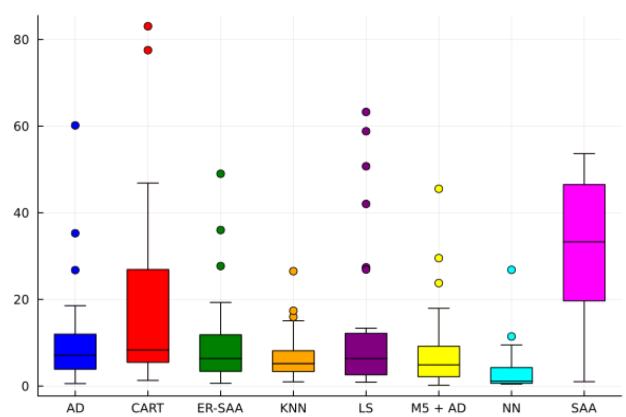

Figure 1: Comparison of our method (**NN**) with two Predict then Optimize methods (**CART** and **LS**), two Application-Driven Forcasts methods (**AD** and **M5 +AD**) and three Conditional Distribution methods (**ER-SAA**, **KNN** and **SAA**). The methods are described in Homem-de Mello et al. [2024].

For the model, we implemented a feed-forward neural network with three hidden layers of 128 ReLU units each. We used the Adam optimizer to perform the learning, with a step size of 10^{-3}. the surrogate is regularized with 0.01 while the down-stream problem is unregularized. We trained the neural network for 20 epochs, which took less than two hours on a computer equipped with an Intel(R) Core(TM) Ultra 7 155H 3.80 GHz processor.

5 Conclusions and Future Perspectives

Current limitations The work presented in this report is still in the early stages. On the experimental side, we neither analyze runtime benefits or test large/real-world instances. On the conceptual side, the method currently only targets convex two-stage problems; settings with integer or nonconvex recourse would require differentiable relaxations. On the computational side, training entails solving a log-barrier deterministic equivalent at each gradient step; practical scaling may hence depend on warm starts and decomposition, both of which is yet to be implemented. Future work will aim to address these challenges, and to leverage classical LP theory to provide solid theoretical arguments for the benefits of the log-barrier smoothing procedure compared with more commonly used regularizers.

References

Dimitris Bertsimas and Nathan Kallus. From predictive to prescriptive analytics. *Management Science*, 66(3):1025–1044, 2020.

Yunxiao Deng and Suvrajeet Sen. Predictive stochastic programming. *Computational Management Science*, 19(1):65–98, 2022. URL https://ideas.repec.org//a/spr/comgts/v19y2022i1d10.1007_s10287-021-00400-0.html.

Paul Grigas, Meng Qi, and Zuo-Jun (Max) Shen. Integrated conditional estimation-optimization. arXiv:2110.12351, 2021.

Tito Homem-de Mello, Juan Valencia, Felipe Lagos, and Guido Lagos. Forecasting Outside the Box: Application-Driven Optimal Pointwise Forecasts for Stochastic Optimization, 2024. URL http://arxiv.org/abs/2411.03520.

David Islip, Roy H. Kwon, Sanghyeon Bae, and Woo Chang Kim. Contextual Scenario Generation for Two-Stage Stochastic Programming, 2025. URL http://arxiv.org/abs/2502.05349.

Rohit Kannan, Güzin Bayraksan, and James R. Luedtke. Technical Note: Data-Driven Sample Average Approximation with Covariate Information. *Operations Research*, 2025. ISSN 0030-364X. doi: 10.1287/opre.2020.0533. URL https://pubsonline.informs.org/doi/10.1287/opre.2020.0533.

Wai-Kei Mak, David P. Morton, and R. Kevin Wood. Monte carlo bounding techniques for determining solution quality in stochastic programs. *Operations Research Letters*, 24(1):47–56, 1999. doi: 10.1016/S0167-6377(98)00054-6.

Jayanta Mandi, James Kotary, Senne Berden, Maxime Mulamba, Victor Bucarey, Tias Guns, and Ferdinando Fioretto. Decision-focused learning: Foundations, state of the art, benchmark and future opportunities. *Journal of Artificial Intelligence Research*, 80:1623–1701, 2024.

Utsav Sadana, Abhilash Chenreddy, Erick Delage, Alexandre Forel, Emma Frejinger, and Thibaut Vidal. A survey of contextual optimization methods for decision-making under uncertainty. *European Journal of Operational Research*, 320(2):271–289, 2024. doi: 10.1016/j.ejor.2024.03.020.

Xuecheng Tian, Bo Jiang, King-Wah Pang, Yu Guo, Yong Jin, and Shuaian Wang. Solving Contextual Stochastic Optimization Problems through Contextual Distribution Estimation. *Mathematics*, 12(11):1612, 2024. ISSN 2227-7390. doi: 10.3390/math12111612. URL https://www.mdpi.com/2227-7390/12/11/1612.

Stein W. Wallace. Decision Making Under Uncertainty: Is Sensitivity Analysis of Any Use? *Operations Research*, 48(1):20–25, 2000. ISSN 0030-364X, 1526-5463. doi: 10.1287/opre.48.1.20.12441. URL https://pubsonline.informs.org/doi/10.1287/opre.48.1.20.12441.