OPTIMIZATION
https://doi.org/10.1080/02331934.2025.2451812

Taylor & Francis
Taylor & Francis Group

OPEN ACCESS Check for updates

Compact uncertainty sets for robust optimization based on bootstrapped Dirichlet process mixture model

Andreas Neofytou, Bin Liu and Kerem Akartunalı

Department of Management Science, University of Strathclyde, Glasgow, UK

ABSTRACT

This study utilizes unsupervised machine learning to improve the performance of robust optimization through construction of compact uncertainty sets. Robust optimization aims to find solutions that are resilient to uncertainties and variations in input parameters, allowing decision-makers to make informed decisions in the face of uncertain conditions. However, traditional optimization approaches often assume known and fixed datasets, which fails to reflect the inherent uncertainties present in real-world problems. Accurate construction of compact and reliable data-driven uncertainty sets is a critical challenge that directly impacts the effectiveness of robust optimization. To address this challenge, we propose a Dirichlet process mixture model for clustering to construct a data-driven uncertainty set suitable for robust optimization problems, allowing for more accurate uncertainty modelling. This data-driven uncertainty set is constructed by intersecting the l_1 and l_∞ norms for each predicted cluster and then merging these multiple basic convex uncertainty sets to create a comprehensive representation. This approach results in uncertainty sets based on clustered data that flexibly capture a compact region of uncertainty in a nonparametric manner. An innovative aspect is the introduction of statistical bootstrap to obtain a more robust clustering solution and outcome. The proposed method is applied to production planning problems and a comparative analysis with existing approaches highlights its advantages. Our method demonstrates effectiveness in improving the accuracy and robustness of solutions in robust optimization by constructing more compact uncertainty sets.

ARTICLE HISTORY
Received 4 September 2024
Accepted 4 January 2025

KEYWORDS
Decision-making under uncertainty; robust optimization; data-driven uncertainty set; Dirichlet process mixture model

2020 MATHEMATICS SUBJECT CLASSIFICATIONS
90B50; 90C17

1. Introduction

In today's dynamic and uncertain environments, the effective utilization of machine learning techniques has become increasingly important [1,2]. These methods have been applied in numerous applications across various domains, including supply chain management among others [3,4]. The integration of

CONTACT Bin Liu b.liu@strath.ac.uk

machine learning algorithms allows organizations to utilize enormous amounts of data to gain valuable insights and make informed decisions [2]. However, traditional optimization methods often face significant challenges in handling the inherent uncertainties present in real-world scenarios [5]. Recently, to overcome these challenges and harness the full potential of machine learning, researchers have turned to robust optimization techniques [6] to tackle 'decision-making under uncertainty'. Specifically, researchers integrate robust optimization principles into the analysis of big data and apply advanced machine learning algorithms to explore robust and reliable strategies that effectively handle uncertainties, maximize performance and achieve optimal outcomes in these dynamic environments [6–8].

Robust optimization aims to find optimal solutions that are resilient to uncertainties and variations in input parameters [9–11]. The uncertainty set, the heart of robust optimization, defines the range of possible values that the uncertain parameters can take [9,12]. Traditionally, robust optimization methods rely on pre-defined uncertainty sets, mainly because they simplify the complexities of real-world problems. However, recent advancements in machine learning techniques offer a huge opportunity to construct more accurate data-driven uncertainty sets and thereby enhance the performance of optimization solutions [13]. Specifically, diverse machine learning techniques can be employed to tighten uncertainty sets which imply reducing the pessimism of robust solutions [14]. For this work, a machine learning integration for two-stage Robust Optimization problems is studied to enhance decision-making in the presence of uncertainty. This approach accounts the dynamic nature of real-world problems by incorporating adaptability.

This research paper focuses on leveraging unsupervised learning to improve uncertainty modelling in robust optimization. The primary objective is to propose an adaptive framework that starts from data leading to high-quality robust decisions and predictions [8]. Particularly, we aim to develop adaptive robust optimization models and machine learning algorithms that can provide solutions that are not overly sensitive to uncertainties in input parameters. This is accomplished by focusing on the uncertainty set construction. With that we aim to merge machine learning and robust optimization, and thus improve the decision-making process by bridging the gap between a good prediction and a good decision [6,9,12,13,15,16].

We apply Bootstrapped Dirichlet Process Mixture model to construct data-driven uncertainty sets. These uncertainty sets capture the variability present in the underlying data distributions without relying on predefined assumptions [17,18]. The unsupervised learning model learns the parameters of the Gaussian mixture components, such as means and covariances, from the data. However, achieving precise inference for the mixture parameters and weights presents an intractable challenge. Consequently, it becomes vital to identify optimal parameters and reasonably accurate coefficients within a novel modified uncertainty

set [19]. Moreover, there is a need to approximate a distribution that closely resembles the true data distribution, enabling the upcoming accurate construction of the uncertainty set [20,21]. Our goal is also to refine the model such that it identifies significant patterns without being influenced by overfitting, thereby ensuring that it recognizes genuine clusters in new and unseen uncertain data. To avoid this overfitting issue during the clustering process, we propose incorporating an outlier detection mechanism [20], which also enhances the model's generalization capability. To tackle the robustness, we employ a bootstrap resampling with replacement method for uncertainty set construction. This the first time we have shown that the integration of bootstrap resampling in this specific process leads to interesting improved results in some case studies. Consequently, we can flexibly model the distribution of exported uncertain parameters and adjust the uncertainty sets based on observed data patterns, allowing for a robust data-driven representation of uncertainty. Our work is quite versatile and applicable to various types of data, including real-time, historical, or synthetic datasets. In domains such as production planning, where data patterns can demonstrate a range of characteristics including symmetry, asymmetry, wide spread, and typical or non-typical distributions [4,22], our approach offers a robust solution because we approximate the best parameters based on the machine learning predictions.

Our interesting and novel research aims to address the limitations of existing approaches in constructing uncertainty sets and advance the state-of-the-art uncertainty modelling for robust optimization utilizing machine leaning. The major novelties of this paper are summarized as follows:

- *Integration of a modified Unsupervised Learning framework in Data-Driven Robust Optimization:*
  We utilize a modified combination of an unsupervised learning technique, variational inference, stick-breaking and outlier detection to estimate the parameters of the Gaussian Mixture Model (GMM) from observed data. Variational inference enables efficient approximation of the posterior distribution, while stick-breaking simplifies the process of allocating clusters within the GMM. Additionally, the inner model's outlier detection identifies the influence of anomalous data points, ensuring robust parameter estimation.
- *Introduction of Bootstrap Resampling for Enhanced Robustness:*
  We employ bootstrap resampling to further reinforce the robustness of the exported parameters. It essentially provides a way to estimate the uncertainty of the clustering results. Consequently, we obtain more robust estimates and improve the overall performance of our approach.
- *Development of more compact and adaptive data-driven uncertainty sets:*
  For data-driven uncertainty set construction, to ensure the desired level of conservativeness in the solution, we devised an adaptive technique to compute the minimum adaptive scale parameter. We manage to capture the variability

in uncertainties and identify the fundamental patterns. In addition, we showed that the final constructed uncertainty set is given by a finite union of convex subsets and their intersected norms.

- *Exploration of different experimental settings and data structures to showcase the efficacy of the proposed method:*
  We tested our approach under various experimental settings and data structures to prove its superiority in solving (two-stage) robust optimization problems. Additionally, we evaluated the performance of the proposed data-driven uncertainty set in comparison with traditional state-of-the-art and data-driven methods in terms of solution quality, robustness, computational time and different coverage rates. The findings of this study clearly show that the constructed uncertainty set provides a more decent objective compared to alternative sets, while still retaining its robustness.

The remainder of this paper is organized as follows: Section 2 provides a comprehensive literature review on the intersection of machine learning with robust optimization, mainly in uncertainty set construction. Section 3 presents the problem statement, defining the challenges and objectives of the study. Section 4 outlines the proposed methodology for constructing data-driven uncertainty sets using the unsupervised learning technique. Section 5 presents the numerical examples, experimental setup and comparative results. Finally, Section 6 discusses the findings, conclusions, limitations, and future research directions.

2. Literature review

In the context of robust optimization and data-driven uncertainty set construction, unsupervised learning techniques play a vital role [23]. The target is to discover patterns and relationships within unlabelled data, without any guidance [2], and to group similar data points, in order to construct relevant uncertainty sets for each cluster [24]. Additionally, unsupervised anomaly detection techniques supports this process by identifying outliers, which is vital for handling extreme cases in uncertainty modelling [25].

Instead of enhancing the algorithmic performance [26–29], diverse machine learning models have been applied to generate compact uncertainty sets from historical data that lead to the learning of high quality representations, which is the aim of our research. In [30] an unsupervised deep learning method is utilized to construct non-convex uncertainty sets and the trained neural networks was integrated into the robust optimization model. Their uncertainty set was defined as a sphere of radius R and they trained their neural network employing one-class classification, which minimizes the empirical centred total variation of the projected data points [31]. After completing the training of the network, adjustments are made to the radius R of the uncertainty set with the aim of achieving a

desired coverage of the dataset. Some other ideas use principal component analysis and kernel smoothing in [32] for similar goal. In [33] the novel idea was to apply support vector clustering and in [8] statistical hypothesis testing to construct the data-driven uncertainty sets. Attempting to bridging the gap between machine learning and robust optimization, [34] proposed a data-driven approach for the construction of an uncertainty set using kernel-based support vector clustering. However, kernel methods can face significant challenges when applied to high-dimensional data [34], where the computational cost grows significantly and leads to overfitting. To tackle the decision-making under uncertainty and find a balance between robustness and performance, the authors in [14] used machine learning predictive models to guide the optimization process. In [35], the authors utilized clustering algorithms to analyse historical data, identify patterns and thus create clusters that represent different scenarios of uncertainty and, consequently, generate uncertainty sets.

Exploring all the possible unsupervised machine learning algorithms, a relevant work in [36] applied k-means to hedge the clustering result against unstructured errors in the observed data. k-means is the most well known unsupervised method in the machine learning world and we aim to compare our results with that. However, it has several limitations, such as when the data points in clusters overlaps other clusters, making this approach not performing well. Furthermore, the number k, the number of clusters, should be determined beforehand which means that different initialization can lead to different final cluster assignments and centroids. It heavily relies on the original data representations and thus suffers from the curse of dimensionality [2,37]. Also, k-means may not perform well when the clusters are non-spherical since it assumes that the data points are distributed in a spherical shape [36].

In [38], the authors addressed uncertainty in optimization problems by focusing on the expected value of the uncertain parameters. They applied clustering and ensured that solutions are optimized against average and weighted scenarios rather than worst-case extremes. The result was less conservative solutions by utilized k-means for their unsupervised learning approach. The primary drawback of this mean robust optimization approach is the accurate estimation of mean values. Furthermore, in their experimental settings they compare different number of clusters for the k-means and they determine the optimal number of clusters with the elbow method to compare the objective. However, they need to define the optimal number of clusters beforehand for every scenario.

For worst-case and predefined uncertainty sets, several attempts have been made with norm bounds, l_1 or l_∞ norms. Prior research has typically been limited to either offering empirical validation or providing certified assurances of model robustness in constrained situations [34]. Overall, other methodologies involved box set, ellipsoidal set, bounded by norms or polyhedral uncertainty sets [17,39]. However, these methods of l_1 and l_∞ norms on their own have some limitations such as unable to capture the specific shape of uncertainty, as it only considers

the magnitude of the individual parameters for l_1 norm, as well as l_∞ norm may not capture correlations beyond the largest individual parameter [39].

In [40], the authors attempted to create two data-driven polyhedral uncertainty sets that directly utilize l_1 and l_∞ norms respectively for their own purpose. Their data-driven uncertainty set was a union of intersected polyhedral uncertainty sets, where each specific norm uncertainty set covers the data belonging to each calculated component. As a result, it forms the final uncertainty set with the union of all constructed l_1 and l_∞ norms. A similar but more innovative attempt was made by Ning and You [20] with the intersection of the two l_1 and l_∞ norms. The suggested polyhedral uncertainty set comprises again a combination of basic uncertainty sets, extending beyond the constraints of a 'parallel-piped' set. Subsequently, the resultant robust model produces solutions of exceptional quality. Instead of the l_1 uncertainty which relies solely on the l_1 norm, this new data-driven uncertainty set is defined using both the l_1 norm and l_∞ norm. When the intersection of the two norm balls are constructed, the final data-driven uncertainty sets differ and lead to distinct optimization solutions due to the intersection. Both works use different scale factors to adapt the data-driven uncertainty set with their data, Dirichlet Process Gaussian Mixture Model as the unsupervised learning algorithm to calculate the important parameters and Variational Inference to approximate the posterior prior distribution of latent variables. This is because the exact inference for the mixture parameters was intractable, thus their uncertainty set construction relied on those approximated parameters and predefined probability guarantees.

In [41] the authors challenge the above model's adaptability and clustering. They tried to improve the performance of the Dirichlet Process Mixture Model (DPMM) with the introduction of deep neural networks in a novel Deep DPMM. This is a deep unsupervised method that simultaneously performs clustering and learn more expressive features. However, they employed expectation-maximization, instead of Variational Inference, and used Gibbs sampling to sample from the posterior. In [42] apart from robust optimization, the authors introduce recursive expectation-maximization algorithm ideal for parameter estimation in DPMMs. The algorithm offers significantly improved computational efficiency compared to Variational Bayes and Inference methods but with its own limitations. Deep unsupervised learning was also employed in [23] to learn and extract hidden structures and anomalies from data with non-convex uncertainty sets for more robust solutions.

Most works focused on novel approaches that integrate unsupervised machine learning with robust optimization to enhance decision-making under uncertainty. The concept of just account exactly all the uncertain data points and scenarios does not safeguard again data perturbations, uncertain data collection, noise or parameters deviation. We will take it a step further and attempt to further improve the robustness of solutions under perturbations, challenging the concept

of balancing optimality and robustness. Our idea is to use a resampling method in the uncertainty set construction, that to the best of our knowledge has not been employed before. This provides more reliable estimates of parameters by generating multiple samples from the data [43]. While the predictive analytics phase of unsupervised machine learning helps understand the data and make predictions about the underlying structure, the prescriptive analytics phase of robust optimization and uncertainty set construction will utilize these insights not only to prescribe optimal decisions, but also robust decisions in the face of uncertainty. This will improve the decision making under uncertainty. To this end, our work seeks to address current challenges and limitations in uncertainty modelling by using clustering to construct reliable nonparametric data-driven uncertainty sets for robust optimization problems.

3. Problem statement

In this section, we formally define the research challenge in the uncertainty modelling and the nonparametric uncertainty set construction for two-stage robust optimization. We address the challenge of constructing uncertainty sets where the decision-makers lack exact knowledge of the underlying distributions of the data and its variations. Previous studies assume a predefined shape for uncertainty sets, which, however, leads to suboptimal solutions and increased risk in decision-making processes. To tackle this challenge, we propose a novel approach called Bootstrapped Dirichlet Process Mixture Model, which will guide the construction of the data-driven uncertainty set and thus improve the decision making process under uncertainty.

Specifically, the research questions to tackle include:

RQ.1. *How to strike the right balance between optimality and conservativeness in the uncertainty set construction?*

RQ.2. *How to construct compact, nonparametric and non-convex uncertainty sets?*

RQ.3. *How to derive a method to approximate the unknown distribution, specifically the posterior prior distribution of latent variables?*

RQ.4. *How to further enhance the robustness of the ML parameters that influence the decisions?*

From the theoretical perspective, a simple illustration of optimization approach is given by the objective function $J(x)$, which represents the quantity that needs to be minimized in the optimization problem. A single data point p over the function tries to find the local or global minimum $\min_p J(p)$. Let us assume that the parameters of the model involve some kind of uncertainty, for instance $p \rightarrow p + \Delta p$, where Δp is a derivation from noise as mentioned. In this case, we do not have a single point but instead a disk of confidence interval that

tries to find the best worst-case in

$$\min_{p} \max_{\Delta p} J(p + \Delta p) \tag{1}$$

where $\Delta p \in U$ in uncertainty set. The difference between this robust solution in contrast to nominal approach is the sensitivity to uncertainty. The nominal solution is highly sensitive to any parametric uncertainty or changes in coefficients, while the robust solution does not give such high quality solution but it is robust against these uncertainties, parameter variations and data perturbations, everything inside the Δp. As a result we have the general single stage robust optimization formulation $\min_x \max_{u \in U} f(x, u)$ and two stage robust optimization, also called adaptive robust optimization, $\min_{x,y} \max_{u \in U} f(x, y, u)$.

Consider the following two-stage robust optimization problem:

$$\begin{aligned} &\min_{x,y} \max_{u \in U} \left(c^\top(u)x + b^\top y(u) \right) \\ &\quad s.t.\ A(u)x + By(u) \leq d, u \in U \end{aligned} \tag{2}$$

where $c^\top$ and A are involved in the uncertainty U, while $b^\top$ and B are not. $x \in X$ denotes the set of initial decision variables that are determined before the uncertainty u becomes known and $y \in Y(x, u)$ represents the set of secondary decisions which can be made after the uncertainty has been revealed [13]. The initial decisions, known as 'here-and-now' decisions, and the secondary decisions, termed as 'wait-and-see' decisions, can both encompass continuous or discrete variables.

Overall, the primarily target in two-stage (adaptive) robust optimization is to calculate a first-stage solution x_i such that for every uncertain scenario u_i in uncertainty set $U = \{u_1, u_2, \ldots\}$, there is a feasible second-stage solution y_i so that the worst possible objective value over all scenarios is minimized. Hence, the main challenge is to construct such precise, but also robust, uncertainty set U that will influence the solution quality.

As mentioned, in robust optimization, the uncertainty set U is often predefined as a single convex finite set and the uncertain parameters u are expressed using a parametric distribution with a fixed set of parameters. For instance, u might have a predefined bound or follow a multivariate Gaussian distribution with fixed mean and covariance parameters.

$$\begin{aligned} U &= \{u | u \in R^m, \|u\| \leq \epsilon\} \\ U &= \{u | u \sim N(\mu, \Sigma)\} \end{aligned} \tag{3}$$

where ϵ is a predefined bound on the uncertainty, and μ and Σ are fixed parameters of the Gaussian distribution. The uncertainty set describes the uncertainty information and it is often constructed based on a priori information and simple assumptions, and thus implying overly conservative solutions [20].

The key difference between parametric and nonparametric uncertainty set construction lies in how the uncertain parameters are represented. Nonparametric uncertainty set construction does not assume a fixed parametric form. Instead, it learns the uncertainty set directly from data, represented as a set of $\{u_i\}_{i=1}^{n}$ and from the adoption of the specific nonparametric model. Each u_i represents a realization of the uncertain parameters, and together they form the nonparametric uncertainty set U. The nonparametric uncertainty set is represented as:

$$U_j = \{u_i | u_i \in R^m, i = 1, \ldots, n\}$$
$$U_{final} = U_1 \cup U_2 \cup \cdots \cup U_k \tag{4}$$

where k is the number of components of the uncertain data, n the number of data points of that specific j component and u_i represents a realization of the uncertain parameters. It allows the number j of mixture components U_j to grow with the data, providing flexibility in modelling without specifying a fixed number of parameters k.

Using a single fixed set to describe uncertainty by fitting data and adjusting parameters is not effective in dynamic environments. As such, we aim to to derive the 'optimal' polyhedral uncertainty set to capture the entire space. In addition, nonparametric machine learning models have a flexible number of parameters. Even though convex optimization is always easier to solve, it presents these highly conservative solutions because it includes unrealistic scenarios. The idea is to combine many simple convex sets in Equation (4) in a non-convex way to cover the entire region and exclude unnecessary space, than previous conservative 'one-set-fits-all' uncertainty set [20,39]. This is recently employed by several authors, but not entirely explored due to the variety and complexity of different machine learning models and their capabilities.

In general, the overall target for compact, efficient and robust uncertainty modelling is to strike a balance between capturing sufficient uncertainty, to account for variations in the data, while avoiding excessive shrinkage that could lead to overfitting [40]. That said, we need an adaptive scale parameter that can balance between conservativeness and optimality, but also play a role in avoiding overfitting and ensuring the robustness. We will derive a method to construct for each component this crucial scaling parameter for controlling the clustering behaviour and conservativeness in the following section.

4. Methodology

In this section, we develop a Dirichlet process mixture model to describe the data pattern for situations where the number of clusters are unknown in the dataset. Subsequently, stick-breaking is employed to flexibly represent the uncertainty structure, through obtaining the weights of the mixture model by finding the best number of components. The final vital step is to approximate the posterior

distribution of the parameters in the DPMM. Although a handful optimization methods have been tested, Variational Inference is used to estimate the relative distributional parameters, by approximating the posterior distribution and by obtaining an approximating distribution that is computationally tractable [21,44,45]. We then proceed with the heart of the research, the formulation and construction of the final data-driven uncertainty set. This includes the calculation of the conservativeness with adaptive scale parameter, reinforcing the robustness with bootstrapping and outlier detection. Additionally, we provide probabilistic guarantees for the robust data-driven uncertainty set.

4.1. Dirichlet process Gaussian mixture model to describe the data pattern

To accurately model a given dataset, we employ the Dirichlet process mixture model – a powerful nonparametric Bayesian statistical model. It is worth noting that despite being labelled as ' nonparametric' this machine learning model does have parameters. However, the parameter space for this model is infinite-dimensional, allowing for greater flexibility in representing uncertainty [40]. Infinite-dimensional parameter can significantly increase the computational cost. To address this computational issues associated with the infinite-dimensional parameter space, significant interest will be given to variational inference which is a probabilistic modelling technique, enabling to estimate the relative distributional parameters.

Denote the data as $u = (u_1, \ldots, u_N)$. The probability distribution of u, $p(U|\theta)$, is described using a Gaussian mixture model (GMM):

$$p(u|\pi, \theta) = \sum_{k=1}^{M} \pi_k N(u|\mu_k, \Sigma_k) \tag{5}$$

where $\theta = (\mu_k, \Sigma_k)_{k=1}^{M}$, $\sum_{k=1}^{M} \pi_k = 1$, $\pi_k \geq 0$ the weight for i component, $N(\cdot) \sim$ Gaussian Distribution, μ_k mean and Σ_k covariance [40,46].

Due to the fact that the number of Gaussian components is unknown in GMM, if the value of M is not properly determined, there will be a significant disparity between the observed data distribution and the GMM distribution. To address this issue, we proposes the use of DPMM, which employs the Dirichlet process as the prior distribution $p(\theta)$ to determine the number of components in the model, which is considered as an infinite mixture model, thus eliminating the need to specify the number of components in advance [22,40].

In Figure 1(a) we can observe that the attempt to fit a single Gaussian distribution to approximate two uncertain parameters, blue and orange, fails miserably since we do not have a accurate representation of the uncertainty with only two final exported parameters, $\theta = (\mu, \Sigma)$. However, in Figure 1(b) the resulting predicted and exported parameters $(\pi_k, \mu_k, \Sigma_k)_{k=1}^{M}$ from Equation (5) automatically determine the number of components by considering infinite mixtures.

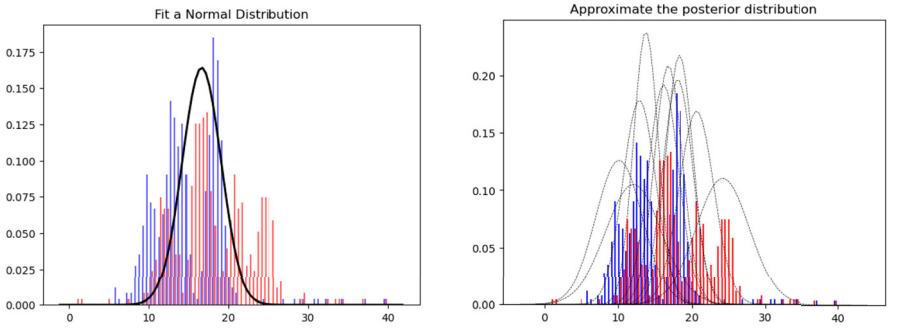

Figure 1. Fitting Gaussian Mixture Model(s) to find the true representation of the data. (a) Attempting to fit a Normal Distribution to represent the data, (b) Approximate the true posterior distribution representation.

Based on the DPMM mixture technique [46], each uncertain parameter u_i is generated by first choosing a component that is distributed according to $\pi = (\pi_1, \ldots, \pi_M)$ and then the u_i is generated from choosing the Gaussian component with the parameter θ_i, which is generated from a prior distribution G that follows a random draw in the Dirichlet process $DP(\alpha, G_0)$ [40]:

$$\begin{aligned} \theta &\sim G \\ G &\sim DP(\alpha, G_0) \\ x_k &\sim N(u|\mu_k, \Sigma_k) \end{aligned} \tag{6}$$

where $\alpha \sim Gamma(a_0, b_0)$ to avoid the requirement of direct specification of the concentration parameter α and G_0 is the base distribution. The concentration parameter α has a tremendous effect on the number of components within the data [24]. The mean μ_k and the covariance Σ_k are unknown and we have to set the base distribution G_0 as a Normal–Wishart distribution:

$$G_0 \sim N(\mu|\mu_0, (\beta_0\Lambda)^{-1}) W_k(\Lambda|\Psi_0, \upsilon_0) \tag{7}$$

where $W_k(\cdot)$ denotes the Wishart distribution, Λ denotes the precision and μ_0, β_0, Ψ_0, υ_0 are the hyperparameters of the base distribution G_0.

4.1.1. Cluster allocation with stick-breaking

The distinct values of $\theta = (\theta_1, \ldots, \theta_n)$ lead to the division of the dataset into clusters, where each cluster k consists of elements with the same value θ_k^*, and the distribution that describes these divisions is known as the Chinese restaurant process [40]. Specifically, a dataset is divided into an infinite number of clusters, and each cluster can accommodate an infinite number of data points [47]. As a sequence of data points $\{\theta\}_{k=1}^{N-1} = (\theta_1, \ldots, \theta_{N-1})$ is observed, each data point is assigned to a cluster and the Nth data point, θ_N, has the option to either join an

existing cluster or form a new one. This process is provided and described in the following form of distribution:

$$\theta_N = \begin{cases} \dfrac{N_c}{\alpha + N - 1}\delta(\theta_N = \theta_k^*), & \text{for an existing cluster} \\ \dfrac{\alpha}{\alpha + N - 1}\theta,\ \theta \sim G_0, & \text{for a new cluster} \end{cases} \tag{8}$$

where N_c represents the total number of data points assigned to cluster k (i.e. the total number of samples that have a value equal to θ_k^*) and δ is the delta function. It can be observed that the posterior distribution when assigning a value θ_N: there is a probability of $\frac{\alpha}{\alpha+N-1}$ for it to be allocated to a new cluster, or a probability of $\frac{N_c}{\alpha+N-1}$ to be assigned to an existing cluster [40]. This means that parameter sets should be automatically grouped into discrete clusters, with adjustments made based on the samples allocated to each cluster.

As a result, the stick-breaking construction is a fundamental concept in Bayesian nonparametric modelling for our cluster allocation [46,47]. It involves breaking a unit-length stick into an infinite number of pieces, each representing the weight assigned to the above potential cluster. The key idea behind stick-breaking is to iteratively break off portions of the stick V_k according to a sequence of probabilities, which are determined by a beta distribution. The stick-breaking construction and the two collections of random variables are provided as follows:

$$\begin{aligned} G &= \sum_{k=1}^{\infty} \pi_k(V)\delta(\theta = \theta_k^*) \\ V_k &\sim Beta(1, \alpha) \\ \theta_k &\sim G_0 \end{aligned} \tag{9}$$

where the generation of the weight $\pi_k(V) = V_k \prod_{j=1}^{k-1}(1 - V_j)$. Initially, to explain this process, we assume to have a stick of length 1, then we break at V_1 with corresponded weight of π_1 which is the length of the broken-off portion. Subsequently, we repeatedly break the remaining portion to obtain $\{\pi_i : i = 1, 2, \ldots\}$. Simultaneously, for each broken-off portion of the stick, a corresponding cluster is created, represented by a unique parameter value drawn from a base distribution G_0. This allocates weights in an effective manner to an infinite number of potential clusters, with the weights determined by the stick lengths and the clusters defined by the drawn parameter values. The function $\delta(\theta = \theta_k^*)$ represents an indicator function centred on the optimal θ_k^* [40].

4.1.2. Optimizing posterior approximations with variational inference

To approximate the posterior distribution of latent variables we will use Variational Inference (VI), an optimization-based approach applied to fit probabilistic models. While Variational Inference involves optimization, it does not explicitly have a step size parameter or a learning rate such as gradient-based optimization

algorithms. Instead, we have parameters that control the optimization process, such as the number of components in the data, the concentration parameter and other associated hyperparameters. The target is to approximate the posterior distribution through obtaining an approximating distribution that is computationally tractable. This is done by minimizing the divergence between the true posterior distribution and the approximating (variational) distribution [24].

This posterior distribution is given below:

$$q(\alpha, Z, V, \mu, \Sigma) = q_1(\alpha) \prod_{i=1}^{N} q_2(Z_i) \prod_{k=1}^{M-1} q_3(V_k) \prod_{k=1}^{M} q_4(\mu_k, \Sigma_k) \tag{10}$$

where $q_1(\alpha)$ denotes the previously mentioned Gamma distribution in Equation (6), $q_2(Z_i = k) = \frac{\rho_{ik}}{\sum_{j=1}^{M} \rho_{ij}}$, where ρ is the not-yet normalized probability component for data point i and cluster k [20], $q_3(V_k)$ the Beta distribution and $q_4(\mu_k, \Sigma_k)$ the Normal-Wishart distribution from Equation (7) [20,46].

The Dirichlet process mixture model from the previous subsection gives the distribution $p(U, \alpha, Z, V, \mu, \Sigma)$, the joint distribution of both the observed data U and the parameters $(\alpha, Z, V, \mu, \Sigma)$. The posterior distribution combines the prior distribution $p(\alpha, Z, V, \mu, \Sigma)$ with the likelihood of the observed data $p(U|\alpha, Z, V, \mu, \Sigma)$ to update those beliefs. The overall target of that inference is to find the posterior distribution of the latent variables and parameters $(\alpha, Z, V, \mu, \Sigma)$ given the observation uncertain parameters U, expressed as $p(\alpha, Z, V, \mu, \Sigma|U)$, which represents the true posterior distribution and the updated belief after observing the data, combining the prior and the likelihood. $q(\alpha, Z, V, \mu, \Sigma)$ is the variational distribution, or variational mean-field approximation [46], we are trying to approximate. The parameters of each term in Equation (10) are estimated by minimizing an upper bound on the KL-divergence between the true posterior and the approximated variational distribution [46]:

$$\min E\left(\frac{q(\alpha, Z, V, \mu, \Sigma)}{p(\alpha, Z, V, \mu, \Sigma|U)}\right) \tag{11}$$

We note that the above KL divergence is a distance metric between the distributions $q(\cdot)$ and $p(\cdot)$, consequently the goal of the variational algorithm is to minimize the KL divergence to achieve the optimal approximation, which is equivalent to maximizing the evidence lower bound (ELBO) [25], as detailed in Algorithm 1.

Figure 2 elaborates the Variational Inference process. This iterative process continues until convergence is reached. Stick-breaking allows for automatic determination of the number of clusters based on the available data. For the DPMM, we are able to estimate the optimal number of components, M, along with the exported parameters, $(\alpha, Z, V, \mu, \Sigma)$. Finally, we employ the useful approximate parameters, $(\pi_k, \mu_k, \Sigma_k)_{k=1}^{M}$, for the construction of the data-driven uncertainty set.

Algorithm 1 Variational Inference Algorithm

Input uncertainty data set u_i and model $p(\cdot)$
Initialize Variational factors $q(\alpha, Z, V, \mu, \Sigma)$ in Equation (10)
while $ELBO \not\rightarrow$ converge **do**
    Update $q_1(\alpha)$
    for i = 1,$\cdots$,N **do**
        Update $q_2(Z_i)$ in Equation (10)
    end for
    for k = 1,$\cdots$,M **do**
        Update $q_3(V_k)$ in Equation (10)
        Update $q_4(\mu_k, \Sigma_k)$ in Equation (10)
    end for
    Compute $ELBO = E\{\log p(U, \alpha, Z, V, \mu, \Sigma)\} - E\{\log q(\alpha, Z, V, \mu, \Sigma)\}$
end while
Result: Variational Mean-Field Approximation $q(\alpha, Z, V, \mu, \Sigma)$ in Equation (10)

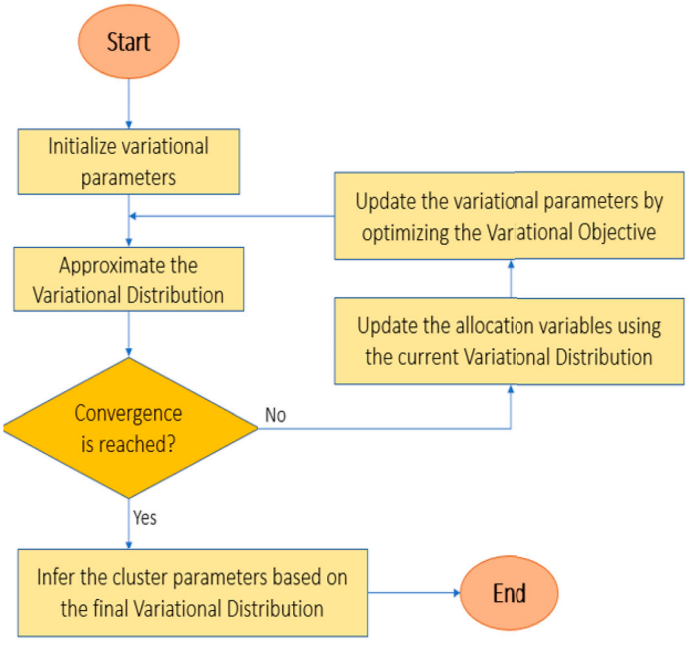

Figure 2. The process for Variational Inference in DPMM.

4.2. Construction of the data-driven uncertainty set

The target is to construct a compact uncertainty set U directly from data that captures the realization of uncertain parameters in u, in Equation (4). This computational tractable data-driven uncertainty set is constructed based on

the previous approximated parameters of the distribution in Equation (10). We iteratively solve the Algorithm 1, initializing the prior, running the variational inference, and exporting a set of optimal parameters $\{\mu_k, \Sigma_k\}_{k=1}^{M}$. Based on these extracted GMM components and parameter estimates, the uncertainty set is constructed as a combination of multiple diverse uncertainty sets. Mathematically, the proposed union of data-driven uncertainty sets can be expressed as follows:

$$\mathcal{U} = \bigcup_{\pi_k \geq \pi^*} \big(u \| \|\tau_k(u-\mu_k)\|_1 \leq (1+\sigma_{\mu_k})\Gamma^{\alpha}_{1,k}, \\ \|\tau_k(u-\mu_k)\|_\infty \leq (1+\sigma_{\mu_k})\Gamma^{\alpha}_{\infty,k}\big) \tag{12}$$

where u is the uncertainty introduced. The threshold π^* is usually set to 5% [20,40], while π_k is the weight of the k component for $k = 1, 2, \ldots, M$, meaning the probability of that k mixture component. It is calculated from Algorithm 1 and $q_3(V_k)$ of Equation (10), where $q_3(V_k) = Beta(\tau_k, \nu_k)$, and thus $\pi_k = \frac{\tau_k}{\tau_k+\nu_k}\prod_{j=1}^{k-1}\frac{\tau_j}{\tau_j+\nu_j}$, for $k = 1, \ldots, M-1$ and $\pi_M = 1 - \sum_{k=1}^{M-1}\pi_k$ [20,46]. Components with a weight π_k greater than the threshold π^* are included in the uncertainty set. The coverage rate α is a predetermined value representing the desired extent of the uncertainty set, for example $\alpha = 1$ means 100% coverage rate. This solely influences the adaptive scale parameters $\Gamma^{\alpha}_{1,k}$ and $\Gamma^{\alpha}_{\infty,k}$, which are associated with each cluster and determine the structure and size. In detail, the coverage rate α is set in advance according to the predefined data coverage rate the user prefers, how conservative or flexible the user wants to be. For simplicity, we will proceed with the full data coverage rate in this study and it aligns well with [40], where the authors chose 99.99% predefined data coverage rate. Thus, we will choose to include all relevant (non-outlier) data points within the uncertainty range, while we already set 95% coverage in the distribution before to focus on the most significant uncertainty. μ_k the mean and σ_{μ_k} the standard deviation of the mean belong to the bootstrap resampling step in the following subsection. σ_{μ_k} slightly expand this boundary to account for small perturbations or variations in the data.

Our construction is based on the classical uncertainty $U = \left\{z \| \| \frac{(z-z_0)}{\sigma}\|_p \leq \rho\right\}$ [7,8,37,39]. For our uncertainty set construction, τ_k will have a new role and will be an adaptive parameter that changes depending on the type, structure, and variability of each component around the mean μ_k. We assign it as $\tau_k = \Sigma_k^{-1}$, which is the value of the corresponding inverse covariance matrix for each component, or as $\tau_k = (L_k^{-1})^T$, which is the Cholesky decomposition of Σ_k^{-1}, where $\Sigma^{-1} = (LL^T)^{-1} = (L^T)^{-1}L^{-1}$. This is because in some cases the cluster data points are spread in an unbalanced way around the μ_k and there is uneven variability. We need a more accurate and adaptive manner to better capture the structure of the cluster, as well as less conservative bounds compared to fixed uncertainty sets, leading to better clustering performance. The uncertainty set construction

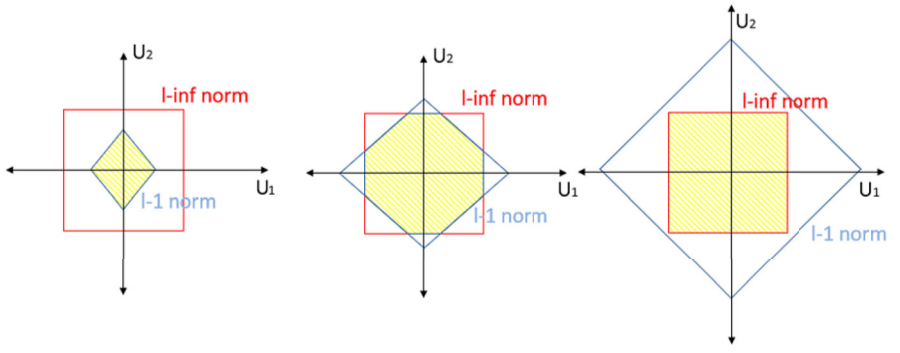

Figure 3. Intersection of l_1 and l_∞ norms.

is inspired by Dai et al. [40], where the authors used a similar way to fix the orientation of their components but utilized the upper triangular matrix derived using the Cholesky decomposition of the result of the inverse covariance inference for each component. This representation is especially useful for estimating distances between observations in these Gaussian processes. However, the authors did not integrate the intersection of the norms into a holistic uncertainty set, utilizing each norm on its own. In general, as presented in Figure 3, for each component/cluster, we construct the intersection of: $\{\|\tau_k(u-\mu_k)\|_1 \leq (1+\sigma_{\mu_k})\Gamma^{\alpha}_{1,k}\}$ and $\{\|\tau_k(u-\mu_k)\|_\infty \leq (1+\sigma_{\mu_k})\Gamma^{\alpha}_{\infty,k}\}$. Subsequently, we take the union of all the constructed data-driven uncertainty sets. Our framework is a combination of both probabilistic and worst-case approaches.

The union of intersected uncertainty sets results in significant challenges, and thus we divide the problem into manageable subproblems. Particularly, we employ the master-subproblem framework combined with Column-and-Constraint Generation to solve the optimization problem. This approach is used by several authors [20,29,40], because it is easy to implement and it is particularly effective due to its ability to handle non-convexity.

4.2.1. Calculating the conservativeness

The level of conservativeness in the uncertainty set is determined by the scale parameters $\Gamma^{\alpha}_{1,k}$ and $\Gamma^{\alpha}_{\infty,k}$. However, when dealing with the U uncertainty set, there is no straightforward method to compute the scale parameter and therefore we need to choose the best hyperparameter Γ for U. To address this, we have propose a modified methodology from [40] to determine the minimum scale parameter that achieves the desired data coverage rate. In this proposed method, we input the mean value μ_k and τ_k of each component k into the equation below, and then calculate ξ_i for each data u_i:

$$\xi_i = \|\tau_k(u_i - \mu_k)\| \tag{13}$$

where τ_k is defined as above and the $\|\cdot\|$ norm changes depending the norm we are calculating, $\|\cdot\|_1$ for Γ_1 and $\|\cdot\|_\infty$ for Γ_∞. For the desired data coverage

rate α, $\Gamma^{\alpha}_{1,k}$ and $\Gamma^{\alpha}_{\infty,k}$ are considered as the N_1^{th} smallest ξ_i, which is calculated as follows:

$$N_1 = round(N_k \times \alpha) \tag{14}$$

where $round(x)$ is a function that returns the closest integer to x. For each component k, N_k is the number of total data points.

For illustration, we sketch the the calculation process in Figure 4 where 10 data points with each representing 1 identified component. Red point is considered the mean μ and blue points are the data points u_i. $\xi_i = \|(u_i - \mu)\|$ calculates the distance of each point from its mean. We get a sorted array with the distance for each ξ_i. For $\alpha = 1$, we apply these to the $N_1 = round(N_k \times \alpha) = 10$. So we select the 10th smallest ξ from our sorted array which is the d_{10}. As can be observed, selecting $\xi_{10} = d_{10}$ as our Γ_∞ for that component, we include all the data points in our green dashed box lines. If we select $\alpha = 0.8$ coverage rate, we apply these to the $N_1 = round(N_k \times \alpha) = 8$. Hence we select the 8th smallest ξ, thus the distance with d_8 as our Γ_∞ for that component, observing that we exclude 2 data points from the red dashed line box. The same will be applied for Γ_1 for that component and the same for every predicted component.

4.2.2. Bootstrapped resampling method

To tackle the trade off between optimality and conservativeness, we establish a bootstrapping method to further enhance the robustness, which, to the best of our knowledge, has not been attempted yet. We employ bootstrap resampling as a robust statistical technique to estimate the variability of the exported parameters $\{\mu_k, \Sigma_k\}_{k=1}^{M}$ of the DPMM for each k cluster and M the predicted maximum number of clusters. Our bootstrap approach involves randomly sampling n observations, size of the dataset, from the observed dataset X with replacement to

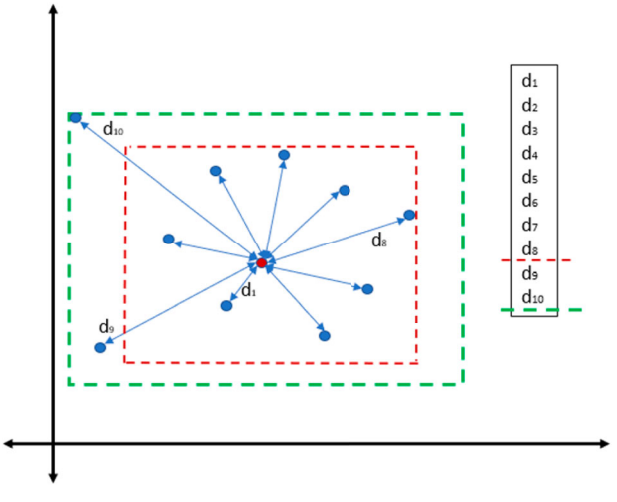

Figure 4. Calculation of Gamma for l_∞ norm.

create bootstrap samples X^*. Repeating this process $L = 1000$ times, we obtain a collection of bootstrap samples $X_1^*, X_2^*, \ldots, X_L^*$ upon which we apply our DPMM to each single subset to obtain the estimated mean $\mu_{X_j^*}\ \forall\, 1 \leq j \leq L$. Subsequently, we take the mean of all the estimated means resulting the final mean all the collection of bootstrap samples, μ^*, which we will replace instead of the normal non-bootstrapped exported μ in Equations (12) and (13). For simplicity, in those equation we will refer them as μ but they are actually the bootstrapped mean μ^*. In Figure 5 we present the histogram of the means for each uncertain parameters, the mean of those mean, observing the standard deviation and presenting the final cluster behaviour of a single cluster mean in contrast to normal DPMM. Furthermore, the confidence level and standard deviation σ_{μ^*} of the estimated final bootstrapped mean is added as a percentage to Γ to enhance the robustness by accounting the actual deviation of the bootstrapped parameters. This implies that we do not solely rely on the calculation of the conservativeness from above method in Equation (13), but we also further enhance the robustness of our statistical inference [43].

The algorithm of the bootstrapping resampling is presented in Algorithm 2. Specifically, we calculate the standard deviation of the bootstrapped resampled means, $\sigma_{\mu_k^*}$ for each k cluster, and add it algebraically to the Γ_k in Equation (12) which slightly expand the boundary to account for any potential small variation in the data. This standard error represents the single standard deviation of the bootstrap mean and holds the variability associated with the estimate of the mean obtained through the bootstrapping process. $(1 + \sigma_{\mu_k})\Gamma^{\alpha}_{\infty,k}$ provides us a controlled expansion of the modified uncertainty set, ensuring that it is neither too conservative nor too restrictive, without overestimating the uncertainty. Although, $2 * \sigma$ in $(1 + 2\sigma_{\mu_k})\Gamma^{\alpha}_{\infty,k}$ covers approximately 95% of the data distribution in distance and offers more robustness, it is not necessary since $\Gamma^{\alpha}_{\infty,k}$ already captures the essential spread of cluster's data. This would make our model overly conservative.

Algorithm 2 Bootstrap Procedure

1: Initialize X dataset and $StoredParameters = \emptyset$
2: **for** j = 1,…,L **do**
3: Take n random observations from X with replacement
4: Create collection of bootstrap estimation X_j^*
5: Solve the DPMM with X_j^* data
6: Export estimated parameter $\mu_{X_j^*}$
7: Store $\mu_{X_j^*} \rightarrow StoredParameters$
8: **end for**
9: Calculate the mean of the $StoredParameters$, μ^*
10: Return: μ^* and σ_{μ^*}

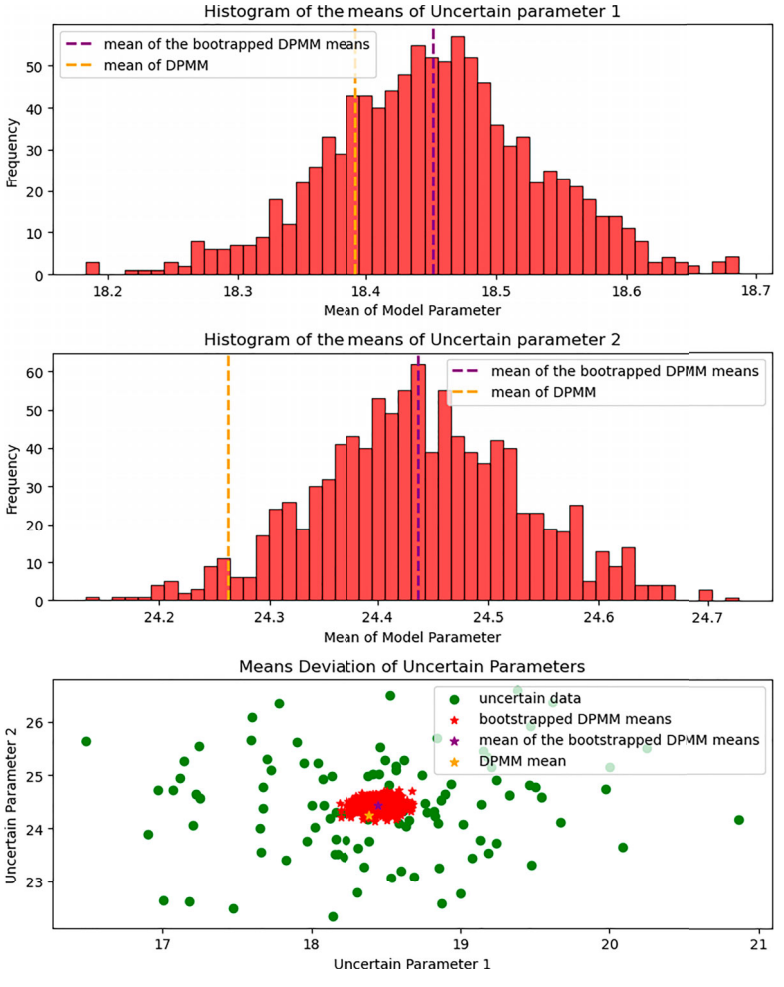

Figure 5. Bootstrapping method with 1000 iteration resampling for a single cluster. How to calculate the means of bootstrapped μ for one single cluster and the μ_k^* of that k cluster. Comparison is also made with DPMM mean without bootstrapped resampling.

However, even though this novel introduction and extra step of bootstrapped resampling results in slightly higher computational time in contrast to normal DPMM approach, it has a huge impact on the solutions. We may argue that the parameters already have well-defined confidence intervals in the data distribution, thus additional bootstrapping may be redundant and leads to this extra computational effort, but we assume the computational resources are available. As a result, for the exported parameter μ, the mean location of the uncertainty set will shift to represent a more generalized centre of the data and will ensure that the uncertainty set doesn't rely on one specific data sample.

To summarize, we propose a two-layer bootstrap framework to construct more reliable uncertainty sets and address uncertainty from different directions and

perspectives. The first layer of bootstrapping within the DPMM, which estimates distributions over mixture components using resampling. The second layer involves the introduction of statistical bootstrap, essentially bootstrap of bootstrapping, which further updates the uncertainty set by resampling the previous estimated parameters. The reason we employ this additional layer of resampling is to further reinforce the robustness, better capture the variability and obtain a more robust clustering solution.

4.2.3. Outlier detection

Here, we present the procedure on how our framework identifies anomalous data points in our dataset. Particularly, our approach utilizes the likelihood values from the previous DPMM to evaluate each data point. In this variational inference step, the model identifies which cluster each data point belongs to and, specifically, the main target is to identify data points with low likelihood values which implies potential outliers. The outlier detection algorithm is summarized in Algorithm 3 and provides a framework for identifying and isolating anomalous data points based on their likelihood values. If a data point has a low posterior probability of belonging to any cluster, it is considered an outlier. This is determined by the *threshold*, usually 5%, to decide whether a point is an outlier in our dataset. This will also help us in terms of generalization and robustness on unseen data, since any future data points that form new clusters with very few members are considered outliers. Our method is quite similar to the inner-outer layers system from [20], which provides us robustness to any potential variation in the resulting *CleanedDataset*.

Algorithm 3 Outlier Detection method of DPMM

Fit DPMM to the data c ∈ X
Initialize $CleanedDataset = \{c_1, c_2, \ldots, c_N\}$
Initialize $OutlierDataset = \emptyset$
Calculate the log probability of each data point LogProb(c_i) using the relevant score samples function
We set *threshold*
for i = 1,…,N **do**
    if LogProb(c_i) ≤ *thresshold* **then**
        $c_i \rightarrow$ *Outlier*
        Remove c_i from *CleanedDataset*
        Store c_i to *OutlierDataset*
    end if
end for
Return: *CleanedDataset*

4.3. Probabilistic guarantees of the proposed method

The following outlines the probabilistic guarantees and its importance to show the robustness of our approach. The algorithms presented–Algorithm 1, Algorithm 2 and Algorithm 3 – are justified by their role to estimate the important parameters for constructing the final data-driven uncertainty set. In general, this uncertainty set comes with a predetermined level of probability, which shows the likelihood that the true parameter values will fall within the proposed uncertainty set. The robust complexity, $\rho(U)$, will be used to compute how hard and how difficult we satisfy a robust constraint given the uncertainty set U.

In the outlier detection process of Algorithm 3, we create an uncertainty set which is free from data points that would lead to extreme results. We will now provide the theoretical properties of the intersection and union of the uncertainty sets, because we want to ensure that we are working with reliable uncertainty sets. Particularly, within this context of data-driven uncertainty sets, the concept of intersection between the l_1 and l_∞ norms is explored. The intersection $l_1 \cap l_\infty$ set and uncertainty $u \in R^L$ is defined as:

$$U_{l_\infty \cap l_1} = \{u : \|u\|_\infty \leq 1, \|u\|_1 \leq \rho\} \tag{15}$$

Based on the previous work from [39], we have the following:

Proposition 4.1: *For two uncertainty sets, U_1 and U_2 such that $ri(U_1) \cap ri(U_2) \neq \emptyset$, the robust complexity of $U_1 \cap U_2$ satisfies $\rho(U_1 \cap U_2) \geq \min_{i=1,2} \rho(U_i)$.*

Proof: Denote δ^* as the support function of U and the relative interior of U, $ri(U) := \{x \in U : \forall\ y \in U \exists\ \lambda > 1,\ y + \lambda(x - y) \in U\}$, where U is any non-empty convex set. If $U = U_1 \cap U_2$ with $ri(U_1) \cap ri(U_2) \neq \emptyset$, we have $\delta^*(y|U) = \min_\nu(\delta^*(\nu|U_1) + \delta^*(y - \nu|U_2))$. By definition of the robust complexity $\delta^*(\cdot|U_i) \geq \rho(U_i)\|\cdot\|_2$, it follows:

$$\delta^*(y|U) \geq \min_{i=1,2} \rho(U_i) \min_\nu \{\|\nu\|_2 + \|y - \nu\|_2\} \geq \min_{i=1,2} \rho(U_i)\|y\|_2 \qquad \blacksquare$$

To generate a flexible uncertainty set construction, we use the unique characteristics of l_1 and l_∞ norms. The union of the norms, $U_1 \cup U_2, \ldots$ offers a comprehensive approach to include a range of uncertainties. For $l_1 + l_\infty$ set and $u \in R^L$, define:

$$U_{l_\infty + l_1} = \{u = u_1 + u_2 : \|u_1\|_\infty \leq \rho_1, \|u_2\|_1 \leq \rho_2\} \tag{16}$$

Proposition 4.2: *For two uncertainty sets, U_1 and U_2, the robust complexity of the Minkowski sum $U_1 + U_2$ satisfies $\rho(U_1 + U_2) \geq \rho(U_1) + \rho(U_2)$.*

Proof: Given $U = U_1 + U_2$, it follows $\delta^*(y|U) = \delta^*(y|U_1) + \delta^*(y|U_2)$. By definition of the robust complexity $\delta^*(\cdot|U_i) \geq \rho(U_i)\|\cdot\|_2$, we have:

$$\delta^*(y|U) \geq \rho(U_1)\|y\|_2 + \rho(U_2)\|y\|_2.$$

Taking the minimum over all y such that $\|y\|_2 = 1$ proves the above. ■

Algorithm 1 generates the parameters for the uncertainty set construction, which are then further reinforced their robustness and generate more robust estimates with Algorithm 2. Furthermore, we identify the actual mixture components that contribute to this process. Below, we show how the union of these exported uncertainty sets, which each represent a mixture component, contributes to robust constraints and that the final uncertainty set U provides probabilistic guarantees.

Based on the previous work from [46], we have the following:

Proposition 4.3: *For any robust constraint $c^\top x \leq b$, $\forall\ c \in U$, if the uncertainty set U can be expressed as a union $U(\epsilon) = \cup_{k=1}^{M} U_k$, then the robust constraint is equivalent to $c^\top x \leq b\ \forall\ c \in U_k$, $k = 1, \ldots, M$.*

Proof: Let the uncertainty set U be expressed as a union of subsets $U(\epsilon) = \cup_{k=1}^{M} U_k$. For any $c \in U$, there exists at least one subset U_k such that $c \in U_k$. Therefore, the constraint $c^\top x \leq b$ must hold for all $c \in U_k$ for each $k = 1, \ldots, M$. Hence, the original robust constraint $c^\top x \leq b\ \forall\ c \in U$ is equivalent to $c^\top x \leq b\ \forall\ c \in U_k, \forall\ U_k$ [46]. ■

Proposition 4.4: *The probability of feasibility of the constraint $Pr(c^\top x \leq b)$ is bounded below ($\geq$) by $(1-\epsilon)\sum_{k:\pi^k \geq \pi_*} \pi_k$.*

Proof: Considering the probability of feasibility, denote $\Gamma \geq 0$ as the adaptive scale parameter that satisfies $Pr(\|\xi\| \leq \Gamma_k) \geq 1 - \epsilon$ and $\xi \sim t_n(0, I)$, following a Student's t-distribution with n degrees of freedom, the overall probability of feasibility is the weighted sum of the probabilities over the subsets. Hence, it is bounded below by $1 - \epsilon \geq (1-\epsilon)\sum_{k:\pi_k \geq \pi^*} \pi_k$ [46]. ■

5. Experiments

This section provides a series of numerical examples to test the performance of the proposed method. To implement the examples, we coded in Python with relevant libraries, such as scikit-learn for machine learning and RSOME (Robust Stochastic Optimization Made Easy) [48] for Robust Optimization. All the optimization problems were solved with Gurobi and implemented on a university laptop with an Intel(R) Core(TM) i5-1145G7 @ 2.60GHz and 16GB RAM.

5.1. Motivating example – a toy problem

In this section we provide an illustrative motivating example adapted from [20] to test the proposed data-driven uncertainty set and the data-driven robust optimization framework. The motivating example is defined as follows:

$$\begin{aligned} &\min_{x} 3x_1 + 5x_2 + \max_{u \in U} \min_{y} 6y_1 + 10y_2 \\ &s.t. \\ &x_1 + x_2 \leq 70 \\ &x_1 + y_1 \geq u_1 \\ &x_2 + y_2 \geq u_2 \\ &x_1, x_2, y_1, y_2 \geq 0 \end{aligned} \tag{17}$$

This is a classical two-stage adaptive robust optimization and *minmax* problem where the scope is to minimize x and y, while maximizing the uncertainty u. x_1 and x_2 are the first-stage decision variables, while y_1 and y_2 are the second-stage decision variables. u_1 and u_2 are the uncertain parameters of a possible production plan that perform well across all possible realizations within the defined uncertainty set U.

5.1.1. Dataset and scenarios

The experimental setting for the above problem is to create two uncertain parameters, u_1 and u_2, which for example can be considered as uncertain demand in a production planning problem. We create 500 data points which are split into 3 distinct scenarios of different data structures and variability, such as symmetric, asymmetric, correlated, non-correlated patterns and no patterns. The purpose of selecting multiple and different scenario-based approaches is to capture a wide range of possible outcomes, so these 500 points represent each time one scenario and different outcomes. We note from Equation (17) that potential supervised learning predictions can determine the problem's constraint boundary and shrink the solution based on those forecasting, but we didn't examine and expand for this experiment. Our goal is to ensure the robustness and effectiveness of our approach by testing the performance of our proposed data-driven uncertainty set. We aim to determine if it can accurately detect and exclude outliers, as well as capture the correct number of components and the shape of each cluster within the data and across different scenarios. We will also compare the objective, solutions, robustness, computational time and coverage rate. The scenarios are as follow:

- SCENARIO 1: We create 5 obvious distinct separate clusters and some outliers with 5% of all the data – 25 data points.

Table 1. Comparisons of different data-driven approaches.

	Box Uncertainty Set	Gamma Uncertainty Set	Proposed Method with 1 pre-defined component	*k*-means Uncertainty Set	Proposed Method with 5 predicted exported components
Minimize Objective	**347.43**	**391.56**	**336.27**	**242.05**	**228.90**
First-Stage Decisions	$x_1 = 30.09$	$x_1 = 14.89$	$x_1 = 14.68$	$x_1 = 18.53$	$x_1 = 18.45$
	$x_2 = 39.90$	$x_2 = 30.98$	$x_2 = 28.67$	$x_2 = 27.68$	$x_2 = 26.99$
Second-Stage	$y_1 = -1.59$	$y_1 = 17.64$	$y_1 = 14.93$	$y_1 = 6.97$	$y_1 = 6.19$
Decisions	$y_2 = 0$	$y_2 = 8.61$	$y_2 = 6.91$	$y_2 = 0.61$	$y_2 = 0.13$
Coverage Rate	100%	96%	96%	95%	95%
Computation time (s)	3	3	4 (+4 for bootstrap resampling)	4 4	4 (+20 for bootstrap resampling)

- SCENARIO 2: We test the same problem and data on a more symmetric data structure with normal linear pattern.
- SCENARIO 3: We evaluate our uncertainty set on wide asymmetric data variation spread, with noise such that the data appears at irregular intervals.

5.1.2. Results

For SCENARIO 1, a detailed table is provided in Table 1 with respect to different uncertainty set construction methods, coverage rate, objective, first-stage and second-stage decisions. Different types of traditional uncertainty sets, such as U_{box} and U_{budget} , as well as other unsupervised learning methods such as *k*-means, are constructed based on the same uncertainty data to test against our framework and approach.

We set the *threshold* as 5%. With the proposed data-driven uncertainty set, our nonparametric unsupervised learning model correctly captures the true number of components of the data, $M = 5$. The output of the means $\{\mu\}_{k=1}^{5}$ $=$ [13.83, 16.88], [18.39, 24.24], [10.19, 16.23], [18.16, 20.70], [12.96, 12.16], while the output of $\{\Gamma_1\}_{k=1}^{5} =$ [3.17, 3.35, 4.36, 4.66, 4.53] and for $\{\Gamma_\infty\}_{k=1}^{5} =$ [1.93, 2.47, 2.26, 2.50, 2.83], $\forall\, k = 1, 2, 3, 4, 5$ and $\alpha = 100\%$ coverage rate.

In Figure 6 we present the process to construct the data-driven uncertainty set, with identification of the outliers and remove them in Figure 6(b) and then our method determines the optimal number of components, with their means in Figure 6(c) for uncertain parameters u_1 and u_2.

In Figure 7 we experiment with the traditional uncertainty sets that are widely used in Robust Optimization, Box (Figure 7(a)) and Budget/Gamma (Figure 7(b)) Uncertainty Sets, which shows overly conservativeness. In this example, the solution being on the edge makes sense because the box set pushes the solution to the limits of the variable ranges to minimize the objective function within those extreme points, from the first constraint. We also experiment our approach with different pre-defined number of components and test the impact on the constructed data-driven uncertainty set. In Figure 7(c), after we identify and remove the outliers noted as data points with $(*)$, we export the uncertainty set with 1 pre-defined component, while in Figure 7(d) we export the uncertainty

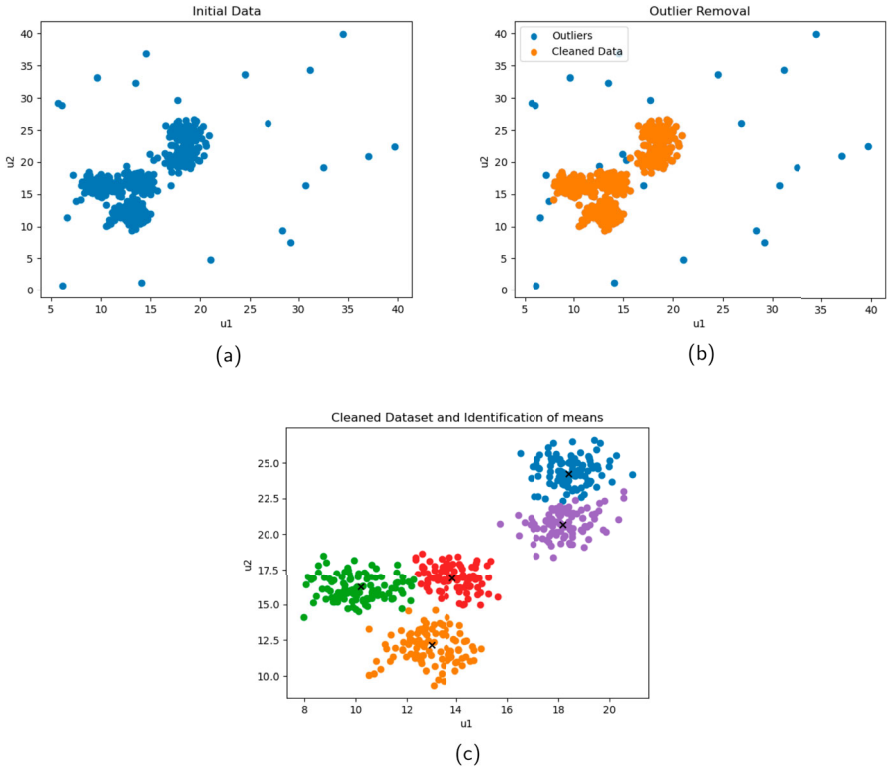

Figure 6. SCENARIO 1: Process for the construction of the Data-Driven Uncertainty Set. (a) initial data, (b) Outlier identification, (c) Inference in black the parameter mean of each predicted cluster.

set with 3 pre-defined component where we can observe that it is less conservative than the previous one. In Figure 7(e,f) we automatically determine the optimal number of components, where optimal number of components are 5, but we examine each norm separately and their affects on our approach, with the same predicted number of components, means, covariance, but different norms and Gammas. We observe that for solely l_∞ norm that the initial 'here-and-now' first-stage decisions are positioned at the top right of the box. It can be observed that out approach produces no 'wait-and-see' second-stage decisions with $y_1 = 0$ and $y_2 = 0$, resulting in a less conservative approach, as it does not introduce robustness against uncertainty and no flexibility. This shows that no robustness is introduced into the model, and it essentially treats the problem as a deterministic optimization. Consequently, no irreversible decisions are selected in the first-stage decisions, which is not necessarily wrong because it is allowed in our constraints, but we would prefer to have these future adjustments that makes our framework more robust. The second-stage decisions equal to zero essentially eliminates the flexibility provided because it is confident that the nominal solutions x_1 and x_2 are robust enough to handle all possible scenarios without the need for adjustments and deviations. In general, larger values of y_1 and y_2 lead to

a more conservative robust solution but might sacrifice some level of optimality. However, the solution of l_1 norm uncertainty set is optimal and inside the diamond shape, with both first and second-stage decisions robust enough. The data distribution is formed in a way that the l_1 norm, $|u_1| + |u_2|$, sufficiently captures the spread and the l_∞ norm, $\max(|u_1|, |u_2|)$, does not provide additional constraints.

As a result, our comprehensive framework and final concrete uncertainty set modelling suggests that the optimal number of components are 5, and thus we perfectly capture all the uncertain information and region in Figure 8, constructing and producing the best approach of our data-driven framework, the union of the above norms in a very nice uncertainty representation. This data-driven influenced uncertainty set accounts for deviations which lead to solutions that balance robustness rather than just pushing to the extremes of individual variables such as the previous conservative classical set. That's why the solution is more central, balancing deviations from multiple directions and avoiding pushing any single variable to its extreme, resulting in finding a solution that remains feasible under all possible scenarios within the uncertainty set.

To test the power of our framework against other existing unsupervised approaches, we implemented the *k*-means method to infer the parameters that we integrate to the same proposed data-driven uncertainty set. Since the drawback of *k*-means is that we have to determine beforehand the number of components, we utilize in Figure 9(a) the 'elbow method' on the same dataset without outlier, since *k*-means is extremely sensitive to that. We found that the optimal numbers of components can be 2, since we can split the data into two clusters (of two and three), or 5, as we correctly predicted. In Figure 9(b) we constructed the data-driven uncertainty set with *k*-means and we observe it is quite similar to our method. Delving deeper in Figure 9(c), we compared the main parameters influencing the efficiency of the uncertainty set, specifically the means. We found that the centroids from the *k*-means algorithm differ from our DPMM centres. While *k*-means generally performs well when the components are distinct from each other, our methodology outperformed it by paying closer attention to the actual weight of the data points in each cluster and their predominant locations within each component. We anticipate *k*-means to fail in other scenarios where data are asymmetrical and exhibit spread variation due to its inherent drawbacks, as well as the need to find and determine beforehand the optimal number of clustering.

In Figure 10 we present the objective with different coverage rates among the traditional and proposed data-driven uncertainty sets. As shown, the proposed data-driven uncertainty set with automatically defined 5 components results in lower objective values across various coverage rates. In contrast, the other methods yield higher objective values due to the overly conservative creation of the uncertainty set and unnecessary space. The *k*-means method was the closest to

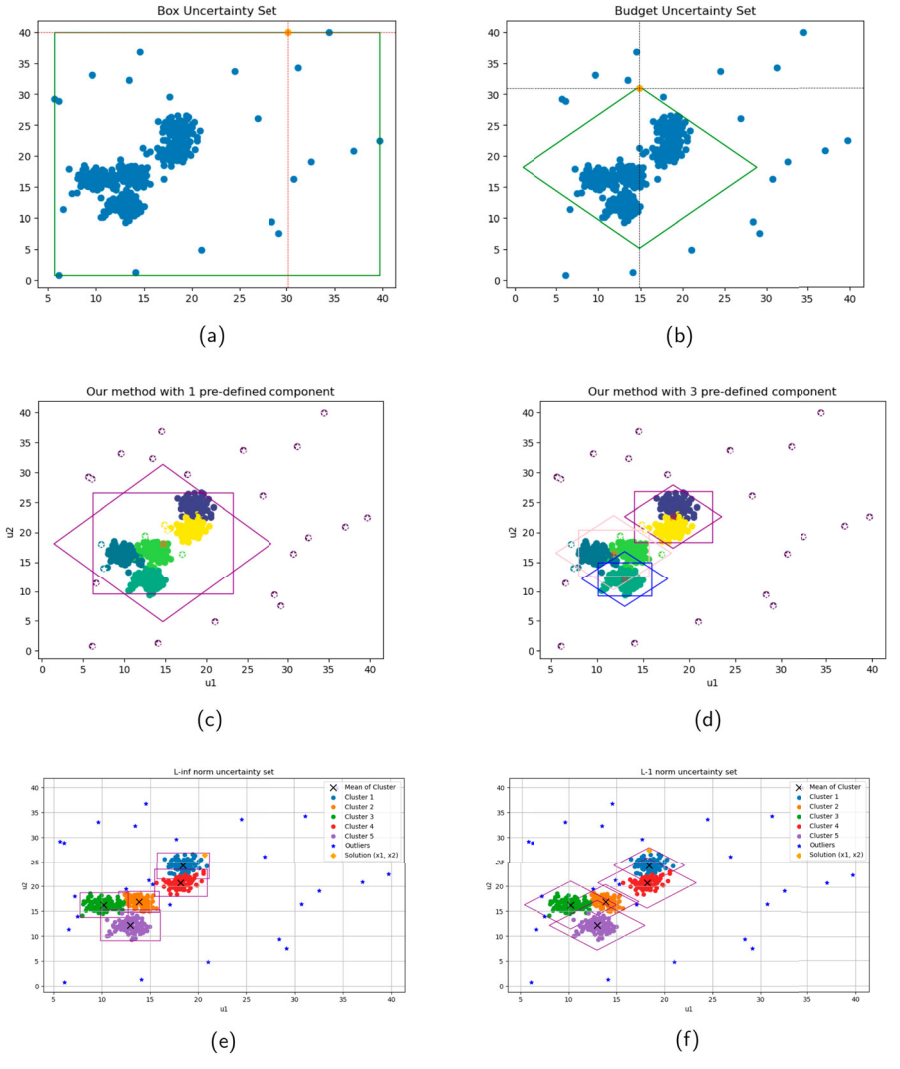

Figure 7. (a) and (b) are Traditional Box and Budget Uncertainty Sets. (c) is our approach with 1 predefined component, (d) our approach with 3 pre-defined components. (e) is our approach with just l_∞ norm uncertainty set and (f) with just l_1 norm uncertainty set.

our approach, largely because we use the same data-driven uncertainty set and adaptive scale parameters.

In the end, we will compare our bootstrap approach in contrast to the non-bootstrap one in Figure 11. We observe that the the bootstrap approach, straight line set, is more robust in contrast to the non-bootstrapped approach, dashed line set. Even though the objective is slightly higher (228.90 in contrast to 224.74), the robustness of our uncertainty set makes it a far better choice for generalization because we account the actual deviation of means and uncertain data points. This makes our model robust enough to cover any potential data or parameters perturbations, but also precise and accurate, because we let the data

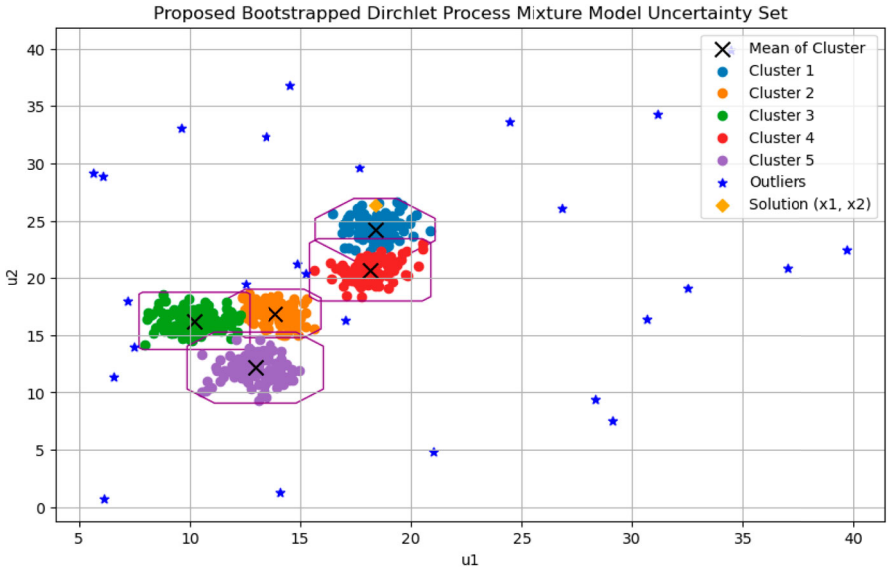

Figure 8. Proposed Bootstrap DPMM approach to construct the optimal number of components automatically.

behaviour and deviation influence the extension of the set. This is our trade off between conservativeness and optimally. For SCENARIO 2, we test the same motivational example on a more symmetric data structure with a normal linear pattern. Our aim is to observe the outcome of our proposed framework on different data structures, as shown in Figure 12. Once again, our method accurately captures the shape of the data, constructing a well-fitted data-driven uncertainty set that includes all necessary information and possible scenarios. This allows us to incorporate it into the same robust optimization problem and achieve the best objective.

Lastly, for SCENARIO 3, we will evaluate our uncertainty set on a dataset characterized by wide asymmetric variation and noticeable noise, where data appears irregularly, as shown in Figure 13. In Figure 13(c), we observe different clustering behaviour from *k*-means model compared to our proposed approach in Figure 14, indicating different parameter inferences for the uncertainty set construction that influence the solutions and objectives. Once again, for different data structure and different range of the two uncertain parameters than previous scenarios, our proposed machine learning methodology and data-driven uncertainty set construction method effectively capture the uncertainty, identify outliers, and produce less conservative objective of 121.04 instead of 122.38 from k-means, where the other two classical traditional sets produce much higher objective as expected.

To evaluate the robustness of our approach we perform our method on out-of-sample and unseen data with train-test split. Then, we perform cross-validation 5 times to evaluate the generalizability of our uncertainty modelling. We split

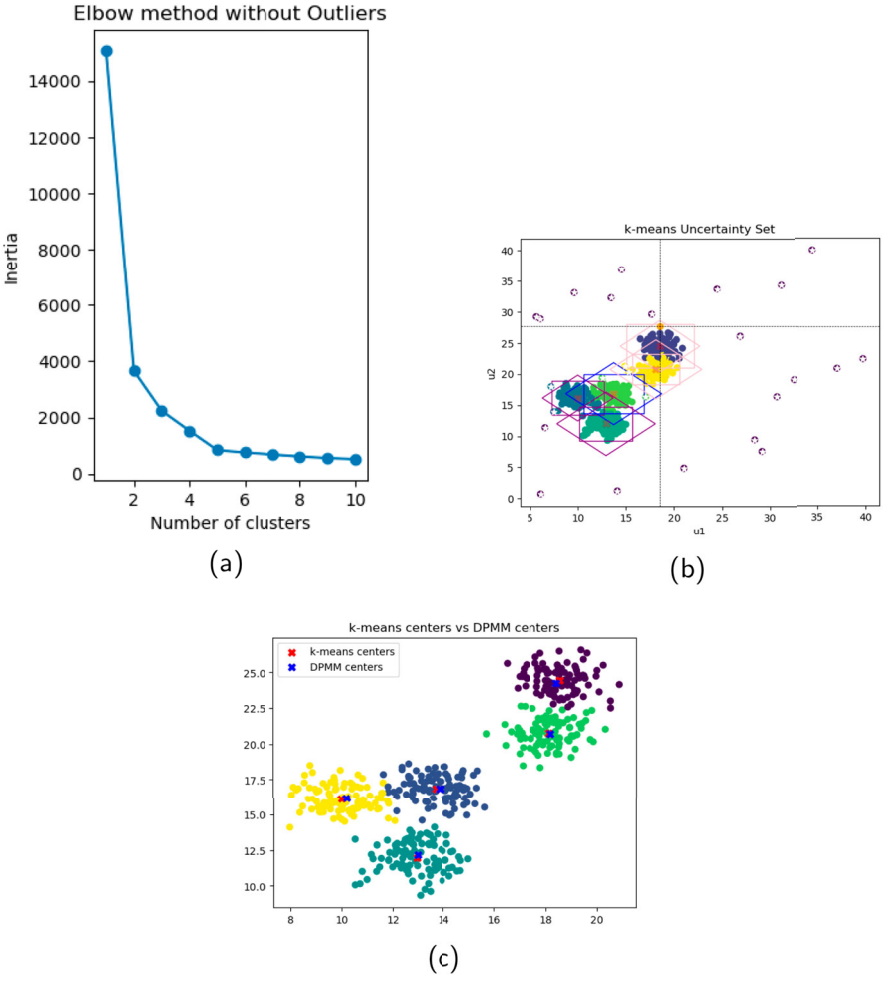

Figure 9. *k*-means algorithm, resulting uncertainty set construction and comparison.

the data to 400 training data and 100 test data, applied our above method to the train and evaluate on the test data. The results in Figure 15 are promising as expected on average of the 5 folds and running the data sample 100 times. For this example, we present in eclipse the previous uncertainty set as an approximation of how our uncertainty set construction performs. The test data fall inside our final combined trained approach with average *P*-Value of 0.77 which implies that that our model's uncertainty set is statistically similar to the true data distribution, indicating a good fit. At last, for SCENARIO 1, to further test the robustness and adaptability of our approach, we will add additional 100 data points, to see how our method's uncertainty set reacts to that, how the uncertainty modelling shifts in terms of the objective and solution in an iterative loop. Intentionally we add 2 more clear cluster to the existing dataset to evaluate if our clustering model will adapt and capture them, visualize dashed eclipses as

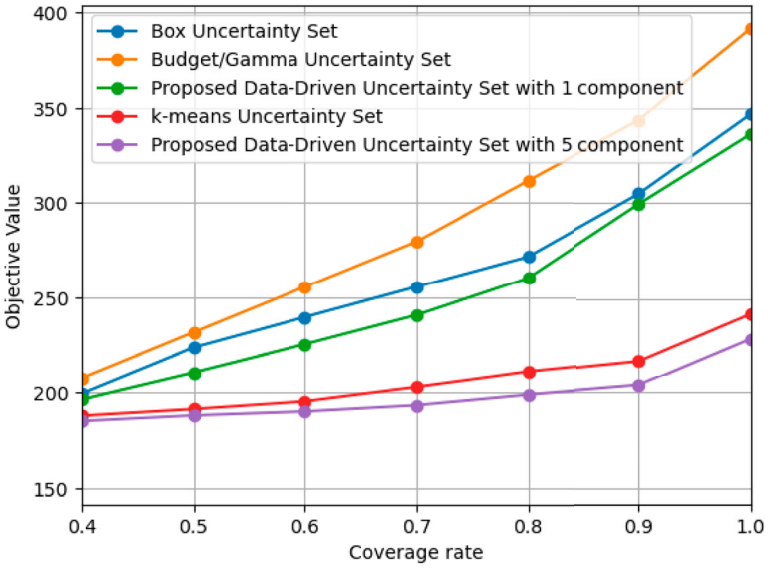

Figure 10. Coverage Rate Comparisons over different percentages of the same data.

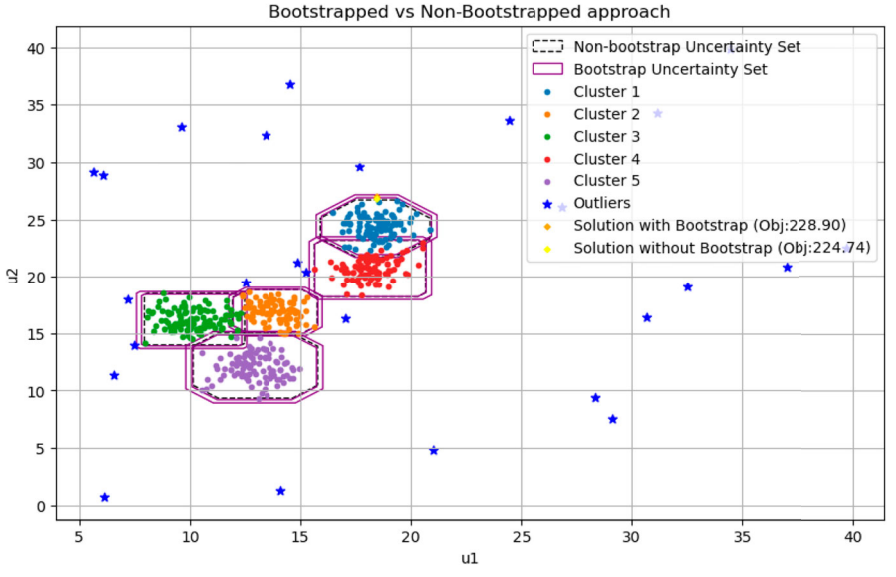

Figure 11. Bootstrap vs. Non-Bootstrap approach: difference between uncertainty set size, solutions and objectives.

approximation of our uncertainty set. We observe from Figure 16 that after 15 new data points, the model predicted and marked the majority as outliers without forming any new cluster or change solutions (orange point), but instead just adjusting the existing uncertainty set to the new modified distribution. After 35 data points, it predicted them correctly as new components and started forming 2 new clusters, changing the solution by trying to find the best approximation

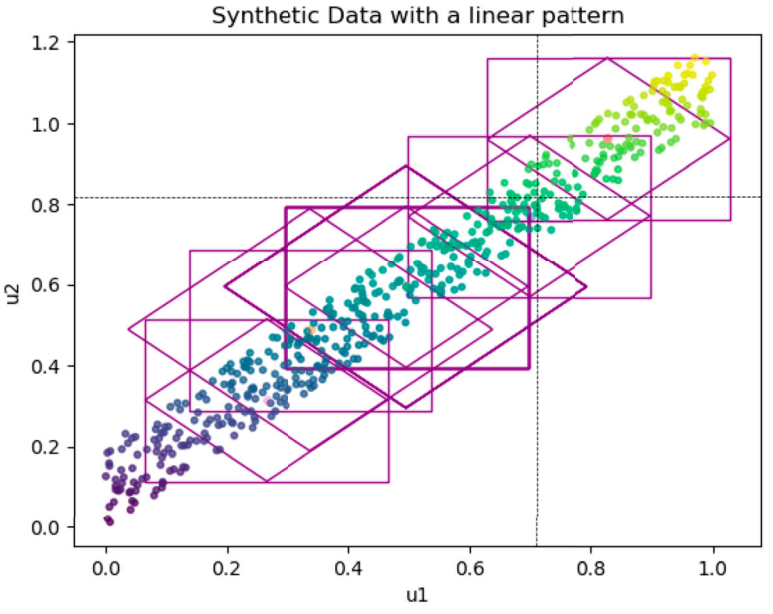

Figure 12. SCENARIO 2: Our framework works well on correlated dataset with linear pattern.

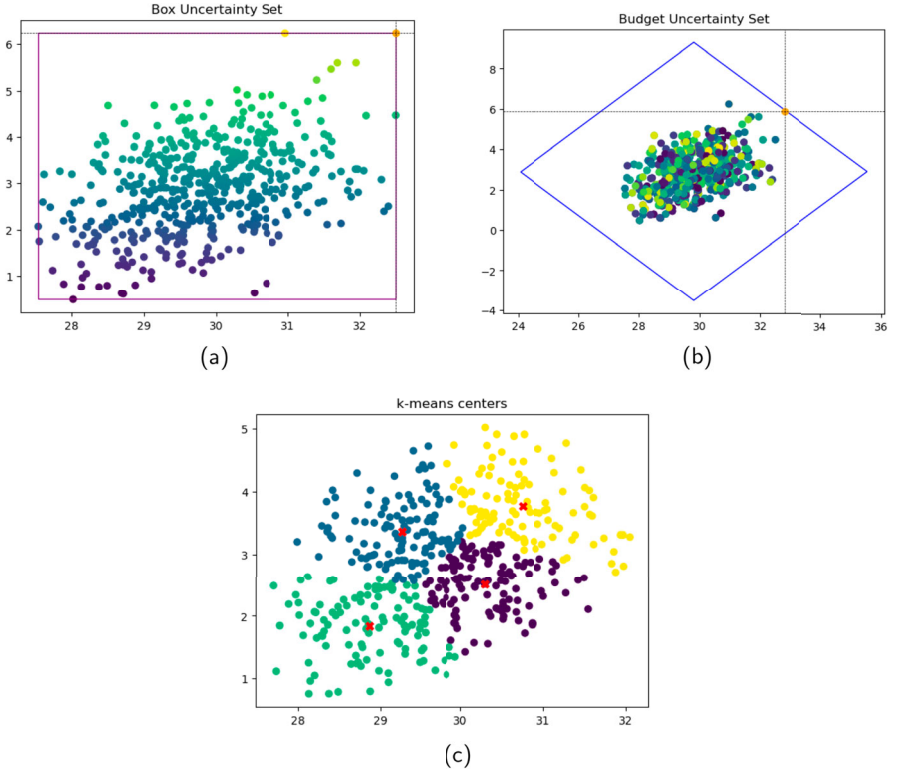

Figure 13. SCENARIO 3: Asymmetric dataset variation spread with noise in irregular intervals. (a) Traditional Box Uncertainty Set, (b) Traditional Budget Uncertainty Set, (c) Clustering behaviour of *k*-means after its own outlier detection.

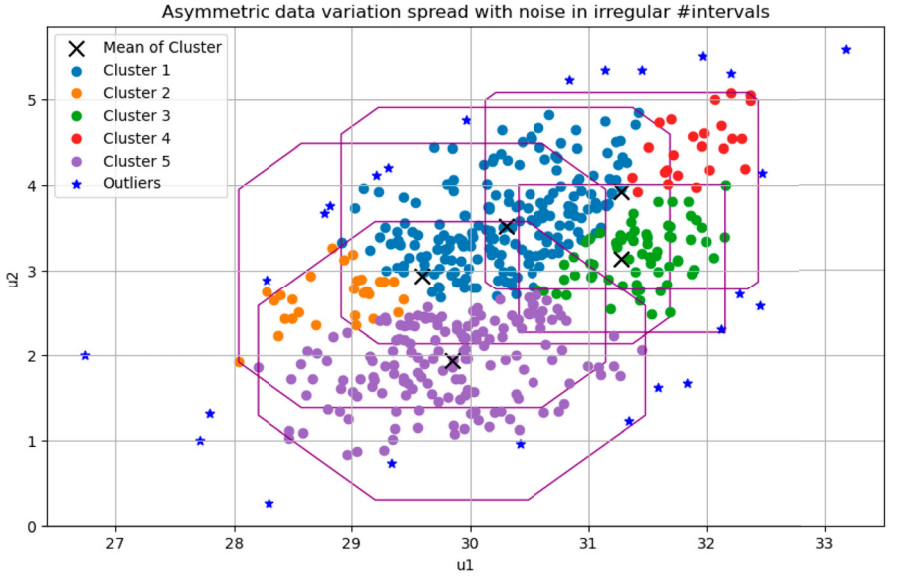

Figure 14. SCENARIO 3: Proposed data-driven uncertainty sets on dataset with data variation.

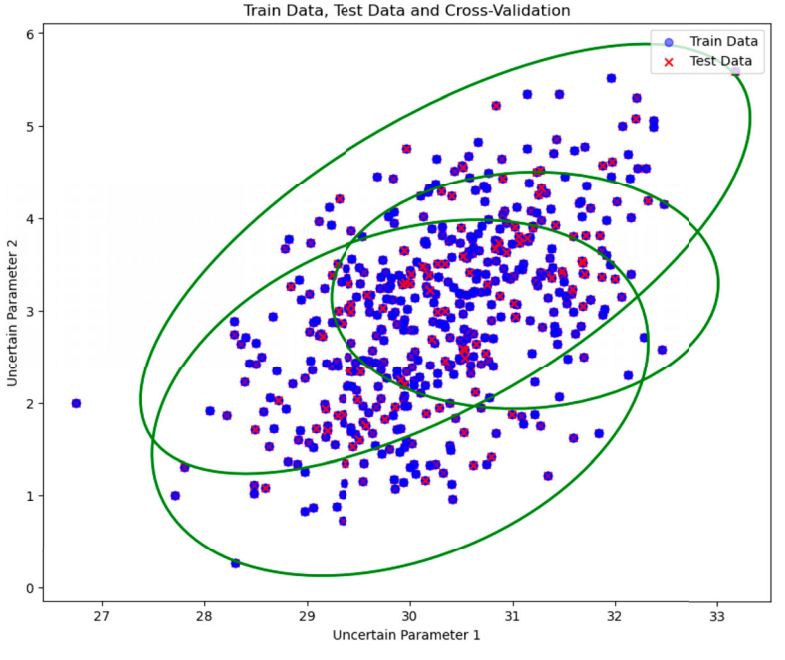

Figure 15. Train-test split and cross validation. Testing data fall inside the trained method and uncertainty set.

distribution and parameter that can express. In the end, after all the 100 data points are introduced, our method, changes the solution and final uncertainty set construction representation based on the constraints, proving that it can be

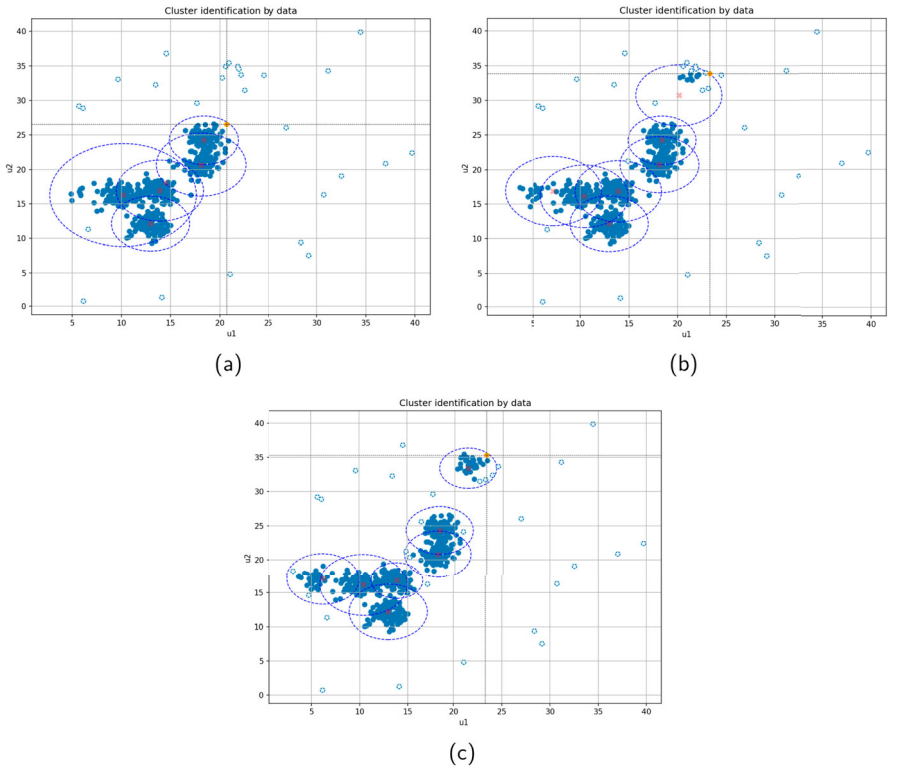

Figure 16. Testing on SCENARIO 1 to see how our approach behaves with additional 100 unseen out-of-sample data points over time: (a) Results after 15 new data points, (b) Results after 35 new data points, (c) Results after 100 new data points.

adaptive to new unseen data and not just solving the uncertain data without any knowledge.

5.2. Production planning example

5.2.1. Experimental settings

We will now consider a production planning problem in the context of a manufacturing company that produces electronic devices. The aim is to optimize a company's production decisions to meet customer demand while minimizing costs, and the objective is to determine the optimal production quantities and scheduling decisions to minimize costs. To address this problem, we applied a robust optimization technique considering uncertainties in demand, decisions, and production capacity. Our goal is to construct a compact uncertainty set by considering a range of possible scenarios and finding a production plan that performs well in all of those scenarios. See [49,50] for further reading. We present a simplified formulation of the production planning problem with additional details:

(1) Decision Variables:
   (a) *Production Quantities:* – x_t represents the quantity of devices produced at time period t.
   (b) *Inventory Levels:* – I_t represents the inventory level of devices at the end of time period t.
   (c) *Setup Variables:* – z_t is a binary variable that indicates whether a production setup is initiated at time period t. – v_t is a binary variable that indicates whether a production setup is completed at time period t.

(2) Robust Objective Function: The objective function is to minimize the costs under this worst-case scenario. For the production and inventory costs, the robust objective function can be written as:

$$\min_{x,I} \max_{d \in \mathcal{U}} \sum_t \left(C_{\text{prod}} \cdot x_t + C_{\text{hold}} \cdot I_t \right)$$

where C_{prod} is the production cost per device, and C_{hold} is the holding cost per device per time period.

(3) Constraints:
   (a) *Demand Constraint:* $I_t = I_{t-1} + x_t - D_t \; \forall \; t$
   (b) *Production Capacity Constraint:* $x_t \leq C_{\text{cap}} \; \forall \; t$
   (c) *Setup Variables Constraint:* $z_t - v_t = 0 \; \forall \; t, v_{t-1} - z_t \geq 0 \; \forall \; t$
   (d) *Setup Time Constraint:* $\sum_t (v_t - z_t) \leq T_{\max}$

The traditional Box uncertainty set for the demand, as explained above, can be defined as a range that encompasses possible values. For instance, let us define the uncertain demand at time period t as D_t and define its uncertainty set as:

$$\mathcal{U}_t = \{D_t \mid D_t \in [D_t^{\min}, D_t^{\max}]\} \tag{18}$$

where $[D_t^{\min}, D_t^{\max}]$ represents the lower and upper bounds of the demand at time period t. But this uncertainty set as explained is highly conservative and will not give us the optimal results. As a result and for comparison purposes, we will proceed with our proposed constructed data-driven uncertainty set in Equation (12).

5.2.2. Data, analysis and results

We consider the following scenario and uncertain synthetic demand data as the experiment: We have two products A and B, such as electronic devices. In this scenario, the demand for these devices fluctuates seasonally due to changes in customer preferences and market trends. Winter and Summer months exhibit higher demands than other months, over a period of 365 days. The demand D_t for each time period t is uncertain and vary. The minimum demand for a period t is denoted as $D_t^{\min}$ and the maximum $D_t^{\max}$. We set the production cost per device to $C_{\text{prod}} = 100$ and the holding cost per device per time period $C_{\text{hold}} = 10$. The

production capacity is determined $C_{\text{cap}} = 1000$ per time period t and the maximum setup time predicted is $T_{\max} = 3$. We have also experimented with different numerical values for these. For each time period t, d_1 represents the uncertain demand for electronic device A and d_2 represents the uncertain demand for electronic device B. In this experiment, we will have 12 solutions for 12 days for each time period $t = 1, 2, \ldots, 12$, instead of 1 like a usual robust optimization approach.

Initially, we addressed the experiment using conservative approaches with a box uncertainty set and Budget uncertainty set. In Figure 17, the visualization of the data over the 12 time periods, derived using the classical uncertainty sets, are shown. The overall 12 solutions (orange points) for both Product A and Product B using the Box and Budget Uncertainty set are shown there. Our framework chose to produce at the same maximum demand for each time period for each product, highlighting an extremely conservative approach. Budget approach performed slightly better than Box, but it was still highly conservative, disregarding seasonal fluctuations in demand and resulting in extremely conservative approach. Finally, the results of our approach are presented in Figure 18. Our ML algorithm initially captured the demand structure by approximating the data distribution variability and proposing the optimal number of components, the mean and covariance of each cluster, along with Gammas of l_1 and l_∞ norms for each component. This approach produced a less conservative uncertainty

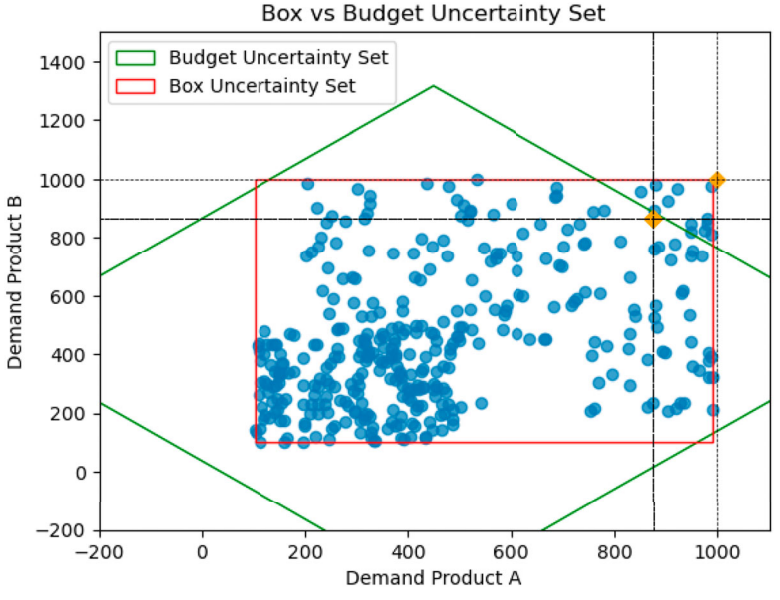

Figure 17. Traditional Uncertainty Sets for Production Planning experiment. Box vs Budget Uncertainty Sets.

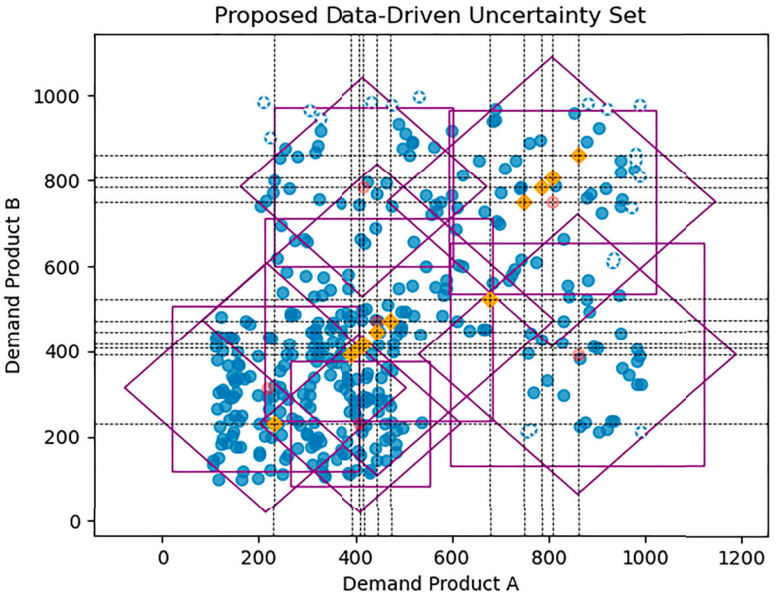

Figure 18. Proposed data-driven approach for experiment 2 over different time periods ($t = 12$).

set, avoiding unnecessary space, identifying outlier (marked as '$*$') and infeasible scenarios/solutions. For this experiment, the proposed uncertainty sets were manually constructed from scratch and adjusted accordingly based on the data. As such, our methodology identified the best predicted demand for each product, accounting for seasonal fluctuations and the original demand's possible interval levels.

When implemented, the objectives, first-stage decisions and inventory levels are presented in Table 2 for Product A and Product B. The solutions remained within the bounds of the constructed data-driven uncertainty sets, where bootstrap produce more robustness, ensuring no solutions fell outside the range of actual production and data points. The framework adjusted and adapted for subsequent 12 time periods, providing the best solutions for both electronic products. In addition, the objectives for the demand of Product A and Product B were significantly lower compared to the previous highly conservative methods. Lastly, the inventory level is lower due to the varying production in each period (t), which accounts for demand fluctuations. This approach is less conservative but remains robust across the different scenarios.

5.3. Robust vehicle pre-Allocation example

In this example, we solve a modified vehicle pre-allocation problem, initially introduced by Hao et al. [51]. Given I supply nodes and J demand nodes in an urban area, after random demand $d_j, j \in [J]$ is realized, the operator allocates x_{ij}

Table 2. Results and Comparison with existing approaches.

	Product A	Product B
Demand	d_A = [876, 638, 666, 522, 774, 314, 514, 424, 212, 814, 450, 388]	d_B = [234, 450, 452, 486, 608, 866, 860, 500, 232, 786, 258, 580]
Minimum Demand	$(d_A)_{min} = 212$	$(d_B)_{min} = 232$
Maximum Demand	$(d_A)_{max} = 876$	$(d_B)_{max} = 866$
Minimum/Maximum Demand estimated after final period	[2544, 10512]	[2784, 10392]
Objective with Box Uncertainty Set	**1,567,879**	**1,532,380**
Initial Solution with Box Uncertainty Set	[752, 1000, 876, 876, 876, 876, 876, 876, 876, 876, 876, 876]	[732, 1000, 866, 866, 866, 866, 866, 866, 866, 866, 866, 866]
Inventory Levels with Box Uncertainty Set	[752, 1752, 2628, 3504, 4380, 5256, 6132, 7008, 7884, 8760, 9636, 10512]	[732, 1732, 2598, 3464, 4330, 5196, 6062, 6928, 7794, 8660, 9526, 10392]
Objective with Gamma/Budget Uncertainty Set	**1,219,960**	**1,207,860**
Initial Solution with Gamma Uncertainty Set	[1252, 1000, 876, 876, 876, 876, 876, 876, 876, 876, 876, 876]	[1232, 1000, 866, 866, 866, 866, 866, 866, 866, 866, 866, 866]
Inventory Levels with Gamma Uncertainty Set	[1252, 2252, 3128, 4004, 4880, 5756, 6632, 7508, 8384, 9260, 10136, 11012]	[1232, 2232, 3098, 3964, 4830, 5696, 6562, 7428, 8294, 9160, 10026, 10892]
Objective with Proposed Data-Driven Uncertainty Set	**705,537.84**	**655,962.93**
Initial Solution with Proposed Data-Driven Uncertainty Set	[0, 678.78, 860.61, 392.53, 409.75, 230.54, 447.27, 473.49, 416.86, 784.74, 807.53, 750.36]	[0, 523.86, 860.61, 392.53, 409.75, 230.54, 447.27, 473.49, 416.86, 784.74, 807.53, 750.36]
Inventory Levels with Proposed Data-Driven Uncertainty Set	[0, 678.78, 1539.39, 1931.93, 2341.68, 2572.22, 3019.50, 3492.99, 3909.85, 4694.60, 5502.14, 6252.50]	[0, 523.86, 1384.47, 1777.01, 2186.76, 2417.30, 2864.57, 3338.07, 3754.93, 4539.68, 5347.22, 6097.58]

vehicles from supply node $i \in [I]$ to demand node $j \in [J]$ at a unit cost c_{ij}. The detailed parameters are listed as below:

- Number of supply nodes $I = 1$
- Number of demand nodes $J = 10$
- Revenue coefficients $r = (4.50, 4.41, 3.61, 4.49, 4.38, 4.58, 4.53, 4.64, 4.58, 4.32)$
- Cost coefficients $c_j = 3$, where $j = 1, 2, \ldots, J$
- Maximum supply of vehicles $q_i = 400$, where $i = 1, \ldots, I$

The vehicle pre-allocation was solved by the robust and sample robust optimization [48], where the decision under demand uncertainty was formulated by solving the robust optimization problem below:

$$\min_{x,y} \max_{u \in U} \left(\sum_{i=1}^{I} \sum_{j=1}^{J} (c_{ij} - r_j) x_{ij} + \sum_{j=1}^{J} r_j y_j \right)$$

$$s.t.$$

$$y_j \geq \sum_{i} x_{ij} - u_j \quad \forall\, u \in U, \quad \forall\, j \in [J], \quad \forall\, i \in [I] \tag{19}$$

Table 3. Robust Vehicle pre-allocation results.

	Box uncertainty set	ϵ-neighborhood uncertainty set	Proposed data-driven uncertainty set construction
Objective	−62.59	−103.65	−108.55
Pre-allocation decisions	[0, 0, 0, 0, 0, 39.61, 0, 0, 0, 0]	[0.34, 0.35, 0, 4.28, 0, 69.45, 2.45, 4.57, 5.22, 2.48]	[0, 0.11, 0, 1.55, 0, 75.86, 0.52, 2.76, 1.74, 0]
Coverage rate	100%	100%	100%
Computational time	10 sec	13 sec	31 sec

$$\sum_j x_{ij} \leq q_i$$

$$x_{ij}, y_j \geq 0$$

where the wait-and-see decision y represents the bookkeeping revenue. It was initially solved with a conservative box uncertainty set where the upper and lower bounds are identified using a sample demand dataset ($taxi - rain.csv$). The dataset includes the demands of 10 regions associated each revenue coefficient and 77 day time demands.

Our target is to construct a more compact uncertainty set that will produce precise but also robust results. We compare with two alternative uncertainty set construction approaches [48,51]: the sample average approximation approach with the conservative Box uncertainty set, defined as $U_{\text{box}} = (d|d_{\min} \leq d \leq d_{\max})$, and the two-stage sample robust model approach with the ϵ-neighborhood uncertainty set, which is defined $U_\epsilon = \left(d|d_{\min} \leq d \leq d_{\max}, \|d - \hat{d}_s\| \leq \epsilon\right)$ for a collection $\left(\hat{d}_1, \hat{d}_2, \ldots, \hat{d}_S\right)$, given in [52]. In Table 3 we present the results and the existing approaches in comparison with our algorithm and data-driven uncertainty set construction. We also demonstrate in Figure 19 the difference in quantity allocation for each model for each region. The results indicate that our proposed method perform significantly better than the conservative sample average approximation approach and slightly better than the sample robust approach. However, although our approach exhibits less conservative and more precise allocation, it is computationally more expensive especially in such small dataset. This is due to the bootstrap resampling that increases the computational burden.

5.4. Adaptive robust lot-sizing example

In this experimental setting, we examine a modified lot-sizing problem, initially proposed by Bertsimas and de Ruiter [53] and revisited by Chen et al. [48]. Within a network comprising N stores, the allocation of stock, represented by x_i for each store i, is established prior to the realization of the actual demand at each location. The demand, d, is subject to uncertainty and presumed to belong to a budget uncertainty set.

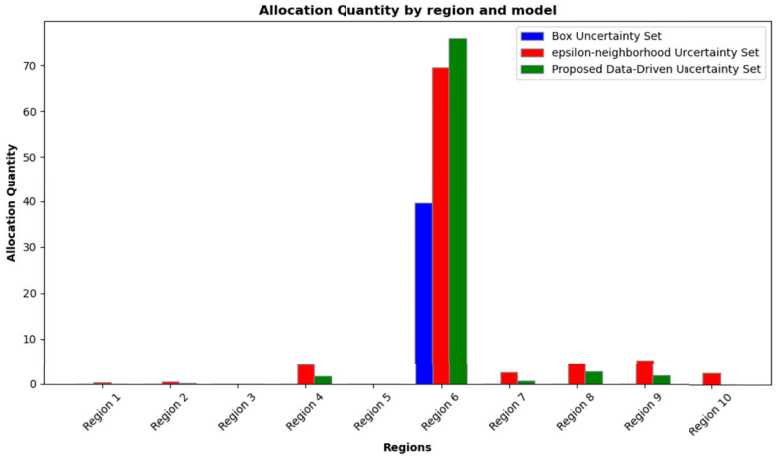

Figure 19. Experiment 3: Allocation Quantity by region and comparison between Box vs epsilon-neighborhood vs proposed data-driven uncertainty sets.

Following the observation of the demand realization, stock y_{ij}, the transportation from store i to j, aims to meet all demand with an associated cost t_{ij}. The goal is to minimize the total cost. The proposed uncertainty set and formulation of this example is given as below:

$$
\begin{aligned}
&U = (d : 0 \leq d \leq d_{max}, d \leq \Gamma) \\
&\min_{x,y} \max_{d} \sum_{i=1}^{N} c_i x_i + \sum_{i=1}^{N} \sum_{j=1}^{N} t_{ij} y_{ij} \\
&s.t. \\
&d_i \leq \sum_{j=1}^{N} y_{ji} - \sum_{j=1}^{N} y_{ij} + x_i d \in U \\
&0 \leq x_i \leq K_i \; i = 1, \ldots, N
\end{aligned}
\tag{20}
$$

For comparisons we set $N = 50$ locations and storage cost $c_i = 20$. t_{ij} is given from the Euclidean distance between the two stores. d_{max} is the maximum demand and K_i is the stock capacity both set to 20. The target is to identify the appropriate stock allocation at each location. The results are presented in Table 4 and the different visualization in the vehicle allocations for each store in Figure 20. For this experiment, the proposed uncertainty set was manually constructed from scratch and adjusted accordingly based on the data.

The experimental results demonstrate superiority of our proposed data-driven uncertainty set in optimizing stock allocation within the modified lot-sizing problem. Despite the increase in computational time, our method consistently outperforms the traditional budget uncertainty set in terms of objective cost, achieving a lower total cost of 3633.19 compared to 3647.01. The coverage rate

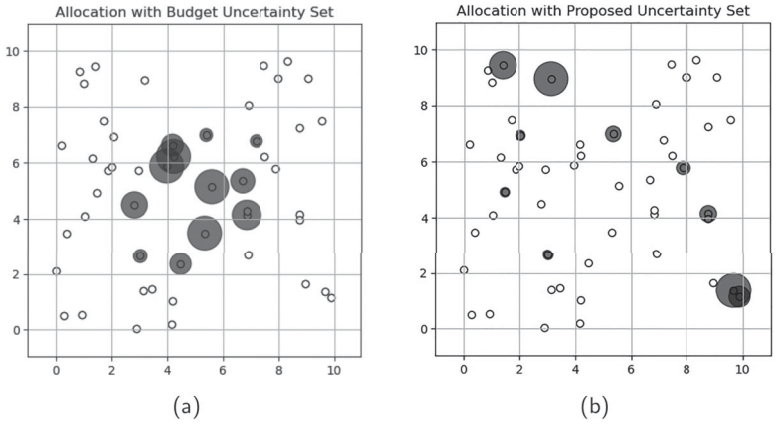

Figure 20. Experiment 4: Lot sizing allocation. Difference allocation between (a) Budget uncertainty set and (b) proposed bootstrapped DPMM uncertainty set.

Table 4. Adaptive robust lot-sizing results.

	Budget uncertainty set	Proposed data-driven uncertainty set construction
Objective	3647.01	3633.19
Coverage rate	100%	100%
Computational time	27 sec	58 sec

remains robust at 100% for both methods, demonstrating the reliability of the stock allocations under varying demand scenarios. The findings of this study clearly show that even though we slightly outperform and cannot fully leverage the unsupervised learning capabilities with only a limited amount of available data, the integration of Bootstrapped DPMM in constructing uncertainty sets enhances the decision-making process when faced with uncertain environments.

6. Conclusion

Our work advances decision-making under uncertainty by integrating unsupervised learning clustering and prescriptive analytics with data-driven uncertainty set construction for robust optimization. This versatile methodology is applicable to various robust optimization problems across multiple domains. We address the limitations of traditional methods by employing Dirichlet process mixture model to automatically shape uncertainty sets in robust optimization problems, where each predicted distribution corresponds to a cluster. This the first time we have shown that the novel and promising introduction of bootstrap resampling within this process significantly improves the efficiency and robustness of our solutions. Our extensive examples demonstrate the effectiveness and superiority of our approach, outperforming existing methods in terms of solution quality. However, the effectiveness of the approach relies on the quality and availability of

data. In scenarios with limited data, our constructed uncertainty sets accurately captured the true underlying uncertainties, resulting in a minor improvement in the objective value but at the cost of increased computation time compared to conservative methods.

Future research can expand on our methodology to build robust optimization models with automated training and validation procedures. Handling data sparsity more effectively and exploring weight allocation on data points or clusters through weighted stick-breaking modifications are areas for further investigation. Additionally, future research can explore the ideal coverage rate to set in advance and the most appropriate method for expanding the boundary of the uncertainty set in order to account for all vital data variations. Finally, advanced deep learning methods can be employed in robust optimization to calculate more optimal coefficients and provide valuable insights.

Acknowledgments

Data availability statement

This manuscript has no associated data.

Disclosure statement

No potential conflict of interest was reported by the author(s).

ORCID

Kerem Akartunalı http://orcid.org/0000-0003-0169-3833

References

[1] Qin SJ. Process data analytics in the era of big data. AIChE J. 2014;60(9):3092–3100. doi: 10.1002/aic.v60.9

[2] Jordan MI, Mitchell TM. Machine learning: trends, perspectives, and prospects. Science. 2015;349(6245):255–260. doi: 10.1126/science.aaa8415

[3] Ben-Tal A, Nemirovski A. Robust solutions of linear programming problems contaminated with uncertain data. Math Program. 2000;88:411–424. doi: 10.1007/PL00011380

[4] Garcia DJ, You F. Supply chain design and optimization: challenges and opportunities. Comput Chem Eng. 2015;81:153–170. doi: 10.1016/j.compchemeng.2015.03.015

[5] Sahinidis NV. Optimization under uncertainty: state-of-the-art and opportunities. Comput Chem Eng. 2004;28(6-7):971–983. doi: 10.1016/j.compchemeng.2003.09.017

[6] Bertsimas D, Sim M. The price of robustness. Oper Res. 2004;52(1):35–53. doi: 10.1287/opre.1030.0065

[7] Bertsimas D, Brown DB, Caramanis C. Theory and applications of robust optimization. SIAM Rev. 2011;53(3):464–501. doi: 10.1137/080734510

[8] Bertsimas D, Gupta V, Kallus N. Data-driven robust optimization. Math Program. 2018;167:235–292. doi: 10.1007/s10107-017-1125-8
[9] Ben-Tal A, Nemirovski A. Robust optimization–methodology and applications. Math Program. 2002;92:453–480. doi: 10.1007/s101070100286
[10] Uğurlu K. Terminal wealth maximization under drift uncertainty. Optimization. 2024;1–19. doi: 10.1080/02331934.2024.2324143
[11] Uğurlu K, Brzeczek T. Distorted probability operator for dynamic portfolio optimization in times of socio-economic crisis. Cent Eur J Oper Res. 2023;31(4):1043–1060. doi: 10.1007/s10100-022-00834-0
[12] Ben-Tal A, El Ghaoui L, Nemirovski A. Robust optimization. Vol. 28. Princeton: Princeton University Press; 2009.
[13] Bertsimas D, Thiele A. Robust and data-driven optimization: modern decision making under uncertainty. In: Models, methods, and applications for innovative decision making, INFORMS, 2006; p. 95–122.
[14] San Juan JL, Sy C. A data-driven target-oriented robust optimization framework: bridging machine learning and optimization under uncertainty. J Ind Prod Eng. 2024;41:1–25.
[15] Jin H, Li Z, Sun H, et al. A robust aggregate model and the two-stage solution method to incorporate energy intensive enterprises in power system unit commitment. Appl Energy. 2017;206:1364–1378. doi: 10.1016/j.apenergy.2017.10.004
[16] Birge JR, Louveaux F. Introduction to stochastic programming. Springer Science & Business Media; 2011.
[17] Faccini D, Maggioni F, Potra FA. Robust and distributionally robust optimization models for linear support vector machine. Comput Oper Res. 2022;147:105930. doi: 10.1016/j.cor.2022.105930
[18] Tay DH, Ng DK, Tan RR. Robust optimization approach for synthesis of integrated biorefineries with supply and demand uncertainties. Environ Prog Sustain Energy. 2013;32(2):384–389. doi: 10.1002/ep.v32.2
[19] Gong J, You F. Optimal processing network design under uncertainty for producing fuels and value-added bioproducts from microalgae: two-stage adaptive robust mixed integer fractional programming model and computationally efficient solution algorithm. AIChE J. 2017;63(2):582–600. doi: 10.1002/aic.v63.2
[20] Ning C, You F. Data-driven adaptive nested robust optimization: general modeling framework and efficient computational algorithm for decision making under uncertainty. AIChE J. 2017;63(9):3790–3817. doi: 10.1002/aic.v63.9
[21] Dai L, You D, Yin X, et al. Distributionally robust dynamic economic dispatch model with conditional value at risk recourse function. Int Trans Electr Energy Syst. 2019;29(4):e2775. doi: 10.1002/etep.v29.4
[22] Ning C, You F. Data-driven adaptive robust unit commitment under wind power uncertainty: a Bayesian nonparametric approach. IEEE Trans Power Syst. 2019;34(3):2409–2418. doi: 10.1109/TPWRS.59
[23] Goerigk M, Kurtz J, et al. Data-driven robust optimization using unsupervised deep learning. Comput Oper Res. 2023;151(106087).
[24] Blei DM, Jordan MI. Variational inference for Dirichlet process mixtures; 2006.
[25] Zhu J, Ge Z, Song Z. Variational bayesian gaussian mixture regression for soft sensing key variables in non-Gaussian industrial processes. IEEE Trans Control Syst Technol. 2016;25(3):1092–1099. doi: 10.1109/TCST.2016.2576999
[26] Bertsimas D, Georghiou A. Design of near optimal decision rules in multistage adaptive mixed-integer optimization. Oper Res. 2015;63(3):610–627. doi: 10.1287/opre.2015.1365

[27] Julien E, Postek K, Birbil SI. Machine learning for *k*-adaptability in two-stage robust optimization, arXiv preprint arXiv:2210.11152; 2022.
[28] Dumouchelle J, Julien E, Kurtz J, et al. Neur2ro: neural two-stage robust optimization, arXiv preprint arXiv:2310.04345; 2023.
[29] Goerigk M, Kurtz J. Data-driven prediction of relevant scenarios for robust optimization, arXiv e-prints arXiv–2203; 2022.
[30] Goerigk M, Kurtz J. Data-driven robust optimization using deep neural networks. Comput Oper Res. 2023;151:106087. doi: 10.1016/j.cor.2022.106087
[31] Ruff L, Vandermeulen R, Goernitz N, et al. Deep one-class classification. In: International Conference on Machine Learning, PMLR, 2018; p. 4393–4402.
[32] Ning C, You F. Data-driven decision making under uncertainty integrating robust optimization with principal component analysis and kernel smoothing methods. Comput Chem Eng. 2018;112:190–210. doi: 10.1016/j.compchemeng.2018.02.007
[33] Shang C, You F. A data-driven robust optimization approach to scenario-based stochastic model predictive control. J Process Control. 2019;75:24–39. doi: 10.1016/j.jprocont.2018.12.013
[34] Shang C, Huang X, You F. Data-driven robust optimization based on kernel learning. Comput Chem Eng. 2017;106:464–479. doi: 10.1016/j.compchemeng.2017.07.004
[35] Arabsheybani A, Arshadi Khamseh A, Pishvaee MS. Sustainable cold supply chain design for livestock and perishable products using data-driven robust optimization. Int J Manag Sci Eng Manag. 2024;19:1–16.
[36] Burgard JP, Moreira Costa C, Schmidt M. Robustification of the k-means clustering problem and tailored decomposition methods: when more conservative means more accurate. Ann Oper Res. 2022;339:1–44.
[37] Bertsimas D, Kim CW. A machine learning approach to two-stage adaptive robust optimization. Eur J Oper Res. 2024;319:16–30.
[38] Wang I, Becker C, Van Parys B, et al. Mean robust optimization, arXiv preprint arXiv:2207.10820; 2022.
[39] Bertsimas D, Den Hertog D, Pauphilet J. Probabilistic guarantees in robust optimization. SIAM J Optim. 2021;31(4):2893–2920. doi: 10.1137/21M1390967
[40] Dai L, You D, Yin X. Data driven robust energy and reserve dispatch based on a nonparametric Dirichlet process Gaussian mixture model. Energies. 2020;13(18):4642. doi: 10.3390/en13184642
[41] Li N, Li W, Jiang Y, et al. Deep dirichlet process mixture models. In: Uncertainty in Artificial Intelligence, PMLR, 2022; p. 1138–1147.
[42] Kimura T, Tokuda T, Nakada Y, et al. Expectation-maximization algorithms for inference in Dirichlet processes mixture. Pattern Anal Appl. 2013;16:55–67. doi: 10.1007/s10044-011-0256-4
[43] Bertsimas D, Van Parys B. Bootstrap robust prescriptive analytics. Math Program. 2022;195(1-2):39–78. doi: 10.1007/s10107-021-01679-2
[44] Yao L, Wang X, Duan C, et al. Data-driven distributionally robust reserve and energy scheduling over Wasserstein balls. IET Gener Transm Distrib. 2018;12(1):178–189. doi: 10.1049/gtd2.v12.1
[45] Chu Y, You F. Integrated scheduling and dynamic optimization of complex batch processes with general network structure using a generalized benders decomposition approach. Ind Eng Chem Res. 2013;52(23):7867–7885. doi: 10.1021/ie400475s
[46] Campbell T, How JP. Bayesian nonparametric set construction for robust optimization. In: 2015 American Control Conference (ACC), IEEE, 2015; p. 4216–4221.

[47] Caron F, Davy M, Doucet A, et al. Bayesian inference for linear dynamic models with Dirichlet process mixtures. IEEE Trans Signal Process. 2007;56(1):71–84. doi: 10.1109/TSP.2007.900167
[48] Chen Z, Sim M, Xiong P. Robust stochastic optimization made easy with RSOME. Manage Sci. 2020;66(8):3329–3339. doi: 10.1287/mnsc.2020.3603
[49] Quezada F, Gicquel C, Kedad-Sidhoum S. Combining polyhedral approaches and stochastic dual dynamic integer programming for solving the uncapacitated lot-sizing problem under uncertainty. Informs J Comput. 2022;34(2):1024–1041. doi: 10.1287/ijoc.2021.1118
[50] Akartunalı K, Dauzère-Pérès S. Dynamic lot sizing with stochastic demand timing. Eur J Oper Res. 2022;302(1):221–229. doi: 10.1016/j.ejor.2021.12.027
[51] Hao Z, He L, Hu Z, et al. Robust vehicle pre-allocation with uncertain covariates. Prod Oper Manag. 2020;29(4):955–972. doi: 10.1111/poms.13143
[52] Bertsimas D, Shtern S, Sturt B. Two-stage sample robust optimization. Oper Res. 2022;70(1):624–640. doi: 10.1287/opre.2020.2096
[53] Bertsimas D, de Ruiter FJ. Duality in two-stage adaptive linear optimization: faster computation and stronger bounds. Informs J Comput. 2016;28(3):500–511. doi: 10.1287/ijoc.2016.0689