A Clustering-based uncertainty set for Robust Optimization

Alireza Yazdani, Ahmadreza Marandi, Rob Basten, Lijia Tan

Department of Industrial Engineering and Innovation Sciences, Eindhoven University of Technology.

*Corresponding author(s). E-mail(s): a.marandi@tue.nl;
Contributing authors: a.yazdani.esfidvajani@tue.nl; r.j.i.basten@tue.nl; l.tan1@tue.nl;

Abstract

Robust Optimization (RO) is an approach to tackle uncertainties in the parameters of an optimization problem. Constructing an uncertainty set is crucial for RO, as it determines the quality and the conservativeness of the solutions. In this paper, we introduce an approach for constructing a data-driven uncertainty set through volume-based clustering, which we call Minimum-Volume Norm-Based Clustering (MVNBC), that leads to less conservative solutions. MVNBC extends the concept of Minimum-Volume Ellipsoid Clustering by enabling customizable regions containing clusters. These regions are defined based on a given set of vector norms, hence providing great flexibility in capturing diverse data patterns. We formulate a mixed-integer conic optimization problem for MVNBC. To address computational complexities, we design an efficient iterative approximation algorithm where we reassign points to clusters and improve the volume of the regions. Our numerical experiments demonstrate the effectiveness of our approach in capturing data patterns and finding clusters with minimum total volume. Moreover, constructed uncertainty sets based on MVNBC result in robust solutions with 10% improvement in the objective value compared to the ones obtained by a recent data-driven uncertainty set. Therefore, using our uncertainty sets in RO problems can generate less conservative solutions compared to traditional uncertainty sets as well as other existing data-driven approaches.

Keywords: Robust Optimization, Uncertainty set, Clustering, Data-driven Optimization, Mixed-Integer Non-linear Programming, Conic Optimization

1 Introduction

The presence of uncertainty can significantly impact the performance of algorithms applied to optimization models [1]. Robust Optimization (RO) is a widespread technique developed to address the challenge of uncertainty in optimization problems, aiming to find a solution safeguarded against all scenarios in the so-called uncertainty set [2].

Let us consider the following convex optimization problem containing uncertainty,

$$\inf_{\boldsymbol{x}\in\mathbb{R}^n} \{g(\boldsymbol{x}) \mid f_j(\boldsymbol{x},\boldsymbol{u}) \leq 0,\ \forall j \in [m]\}, \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the vector of decision variables, $g\colon \mathbb{R}^n \to \mathbb{R}$ is the objective function, $f_j(\cdot,\boldsymbol{u})\colon \mathbb{R}^n \to \mathbb{R}$, $j \in [m]$, are closed, proper, and convex functions for any realization of the uncertain parameter $\boldsymbol{u}$, which lies in the uncertainty se $\mathcal{U} \subset \mathbb{R}^d$, and m is the number of constraints. The robust counterpart of (1) reads as

$$\inf_{\boldsymbol{x}\in\mathbb{R}^n} \{g(\boldsymbol{x}) \mid f_j(\boldsymbol{x},\boldsymbol{u}) \leq 0,\ \forall \boldsymbol{u} \in \mathcal{U}, j \in [m]\}. \tag{2}$$

A main criticism of RO is that it can provide an overly conservative solution [3], meaning that a solution is found that is safeguarded against many scenarios that are unlikely to happen. The role of uncertainty sets in shaping the conservatism of solutions in RO problems is crucial [4, 5]. Although constructing classical uncertainty sets, such as boxes, ellipsoids, or diamonds, without considering the patterns of data requires little computational effort, they may not effectively represent data patterns. Therefore, researchers have started to construct the uncertainty set using data, forming the data-driven RO domain.

Data-driven uncertainty sets are constructed based on historical observations. By considering historical data, the optimization problem is better equipped to deal with similar patterns, leading to less conservative solutions [6]. There are various ways to construct data-driven uncertainty sets, including statistical hypothesis tests [7], kernel-based support vector clustering [8], principle component analysis [9, 10], and neural networks [11]. We further discuss the literature in Section 1.1.

In our work, we present a method for defining uncertainty sets based on available data. The fundamental idea is to use a clustering method to construct the uncertainty set. Clustering is one of the methods in Machine Learning (ML) to capture data characteristics. This method is known to be one of the most explainable ML methods [12]. One of the approaches to clustering data is volume-based clustering.

We formulate a mathematical optimization problem for constructing an uncertainty set incorporating clustering. The formulated problem is a Mixed Integer Non-Linear Optimization (MINLO) aiming to find the best clusters of data by minimizing the multi-dimensional volume of the clusters and being able to detect outliers at a desired rate. To solve the problem, we first reformulate the MINLO problem as a Mixed Integer Conic Optimization (MICO), and we then develop an algorithm that reassigns points to clusters iteratively and reduces clusters' volumes throughout this process. The first numerical results demonstrate that the approach outperforms some

typical clustering methods in finding clusters with minimum volumes. The results of the second numerical experiment also show the effectiveness of our constructed uncertainty set in RO, providing about 10% better objective value and highlighting the potential for our data-driven uncertainty set construction approach to be applied to a wide range of RO problems.

To explain our contribution in more detail, we first discuss the related literature in Section 1.1, afterwhich, we explain our contributions in Section 1.2 explicitly. Finally, we introduce the structure of the remainder of this paper in Section 1.3.

1.1 Related works

In this section, we first review the existing literature on constructing uncertainty sets from data and identify a gap between the two streams of approaches. Next, we explore different clustering methods to determine the most suitable one for constructing uncertainty sets and conclude that volume-based clustering is adopted as our approach. We then review the literature on minimum-volume ellipsoid clustering, which is the type of clustering on which our clustering method builds.

1.1.1 Data-driven uncertainty sets

There are two major streams in the approaches to construct data-driven uncertainty sets: the construction of classical uncertainty sets and the construction of ML-based ones. One of the first papers to construct classical uncertainty sets is Bertsimas et al. [7], where the authors integrate statistical hypothesis tests and goodness-of-fit with prior assumptions on data distributions. Zhang et al. [13] develop classical uncertainty sets like box, ellipsoidal, and polyhedral sets, with adjustable parameters derived from confidence intervals of data distributions. The main disadvantage of considering the classical uncertainty set is that when data do not follow a pattern that can be fitted in classical uncertainty sets, using these sets may lead to overly conservative solutions.

There is quite some variety in the second stream of approaches. Shang et al. [8] use kernel-based support vector clustering (SVC) to construct their uncertainty set. Goerigk and Kurtz [11] employ deep neural networks to discover patterns in data, which could also detect outliers, outperforming the kernel-based SVC. Ning and You [9] utilize principal component analysis and kernel smoothing techniques such as kernel density estimation to create an uncertainty set, which suffers from the curse of dimensionality. Zhang et al. [10] utilize PCA to identify the patterns in the data and create a polyhedral uncertainty set. They employ cutting-plane methods to capture data patterns more accurately. However, this method still takes into account some less-probable-to-happen scenarios. Furthermore, generating the cutting plane component can be computationally complex. Asgari et al. [14] use a modified version of the SVC, called position-regulated support vector clustering, to construct their uncertainty set. However, the kernel function is not capable of capturing multiple patterns in data points.

While classical uncertainty sets are interpretable, and using them results in a computationally efficient robust counterpart, they often struggle to capture data patterns

effectively. ML methods, on the other hand, lack interpretability and can be computationally expensive while being able to detect data patterns. The volume-based clustering approach can balance between computational efficiency and interpretability.

1.1.2 Minimum volume-based clustering

Clustering refers to an unsupervised learning technique in which related data points are grouped together based on some similarities [15]. Fundamental clustering methods can be classified into partitioning methods, hierarchical methods, and density-based methods [16, pp. 383-385]. Partitioning algorithms, like K-means, divide a dataset into K disjoint clusters, ensuring that each cluster contains at least one point and each point belongs to exactly one group. Various algorithms in this class aim to minimize an objective function, such as the total distance to the mean of each cluster, guiding the clustering process [16, pp. 385-390]. Hierarchical clustering methods group individual elements into larger and larger clusters, starting with individual points and gradually combining them based on their similarity. This process creates a tree-like structure that represents the relationships between the data points [17]. In density-based clustering, a cluster is a group of data points that are situated in the data space and are packed closely together in a region of high density. These density-based clusters are separated from one another by contiguous areas of low object density [18].

Volume-based clustering, belonging to the class of density-based clustering, groups data points according to the clusters' volume. Cluster volumes can be defined in different ways. One customary method, known as minimum volume ellipsoid clustering (MVEC), is to define the smallest volume of the ellipsoid, which encloses all points in the cluster. It is first introduced by Rosen [19] as a convex optimization method for separating patterns in data. Barnes [20] provides a heuristic algorithm for finding these patterns using eigenvalue decomposition. MVEC approach is scale-invariant, so it is able to handle asymmetric and unequal clusters [21]. Therefore, this approach is useful for grouping data points with different densities and identifying clusters with overlapping boundaries. In addition, it also has the advantage of identifying outliers.

Henk [22] defines an ellipsoid $\mathcal{E}$ in a d-dimensional Euclidean space $\mathbb{R}^d$ as the image of the ℓ_2-*norm* unit ball $\mathcal{B}^2$, which is an ℓ_2-*norm* ball of radius 1 centered at the origin, under a regular affine transformation. This implies that there exists an invertible matrix $\mathbf{T}^{-1} \in \mathbb{R}^{d\times d}$ and a center $-\mathbf{T}^{-1}\mathbf{t} \in \mathbb{R}^d$ such that the ellipsoid can be obtained by applying $\mathbf{T}^{-1}$ to $\mathcal{B}^2$ and transferring the center of the ball by $-\mathbf{T}^{-1}\mathbf{t}$, i.e.

$$\begin{aligned}\mathcal{E} = \mathbf{T}^{-1}\mathcal{B}^2 - \mathbf{T}^{-1}\mathbf{t} &= \{-\mathbf{T}^{-1}\mathbf{t} + \mathbf{T}^{-1}y : y \in \mathcal{B}^2\} \\ &= \{x \in \mathbb{R}^d : \left\|(\mathbf{T}^{-1})^{-1}(x + \mathbf{T}^{-1}\mathbf{t})\right\|_2 \leq 1\} \\ &= \{x \in \mathbb{R}^d : \|\mathbf{T}x + \mathbf{t}\|_2 \leq 1\},\end{aligned}$$

where $\|\cdot\|_2$ denotes the ℓ_2-norm or Euclidean norm.

A Minimum-Volume Ellipsoid (MVE), also known as the Löwner ellipsoid [22], is a geometric form that minimizes the volume required to contain a given set of points [23]. The volume of an ellipsoid is correlated with the determinant of its transformation matrix, $\det(\mathbf{T}^{-1})$. The exact details of this correlation are given in Section 2.

The computation of minimum-volume ellipsoids involves optimization techniques that minimize a specific objective function involving $\det(\mathbf{T}^{-1})$. To efficiently approximate the parameters of the ellipsoids, namely the transformation matrix $\mathbf{T}$ and translation vector $\mathbf{t}$, various algorithms and heuristics have been proposed, including simulated annealing, genetic algorithms, and tabu search [24], arithmetic operations [25], iterative methods [26], convex optimization approaches [27], re-sampling algorithms [28], and randomized algorithms [29].

Shioda and Tunçel [30] formulate MVEC as a MINLO problem. The objective function they use for their optimization problem does not fully capture the volume, as we discuss in Appendix B. Kumar and Orlin [21] formulate an MVEC problem considering outliers as a mixed integer non-convex problem. Martínez-Rego et al. [27] present a new non-convex formulation that avoids direct evaluation of determinants and does not consider outliers in the estimation process of ellipsoids.

So, in the literature, there are different formulations for a minimum volume-based clustering problem where the regions containing clusters are ellipsoids. Moreover, there are algorithms to solve the MVEC problem; however, their performance is highly case-dependent. To enhance compatibility with diverse data patterns, we extend the MVEC by incorporating general norm-based regions, which we call the *minimum volume norm-based clustering* (MVNBC). This flexibility empowers us to deploy multiple regions tailored to varying spatial patterns of data points. Moreover, leveraging the classical shapes of uncertainty sets as regions for each cluster enables us to develop a computationally tractable optimization problem to find an optimal robust solution.

1.2 Our contribution

We make a five-fold contribution to the literature:

1. We introduce a *norm-based clustering* method that relies on cluster regions' volumes. Extending the concept of Minimum Volume Ellipsoid Clustering (MVEC), we develop Minimum volume Norm-Based Clustering (MVNBC), a mathematical optimization problem that is based on a combination of vector norms in $\mathbb{R}^d$. This method can capture data patterns effectively and detect outliers at a desired rate.
2. We develop a *Mixed Integer Conic Optimization (MICO) formulation* for norm-based clustering, which contains an *exponential* cone and a *logdet* cone for each cluster, and a *quadratic* cone for each data point and each cluster.
3. Recognizing the computational complexities inherent in solving our problem, we introduce an *efficient iterative approximation algorithm* based on iteratively reassigning points to clusters and reducing the volume.
4. We implement MVNBC to *construct a new uncertainty set* for Robust Optimization (RO) and obtain its robust counterpart for convex optimization problems.
5. We conduct an extensive *numerical experiment* to highlight the practical significance of our proposed approach. Our numerical results demonstrate that our algorithm is more effective in finding the patterns in data sets than K-means and GMM. Moreover, results show that using our uncertainty set not only leads to 10% better objective value but also RO problems can be solved faster by orders of magnitude compared to one of the best benchmarks in literature.

1.3 Paper organization

Section 2 provides the notation and mathematical preliminaries used throughout the paper. Section 3 introduces our norm-based clustering optimization problem and its MICO formulation. In Section 4, we provide the solution algorithm for the MICO problem. In Section 5, we provide the robust counterpart reformulation of an RO problem using our uncertainty set. Section 6 contains the numerical experiments in which we use our approach to cluster different data sets and construct uncertainty sets. Finally, we conclude the paper in Section 7 and give potential future research directions.

2 Preliminaries

In this section, we define the notation and mathematical concepts that are used throughout the paper. Next, we recall the definition of the volume of a set and explain how to calculate the volume of a region.

2.1 Notation

We use the following notation throughout the paper.

- Matrices are represented by uppercase bold letters, such as $\mathbf{A}$ or $\mathbf{B}$.
- Vectors are denoted using lowercase bold letters, such as $\mathbf{x}$ or $\mathbf{y}$.
- The transpose of a matrix $\mathbf{A} \in \mathbb{R}^{n\times m}$ is indicated by $\mathbf{A}^T \in \mathbb{R}^{m\times n}$.
- The set of real numbers is denoted by $\mathbb{R}$, the set of strictly positive integer numbers is denoted by $\mathbb{N}$, and other sets are in calligraphic font, e.g., $\mathcal{U}$.
- For any positive integer n, we define the set $[n] := \{1, 2, \ldots, n\}$.
- p-norms (or ℓ_p-norms) of a vector $\mathbf{x} \in \mathbb{R}^d$ are defined and represented by

$$\|\mathbf{x}\|_p = \left(\sum_{i\in[d]} |x_i|^p \right)^{1/p}.$$

  The following are of particular interest:

  - The ℓ_1-*norm*: $\|\mathbf{x}\|_1 = \sum_{i\in[d]} |x_i|$
  - The ℓ_2-*norm* or Euclidean norm: $\|\mathbf{x}\|_2 = \sqrt{\sum_{i\in[d]} |x_i|^2} = \sqrt{\mathbf{x}^T\mathbf{x}}$
  - The ℓ_∞-*norm*: $\|\mathbf{x}\|_\infty = \max_{i\in[d]} |x_i|$

  where $|\cdot|$ is the absolute value function.
- For a $d \times d$ matrix $\mathbf{A}$, the ℓ_p induced matrix norm is defined by

$$\|\mathbf{A}\|_p = \sup\{\|\mathbf{A}\mathbf{x}\|_p \mid \|\mathbf{x}\|_p = 1\}.$$

- The **determinant** of a symmetric matrix $\mathbf{A}$, denoted by $\det(\mathbf{A})$, is the multiplication of its eigenvalues.

- Let $\mathbb{S}^d$ be the set of $d \times d$ symmetric matrices. We use $\mathbf{A} \succeq 0$ to denote that $\mathbf{A}$ is positive semidefinite, $\mathbb{S}^d_+$ to denote the set of symmetric positive semidefinite $d \times d$ matrices, $\mathbf{A} \succ 0$ to denote that $\mathbf{A}$ is positive definite, and $\mathbb{S}^d_{++}$ to denote the set of symmetric positive definite $d \times d$ matrices.
- For $\mathbf{T} \in \mathbb{R}^{d \times d}$, the **matrix transformation** associated to $\mathbf{T}$ is the transformation $\mathrm{T} : \mathbb{R}^d \to \mathbb{R}^d$ defined by $\mathrm{T}(\boldsymbol{x}) = \mathbf{T}\boldsymbol{x}$. This is the transformation that takes a vector $\boldsymbol{x} \in \mathbb{R}^d$ to the vector $\mathbf{T}\boldsymbol{x} \in \mathbb{R}^d$.
- For $\mathbf{t} \in \mathbb{R}^d$, the **vector translation** associated to $\mathbf{t}$ is the translation $\mathrm{T} : \mathbb{R}^d \to \mathbb{R}^d$ defined by $\mathrm{T}(\boldsymbol{x}) = \boldsymbol{x} + \mathbf{t}$. This is the translation that takes a vector $\boldsymbol{x} \in \mathbb{R}^d$ to the vector $\boldsymbol{x} + \mathbf{t} \in \mathbb{R}^d$.
- $\mathcal{S} = \{\boldsymbol{x} \in \mathbb{R}^d \mid \|\mathbf{T}\boldsymbol{x} + \mathbf{t}\|_p \leq 1\}$ is a **norm-based region**, which is a compact set, where $\mathbf{T} \in \mathbb{R}^{d \times d}$ is an invertible transformation matrix and $\mathbf{t} \in \mathbb{R}^d$ is a translation vector.
- $\mathcal{B}^p$ is the unit ℓ_p-norm ball which is defined as $\mathcal{B}^p = \{\boldsymbol{x} \mid \|\boldsymbol{x}\|_p \leq 1\}$
- The **floor function**, denoted as $\lfloor \cdot \rfloor$, takes as input a real number, x, and gives the greatest integer less than or equal to x. In other words, $\lfloor x \rfloor$ is the largest integer n such that $n \leq x$.
- The **modulo function**, denoted as $a \mod b$, calculates the non-negative integer remainder, $r \in \mathbb{N}$ when dividing $a \in \mathbb{N}$ by $b \in \mathbb{N} \neq 0$, where $0 \leq r < |b|$.

2.2 Volume of regions

Let us recall some definitions and theorems from the literature that are essential for this paper. These results clarify the relationship between transformation matrices and volume changes for a given region.

Definition 1 ([31]) Let $\mathcal{S} \subset \mathbb{R}^d$ be a region. Then, the **volume of** $\mathcal{S}$ is $\mathrm{vol}(\mathcal{S}) = \int_{\mathcal{S}} \mathrm{d}x$ (which is the Lebesgue measure of $\mathcal{S}$).

Theorem 1 *(Lemma 1 in [21]) Let* $\mathbf{T}$ *be a* $d \times d$ *matrix, and let* $\mathrm{T}(\boldsymbol{x})\colon \mathbb{R}^d \to \mathbb{R}^d$ *represent the associated matrix transformation:* $\mathrm{T}(\boldsymbol{x}) = \mathbf{T}\boldsymbol{x}$*. If* $\mathcal{S}$ *is a region in* $\mathbb{R}^d$*, then* $\mathrm{vol}(\mathrm{T}(\mathcal{S})) = |\det(\mathbf{T})| \cdot \mathrm{vol}(\mathcal{S})$

Proof See Lemma 1 from [21] for a special case where $\mathcal{S}$ is an ellipsoid, or Appendix (A.1) for a general one. □

Corollary 1 *Let* $\mathbf{T} \in \mathbb{R}^{d \times d}$ *be an invertible transformation matrix and* $\mathbf{t} \in \mathbb{R}^d$ *be a translation vector. Consider the region* $\mathcal{S} = \{\boldsymbol{x} \in \mathbb{R}^d \mid \|\mathbf{T}\boldsymbol{x} + \mathbf{t}\|_p \leq 1\}$*, where* $\mathbf{T} \in \mathbb{R}^{d \times d}$ *and* $\mathbf{t} \in \mathbb{R}^d$*. Then, the volume of* $\mathcal{S}$ *is given by*

$$\mathrm{vol}(\mathcal{S}) = \frac{1}{|\det(\mathbf{T})|} \cdot \frac{(2\Gamma(1 + \frac{1}{p}))^d}{\Gamma(1 + \frac{d}{p})}, \tag{3}$$

where

$$\Gamma(x) = \int_0^\infty s^{x-1} e^{-s} ds$$

is the gamma function.

Proof See Appendix (A.2). □

For more details regarding volumes of unit balls and calculation of gamma function, we refer to [32].

3 Minimum volume norm-based uncertainty set

The objective of our paper is to develop an uncertainty set based on volume-based clustering by utilizing available data. To achieve this, we propose a new mathematical formulation to find the best volume-based clustering of data, which considers various norm-based regions.

3.1 Minimum volume norm-based clustering

The volume-based clustering optimization problem is similar to an assignment problem with a non-linear objective function. It aims to find the optimal assignment of N different points, $\mathbf{a}^i$, $i = [N]$, to multiple clusters and allows some of them to be outliers. The objective function is to minimize the summation of the volumes of the norm-based regions. Similar to the formulation by [21, 27] for the ℓ_2-norm, we formulate the problem as

$$\inf_{\mathcal{S}_{k_p}, w^i, l^i_{k_p}} \quad \sum_{p\in\mathcal{P}} \sum_{k_p\in[K_p]} \mathrm{vol}(\mathcal{S}_{k_p}) \tag{4a}$$

$$\text{subject to} \quad \mathbf{a}^i \in \mathcal{S}_{k_p} \text{ if } l^i_{k_p} = 1, \qquad \forall i \in [N],\ k_p \in [K_p],\ p \in \mathcal{P}, \tag{4b}$$

$$w^i \le \sum_{p\in\mathcal{P}} \sum_{k_p\in[K_p]} l^i_{k_p} \le w^i \sum_{p\in\mathcal{P}} K_p, \qquad \forall i \in [N], \tag{4c}$$

$$\sum_{i\in[N]} w^i \ge (1-e)N, \tag{4d}$$

$$w^i \in \{0,1\}, \qquad \forall i \in [N], \tag{4e}$$

$$l^i_{k_p} \in \{0,1\}, \qquad \forall i \in [N],\ k_p \in [K_p],\ p \in \mathcal{P}, \tag{4f}$$

where:

- $\mathcal{P}$ is the desired set of norms;
- K_p is the number of ℓ_p-norm-based regions for all $p \in \mathcal{P}$;
- $\mathcal{S}_{k_p} := \{\boldsymbol{x} \in \mathbb{R}^d \mid \left\|\mathbf{T}_{k_p}\boldsymbol{x} + \mathbf{t}_{k_p}\right\|_p \le 1\}$ for all desired norm $p \in \mathcal{P}$ and all $k_p \in [K_p]$, where $\mathbf{T}_{k_p} \in \mathbb{R}^{d\times d}$ is a positive definite transformation matrix, $\mathbf{t}_{k_p} \in \mathbb{R}^d$ is the translation vector, and $p \in \mathcal{P}$ indicates the ℓ_p-norm used for regions $k_p \in [K_p]$;
- $l^i_{k_p}$, membership indicator, is a binary variable that equals 1 if the data point $\mathbf{a}^i$ is assigned to region $\mathcal{S}_{k_p}$, and 0 otherwise;

- w^i, assignment indicator, is a binary variable that equals 1 if the data point $\mathbf{a}^i$ is assigned to any cluster, and 0 if it is considered to be an outlier;
- and $e \in [0, 1)$ is the portion of the points that are considered outliers.

Problem (4) aims to minimize the volume of regions, where (4a) represents the objective function to achieve this. Constraint (4b) assigns the points to clusters, (4c) ensures that non-outlier data points are assigned to at least one cluster. Additionally, (4d) ensures that at most e portion of the data points are treated as outliers. To define the assignment indicators, w^i, and the membership indicators, $l^i_{k_p}$, we use binary values, which are enforced by (4e) and (4f).

3.2 Minimum volume norm-based clustering conic formulation

In this section, we provide a MICO problem to find the clusters with minimum volume. (4) is equivalent to the following optimization problem since (4b) can be reformulated as (5b) and (5c) to the following MINLO problem

$$\inf_{\mathcal{S}_{k_p}, l^i_{k_p}, \mathbf{T}_{k_p}, \mathbf{t}_{k_p}, w^i} \quad \sum_{p\in\mathcal{P}} \sum_{k_p\in[K_p]} \operatorname{vol}(\mathcal{S}_{k_p}) \tag{5a}$$

$$\text{subject to} \quad l^i_{k_p} \left\|\mathbf{T}_{k_p}\mathbf{a}^i + \mathbf{t}_{k_p}\right\|_p \le 1, \qquad \forall i \in [N],\ k_p \in [K_p],\ p \in \mathcal{P}, \tag{5b}$$

$$\mathbf{T}_{k_p} \succ 0, \qquad \forall k_p \in [K_p],\ p \in \mathcal{P}, \tag{5c}$$

$$(4c), \ldots, (4f).$$

Problem (5) is a mixed-integer non-convex optimization problem, which poses significant challenges in terms of finding an optimal solution. In the next theorem, we introduce a conic reformulation for this particular problem.

Theorem 2 *Let $\mathbf{a}^i \in \mathbb{R}^d$ be our data points, and $\mathcal{P}$ be the set of desired norms to consider. For any $p \in \mathcal{P}$, we set K_p to be the number of ℓ_p-norm-based regions. Then, the optimization problem (5) is equivalent to*

$$\inf_{\theta_{k_p}, \tau_{k_p}, l^i_{k_p}, \mathbf{T}_{k_p}, \mathbf{t}_{k_p}, w^i} \quad \sum_{p\in\mathcal{P}} \frac{(2\Gamma(1+\frac{1}{p}))^d}{\Gamma(1+\frac{d}{p})} \cdot \sum_{k_p\in[K_p]} \theta_{k_p} \tag{6a}$$

$$\text{subject to} \quad \tau_{k_p} \le \log \theta_{k_p}, \qquad \forall k_p \in [K_p],\ p \in \mathcal{P}, \tag{6b}$$

$$-\tau_{k_p} \le \log\det(\mathbf{T}_{k_p}), \qquad \forall k_p \in [K_p],\ p \in \mathcal{P}, \tag{6c}$$

$$l^i_{k_p} \left\|\mathbf{T}_{k_p}\mathbf{a}^i + \mathbf{t}_{k_p}\right\|_p \le 1, \qquad \forall i \in [N],\ k_p \in [K_p],\ p \in \mathcal{P}, \tag{6d}$$

$$w^i \le \sum_{p\in\mathcal{P}} \sum_{k_p\in[K_p]} l^i_{k_p} \le w^i \sum_{p\in\mathcal{P}} K_p, \qquad \forall i \in [N], \tag{6e}$$

$$\sum_{i\in[N]} w^i \geq (1-e)N, \tag{6f}$$

$$w^i \in \{0,1\}, \qquad \forall i \in [N], \tag{6g}$$

$$l^i_{k_p} \in \{0,1\}, \qquad \forall i \in [N],\ k_p \in [K_p],\ p \in \mathcal{P}. \tag{6h}$$

Proof Since $\mathcal{S}_{k_p} := \{\boldsymbol{x} \in \mathbb{R}^d \mid \left\|\mathbf{T}_{k_p}\boldsymbol{x} + \mathbf{t}_{k_p}\right\|_p \leq 1\}$ for all desired norm $p \in \mathcal{P}$ and all $k_p \in [K_p]$, using Corollay 1, we rewrite (5a) as

$$\begin{aligned}\sum_{p\in\mathcal{P}} \sum_{k_p\in[K_p]} \mathrm{vol}(\mathcal{S}_{k_p}) &= \sum_{p\in\mathcal{P}} \sum_{k_p\in[K_p]} \frac{1}{\det(\mathbf{T}_{k_p})} \cdot \mathrm{vol}(\mathcal{B}^p) \\ &= \sum_{p\in\mathcal{P}} \mathrm{vol}(\mathcal{B}^p) \sum_{k_p\in[K_p]} \frac{1}{\det(\mathbf{T}_{k_p})}.\end{aligned}$$

Let θ_{k_p} be a new variable for each region $k_p \in [K_p]$. Using epigraph reformulation, (5) reads as

$$\inf_{\theta_{k_p}, l^i_{k_p}, \mathbf{T}_{k_p}, \mathbf{t}_{k_p}, w^i} \quad \sum_{p\in\mathcal{P}} \mathrm{vol}(\mathcal{B}^p) \sum_{k_p\in[K_p]} \theta_{k_p} \tag{7a}$$

$$\text{subject to} \quad \frac{1}{\det(\mathbf{T}_{k_p})} \leq \theta_{k_p}, \qquad \forall k_p \in [K_p],\ p \in \mathcal{P}, \tag{7b}$$

$$l^i_{k_p} \left\|\mathbf{T}_{k_p}\mathbf{a}^i + \mathbf{t}_{k_p}\right\|_p \leq 1, \qquad \forall i \in [N],\ k_p \in [K_p],\ p \in \mathcal{P}, \tag{7c}$$

$$w^i \leq \sum_{p\in\mathcal{P}} \sum_{k_p\in[K_p]} l^i_{k_p} \leq w^i \sum_{p\in\mathcal{P}} K_p, \qquad \forall i \in [N], \tag{7d}$$

$$\sum_{i\in[N]} w^i \geq (1-e)N, \tag{7e}$$

$$w^i \in \{0,1\}, \qquad \forall i \in [N], \tag{7f}$$

$$l^i_{k_p} \in \{0,1\}, \qquad \forall i \in [N],\ \forall k_p \in [K_p],\ p \in \mathcal{P}. \tag{7g}$$

Since $\log(\cdot)$ is a strictly increasing function, we replace (7b) with

$$-\log\det(\mathbf{T}_{k_p}) \leq \log\theta_{k_p}, \quad \forall k_p \in [K_p],\ p \in \mathcal{P}. \tag{8}$$

Introducing new variable τ_{k_p}, (8) can be further reformulated as

$$\begin{aligned}\log\det(\mathbf{T}_{k_p}) \leq \tau_{k_p}, &\quad \forall k_p \in [K_p],\ p \in \mathcal{P}, \\ \tau_{k_p} \leq \log\theta_{k_p}, &\quad \forall k_p \in [K_p],\ p \in \mathcal{P}.\end{aligned}$$

□

Theorem 2 shows how an optimal clustering is obtained by reformulating the $\det(\cdot)$ function into a conic form. More explicitly, given $\mathcal{P} \subseteq \{1, 2, +\infty\}$, (6b) is equivalent to $(\theta_{k_p}, 1, \tau_{k_p})$ belonging to the *exponential* cone, and (6c) is equivalent to $\mathbf{T}_{k_p}$ belonging to the *logdet* cone.

Despite having the cones in (6), the problem is not yet a MICO problem due to the products of $l^i_{k_p}$ and $\left\|\mathbf{T}_{k_p}\mathbf{a}^i + \mathbf{t}_{k_p}\right\|_p$ in (6d). We know that if $\left\|\mathbf{T}_{k_p}\mathbf{a}^i + \mathbf{t}_{k_p}\right\|_p$ is bounded from above, then the product has a mixed integer convex quadratic reformulation, due to $l^i_{k_p}$ being binary. However, $\left\|\mathbf{T}_{k_p}\mathbf{a}^i + \mathbf{t}_{k_p}\right\|_p$ may not necessarily be bounded. To see this, let $l^i_{k_p}$ be 1 for only one i and k_p. Let $\mathbf{T}_{k_p}$ be a diagonal matrix where all the diagonal elements are χ, and $\mathbf{t}_{k_p} = -\chi\mathbf{a}^i$. Therefore, (6d) holds for this solution. Furthermore, increasing χ decreases the objective function, implying the norm is unbounded. Even though $\left\|\mathbf{T}_{k_p}\mathbf{a}^i + \mathbf{t}_{k_p}\right\|_p$ cannot be bounded from above, in the next theorem, we provide a necessary condition on an optimal clustering, helping us to bound $\left\|\mathbf{T}_{k_p}\mathbf{a}^i + \mathbf{t}_{k_p}\right\|_p$.

Theorem 3 *Let $0 < \omega < \frac{1}{2}$ be given. Also, let $\left(\theta^*_{k_p}, \tau^*_{k_p}, l^{i*}_{k_p}, \mathbf{T}^*_{k_p}, \mathbf{t}^*_{k_p}, w^{i*}\right)$ be an optimal solution of (6). For any $p \in \mathcal{P}$ and $k_p \in [K_p]$, if $\theta^*_{k_p} > 0$, then $\left\|{\mathbf{T}^*_{k_p}}^{-1}\right\|_p > \gamma$, where*

$$\gamma := \omega \cdot \min_{p\in\mathcal{P}} \min_{\substack{i,j\in[N]\\ i\neq j}} \left\|a^i - a^j\right\|_p .$$

Proof By contradiction, let us assume there exist $p \in \mathcal{P}$ and $k_p \in [K_p]$, where $\theta^*_{k_p} > 0$, and $\left\|{\mathbf{T}^*_{k_p}}^{-1}\right\|_p \leq \gamma$.

Let us denote by $\mathcal{B}^p_\kappa(\mathbf{a})$ the ball $\{\boldsymbol{x} \in \mathbb{R}^d \mid \|\boldsymbol{x} - \mathbf{a}\|_p \leq \kappa\}$. Given this notation, for any $\mathbf{a} \in \mathbb{R}^d$, the ball $\mathcal{B}^p_\gamma(\mathbf{a})$ contains at most one data point. To see this, let us assume the contrary and that there are two points i and j in the ball. So, we have

$$\|\mathbf{a}^i - \mathbf{a}\|_p \leq \gamma, \quad \|\mathbf{a}^j - \mathbf{a}\|_p \leq \gamma.$$

Therefore,

$$\|\mathbf{a}^i - \mathbf{a}^j\|_p \leq \|\mathbf{a}^i - \mathbf{a}\|_p + \|\mathbf{a} - \mathbf{a}^j\|_p \leq 2\gamma < \|\mathbf{a}^i - \mathbf{a}^j\|_p,$$

which is a contradiction, where (from left) the first inequality is the triangle inequality, the second one is because both i-th and j-th data points are in the ball, and the last one is because of the definition of γ.

Furthermore, we have

$$\sup_{\substack{\boldsymbol{x}\in\mathbb{R}^d\\ \|\mathbf{T}^*_{k_p}\boldsymbol{x}+\mathbf{t}_{k_p}\|_p\leq 1}} \left\|\boldsymbol{x} + {\mathbf{T}^*_{k_p}}^{-1}\mathbf{t}_{k_p}\right\|_p = \sup_{\|\boldsymbol{y}\|\leq 1} \left\|{\mathbf{T}^*_{k_p}}^{-1}\boldsymbol{y}\right\|_p \leq \left\|{\mathbf{T}^*_{k_p}}^{-1}\right\|_p \leq \gamma,$$

which shows that the cluster is contained in the ball $\mathcal{B}^p_\gamma(-{\mathbf{T}^*_{k_p}}^{-1}\mathbf{t}_{k_p})$. Therefore, the cluster contains at most one data point since the ball has this feature. If the cluster does not contain any data point, then clearly its volume, and hence $\theta^*_{k_p}$, is 0, which contradicts the assumption. So, let $\mathbf{a}^i$ be the point in the cluster. Then, setting $\mathbf{T}_{k_p}$ to be a diagonal matrix where all the diagonal elements are χ, and $\mathbf{t}_{k_p} = -\chi\mathbf{a}^i$, we have the same cluster of data, which is a singleton, with $\theta_p = \frac{1}{\chi^d}$. Since increasing χ does not change the cluster while reducing the objective function, we have $\theta^*_{k_p} = 0$. This contradicts the assumption, which concludes the proof. □

Theorem 3 asserts that for clusters with positive volume, we have a lower bound on the norm of the inverse of the transformation matrix. Since for any induced matrix norm $\|\cdot\|_p$, we have $\frac{1}{\|\mathbf{A}\|_p} \leq \|\mathbf{A}^{-1}\|_p$, we conclude that we are looking for transformation matrices $\mathbf{T}_{k_p}$, whose norm does not exceed $\frac{1}{\gamma}$. Moreover, we can assume, without loss of generality, that $\|\mathbf{a}^i\|_p < 1$, implying $\|\mathbf{t}_{k_p}\|_p \leq 1$. Therefore, we limit ourselves to the feasible points where $\left\|\mathbf{T}_{k_p}\mathbf{a}^i + \mathbf{t}_{k_p}\right\|_p$ is bounded from above by $\frac{1}{\gamma} + 1$. Therefore, (6) has an outer approximation (enlarging the feasible region) that is a MICO problem.

We emphasize that this new approximation and (6) significantly contribute to the existing literature. More specifically, the authors of [30] put forth a mixed integer nonconvex optimization problem that focuses solely on $p = 2$, while our MICO problem can deal with all computationally tractable norms. Furthermore, [27] offers a mixed integer semidefinite optimization problem that applies to $p = 2$, which is a looser approximation of (6) compared to our formulation.

Even though there are solvers capable of solving MICO problems, our formulations are computationally challenging as the number of cones depends on the number of clusters, and the number of constraints depends on the number of data points. Therefore, in the next section, we develop an algorithm to approximate and optimal solution timely.

4 Solution algorithm

In this section, we introduce a solution algorithm designed to approximate (6) efficiently. The closest algorithm in the literature compared to ours is the one proposed in [27], which is developed for $p = 2$ and is extremely sensitive to the initial solution. However, our algorithm is able to deal with a general vector norm and is not sensitive to the initial solution.

The problem's computational complexity arises from the combination of conic constraints and binary variables, making traditional solution techniques computationally inefficient. To overcome these difficulties and find efficient solutions for this optimization problem, we propose an iterative algorithm based on assignment approaches and duality theory. Firstly, we introduce an overview of the solution algorithm. Next, we introduce our reassignment algorithm, which removes and reassigns points to clusters in each iteration. Finally, we present the exploration algorithm, which adjusts a parameter that is used in the reassignment algorithm to refine the solutions and prevent the reassignment algorithm from getting stuck in local optimums.

4.1 An overview of our solution algorithm

We employ a bilevel algorithm, where the master problem is the assignment of points to clusters, and the sub-problems are identifying the regions that contain clustered points and have the minimum volume. We start with an initial solution. Then, we use an iterative local search procedure to refine the solution. This process yields new membership indicators, $l^i_{k_p}$, which are used to find new regions, enhancing the algorithm's efficiency and ability to find high-quality solutions.

Given fixed membership indicator values, $l^{\bar{i}}_{k_p}$, (6) decomposes into $K := \sum_{p\in\mathcal{P}} K_p$ independent minimum volume norm-based optimizations, each to obtain $\mathbf{T}_{k_p}$ and $\mathbf{t}_{k_p}$, $k_p \in [K_p]$ and $p \in \mathcal{P}$:

$$\inf_{\mathbf{T}_{k_p},\mathbf{t}_{k_p}} \quad -\log\det \mathbf{T}_{k_p} \tag{9a}$$

$$\text{subject to} \quad l^{\bar{i}}_{k_p} \left\|\mathbf{T}_{k_p}\mathbf{a}^i + \mathbf{t}_{k_p}\right\|_p \leq 1, \qquad \forall i \in [N]. \tag{9b}$$

Let us consider the Lagrangian dual problem for (9),

$$\max_{\boldsymbol{\lambda}_{k_p}\geq 0} \inf_{\mathbf{T}_{k_p},\mathbf{t}_{k_p}} \big(-\log\det \mathbf{T}_{k_p} + \sum_{i\in[N]} \lambda^i_{k_p}(1 - l^{\bar{i}}_{k_p} \left\|\mathbf{T}_{k_p}\mathbf{a}^i + \mathbf{t}_{k_p}\right\|_p)\big), \tag{10}$$

where $\boldsymbol{\lambda}_{k_p} \in \mathbb{R}^N$ is the vector of Lagrange multipliers associated with each constraint. Problem (9) contains a Slater's point. More explicitly, the Slater's point can be any region that strictly contains all the points with $l^{\bar{i}}_{k_p} = 1$. A simple way to construct such a region is to take the arithmetic mean of all the points as the center of the region and scale the identity matrix by a large enough constant to ensure all points are contained within the region. Thus, by Theorem 1 in [33], we know strong duality holds and (10) provides us with the same value as (9).

We denote the optimal solutions achieved from these subproblems by $\mathbf{T}^*_{k_p}$ and $\mathbf{t}^*_{k_p}$, $k_p \in [K_p]$ and $p \in \mathcal{P}$. Given these values, we define the master problem as

$$\inf_{r_{k_p},l^i_{k_p},w^i} \quad \sum_{p\in\mathcal{P}}\sum_{k=1}^{K_p} r_{k_p} \tag{11a}$$

$$\text{subject to} \quad \left\|\mathbf{T}^*_{k_p}\mathbf{a}^i + \mathbf{t}^*_{k_p}\right\|_p \cdot l^i_{k_p} \leq 1 + r_{k_p}, \qquad \forall i \in [N],\ k_p \in [K_p],\ p \in \mathcal{P}, \tag{11b}$$

$$w^i \leq \sum_{p\in\mathcal{P}}\sum_{k_p\in[K_p]} l^i_{k_p} \leq w^i \sum_{p\in\mathcal{P}} K_p, \qquad \forall i \in [N], \tag{11c}$$

$$\sum_{i\in[N]} w^i \geq (1-e)N, \tag{11d}$$

$$w^i \in \{0,1\}, \qquad \forall i \in [N], \tag{11e}$$

$$l^i_{k_p} \in \{0,1\}, \qquad \forall i \in [N],\ k_p \in [K_p],\ p \in \mathcal{P}, \tag{11f}$$

where the variable r_{k_p} is the penalty variable used to relax the constraint (11b). Problem (11) is an assignment problem with binary decision variables, $l^i_{k_p}$ and w^i and continuous variable r_{k_p}. For a given region k_p, we use r_{k_p} to shrink or enlarge the region. Therefore, (11) tries to find the best combinations of shrinking some regions, while enlarging others with the hope of finding better clusters.

4.2 Reassignment algorithm

To improve the clusters, we need to identify points to be reassigned to different clusters. In what follows, we explain the points that are interesting to consider. Next, we present the reassigning procedure of data points considering the interesting points.

Points in the boundary zone

We are interested in points in each cluster located on or near the boundary of the regions, which we call their boundary zones. We define the $\text{dist}(\mathbf{a}^i, k_p)$ function for each data point $\mathbf{a}^i$ and each region k_p, $k_p \in [K_p]$, $p \in \mathcal{P}$, as

$$\text{dist}(\mathbf{a}^i, k_p) = \left\| \mathbf{T}_{k_p} \mathbf{a}^i + \mathbf{t}_{k_p} \right\|_p .$$

Let $k^{*^i} = \text{argmin}_{k_p \in [K_p], p \in \mathcal{P}} \text{dist}(\mathbf{a}^i, k_p)$. We identify η data points that have the least values of $|\text{dist}\left(\mathbf{a}^i, k^{*^i}\right) - 1|$ and label them as points in the boundary zone. η is the desired number of points in the boundary zone, which is given by the exploration algorithm.

Points in the overlap

Given the solution to the master problem (11), there might be data points that belong to multiple regions in a given iteration. To enhance the solution of an iteration, we can exploit these overlapping points by incurring a small computation cost. The points that have $\text{dist}(\mathbf{a}^i, k_p) \leq 1$ from both regions, denoted as k_p^1 and k_p^2, represent the points in the overlap of those two regions.

Reassigning procedure

To update the regions, we remove the points in the boundary zone and the points in the overlap of regions from clusters and reassign them to find a better solution. In this procedure, we assume that η is given. Let $\mathbf{T}^*_{k_p}$, $\mathbf{t}^*_{k_p}$ and $l^{i*}_{k_p}$ be the transformation matrix, translation vectors, and membership indicators of the current optimal regions, respectively. We identify points in the boundary zones and the overlap. To solve the master problem (11), we do as follows:

1. We introduce $l'^i_{k_p}$ with the same values as $l^{i*}_{k_p}$ for $i \in [N]$, $k_p \in [K_p]$, $p \in \mathcal{P}$.
2. (Shrinking regions) We remove the points $\mathbf{a}^i$ within the boundary zones and the overlap zones by setting $l'^i_{k_p}$ to zero. We call $l'^i_{k_p}$ as shrunk membership indicators.
3. We solve sub-problems (9) for any k_p, given $l'^i_{k_p}$, $i \in [N]$, and call them shrunk regions with $\mathbf{T}'_{k_p}$ and $\mathbf{t}'_{k_p}$ as their shrunk transformation matrices and shrunk translation vectors, respectively.
4. We update the value of $l'^i_{k_p}$ for all data points and regions:

$$l'^i_{k_p} = \begin{cases} 1, & \text{if } \text{dist}(\mathbf{a}^i, k_p) \leq 1, \\ 0, & \text{if } \text{dist}(\mathbf{a}^i, k_p) > 1. \end{cases}$$

5. We identify points outside the shrunk regions as probable outliers, i.e. points with $\sum_{p\in\mathcal{P}}\sum_{k_p\in[K_p]} l'^i_{k_p} = 0$. We denote the portion of these probable outliers in the total data points as e'.
6. If $e' \leq e$, then $\mathbf{T}'_{k_p}$, $\mathbf{t}'_{k_p}$ and $l'^i_{k_p}$ are the refined values for regions. Otherwise, we will carry out the next step.
7. (Enlarging regions) To select the best region k'_p to enlarge, we calculate the increase in the total volume of regions in case of each region's enlargement. So, for given $k'_p \in [K_p]$ and $p \in \mathcal{P}$:
   (a) We introduce $l''^i_{k_p}$ which is equal with $l'^i_{k_p}$ for $i \in [N]$, $k_p \in [K_p]$, $p \in \mathcal{P}$.
   (b) We calculate $\text{dist}(\mathbf{a}^i, k'_p)$ for all the points and regions
   (c) We identify $\lfloor (e' - e)N/2 \rfloor$ number of datapoints that have minimum positive $\text{dist}(\mathbf{a}^i, k'_p) - 1$ value.
   (d) We set the $l''^i_{k_p}$ to 1 for the points identified from the last step and solve the associated subproblem for $l''^i_{k_p}$ to find the increment in volume of region in case k'_p is selected.
8. We select the cluster k'_p that causes the least increment in the total volume and enlarge it ($\mathbf{T}'_{k_p}$, $\mathbf{t}'_{k_p}$ and $l'^i_{k_p}$ are updated).
9. We go back to Step 6.

Using the reassignment algorithm, we remove and reassign points to the regions gradually to find an approximated solution. In this algorithm, η plays a crucial role. A small constant value for η can cause the algorithm to get stuck in local optimums, and a large value can lead to meaningless approximations. Therefore, we develop an exploration algorithm to adjust the value of η for the reassignment algorithm.

4.3 Exploration algorithm

As mentioned, η is a given parameter in the reassignment algorithm and plays a pivotal role in the final solution. In this section, we propose an algorithm to find a suitable η. The idea behind this algorithm is we start with a large number, and decrease the value after a fixed number of iterations.

Number of points in the boundary zone for each iteration

We require at least $d + 1$ points in $\mathbb{R}^d$ to be able to find minimum volume regions [34]. Since we need to solve subproblems for shrunk regions in each iteration of the reassignment algorithm, η has a maximum limit, which is denoted by $\eta_{\max}$. Values exceeding this threshold could make it impossible to construct one of the shrunk regions as they might have less than $d + 1$ data points.

Assume that the allowed total number of iterations for the algorithm is given by $\mathcal{T}$, and *iter* is the counter of them. We divide this total into phases, each with Ψ iterations. So, we have $\mathcal{T}/\Psi$ phases. Let ψ be the counter of the phases. We set the maximum number of points as

$$\eta^\psi := \frac{\eta_{\max}}{2^{\psi-1}}, \tag{12}$$

which decreases as the phase number ψ grows.

Exploration procedure

Having defined the parameters needed, we take the following steps as an exploration algorithm:

1. Let $\mathbf{T}^*_{k_p}$, $\mathbf{t}^*_{k_p}$, and Obj* be the transformation matrix, translation vectors, and the total volume of the regions, respectively, which are given from an initial solution.
2. If $iter = \mathcal{T}$, and ψ is the current phase, then we terminate the algorithm and return ($\mathbf{T}^*_{k_p}$, $\mathbf{t}^*_{k_p}$) as the solution. Otherwise, $iter \leftarrow iter + 1$, and we calculate η^ψ using (12).
3. We use the reassignment algorithm given $\mathbf{T}^*_{k_p}$, $\mathbf{t}^*_{k_p}$, and

$$\eta = \lfloor (1 - \frac{(iter - 1) \mod \Psi}{\Psi}) \times \eta^\psi \rfloor,$$

   to attain new $\mathbf{T}_{k_p}$, $\mathbf{t}_{k_p}$, and Obj.
4. If Obj $<$ Obj*, then we found a better solutions and we update $\mathbf{T}^*_{k_p}$, $\mathbf{t}^*_{k_p}$, and Obj*.
5. Go to Step 2.

In this algorithm, we adjust the number of boundary points in each iteration to prevent the algorithm get stuck in local optimums. In each phase, we gradually decrease η to fine-tune the solutions, but when a new phase starts, we again use a high value for η to explore more possible clustering. We find better solutions in each phase after doing the reassignment algorithm in each iteration, and at the last iteration of each phase, we find the best solution for that phase. Each phase takes the best solution so far as the initial solution and improves it.

5 RO reformulation

So far, we have explained how to construct the uncertainty set using our approach. In this section, we show how the RO problem (2) can be reformulated using our uncertainty set.

5.1 Our uncertainty set in RO

The uncertainty set derived from (6) is

$$\mathcal{U} = \bigcup_{p \in \mathcal{P}} \bigcup_{k_p \in K_p} \mathcal{U}_{k_p}, \tag{13}$$

where $\mathcal{U}_{k_p} = \{\boldsymbol{u} \mid \|\mathbf{T}^*_{k_p}\boldsymbol{u} + \mathbf{t}^*_{k_p}\|_p \leq 1\}$. Implementing (13) as the uncertainty set for (2), the robust counterpart reads as

$$\inf_{\boldsymbol{x} \in \mathbb{R}^n} \left\{ g(\boldsymbol{x}) \mid f_j(\boldsymbol{x}, u) \leq 0,\ \forall u \in \mathcal{U}_{k_p}, k_p \in [K_p], p \in \mathcal{P}, j \in [m] \right\}. \tag{14}$$

Substituting $\boldsymbol{z} = \mathbf{T}^*_{k_p}\boldsymbol{u} + \mathbf{t}^*_{k_p}$, for each $j \in [m]$, we can reformulate each constraint of (14) as

$$f_j(\boldsymbol{x}, -{\mathbf{T}^*_{k_p}}^{-1}\mathbf{t}^*_{k_p} + {\mathbf{T}^*_{k_p}}^{-1}\boldsymbol{z}) \leq 0,\ \forall \boldsymbol{z} \in \mathcal{Z}_{k_p}, k_p \in [K_p], p \in \mathcal{P}, \tag{15}$$

where $\mathcal{Z}_{k_p} = \{\boldsymbol{z} \mid \|\boldsymbol{z}\|_p \leq 1\}$. Using Theorem 2 in [35], each constraint in (15) reads as

$$\left(-{\mathbf{T}^*_{k_p}}^{-1}\mathbf{t}^*_{k_p}\right)^T \boldsymbol{\zeta}_{k_p} + \delta^*\left(({\mathbf{T}^*_{k_p}}^{-1})^T\boldsymbol{\zeta}_{k_p} \mid \mathcal{Z}_{k_p}\right) - f_{j*}(\boldsymbol{x}, \boldsymbol{\zeta}_{k_p}) \leq 0,\ \forall k_p \in [K_p], p \in \mathcal{P},$$

where $\boldsymbol{\zeta}_{k_p} \in \mathbb{R}^d$ for any $k_p \in [K_p]$ and $p \in \mathcal{P}$, the support function $\delta^*(\cdot)$ and the partial concave conjugate function $f_{j*}(\cdot)$ are as defined in [35]. Then, using Theorem 2 in [36], the constraint reads as

$$\left(-{\mathbf{T}^*_{k_p}}^{-1}\mathbf{t}^*_{k_p}\right)^T \boldsymbol{\zeta}_{k_p} + \left\|({\mathbf{T}^*_{k_p}}^{-1})^T\boldsymbol{\zeta}_{k_p}\right\|_{q_p} - f_{j*}(\boldsymbol{x}, \boldsymbol{\zeta}_{k_p}) \leq 0,\ \forall k_p \in [K_p], p \in \mathcal{P}, \tag{16}$$

where for each $p \in \mathcal{P}$, $\|.\|_{q_p}$ is the dual norm.

Example 1 (Linear constraint in the optimization variable) Consider a linear constraint as

$$\boldsymbol{u}^T\boldsymbol{x} - \beta \leq 0,\ \forall \boldsymbol{u} \in \mathcal{U} \tag{17}$$

where $\boldsymbol{u} \in \mathbb{R}^d$ is the coefficient vector of the variables, $\beta \in \mathbb{R}$ is the right-hand side parameter, and $\mathcal{U}$ is the uncertainty set defined in (13) with $\mathcal{P} = \{2\}$. Note that for $f(\boldsymbol{x}, \boldsymbol{u}) = \boldsymbol{u}^T\boldsymbol{x} - \beta$, we have

$$f_*(\boldsymbol{x}, \boldsymbol{u}) = \begin{cases} \beta, & \text{if } \boldsymbol{x} = \boldsymbol{u}, \\ -\infty, & \text{if } \boldsymbol{x} \neq \boldsymbol{u}. \end{cases}$$

So, using (16), the constraint (17) reads as

$$(-{\mathbf{T}^*_k}^{-1}\mathbf{t}^*_k)^T\boldsymbol{x} + \|{\mathbf{T}^*_k}^{-T}\boldsymbol{x}\|_2 \leq \beta,\ \forall k \in [K].$$

□

6 Numerical experiments

In this section, we evaluate the effectiveness and efficiency of our proposed approach. To make the research accessible and transparent for the broader scientific community, the code is available in a public repository [1].

In our experiments, we made use of Julia 1.10 and JuMP 1.20.0 to pass (9) to Mosek 10.0. To perform K-means clustering, we utilize the *Clustering.jl* package[2] from the JuliaStats library. For GMM clustering, we use the package *GaussianMixtures.jl*[3]. The experiment was performed on a MacBook Pro equipped with an Apple M1 Pro chip that has a total of 10 cores (8 performance and 2 efficiency) and 32 GB of memory.

[1] The code repository is available at https://github.com/alirezaSfid/MVNBC. It can be used for further experimentation and insights.
[2] https://github.com/JuliaStats/Clustering.jl
[3] https://github.com/davidavdav/GaussianMixtures.jl

We first tested if a regular solver could solve (6). The *Pajarito.jl* [37] package is a logical option, so we tested that. However, it turned out that even for small problem instances, the solver was not able to give a reasonable solution. For larger problem instances, it would not be able to give any solution. This is why we only report on the results from our own algorithm.

6.1 MVNBC as a clustering method

The purpose of this section is to visualize the performance of our algorithm to approximate optimal solutions regardless of the initial conditions. We compare our proposed clustering method with two traditional clustering methods, K-means and the Gaussian Mixture Model (GMM). Various datasets have been generated to illustrate the clustering results in a two-dimensional representation. Alongside this, measurements of time and total region volume have been recorded for comparison.

Data generation

We utilize the Gaussian mixture method [38] to generate the data, which effectively represents complex distributions. We generate three random positive definite covariance matrices for three Gaussian distributions. Then, for the center of each Gaussian distribution, we uniformly sample a point from $[-20, 20]^2$. Next, we generate 475 data points uniformly from these three distributions. Afterwards, we generate 25 random points sampled uniformly in the interval of minimum and maximum of the points generated in the previous step to represent some outliers. So, we have 5% outliers in total. Out of all the datasets that can be generated by applying the described method, we select the one that displays complex data patterns to emphasize the difference between the effectiveness of various methods in capturing data patterns.

Visual clustering results

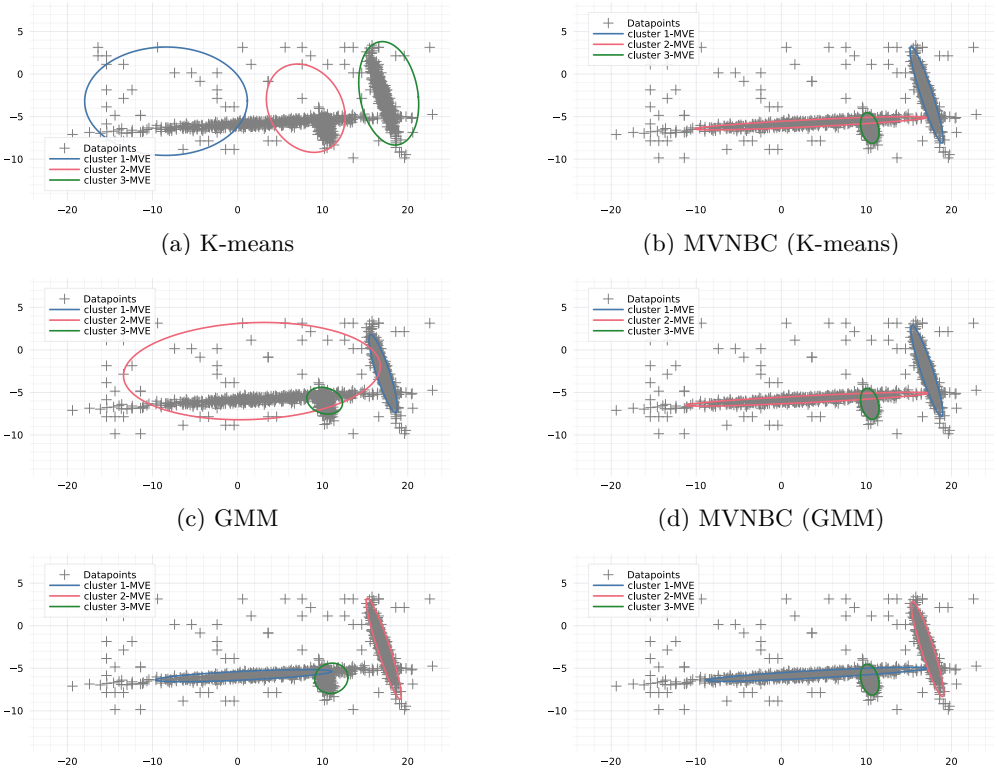
Given generated data sets, we use $K = 3$, the same number of clusters as the number of distributions, for K-means, GMM, and MVNBC approach. This is to evaluate the methods fairly when the optimal number of clusters is known. We conduct experiments with $p = 2$ and $p = \infty$. As K-means and GMM are not able to detect outliers, we consider outliers as the points which have maximum values of $\text{dist}(\mathbf{a}^i, k_p^*)$ (introduced in Section 4.2) and solve subproblems for each clustering membership indices (l_k^i). Three different initial solutions are used for MVNBC: regions based on clustering membership indices from K-means, regions based on clustering membership indices from GMM, and regions based on randomly generated membership indices. In the random case, we randomly set membership indices for three points from the data set to each cluster and then apply MVNBC for those initial solutions. To solve the subproblems with $p = 2$, we use *MinimumVolumeEllipsoids.jl*[4] package, and for the subproblems with $p = \infty$, we solve it using Mosek.

Figure 1 presents the results with $p = 2$. The regions resulting from the K-means algorithm, as shown in Figure 1a, are not effective in capturing data patterns. In

[4] https://github.com/FriesischScott/MinimumVolumeEllipsoids.jl

Figure 1b, we present the results for the MVNBC approach where the initial solution is the K-means, and we set the iteration number to 50 ($\mathcal{T} = 50$). For GMM, we do the same as K-means. The clustering using GMM is shown in Figure 1c. Although GMM can capture data patterns better than K-means, there is a huge cluster, which does not capture the data pattern well. Given this cluster as the initial solution, one can see the effectiveness of the MVNBC method for $\mathcal{T} = 50$ in Figure 1d.

Figure 1e demonstrates the resulting clustering from MVNBC when random initial solutions are provided and $\mathcal{T} = 50$. Figure 1f demonstrates the same for $\mathcal{T} = 100$. Even though in Figure 1eE there is still room for enhancing the clusters to capture data patterns efficiently, the resulting clusters are an effective representation of data. Then, applying more iterations would slightly improve the total volume of the clusters, as shown in Figure 1f.



(a) K-means (b) MVNBC (K-means)

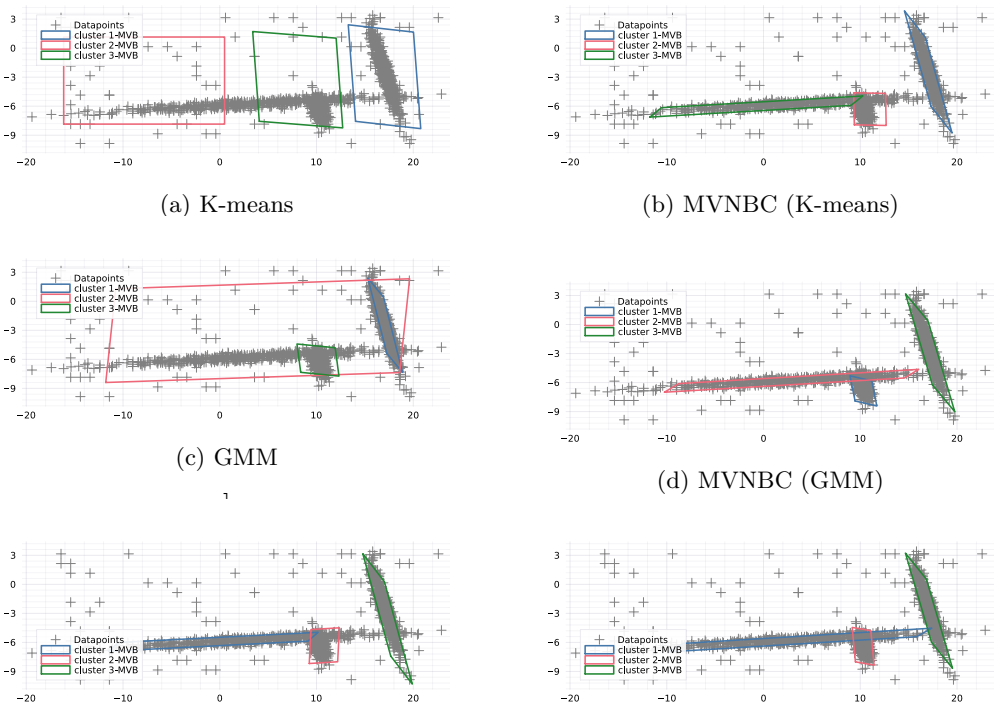
(c) GMM (d) MVNBC (GMM)

(e) MVNBC with 50 iters. (Random) (f) MVNBC with 100 iters. (Random)

Fig. 1: Clustering performance comparison ($p = 2$)

Similarly, Figure 2 shows the result of the same experiments for $p = \infty$. This experiment also provides consistent results, as for $p = 2$. In both cases with $p = 2$

and $p = \infty$, clearly, MVNBC outperforms other clustering methods in capturing data patterns effectively.



(a) K-means

(b) MVNBC (K-means)

(c) GMM

(d) MVNBC (GMM)

(e) MVNBC with 50 iters. (Random)

(f) MVNBC with 100 iters. (Random)

Fig. 2: Clustering performance comparison ($p = \infty$)

Algorithm effectiveness and computational efficiency

The trend of the objective value, representing the optimization progress, is illustrated in Figure 3 after a specified number of algorithm iterations for each experiment. When GMM clustering is employed as the initial solution, the optimization process demonstrates faster progress compared to other MVNBC cases. GMM, which fits multi-variate normal distributions on data, shares statistical equivalence with MVNBC when $p = 2$. In contrast, K-means clusters data solely based on their Euclidean distance to the centers without directly considering data patterns. As we expected, random initial solutions exhibit slower optimization progress compared to others. Nevertheless, the algorithm's effectiveness is observable regardless of the initial solution provided to MVNBC.

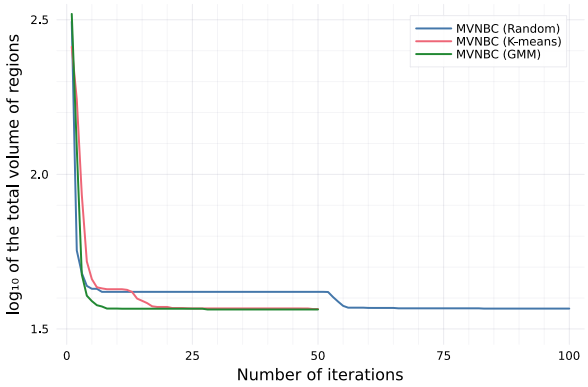


Fig. 3: Total volume of the best clusters for each iteration for $p = 2$

Table 1 provides an overview of the performance metrics of the solution for all methods, including the total volume of resulted regions and clustering times. It is worth noting that the calculation time for cases with $p = \infty$ is relatively high because optimization problems are solved exactly using the solver in each iteration. However, for cases with $p = 2$, the use of *MinimumVolumeEllipsoids.jl* package helps us find optimal solutions for subproblems more quickly. Although MVNBC takes longer to cluster data, it outperforms K-means and GMM by approximately 90% in terms of total cluster volumes.

Table 1: Performance Metrics for Clustering Methods

	Ellipsoid ($p = 2$)		**Diamond** ($p = \infty$)	
	Time (s)	**Total Volume**	**Time (s)**	**Total Volume**
K-means	0.2	331.1	0.2	295.9
GMM	2.8	291.8	0.4	313.5
MVNBC (K-means)	15.2	36.6	124.7	43.4
MVNBC (GMM)	53.5	**36.4**	197.1	39.5
MVNBC (Random) - 50iters.	12.2	41.7	134.8	40.1
MVNBC (Random) - 100iters.	20.3	36.8	259.0	**39.1**

6.2 RO with a linear constraint

In this section, we utilize the identical RO problem and datasets as those employed in [11]. By adopting the same problem formulation and dataset, we aim to directly compare our approach to the methods proposed in [11], which uses Neural Networks (NN), and is one of the best benchmarks in the field.

Problem description

Let us consider the following RO problem:

$$\max_{\boldsymbol{x}} \quad \sum_{i\in[d]} x_i \tag{18a}$$

$$\text{subject to} \quad \mathbf{c}^T\boldsymbol{x} \leq 1000, \qquad \forall \mathbf{c} \in \mathcal{U}, \tag{18b}$$

$$\boldsymbol{x} \in [-1,1]^d, \tag{18c}$$

where $\boldsymbol{x} \in \mathbb{R}^d$ is the vector of decision variables, $\mathbf{c} \in \mathbb{R}^d$ is the vector of uncertain parameters, and $\mathcal{U}$ is the uncertainty set.

Data generation

For this experiment, we use identical data sets as the one used in [11]. They consider three types of data sets. The first type of dataset is generated from a multivariate normal distribution, which we denote as *Gaussian*. The second type consists of two independent Gaussian distributions. For each sampled data point, they randomly choose with equal probability whether to use the first or the second distribution. We denote this data set as *Mixed Gaussian*. Finally, they sample the third set uniformly from a polyhedron constructed in the manner of budgeted uncertainty [39]:

$$\Xi = \left\{ \boldsymbol{\xi} \in \mathbb{R}^d : \xi_i = \underline{\xi_i} + \overline{\xi_i}\varrho_i, \sum_{i=1}^{d} \varrho_i \leq \rho, \boldsymbol{\varrho} \in [0,1]^d \right\},$$

where the lower and upper bounds $\underline{\xi_i}$ and $\overline{\xi_i}$, $i \in [d]$, are chosen randomly and $\rho = \frac{d}{2}$. We call the instances in this category as *Polyhedral*. To generate the data sets, they create 10 configurations for each data type (*Gaussian*, *Mixed Gaussian*, *Polyhedral*), resulting in a total of 30 data sets. For each of these 30 data sets, there is a training data set with 500 data points consisting of 5 percent noise sampled uniformly in $[0, 300]^d$, where $d = 20$ and a test dataset with 10,000 samples.

Results

For each data set, we construct an uncertainty set $\mathcal{U}$, using MVNBC with 1, 2, and 3 clusters when $p = 2$, respectively. We conducted the experiment only for $p = 2$, as we can use the *MinimumVolumeEllipsoids.jl* package for solving subproblems in this case. Solving subproblems for $p = 1$ or $p = +\infty$ is not computationally affordable for this experiment when we use Mosek to solve subproblems. For NN solutions, we use uncertainty sets constructed using NN, which is accessible in [11]'s repository. We construct these uncertainty sets using different rates of outliers, e. We consider cases where $1-e \in \{0.60, 0.65, 0.70, 0.75, 0.80, 0.85, 0.90\}$. We solved (18) for these different uncertainty sets.

We present the quality of the solutions in Table 2 for different e and different methods. In this table, higher values show a better performance. In most experiments,

MVNBC outperforms NN by about 10 percent, which is a notable difference. It is also worth mentioning that MVNBC with $e = 0.05$ shows a better performance than NN with $e = 0.40$, meaning the most conservative uncertainty set constructed using MVNBC outperforms the least conservative one constructed using NN.

Regarding feasibility, almost for all of the test out-of-sample scenarios, (18b) is not violated. It is also noteworthy that the results do not show any preference for the number of clusters we used. We expected, at least for *Mixed Gaussian* case, to see improvement between 1 and 2 clusters. This result is because of the structure of the problem and objective function. As the objective function in (18a) is linear on $\boldsymbol{x}$, the worst-case scenario happens at the boundary of the uncertainty set in 1 cluster case, coinciding with the boundary of one of the clusters in 2 clusters case. Consequently, despite the differing representations of data patterns, the objective values obtained under both uncertainty sets are identical. This equivalence underscores the robustness of uncertainty sets constructed using 2 clusters compared with 1 cluster, indicating that both provide the same conservative solutions.

Table 2: Obejctive values of (18) for different outlier detection rate (e)

		Objective value ($\sum_{i=1}^{d} x_i$)			
Type	$1-e$	**MVNBC-1**	**MVNBC-2**	**MVNBC-3**	**NN**
Gaussian	0.90	9.58	9.56	9.57	8.95
	0.85	9.60	9.60	9.58	8.97
	0.80	9.63	9.63	9.63	9.01
	0.75	9.64	9.64	9.64	9.05
	0.70	9.66	9.62	9.65	9.08
	0.65	9.69	9.65	9.64	9.11
	0.60	9.70	9.66	9.64	9.14
Mixed Gaussian	0.90	8.81	8.83	8.83	7.82
	0.85	8.84	8.85	8.86	7.88
	0.80	8.87	8.89	8.88	7.92
	0.75	8.91	8.90	8.89	7.98
	0.70	8.91	8.92	8.92	8.04
	0.65	8.92	8.96	8.93	8.09
	0.60	8.92	8.97	8.95	8.14
Polyhedral	0.90	8.56	8.56	8.57	8.13
	0.85	8.59	8.61	8.56	8.14
	0.80	8.65	8.63	8.62	8.14
	0.75	8.65	8.63	8.60	8.16
	0.70	8.68	8.63	8.64	8.17
	0.65	8.68	8.64	8.62	8.19
	0.60	8.71	8.69	8.64	8.20

For computational efficiency comparison, we provide the average training and solution times in seconds for different methods in Table 3. We utilized the pre-trained network in the repository of [11]. As we are using a different machine to conduct experiments, we do not have the exact training time information for the NN case. However, we can estimate the training time by considering the correlation between

training times and solution times reported in [11] and the solution time we achieved using our system. This allows us to extrapolate the training times for NN.

Training times for MVNBC-1 and NN require comparably little effort, with averages ranging from 12 to 21 seconds and 2 to 16 seconds, respectively. MVNBC-3 has the longest training times among all, as the size of the optimization problem is larger. There is a significant difference in solution times between NN and MVNBC. The MVNBC approach involves solving a second-order conic optimization problem (see Example 1), which can be solved quickly. On the other hand, the NN approach generates scenarios iteratively. It is clear that MVNBC is much faster than NN by two orders of magnitude. Furthermore, solving time is more important than training time since we only need to train and construct uncertainty sets once based on historical data.

Table 3: Average training time and solution time of problem (18)

Data type	**MVNBC-1**		**MVNBC-2**		**MVNBC-3**		**NN**	
	Training time (s)	Solution time (s)	Training time (s)	Solution time (s)	Training time (s)	Solution time (s)	Training time (s)	Solution time (s)
Gaussian	12.1	<0.1	88.0	<0.1	183.8	<0.1	2.8	102.3
Mix. Gaussian	10.4	<0.1	48.7	<0.1	122.5	<0.1	6.6	133.9
Polyhedral	20.3	<0.1	117.0	<0.1	180.2	<0.1	15.7	124.7

As shown in [11], NN shows a better performance compared to the Kernel method proposed in [8]. They also show that NN outperforms the Box uncertainty set. Therefore, our approach outperforms NN, Kernel, and Box methods.

7 Conclusion

In this paper, we have proposed a norm-based clustering method, called Minimum Volume Norm-Based Clustering (MVNBC). This method aims to minimize the volumes of regions for a desired set of vector norms. Each of these regions contains a cluster of data points. MVNBC can locate these regions in order to capture data patterns effectively and identify outliers at a specified rate. We have developed a Mixed Integer Conic Optimization (MICO) problem for an MVNBC problem. The MICO problem includes a quadratic cone for each data point and region, as well as an exponential cone and a logdet cone for each region. Additionally, we have developed an efficient iterative approximation algorithm, which iteratively reassigns points to new regions to reduce the regions' total volume.

Moreover, we utilized the regions generated from MVNBC as a new data-driven uncertainty set in Robust Optimization (RO). We formulated the robust counterpart for a general convex optimization problem with the presented uncertainty set, which is as tractable as the original problem in many cases. Finally, we have conducted two numerical experiments to showcase the practical significance of our proposed approach. In the first experiment, we evaluated MVNBC as a clustering method to

identify patterns in data sets. Our algorithm demonstrated superior performance when compared to K-means and GMM in finding regions with a minimum total volume containing clusters. In the second experiment, we used our uncertainty set for a typical RO problem and compared its performance with one of the best benchmarks in the field. MVNBC not only improved the objective value by 10%, but also allowed RO problems to be solved significantly faster compared to the benchmark from the literature.

References

[1] Ben-Tal, A., Nemirovski, A.: Robust optimization–methodology and applications. Mathematical Programming **92**, 453–480 (2002) https://doi.org/10.1007/s101070100286

[2] Ben-Tal, A., Nemirovski, A.: Robust solutions of linear programming problems contaminated with uncertain data. Mathematical Programming **88**, 411–424 (2000) https://doi.org/10.1007/PL00011380

[3] Bertsimas, D., Brown, D.B., Caramanis, C.: Theory and applications of robust optimization. SIAM Review **53**(3), 464–501 (2011) https://doi.org/10.1137/080734510

[4] Baron, O., Milner, J., Naseraldin, H.: Facility location: A robust optimization approach. Production and Operations Management **20**(5), 772–785 (2011) https://doi.org/10.1111/j.1937-5956.2010.01194.x

[5] Kang, Z., Marandi, A., Basten, R.J.I., Kok, T.: Robust spare parts inventory management (2023). Available at SSRN: https://ssrn.com/abstract=4553430 or http://dx.doi.org/10.2139/ssrn.4553430

[6] Bertsimas, D., Thiele, A.: Chapter 4. In: Johnson, M.P., Norman, B., Secomandi, N. (eds.) Robust and Data-Driven Optimization: Modern Decision Making Under Uncertainty. INFORMS TutORials in Operations Research, pp. 95–122. INFORMS, Catonsville, MD (2006). https://doi.org/10.1287/educ.1063.0022 . https://pubsonline.informs.org/doi/abs/10.1287/educ.1063.0022

[7] Bertsimas, D., Gupta, V., Kallus, N.: Data-driven robust optimization. Mathematical Programming **167**, 235–292 (2018) https://doi.org/10.1007/s10107-017-1125-8

[8] Shang, C., Huang, X., You, F.: Data-driven robust optimization based on kernel learning. Computers & Chemical Engineering **106**, 464–479 (2017) https://doi.org/10.1016/j.compchemeng.2017.07.004

[9] Ning, C., You, F.: Data-driven decision making under uncertainty integrating robust optimization with principal component analysis and kernel smoothing methods. Computers & Chemical Engineering **112**, 190–210 (2018) https://doi.

org/10.1016/j.compchemeng.2018.02.007

[10] Zhang, S., Jia, R., He, D., Chu, F.: Data-driven robust optimization based on principle component analysis and cutting plane methods. Industrial & Engineering Chemistry Research **61**(5), 2167–2182 (2022) https://doi.org/10.1021/acs.iecr.1c03886

[11] Goerigk, M., Kurtz, J.: Data-driven robust optimization using deep neural networks. Computers & Operations Research **151**, 106087 (2023) https://doi.org/10.1016/j.cor.2022.106087

[12] Moshkovitz, M., Dasgupta, S., Rashtchian, C., Frost, N.: Explainable k-means and k-medians clustering. In: III, H.D., Singh, A. (eds.) Proceedings of the 37th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 119, pp. 7055–7065. PMLR, Virtual Conference (2020). https://proceedings.mlr.press/v119/moshkovitz20a.html

[13] Zhang, Y., Feng, Y., Rong, G.: New robust optimization approach induced by flexible uncertainty set: Optimization under continuous uncertainty. Industrial & Engineering Chemistry Research **56**(1), 270–287 (2017) https://doi.org/10.1021/acs.iecr.6b02989

[14] Asgari, S.D., Mohammadi, E., Makui, A., Jafari, M.: Data-driven robust optimization based on position-regulated support vector clustering. Journal of Computational Science **76**, 102210 (2024) https://doi.org/10.1016/j.jocs.2024.102210

[15] Saxena, A., Prasad, M., Gupta, A., Bharill, N., Patel, O.P., Tiwari, A., Er, M.J., Ding, W., Lin, C.-T.: A review of clustering techniques and developments. Neurocomputing **267**, 664–681 (2017) https://doi.org/10.1016/j.neucom.2017.06.053

[16] Han, J., Kamber, M., Pei, J.: 8 - classification: Basic concepts. In: Han, J., Kamber, M., Pei, J. (eds.) Data Mining (Third Edition), Third edition edn. The Morgan Kaufmann Series in Data Management Systems, pp. 327–391. Morgan Kaufmann, Boston (2012). https://doi.org/10.1016/B978-0-12-381479-1.00008-3

[17] Nielsen, F.: Hierarchical Clustering, pp. 195–211. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-21903-5_8

[18] Campello, R.J.G.B., Kröger, P., Sander, J., Zimek, A.: Density-based clustering. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **10**(2), 1343 (2020) https://doi.org/10.1002/widm.1343

[19] Rosen, J.B.: Pattern separation by convex programming. Journal of Mathematical Analysis and Applications **10**(1), 123–134 (1965) https://doi.org/10.1016/

0022-247X(65)90150-2

[20] Barnes, E.R.: An algorithm for separating patterns by ellipsoids. IBM Journal of Research and Development **26**(6), 759–764 (1982) https://doi.org/10.1147/rd.266.0759

[21] Kumar, M., Orlin, J.B.: Scale-invariant clustering with minimum volume ellipsoids. Computers & Operations Research **35**(4), 1017–1029 (2008) https://doi.org/10.1016/j.cor.2006.07.001

[22] Henk, M.: Löwner-john ellipsoids. Documenta Mathematica **Extra Volume: Optimization Stories**, 95–106 (2012)

[23] Kumar, P., Yildirim, E.A.: Minimum-volume enclosing ellipsoids and core sets. Journal of Optimization Theory and Applications **126**(1), 1–21 (2005) https://doi.org/10.1007/s10957-005-2653-6

[24] Woodruff, D.L., Rocke, D.M.: Heuristic search algorithms for the minimum volume ellipsoid. Journal of Computational and Graphical Statistics **2**(1), 69–95 (1993) https://doi.org/10.1080/10618600.1993.10474600

[25] Khachiyan, L.G.: Rounding of polytopes in the real number model of computation. Mathematics of Operations Research **21**(2), 307–320 (1996) https://doi.org/10.1287/moor.21.2.307

[26] Candela, J.A.: Exact iterative computation of the multivariate minimum volume ellipsoid estimator with a branch and bound algorithm. In: Prat, A. (ed.) COMPSTAT, pp. 175–180. Physica-Verlag HD, Heidelberg (1996). https://link.springer.com/chapter/10.1007/978-3-642-46992-3_16

[27] Martínez-Rego, D., Castillo, E., Fontenla-Romero, O., Alonso-Betanzos, A.: A minimum volume covering approach with a set of ellipsoids. IEEE Transactions on Pattern Analysis and Machine Intelligence **35**(12), 2997–3009 (2013) https://doi.org/10.1109/TPAMI.2013.94

[28] Van Aelst, S., Rousseeuw, P.: Minimum volume ellipsoid. Wiley Interdisciplinary Reviews: Computational Statistics **1**(1), 71–82 (2009) https://doi.org/10.1002/wics.19

[29] Ahipaşaoğlu, S.D.: Fast algorithms for the minimum volume estimator. Journal of Global Optimization **62**(2), 351–370 (2015) https://doi.org/10.1007/s10898-014-0233-8

[30] Shioda, R., Tunçel, L.: Clustering via minimum volume ellipsoids. Computational Optimization and Applications **37**(3), 247–295 (2007) https://doi.org/10.1007/s10589-007-9024-1

[31] Bemporad, A., Filippi, C., Torrisi, F.D.: Inner and outer approximations of polytopes using boxes. Computational Geometry **27**(2), 151–178 (2004) https://doi.org/10.1016/S0925-7721(03)00048-8

[32] Wang, X.: Volumes of generalized unit balls. Mathematics Magazine **78**(5), 390–395 (2005) https://doi.org/10.2307/30044198

[33] Slater, M.: Lagrange multipliers revisited. Cowles Commission Discussion Papers: Mathematics **403**, 1–13 (1950)

[34] John, F.: In: Giorgi, G., Kjeldsen, T.H. (eds.) Extremum Problems with Inequalities as Subsidiary Conditions, pp. 197–215. Springer, Basel (2014). https://doi.org/10.1007/978-3-0348-0439-4_9

[35] Ben-Tal, A., Den Hertog, D., Vial, J.-P.: Deriving robust counterparts of nonlinear uncertain inequalities. Mathematical Programming **149**(1), 265–299 (2015) https://doi.org/10.1007/s10107-014-0750-8

[36] Bertsimas, D., Pachamanova, D., Sim, M.: Robust linear optimization under general norms. Operations Research Letters **32**(6), 510–516 (2004) https://doi.org/10.1016/j.orl.2003.12.007

[37] Coey, C., Lubin, M., Vielma, J.P.: Outer approximation with conic certificates for mixed-integer convex problems. Mathematical Programming Computation **12**(2), 249–293 (2020)

[38] Reynolds, D.: In: Li, S.Z., Jain, A.K. (eds.) Gaussian Mixture Models, pp. 827–832. Springer, Boston, MA (2015). https://doi.org/10.1007/978-1-4899-7488-4_196

[39] Bertsimas, D., Sim, M.: The price of robustness. Operations Research **52**(1), 35–53 (2004) https://doi.org/10.1287/opre.1030.0065

[40] Margalit, D., Rabinoff, J., Rolen, L.: Interactive Linear Algebra. Schoo of Mathematics, Georgia Institute of Technology, Atlanta, GA, USA (2019). https://textbooks.math.gatech.edu/ila/

[41] Sun, P., Freund, R.M.: Computation of minimum-volume covering ellipsoids. Operations Research **52**(5), 690–706 (2004) https://doi.org/10.1287/opre.1040.0115

Appendix A Mathematical Supplements

A.1 Proof of Theorem 1

A brief proof of this theorem is provided in [40] Section 4.3.3. Here is a more elaborate proof for the theorem. Firstly, we need some definitions in order to prove the theorem.

Definition A.1 (See Definition 3.1.22 in [40]) A **transformation** from $\mathbb{R}^n$ to $\mathbb{R}^m$ is a rule, T, that maps each vector, $\boldsymbol{x}$ in $\mathbb{R}^n$, to a vector, $\mathrm{T}(\boldsymbol{x})$ in $\mathbb{R}^m$. The notation $\mathrm{T}(\boldsymbol{x})\colon \mathbb{R}^n \to \mathbb{R}^m$ means "T is a transformation from $\mathbb{R}^n$ to $\mathbb{R}^m$."

Definition A.2 (See Definition 3.3.1 in [40]) A **linear transformation** is a transformation $\mathrm{T}(\boldsymbol{x})\colon \mathbb{R}^n \to \mathbb{R}^m$ satisfying

$$\mathrm{T}(\boldsymbol{u}+\boldsymbol{v}) = \mathrm{T}(\boldsymbol{u}) + \mathrm{T}(\boldsymbol{v}), \tag{A1}$$

$$\mathrm{T}(c\boldsymbol{u}) = c\mathrm{T}(\boldsymbol{u}) \tag{A2}$$

for all vectors, $\boldsymbol{u}$ and $\boldsymbol{v}$ in $\mathbb{R}^n$, and all scalars c.

Definition A.3 (See Definition 3.3.13 in [40]) The **standard coordinate vectors** in $\mathbb{R}^n$ are the n vectors

$$e_1 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}, \quad e_2 = \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \\ 0 \end{pmatrix}, \quad \ldots, \quad e_n = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}. \tag{A3}$$

The i-th entry of e_i equals 1, and the other entries are zero.

Theorem A.1 (See Section 3.3.3 and page 148 in [40]) *Let* $\mathrm{T}\colon \mathbb{R}^n \to \mathbb{R}^m$ *be a linear transformation. Let* $\mathbf{T}$ *be the* $m \times n$ *matrix*

$$\mathbf{T} = \begin{pmatrix} | & | & & | \\ \mathrm{T}(e_1) & \mathrm{T}(e_2) & \ldots & \mathrm{T}(e_n) \\ | & | & & | \end{pmatrix} \tag{A4}$$

Then T *is the* ***matrix transformation associated with*** $\mathbf{T}$*: that is,* $\mathrm{T}(\boldsymbol{x}) = \mathbf{T}\boldsymbol{x}$.

Proof See Section 3.3.3 and page 149 in [40] □

Definition A.4 (See Definition 4.3.1 in [40]) The **paralellepiped** determined by n vectors $\boldsymbol{v}_1$, $\boldsymbol{v}_2$, $\mathbf{t}$, $\boldsymbol{v}_n$ in $\mathbb{R}^n$ is the subset

$$P = \{a_1\boldsymbol{v}_1 + a_2\boldsymbol{v}_2 + \cdots + a_n\boldsymbol{v}_n | 0 \le a_1, a_2, \ldots, a_n \le 1\} \tag{A5}$$

Theorem A.2 (See Theorem 4.3.6 in [40]) *Let* $\mathbf{v}_1$, $\mathbf{v}_2$, ..., $\mathbf{v}_n$ *be vectors in* $\mathbb{R}^n$, *let* P *be the parallelepiped determined by these vectors, and let* $\mathbf{T}$ *be the matrix with rows* $\mathbf{v}_1$, $\mathbf{v}_2$, ..., $\mathbf{v}_n$. *Then the absolute value of determinant of* $\mathbf{T}$ *is the volume of* P*:*

$$|\det(\mathbf{T})| = \mathrm{vol}(P) \tag{A6}$$

Proof (See Section 4.3.2 from [40]) □

proof of Theorem 1 Let $\mathcal{C}$ be the unit cube in d dimesnion, $\mathbf{v}_1$, $\mathbf{v}_2$, …, $\mathbf{v}_d$ be the columns of $\mathbf{T}$, and P be the paralellepiped determined by these vectors, i.e. $\mathrm{T}(\mathcal{C}) = P$ and $\mathrm{vol}(\mathrm{T}(\mathcal{C})) = |\det(\mathbf{T})|$ (from Theorem A.2). Let $\varepsilon > 0$ and $\varepsilon\mathcal{C}$ be the cube with side lengths ε, i.e., the parallelepiped determined by the vectors $\varepsilon e_1, \varepsilon e_2, \ldots, \varepsilon e_d$ and εP defines similarly. We have

$$\mathrm{vol}(\varepsilon P) = \varepsilon^d \mathrm{vol}(P) = \varepsilon^d |\det(\mathbf{T})| \tag{A7}$$

where $\mathrm{vol}(P) = \mathrm{vol}(\mathrm{T}(\mathcal{C}))$. The volume of $\varepsilon\mathcal{C}$ is ε^d, because we scaled each of the d standard vectors by a factor ε. We obtain that the volume of εP equals to $\varepsilon^d |\det(\mathbf{T})|$.
Moreover, by Definition A.2, for any $\boldsymbol{x} \in \mathbb{R}^d$:

$$\mathrm{T}(\boldsymbol{x} + \varepsilon\mathcal{C}) = \mathrm{T}(\boldsymbol{x}) + \mathrm{T}(\varepsilon\mathcal{C}) = \mathrm{T}(\boldsymbol{x}) + \varepsilon P \tag{A8}$$

Since a translation does not change the volumes, T scales the volume of a translate of $\varepsilon\mathcal{C}$ by $|\det(\mathbf{T})|$ (from Theorem A.2), i.e.

$$\mathrm{vol}(\mathrm{T}(\boldsymbol{x} + \varepsilon\mathcal{C})) = \mathrm{vol}(\mathrm{T}(\varepsilon\mathcal{C})) = \varepsilon^d |\det(\mathbf{T})| \tag{A9}$$

Furthermore, we know

$$\mathrm{vol}(\mathcal{S}) = \oint_{\mathcal{S}} \mathrm{d}s = \int \cdots \int_{x \in \mathcal{S}} \mathrm{d}x \tag{A10}$$

where $\mathcal{S}$ is a region in $\mathbb{R}^d$ and

$$\mathrm{d}s = \mathrm{d}e_1 \cdot \mathrm{d}e_2 \cdot \cdots \cdot \mathrm{d}e_d = \lim_{\varepsilon \to 0} \varepsilon e_1 \cdot \varepsilon e_2 \cdot \cdots \cdot \varepsilon e_d = e_1 \cdot e_2 \cdot \cdots \cdot e_d \lim_{\varepsilon \to 0} \varepsilon^d. \tag{A11}$$

The translation of this subset ($\mathcal{S}$) by T is another subset ($\mathcal{V}$), which consists of translated $\mathrm{d}s$. The volume of this subset ($\mathcal{V}$) is calculated by:

$$\mathrm{vol}(\mathcal{V}) = \oint_{\mathcal{V}} \mathrm{d}v = \int \cdots \int_{x \in \mathcal{V}} \mathrm{d}x \tag{A12}$$

where

$$\begin{aligned}
\mathrm{d}v &= \mathrm{T}(\mathrm{d}s) && \text{(A13)}\\
&= \mathrm{T}(e_1 \cdot e_2 \cdot \cdots \cdot e_d \lim_{\varepsilon \to 0} \varepsilon^d) && \text{(A14)}\\
&= \lim_{\varepsilon \to 0} \varepsilon^d \times \mathrm{T}(e_1 \cdot e_2 \cdot \cdots \cdot e_d) && \text{(A15)}\\
&= \lim_{\varepsilon \to 0} \varepsilon^d \times |\det(\mathbf{T})| \times e_1 \cdot e_2 \cdot \cdots \cdot e_d && \text{(A16)}\\
&= |\det(\mathbf{T})| \times \mathrm{d}s. && \text{(A17)}
\end{aligned}$$

So,

$$\begin{aligned}
\mathrm{vol}(\mathcal{V}) = \oint_{\mathcal{V}} \mathrm{d}v = \oint_{\mathcal{S}} \mathrm{T}(\mathrm{d}s) = \oint_{\mathcal{S}} |\det(\mathbf{T})| \cdot \mathrm{d}s = |\det(\mathbf{T})| \cdot \oint_{\mathcal{S}} \mathrm{d}s && \text{(A18)}\\
= |\det(\mathbf{T})| \cdot \mathrm{vol}(\mathcal{S}) && \text{(A19)}
\end{aligned}$$

□

A.2 Proof of Corollary 1

Proof Given $\boldsymbol{x} \in \mathbb{R}^d$, let us define $\boldsymbol{y} = \mathbf{T}\boldsymbol{x} + \mathbf{t}$. Since $\mathbf{T}$ is invertible, we have

$$\boldsymbol{x} = \mathbf{T}^{-1}(\boldsymbol{y} - \mathbf{t}) \tag{A20}$$

therefore, $\mathcal{S}$ can be represented as

$$\mathcal{S} = \{\mathbf{T}^{-1}(\boldsymbol{y} - \mathbf{t}) \mid \|\boldsymbol{y}\|_p \leq 1\} \tag{A21}$$

$$= \mathrm{T}(\mathcal{B}^p), \tag{A22}$$

where

$$\mathrm{T}(\boldsymbol{x}) = \mathbf{T}^{-1}\boldsymbol{x} - \mathbf{T}^{-1}\mathbf{t} \tag{A23}$$

As translation ($\mathbf{T}^{-1}\mathbf{t}$) does not change the volume, using Theorem 1, we have

$$\mathrm{vol}(\mathcal{S}) = |\det(\mathbf{T}^{-1})| \cdot \mathrm{vol}(\mathcal{B}^p) = \frac{1}{|\det(\mathbf{T})|} \cdot \mathrm{vol}(\mathcal{B}^p), \tag{A24}$$

where $\mathcal{B}^p$ is the unit ball of ℓ_p-norm in $\mathbb{R}^d$, i.e., $\mathcal{B}^p = \{\boldsymbol{x} \in \mathbb{R}^d \mid \|\boldsymbol{x}\|_p \leq 1\}$. So, the volume of the region $\mathcal{S}$ is inversely proportional to the determinant of $\mathbf{T}$. The fact that the volume of the unit ball $\mathcal{B}^p$ in $\mathbb{R}^d$ is equal to (See e.g., [32])

$$\mathrm{vol}(\mathcal{B}^p) = \frac{(2\Gamma(1+\frac{1}{p}))^d}{\Gamma(1+\frac{d}{p})}, \tag{A25}$$

concludes the proof. □

Appendix B Limitations and Critique of [30]

[30] introduces a Minimum Volume Ellipsoid Clustering (MVEC) and formulates it as a MICO problem. The challenge lies in distributing N points in $\mathbb{R}^d$ among K clusters, each defined by the total volume of its covering ellipsoid. Essentially, the objective is to arrange the data points to minimize the sum of the volumes of the Minimum Volume Ellipsoid (MVE) associated with each cluster. The MVE serves as a metric to characterize the region of a cluster. The primary aim is to reduce the volume of MVE for each cluster while also guaranteeing that all points are contained within the same cluster:

$$\min_{\mathcal{C}_k} \quad -\sum_{k=1}^{K} \ln\det(\boldsymbol{M}_k) \tag{B26a}$$

$$\text{subject to} \quad (\boldsymbol{M}_k\mathbf{a}^i - \boldsymbol{z}_k)^T(\boldsymbol{M}_k\mathbf{a}^i - \boldsymbol{z}_k) \leq 1, \quad \forall i \in \mathcal{C}_k,\ k = [K], \tag{B26b}$$

$$\bigcup_k C_k = [N], \tag{B26c}$$

$$\boldsymbol{M}_k \succeq 0. \tag{B26d}$$

where $\mathcal{C}_k$ is the allocation of points to the k-th cluster, $[K]$,
$\boldsymbol{M}_k \in \mathbb{R}^{d\times d}$ and $\boldsymbol{z}_k \in \mathbb{R}^d$ are the parameters of the k-th ellipsoid.

In this section, we discuss the inadequacies of the formulation (B26) with respect to its objective function and two limitations. Specifically, we identified that the objective function is erroneous and further noted two shortcomings that need to be addressed.

B.1 Main inadequacy: inaccurate objective function

The primary adequacy pertains to the utilization of a summation of logarithmic terms in the objective function (B26a) instead of a summation of the volume of ellipsoids. The problem with this approach is that the objective function in (B26) is not the summation of volumes, which is actually $\sum_k \frac{1}{\det \boldsymbol{M}_k}$, but rather the summation of the logarithmic terms $-\sum_k \ln\det \boldsymbol{M}_k$. Although the logarithmic function preserves the ordering of numbers, the use of logarithmic terms in the MVE formulation cannot be justified. This is due to the fact that the logarithmic function is not linear, and the summation of logarithms does not preserve the ordering of numbers. To illustrate the issue with their objective function, we present a simple numerical example.

Consider four ellipsoids in 2D, where each ellipsoid is represented by the equation:

$$\mathcal{E}_k = \{\boldsymbol{x} \in \mathbb{R}^2 \mid \|\mathbf{T}_k\boldsymbol{x} + \mathbf{t}_k\|_2 \leq 1\}, \quad \forall k \in \{1,2,3,4\},$$

where $\det(\mathbf{T}_k)$ represents the determinant of matrix $\mathbf{T}_k$, for $k \in \{1,2,3,4\}$, such that:

$$\begin{aligned}
\det(\mathbf{T}_1) &= \frac{1}{2}, \\
\det(\mathbf{T}_2) &= \frac{1}{2}, \\
\det(\mathbf{T}_3) &= \frac{1}{3}, \\
\det(\mathbf{T}_4) &= \frac{2}{3}.
\end{aligned}$$

Using equation (3), we have the following volumes for the four ellipsoids:

$$\begin{aligned}
\mathrm{vol}(\mathcal{E}_1) &= \frac{\pi}{2}, \\
\mathrm{vol}(\mathcal{E}_2) &= \frac{\pi}{2}, \\
\mathrm{vol}(\mathcal{E}_3) &= \frac{\pi}{3}, \\
\mathrm{vol}(\mathcal{E}_4) &= \frac{2\pi}{3}.
\end{aligned}$$

The objective function for $\mathcal{E}_1$ and $\mathcal{E}_2$ as clusters is equal to:

$$-\ln\det(\mathbf{T}_1) - \ln\det(\mathbf{T}_2) = -\ln\frac{1}{2} - \ln\frac{1}{2} = -1.386$$

The objective function for $\mathcal{E}_3$ and $\mathcal{E}_4$ as clusters is equal to:

$$-\ln\det(\mathbf{T}_3) - \ln\det(\mathbf{T}_4) = -\ln\frac{1}{3} - \ln\frac{2}{3} = -1.504$$

However, it should be noted that even though the objective functions for $\mathcal{E}_1$ and $\mathcal{E}_2$ as well as $\mathcal{E}_3$ and $\mathcal{E}_4$ are equal, their volumes are different. Despite this difference, the sum of volumes for $\mathcal{E}_1$ and $\mathcal{E}_2$ is equal to the sum of volumes for $\mathcal{E}_3$ and $\mathcal{E}_4$.

In their formulation, the authors utilize a summation of logarithmic terms within the objective function. Each of these terms has an inverse correlation with the volume of each ellipsoid. However, a critical issue arises when comparing the logarithms of individual variables. Although, it is correct that if $\det(\mathbf{A}) > \det(\mathbf{B})$, then $\ln\det(\mathbf{A}) > \ln\det(\mathbf{B})$ and this has been used for MVE formulation in literature [23, 41], controdictory, the paper implicitly assumes that if $\det(\mathbf{A}) + \det(\mathbf{C}) > \det(\mathbf{B}) + \det(\mathbf{D})$, then it follows that $\ln\mathbf{A} + \ln\mathbf{C} > \ln\mathbf{B} + \ln\mathbf{D}$. This assumption is not valid, as the logarithm function is not linear. Logarithms do not preserve the ordering of numbers, and as a result, the inequality $\ln\mathbf{A} + \ln\mathbf{C} > \ln\mathbf{B} + \ln\mathbf{D}$ cannot be directly inferred from $\det(\mathbf{A}) + \det(\mathbf{C}) > \det(\mathbf{B}) + \det(\mathbf{D})$.

B.2 Limitation 1: Only ellipsoidal clusters

The primary limitation in [30]'s formulation of the clustering method lies in its exclusive applicability to clusters characterized by ellipsoidal regions. While ellipsoids have conventionally facilitated MVE computations in the literature, it is essential to acknowledge scenarios where alternative geometric configurations may better capture the intrinsic pattern of the data. The reliance solely on ellipsoidal regions may lead to suboptimal models when confronted with diverse datasets. The literature associated with [30] has historically favored ellipsoids, resulting in a technical infrapattern for MVE computations. However, broadening the scope to encompass a more diverse range of geometric regions can enhance the optimization process, thereby ensuring accuracy and efficiency.

In result, we propose a novel approach termed Minimum Volume Norm-based Clustering (MVNBC), wherein each cluster can exhibit different norm-based regions. By adjusting the containment constraint and objective function values for the volume of each region, MVNBC offers a versatile solution that goes beyond the limitations of exclusively relying on ellipsoidal configurations. This innovative method allows for a more comprehensive representation of the underlying data pattern, addressing the shortcomings associated with a singular focus on ellipsoidal regionss.

B.3 Limitation 2: Absence of outlier detection

The other limitation of the model, as described in [30], is its lack of outlier detection. This hinders its practical usefulness, as outliers - data points that deviate significantly from expected values - can greatly affect the accuracy and reliability of the model. Decision-makers must be aware of the prevalence of outliers in their data or potential scenarios, in order to assess the model's performance and adjust their strategies accordingly. To enhance the model's applicability and ensure accurate predictions across diverse scenarios, outlier detection mechanisms are necessary within the clustering methodology.

Furthermore, the absence of outlier detection in the model poses a practical challenge, as decision-makers may struggle to distinguish between data noise and other

important data points. While some scenarios, such as gambling decisions, may not require a thorough analysis of outliers, in contexts like healthcare, even seemingly insignificant data points can hold crucial importance. The inclusion of outlier detection mechanisms is therefore crucial for distinguishing and handling outliers, ultimately improving the model's practical usefulness. By designating certain data points or scenarios as outliers, the model can reduce conservativeness in optimal decision values.

Given the variability in safeguard guarantees based on specific problem circumstances, our proposed model acknowledges the feasibility constraints in detecting anomalies, providing a nuanced and adaptable approach to outlier detection.