Proceedings of the IISE Annual Conference & Expo 2025
E. Gentry, F. Ju, X. Liu, eds.

A Computational Study on the Impact of Incrementally Changing Stochastic Probability Distributions to Optimization Model Outputs

John Nichols, Jennifer A. Pazour, Sandipan Mishra, John E. Mitchell
Rennselaer Polytechnic Institute
Troy, New York

Abstract

Stochastic programming is used to optimize decisions when there exist uncertain parameters, represented as probability distributions. In this work we are interested in stochastic programming problems where additional information about these uncertain parameters is revealed incrementally. In such cases, the probability distributions for the stochastic input parameters get updated. We deploy a data-driven approach that uses machine learning and common statistical distance measures, such as Kullback-Leibler divergence, total variation distance, and Wasserstein distance, to predict how changes in probability distribution updates impact the outputs of stochastic optimization models, i.e., on the optimal decision variable values and the associated objective function values. Statistical analysis of results from a computational study using a specific two-stage stochastic mixed integer optimization model illustrates that a combination of statistical distance measures and other input parameter data can be used to predict changes in objective function values, and optimal first stage decision variable values. Specifically, for this study, the statistical distance method that is most useful at predicting changes is different depending on the dependent variable.

Keywords
Statistical distance measures, stochastic programming, machine learning, uncertainty quantification

1. Introduction
Probability distributions are used to represent the possible values of uncertain input parameters in stochastic programming. In this work we are interested in stochastic programming problems where additional information about input parameters, but specifically uncertain parameters, is revealed incrementally. In such cases, a stochastic optimization model could be solved dynamically as new information about uncertain parameters becomes available. Yet, when the solve time is large, it would be useful to understand the impact of changing input information on the stochastic optimization model's outputs. Such prediction capabilities could guide whether the model needs to be resolved or not. Further, if additional data collection is costly, the ability to predict the impact of changes to stochastic inputs to the model's outputs can also help guide the collection of additional targeted data.

The goals of this paper are to explore if we can predict how changes to uncertain parameters' probability distributions affect changes to a stochastic optimization model's output, and which input parameters have the most influence on the predictions. To do so, we conduct a computational study to investigate if established statistical distance measures, specifically Kullback-Leibler divergence, Wasserstein distance, and total variation distance, combined with changes to other deterministic input parameters, can be used as inputs to a machine learning method that predicts impacts to a stochastic optimization model's outputs. We also explore which statistical distance measure is best at predicting the changes to the output when the underlying probability distributions for the uncertain parameters are changed, and what has the most influence on these predictions.

The rest of the paper is organized as follows. Section 2 is a review of relevant literature. Section 3 describes our computational study. Section 4 presents our results and answers to our motivating research questions.

2. Literature Review
Stochastic input uncertainty and its impact on model outputs is a well-studied area in stochastic simulations, see [1] for a broad review of research in stochastic simulation under input uncertainty and sensitivity analysis for input certainty reduction. An illustrative example is [2], which creates a distributionally robust stratified sampling approach to handle stochastic simulation under input uncertainty. Their work uses statistical distance measures such as the Wasserstein

distance and the L_2-norm to frame a robust optimization approach that minimizes the maximum of the worst-case estimator variances and builds ambiguity sets for stochastic simulation. They test their approach by comparing the different levels of variance reduction when using the different ambiguity sets. In [3], the impact of input uncertainty on stochastic simulations is quantified through an upper-bound and a lower-bound for the worst-case expected performance by using a robust simulation method and a ranking and sampling method. These bounds are shown to be related to statistical distance measures between the nominal and worst-case probability distributions.

We focus the remainder of the literature review on stochastic input uncertainty quantification associated with stochastic optimization models. [4] analytically examines the behavior of stochastic programming problems when there are perturbations in the probability distributions of the uncertain data used in the model. This work looks at quantitative stability of a stochastic programming problem (without integer variables), i.e., how much the optimal solution changes when the input parameters are slightly perturbed. The approach assumes that the probability distribution is an element of a metrized space of probability measures, with some topological requirements. [5] investigates the worst-case approach to approximating error in the objective function of a stochastic program. That work uses Kullback-Leibler divergence to measure model misspecification and measure the worst-case performance impacts of such misspecifications, but notably they do not resolve the optimization model nor do they look at the changes in the solution to the optimization model. Broadly speaking, these works take an analytical approach to measuring the effects of changes in the probability distributions used in stochastic optimization.

In contrast, [6] uses a data-driven approach to solving stochastic optimization problems that have distributional ambiguity. The observed stochastic costs are used to construct confidence regions for estimators. These estimators then approximate a solution to the stochastic programming problem. The reported computational results show that this approach provides a better estimate of the objective function value of the baseline model than Bayesian and robust optimization methods. [7] examines a distributionally robust stochastic optimization (DRSO) problem, an optimization under uncertainty problem that seeks to find the best hedge against a set of probability distributions. Their data-driven DRSO approach using a dual formulation yields a worst-case distribution, which can be connected with robust optimization methods. The authors also look at different statistical distance measures, and discuss why some measures have shortcomings with their DRSO approach.

Our work has similar motivations and components to these papers, but wants to use the statistical distance measures as the estimators for a stochastic optimization model's output. Specifically, here we are interested in understanding the effect of the changes in the probability distribution(s) on the solutions and the objective function values of stochastic mixed integer linear programs (MILPs) where both the objective function and the constraints are stochastic in nature. As such, it is difficult to employ analytical methods to evaluate the impact of changes in the underlying probability distributions on the solution and the associated optimal value of the objective function. Hence, there is room for a general, numerical approach that can predict changes to model's output by using statistical distance measures.

3. Computational Study

We design a computational study to answer two primary research questions.

1. Given that there are several established statistical distance measures as noted in [8], when used in prediction algorithms, which one of these measures provides the best estimate on how changes to stochastic inputs influence changes to the objective function or the decision variable value of the solved stochastic optimization model?
2. Which features have the most influence on the prediction?

3.1 Stochastic Optimization Model and Input Parameter Generation

For this computational study, we consider a specific stochastic MILP used to optimize production decisions and formulate it as a scenario-based approach [9]. The problem determines, which out of a large set of potential products to launch, and the initial quantity to produce of launched products that maximizes a manufacturing firm's expected profits in the face of uncertain demand. The demand for each of the potential products is uncertain and through different forecasting techniques can be represented as a probability distribution. The firm's decisions are also influenced by other input parameters, such as the product's production costs, resource consumption, and profits, which are assumed deterministic input parameters. In what follows, we describe the sets, input parameters, decision variables, objective function, and constraints, and also describe how we set these input parameters in our computational study.

Table 1: Deterministic Input Parameters' Notation, Description, and Computational Study Data Generation

Input Parameter	Description	Data Generation
A_i	fixed cost to start production of product $i \in I$	$\sim U[0.1, 5]$
B_i	disposal cost per excess unit of product $i \in I$	$\sim U[0.0001, 0.005]$
F_i	profit per unit of product $i \in I$	$\sim U[0.05, 0.6]$
$G_{i,q}$	amount of resource $q \in Q$ used in product $i \in I$	$\sim U[0.001, 0.05]$
$H_{q,s}$	amount of resource $q \in Q$ available for production at stage $s \in S$	$\sim U[0.05, 5]$
$J_{q,s}$	cost per unit of resource $q \in Q$ for production at stage $s \in S$	$\sim U[0.0001, 0.005]$
P_w	probability of scenario $w \in W$ happening	[0.25, 0.25, 0.25, 0.25]

Sets:

- $I :=$ set of potential products. In our experiments, we consider $|I| = 4$ products.
- $Q :=$ set of resources. In our experiments, we consider $|Q| = 5$ resources.
- $S :=$ set of stages. In our experiments, we consider $|S| = 2$ stages, initial and expedited.
- $W :=$ set of scenarios. In our experiments, we consider $|W| = 100$ scenarios.

Input Parameters:

- Stochastic Input Parameters: $D_{i,w} :=$ demand of potential product $i \in I$ in scenario $w \in W$, $D_{i,w} \sim p_i$, for $i \in I$ for $w \in W$.
- In Table 1, we provide the deterministic input parameters.

Decision Variables:

- $y_i := 1$ if the firm decides to launch new product $i \in I$, 0 otherwise.
- $x_i :=$ how much to make for the first-stage batch of new product $i \in I$.
- $z_{i,w} :=$ how much to make for the second-stage batch of new product $i \in I$ after demand is known in scenario $w \in W$.
- $s_{i,w} :=$ amount of product $i \in I$ sold in scenario $w \in W$.
- $e_{i,w} :=$ amount of excess product $i \in I$ in scenario $w \in W$.

$$\text{Max} \sum_{i \in I} -A_i y_i - \sum_{i \in I}(\sum_{q \in Q}(J_{q,1} G_{i,q} x_i)) + \sum_{w \in W} P_w (\sum_{i \in I}(-e_{i,w} B_i + s_{i,w} F_i - O \cdot \sum_{q \in Q}(J_{q,2} G_{i,q} z_{i,w}))) \tag{1}$$

$$\text{s.t.} \quad x_i \leq M_i y_i \quad \forall_{i \in I} \tag{2}$$

$$\sum_{i \in I} G_{i,q} x_i \leq H_{q,1} \quad \forall_{q \in Q} \tag{3}$$

$$z_{i,w} \leq M y_i \quad \forall_{i \in I} \quad \forall_{w \in W} \tag{4}$$

$$\sum_{i \in I} G_{i,q} z_{i,w} \leq H_{q,2} + (H_{q,1} - \sum_{i \in I} G_{i,q} x_i) \quad \forall_{q \in Q} \quad \forall_{w \in W} \tag{5}$$

$$s_{i,w} \leq x_i + z_{i,w} \quad \forall_{i \in I} \quad \forall_{w \in W} \tag{6}$$

$$s_{i,w} \leq D^k_{i,w} \quad \forall_{i \in I} \quad \forall_{w \in W} \quad \forall_{k \in K} \tag{7}$$

$$e_{i,w} \geq x_i + z_{i,w} - D^k_{i,w} \quad \forall_{i \in I} \quad \forall_{w \in W} \quad \forall_{k \in K} \tag{8}$$

$$x_i \geq 0 \quad \forall_{i \in I} \tag{9}$$

$$y_i \in \{0, 1\} \quad \forall_{i \in I} \tag{10}$$

$$z_{i,w} \geq 0 \quad \forall_{i \in I} \quad \forall_{w \in W} \tag{11}$$

$$e_{i,w} \geq 0 \quad \forall_{i \in I} \quad \forall_{w \in W} \tag{12}$$

$$s_{i,w} \geq 0 \quad \forall_{i \in I} \quad \forall_{w \in W} \tag{13}$$

Equation (1) is the objective function and maximizes expected profit. The first two terms capture the fixed costs to launch a product and the cost to produce the launched products in the first-stage, which needs to occur before demand is known. The last term represents the expected profits of the second stage decisions, which capture excess product costs, profit of sold products, and expedited second-stage production costs which are multiplied by $O = 3$ to make it more costly to produce. Constraint (2) enforces the company only produces products if the decision is made to launch

the product, with M being a sufficiently large number. Constraint (3) is an upper limit to how much of any product can be produced in the first-stage given the availability of resources. Constraint (4) enforces that only products flagged for launch will be made in the expedited production run. Constraint (5) enforces the upper bounds of a product's expedited production run in the second-stage but adds any leftover resources not used in the initial run. Constraints (6) and (7) set the upper bounds of how much of product i is sold for profit. Constraint (8) captures the disposal costs, in that the lower bound of $e_{i,w}$ is either 0 or the amount of product i made in both stages minus the demand for product i. Note that since Equation (1) is to maximize profit and that $-e_{i,w}B_i$ has a negative sign, the model will try to make $e_{i,w}$ as low as possible to reduce disposal costs.

In our computational experiments, each time we solve the model with new input parameter values, we denote it as a replication. We generate the deterministic input parameters by drawing a new sample for each replication from the distributions as displayed in Table 1. For the stochastic input parameters, we capture that each of the potential products' demands can be represented as a probability distribution p_i, and we assume p_i can be represented as a metalog distribution [10]. Thus, we obtain $D_{i,w}$ by sampling from product i's specific metalog distribution, p_i. We generate a different metalog distribution in our computation experiments by drawing five random samples from $U[0,100]$ and then sorting them and mapping them to the corresponding set of probabilities of $[0.25, 0.33, 0.50, 0.66, 0.75]$. These five samples are fit to a metalog distribution. In our computational experiments, we set a static, default metalog distribution for each of the 4 products, and then we change one product's metalog distribution while leaving the other three products the same. For each product, we create 10 different metalog distributions. With 4 products, this results in 40 separate replications. We solve the stochastic model for each of these 40 replications using Gurobi 12.0.0, called using the Pyomo library in Python.

3.2 Prediction Data Set and Random Forest Regression Prediction Methodology

Given the goal of this paper is to understand the impact of changes of input model parameters to model outputs, we generate a new data set that records all pairwise combinations of the 40 replication runs. This results in a prediction data set of $\binom{40}{2}$ different pairwise combinations. For each row of this data set, we know the values for each of the input parameters, and the outputs of the solved optimization model, and we use this data to calculate our independent and dependent variables used in our prediction algorithm.

Specifically, for independent variables, we calculate for each pairwise combination, statistical distance measures for each of the products' stochastic input parameters. Statistical distance measures quantify how much two probability distributions diverge from one another, with common measures described in [8]. Generally smaller values indicate that the two probability distributions are similar, while larger values mean that the two are different. In our computational experiments, we use Wasserstein distance, Kullback-Leibler divergence, and total variation distance measures. Thus, for each of these three statistical measures, we calculate the statistical distance for product i probability distributions in replications r and v, with $r, v \in R$ and $r \neq v$. The statistical distance measurements of replications r and v for product $i \in I$ are represented by $WD_i^{r,v}$ for Wasserstein distance, $KL_i^{r,v}$ for Kullback-Leibler divergence, and $TVD_i^{r,v}$ for total variation distance.

We also consider the impact on the change in the deterministic input parameters in our prediction. Our set of independent variables consist of: $A_i^r - A_i^v$, $B_i^r - B_i^v$, $F_i^r - F_i^v$, $G_{i,q}^r - G_{i,q}^v$, $H_{q,s}^r - H_{q,s}^v$, $J_{q,s}^r - F_{q,s}^v$, plus the previously described statistical distance measures.

The variables we are interested in predicting are (i) the change in objective function values, and (ii) the change in first-stage decision variables. Thus with replications r and v, $r \neq v$, in (i) we define our dependent variable as $f_r - f_v$, and in (ii) we consider two separate dependent variables, $|y_2^r - y_2^v|$ and $|y_4^r - y_4^v|$. We examine y_2 and y_4 in our computational experiments, because their optimal values varied from replications, whereas in all the replications, $y_1 = 0$ and $y_3 = 1$.

We use a random forest regression model to make predictions, and do so by deploying the scikit learn Python library. For each of our defined dependent variables, we create four separate random forest regression models that consider all the deterministic independent variables and plus each of the previously described statistical distance measures, as well as a model that includes all three measures. To create and test our prediction, 80% of the pairwise data set is used for training while 20% is used for testing. When splitting the data into training and test sets with the train_test_split function from the scikit learn Python library, we ensure for reproducibility that the same splits are generated. We then calculate the permutation feature importance for each regression model to give a value for how much influence each independent variable has on the prediction.

4. Results

In this section we present the results from our computational study, organized by subsections corresponding to our previously defined research questions.

4.1 Statistical Distance Measures and Predicting Changes to Output

We first explore which specific statistical distance measure is the best at predicting changes to the optimal objective function values and optimal first stage decision variables values when stochastic input parameter probability distributions are changed. Table 2 provides the mean squared error (MSE) and R^2 scores, calculated based on the difference between the predicted output values and the actual output values in the test set. When using $f_r - f_v$, the prediction algorithm with total variation distance (TVD) achieved both the lowest MSE and highest R^2 scores. Yet the Kullback-Leibler divergence (KL) prediction algorithm was the best for $|y_2^r - y_2^v|$, and the Wasserstein distance (WD) prediction algorithm was the best for $|y_4^r - y_4^v|$. While improvements in prediction can be achieved for all dependent variables when considering all three statistical distance measures as independent variables, the improvement for predicting the objective function values over just using the best measure is relatively small, whereas there is more value in using all measures for predicting decision variable value changes.

Table 2: MSE and R^2 Values of the Random Forest Regression Models

Dependent Variable →	$f_r - f_v$		$\lvert y_2^r - y_2^v \rvert$		$\lvert y_4^r - y_4^v \rvert$	
Method ↓	MSE	R^2	MSE	R^2	MSE	R^2
Wasserstein	0.46	0.96	0.1	0.58	0.03	0.87
Kullback-Leibler	0.36	0.97	0.01	0.95	0.04	0.85
Total variation distance	0.22	0.98	0.1	0.58	0.04	0.82
All measures	0.08	0.99	0.01	0.97	0.02	0.89

To illustrate how well our prediction models can predict a stochastic model's objective function, Figure 1 plots for the test set, the actual objective function values pairwise difference to their predicted values. For the decision variables y_2 and y_4, Figures 2 and 3 are confusion matrices with TP for true positive, FN for false negative, FP for false positive, and TN for true negative. The predictions roughly follow the outputs from solving the stochastic optimization model.

Figure 1: Objective function values pairwise differences in the test set compared to their predicted values.

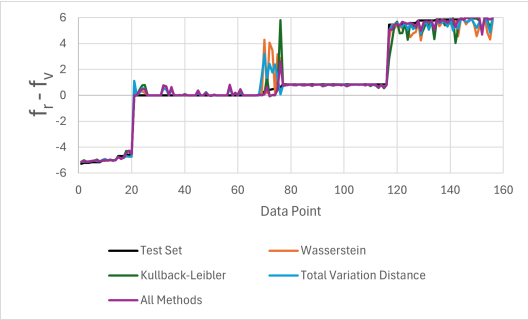

Figure 2: Confusion matrix of $|y_2^r - y_2^v|$ predictions with Kullback-Leibler

$\lvert y_2^r - y_2^v \rvert$		Predicted	
		Made	Not Made
Actual	Made	TP = 80	FN = 0
	Not Made	FP = 2	TN = 74

Figure 3: Confusion matrix of $|y_4^r - y_4^v|$ predictions with Wasserstein

$\lvert y_4^r - y_4^v \rvert$		Predicted	
		Made	Not Made
Actual	Made	TP = 60	FN = 0
	Not Made	FP = 5	TN = 91

4.2 Independent Variables and Feature Importance

Next, we explore which specific independent variables have the most influence on predicting changes to the optimization model's outputs for the objective function and optimal decision variable values. We calculate and compare the permutation feature importance scores for each independent variable, for their developed prediction models. The permutation feature importance is a general method that calculates the expected loss when the feature is permuted to its original value [11]. Higher values mean that a feature has a more significant impact on the prediction.

In Table 3, the top four permutation importance scores are reported for each of the dependent variables. These results are based on using the best performing prediction model for each dependent variable. For each of the dependent

Table 3: The top four permutation features for each dependent variable using the Wasserstein prediction model.

$f_r - f_v$: Total Variation Distance		$\lvert y_2^r - y_2^v \rvert$: Kullback-Leibler		$\lvert y_4^r - y_4^v \rvert$: Wasserstein	
Feature	Importance	Feature	Importance	Feature	Importance
TVD_4	1.65	KL_2	1.67	WD_4	1.75
TVD_3	1.30	KL_1	1.49	$G_{2,3}^r - G_{2,3}^v$	0.01
TVD_1	0.10	$A_3^r - A_3^v$	0.01	$G_{1,3}^r - G_{1,3}^v$	0.01
TVD_2	0.07	$J_{3,1}^r - J_{3,1}^v$	0.01	$A_4^r - A_4^v$	0.01

variables, the highest permutation importance score is the statistical distance measure for a specific product. Thus, using statistical distance measures as a way to predict changes to the objective function values and optimal decision variables of stochastic optimization models has promise. For $|y_2^r - y_2^v|$ and $|y_4^r - y_4^v|$, the top four permutation features were a combination of statistical distance measures and changes to deterministic input parameters. While statistical distance measures are useful in prediction of stochastic optimization model outputs, methods should not consider them alone, but instead capture information about the stochastic optimization model. When predicting changes in whether product 2 was made, the Kullback-Leibler divergence of product 1's demand was the second most important feature. This is due to the limited and shared resources captured in the model. When predicting optimal decision variable values, input parameter information changes about the set of potential products can improve the prediction.

5. Conclusions and Future Work

We created a data-driven approach that uses random forest regression with different statistical distance measures to predict how changes to stochastic parameters can estimate changes to a stochastic optimization model's optimal objective function and decision variable values. This is a preliminary work that considers a single optimization model, further work can expand to varied models and also look at more statistical distance measures and cross-validate the machine learning models. Such predictions can be used to guide targeted data collection and reoptimization decisions.

Acknowledgements

This research is funded by Office of Naval Research Grant N00014-22-1-2542.

References

[1] Canan G Corlu, Alp Akcay, and Wei Xie. Stochastic simulation under input uncertainty: A review. Operations Research Perspectives, 7:100162, 2020.

[2] Seung Min Baik, Eunshin Byon, and Young Myoung Ko. Distributionally robust stratified sampling for stochastic simulations with multiple uncertain input models. arXiv preprint arXiv:2306.09020, 2023.

[3] Lewei Shi, Zhaolin Hu, and Ying Zhong. Input parameter uncertainty quantification with robust simulation and ranking and selection. In 2024 Winter Simulation Conference (WSC), pages 3614–3625. IEEE, 2024.

[4] Werner Römisch and Rüdiger Schultz. Distribution sensitivity in stochastic programming. Mathematical Programming, 50:197–226, 1991.

[5] Henry Lam. Robust sensitivity analysis for stochastic systems. Mathematics of Operations Research, 41(4):1248–1275, 2016.

[6] Steffen Rebennack. Data-driven stochastic optimization for distributional ambiguity with integrated confidence region. Journal of Global Optimization, 84(2):255–293, 2022.

[7] Rui Gao and Anton Kleywegt. Distributionally robust stochastic optimization with wasserstein distance. Mathematics of Operations Research, 48(2):603–655, 2023.

[8] Mark Kelbert. Survey of distances between the most popular distributions. Analytics, 2(1):225–245, 2023.

[9] Alexander Shapiro and Andy Philpott. A tutorial on stochastic programming. Manuscript. Available at www2.isye.gatech.edu/ashapiro/publications.html, 17, 2007.

[10] Thomas W Keelin. The metalog distributions. Decision Analysis, 13(4):243–277, 2016.

[11] Francesco Cappelli, Gianfranco Castronuovo, Salvatore Grimaldi, and Vito Telesca. Random forest and feature importance measures for discriminating the most influential environmental factors in predicting cardiovascular and respiratory diseases. International Journal of Environmental Research and Public Health, 21(7):867, 2024.