# A Robust Multi-Objective Approach for Clustering

Omid Nohadani and Andrew T. Zheng

**Abstract**—We propose a multi-objective method that leverages robust optimization for hierarchical clustering (rMOC). rMOC chooses clusters based on a weighted sum of data-intrinsic objective functions, determining a threshold that is most robust to uncertainties in these weights. We compare this method to the reference methods of $K$-means and Gaussian mixture models. In terms of misassignment rate, rMOC outperforms both other methods on several benchmark datasets, in part because of its independence of data-structure. Furthermore, both reference methods require solving an NP-hard problem. In contrast, rMOC exhibits $O(nm)$ complexity for data size $n$ and $m$ candidate thresholds. Additionally, rMOC can process data sequentially, reducing memory usage to a constant, as opposed to $O(n)$ of the reference methods. We also employ rMOC in a case study on data from nanoHUB, a nanotechnology research web platform. *Power* and *casual users* are consistently identified, offering insights into actual user behavior. This unsupervised learning method is generic, independent of data structure, and scales well, hence is suitable for a large variety of applications and can be enhanced by modular application of dimensionality reduction.

**Index Terms**—Clustering, unsupervised learning, multi-objective optimization, robust optimization.

## 1 Introduction

Recent years have seen the rampant proliferation of unstructured data, most noticeably in the context of web-based services. These data often lack predefined labels, defying traditional regression and supervised learning techniques. Instead, its continuous collection exhibits structures that can reveal volumes about user behavior, usage patterns and other implicit attributes.

The class of techniques used to uncover these insights is referred to as "unsupervised learning." Clustering techniques comprise a particularly active subfield of unsupervised learning, allowing practitioners to identify groups of data points that are internally similar but dissimilar across groups (for an overview, see [1; 2]). Such methods have successful large-scale applications in fields as diverse as fraud detection [3], tumor detection [4], and recommendation systems [5], amongst others. Given the large applicability of clustering methods, there exists a great diversity of existing techniques, which typically follow one of several paradigms:

*Mixture models* assume that each cluster follows a certain family of distributions. Maximum likelihood estimation is applied to determine the corresponding distribution parameters. Gaussian mixture models are a particularly popular model of this type.

*Distance-based models* define a distance metric between data points based on a set of features, then assign points to clusters to minimize the distance between points in a cluster and the cluster itself, often defined as the centroid. This process is often applied iteratively until convergence, as in the case of $K$-means clustering [6]. This and similar types of methods are particularly effective, when datasets are isotropic and of similar size.

*Hierarchical* models either partition the dataset recursively or build up clusters from smaller components. These models typically impose some criteria for merging or dividing clusters, such as a maximum distance between any two points in a cluster [2].

*Graphical* models represent the data as undirected graphs, with single observations as vertices. Edges exist between two vertices, when they are "similar" enough to each other based on some similarity metric. These graphs are then clustered using one amongst a diverse range of approaches: by checking for clusters that satisfy certain constraints on density (DBSCAN as in [7]), assessing flow capacity within and between clusters (Markov Clustering as in [8]), or iteratively propagating "importance" through the network (Affinity Propagation as in [9]).

Each of these families of techniques has found effective applications. However, the requirement of a particular distribution or distance metric has a number of disadvantages: clustering results are highly sensitive to the choice of these hyperparameters, leading to difficulty in tuning the models. Particularly in the case of distance-based models, these choices also often imply an ellipsoid shape, which can be inconsistent with the actual structure of the data. Graphical clustering approaches circumvent the limitations on cluster shape, but at the cost of a computational complexity that is infeasible for larger problems (affinity propagation constitutes a notable exception to this point) [2]. "Correctness" of these choices is difficult to ascertain, especially in higher-dimensional problems (see e.g., reviews on $K$-means in [8] and more recently in [10]).

When different criteria are employed for clustering, the multi-objective optimization problem is often solved with single-objective tools. The scalarized single-objective problem depends on parameters that were not included in the

• *O. Nohadani and A. Zheng are with the Department of Industrial Engineering & Management Sciences, Northwestern University, Evanston, IL 60208.*
*E-mail: nohadani@northwestern.edu & azheng92@gmail.com.*

original problem. The problem is then solved repeatedly for different parameter combinations, resulting in a collection of Pareto optimal solutions.

Most scalarization approaches can be divided into: (i) minimizing an aggregate of the objective, (ii) transforming objectives into constraints, and (iii) optimizing the distance to a reference point [11]. The most common type is through a weighted sum of (positively) weighted objectives. For an in-depth discussion of multi-criteria approaches in data-mining, we refer to the review in [12] and the references within. It is argued that while the latter approach is most common for data-mining purposes and is rather ad-hoc, the former two are based on first-principles [12]. Most work in this domain relates to supervised learning or takes an evolutionary approach. The unsupervised approach was divided into two steps of generation of solutions for complementary objectives and selection of the best among alternative solutions [13]. The common assumption is that well partitioned clusters also perform well under differing criteria (e.g., fitness, correctness, and connectedness), suggesting a linear (or non-linear) superposition of criteria for obtaining optimal solutions.

### Contributions

In this paper, we propose a novel unsupervised clustering technique that addresses some of these deficiencies and provides additional functionality. Our approach employs multi-objective optimization to determine a hierarchical clustering, optimizing for a number of desirable properties (stability, separation, and anomaly detection) related to the structure of features in each cluster. This work provides alternative optimization criteria and an efficient algorithm for finding optimal partitioning in the presence of uncertainty about the relative importance of each objective.

In contrast to many existing classes of models, this approach is free of assumptions on the distributions of the inputs, as well as free of the need to define a distance metric. This approach can be applied recursively to an arbitrary depth to determine a hierarchy among clusters, and allows the user to select the number of clusters after clustering instead of requiring it *a priori*. This approach is also (to our knowledge) one of the first to offer a mechanism by which an unsupervised clustering algorithm can simultaneously optimize for multiple, potentially competing, objective functions. This can also be viewed as a robust optimization-based, multi-objective alternative to objectives such as the Akaike and Bayesian Information Criteria, which combine a data fitting objective with a complexity penalty based on information-theoretic principles [14].

The remainder of this paper is structured as follows. Section 2 introduces the clustering method. We test the proposed approach empirically both for benchmarking, using publicly available data (Section 3), and for real-world relevance, employing usage data from nanoHUB, the largest provider of web-based nanotechnology research and education (Section 4). The latter data set contains usage history for over 300,000 users on 330 scientific simulation tools.

## 2 The Robust Multi-Objective Clustering Method

The robust multi-objective clustering algorithm (rMOC) determines a split (the "threshold") on a single variable (the "categorizing" variable) by optimizing a number of objective functions. These objective functions guide the algorithm towards a threshold that emphasizes certain characteristics of the distributions of both the categorizing variable and several non-categorizing variables. The algorithm determines the values of these objective functions, and chooses the threshold whose value is least sensitive to the relative weighting of these objective functions. This process can then be applied recursively to determine a hierarchical partitioning of the feature space.

This work generalizes the method in the case-study in [15], which proposed a methodology for clustering that forms the basis of our approach in this paper. The method proposed in that study selects categorizing and non-categorizing variables, but selects the clustering threshold based on a single objective: the stability of the categorizing variable. The analyst can then assess the validity of the cluster by plotting the cumulative distribution functions (CDFs) of each non-categorizing variable for each cluster and observing their differences.

In this paper, we develop a mathematical framework for the validation step, which is generalized and allows for automation. First, we identify and quantify characteristics of the distributions of the non-categorizing variable over the clusters: separation and anomaly detection (defined in Section 2.2). Importantly, these are also intrinsic characteristics that an expert would typically observe and determine to be desirable when performing the validation manually. Second, we incorporate these characteristics into our optimization method as a secondary set of objective functions. Finally, we provide a robust optimization framework to combine and optimize these objective functions that is both computationally efficient and robust to uncertainty over their relative weights.

### 2.1 Variables

Based on the desired scope of the analysis (e.g., if we aim to gain insight into the behavior of different groups of users of a web service), data intrinsic features are represented by the *variables* $X \in \mathbb{X}$ in this framework, where $\mathbb{X}$ is the set of features available in a given dataset.

For the sake of simplicity of the exposition, we assume the variables to be one-dimensional and discrete. However, the proposed framework also applies to continuous data and we describe the differences when necessary. The collection of all variables provides a multifaceted description of the underlying nature of the data (e.g., user behavior in cloud services). This set $\mathbb{X}$ is divided into one categorizing $X_c$ and many non-categorizing (or validating) $X_a$ variables. Furthermore, we assume that a critical value of the categorizing variable suffices to determine the boundary between two categories. Let this threshold value be denoted as $X_c = t^*$. When categorizing the data, each variable may display signatures that differentiate between cluster. To measure these signatures, we employ different and data-intrinsic objective functions.

### 2.2 Objectives

First, the respective relative frequencies are expressed via a probability mass function (PMF), denoted with $p_X$. Three universal objective functions $f_i$ are defined that probe the distinction amongst the clusters:

**Stability** quantifies the rate of change in the categorizing variable's PMF, identifying sharp changes that indicate transitions between portions of the distribution that are susceptive to varying $t$ and plateauing sections. For the categorizing variable, it is expressed as the first derivative of the categorizing variable's PMF, calculated using the forward finite differences method

$$f_1 = \text{stability}(t) := \frac{\delta p_{X_c}(X_c = t)}{\delta x}, \tag{1}$$

where $\delta x$ is the difference between two neighboring values of $X_c$ in the discrete case. In the continuous case, in our implementation, we find this derivative by first estimating the PDF using Gaussian kernels, then summing the derivatives of each kernel at $t$. Different choices of kernel and bandwidth may be selected based on the application, but offered no advantage in our setting. Maximizing this objective determines a split where the distribution of the categorizing variable exhibits a discontinuity.

**Separation** between the PMFs of the validating variables $X_a$ in each cluster is defined as the absolute difference in area under the two PMFs of category A, for which $X_c \leq t$, and category B, for which $X_c > t$:

$$\begin{aligned} f_{2,a} &= \text{separation}(t, a) \\ &:= \sum_x \Big| P(X_a = x | X_c \leq t) - P(X_a = x | X_c > t) \Big|. \end{aligned} \tag{2}$$

This offers a measure of the degree to which the threshold "separates" the distributions of the validating variables of the respective neighboring categories. Note that the set of validating variables $\mathbb{X}_a$ may be large, leading to correspondingly many separation objectives, indexed via $a$. In the continuous case, we apply an analogous approach using kernel density estimation to approximate the probability density functions (PDFs) of each category and integrating over the absolute difference.

**Anomaly Detection** serves to differentiate specific characteristics available in one cluster, but not in the other. It is measured as the separation of the derivative of the PMF between the two clusters:

$$\begin{aligned} f_{3,a} &= \text{anomaly}(t, a) \\ &:= \sum_x \Big| \frac{\delta P(X_a = x | X_c \leq t)}{\delta x} - \frac{\delta P(X_a = x | X_c > t)}{\delta x} \Big|. \end{aligned} \tag{3}$$

In the discrete case, the forward finite differences method is applied to calculate the derivatives. In the continuous case, we first apply kernel density estimation to approximate the PDFs, then sum the derivatives of the Gaussian kernels.

This objective determines the aggregated difference in conditional stability and offers a measure of the degree to which the clustering can reveal anomalies in the distribution of a validating variable. Such anomalies occur, whenever a subgroup within a category exhibits a collective behavior that departs from the overall trend of the parent category.

### 2.3 Optimization

In order to determine an optimal threshold $t^*$ for all objectives $f_i$ simultaneously, we scalarize them by using the weighted sum approach, following the commonly suggested multi-criteria optimization approach, as described in [11]:

$$G(\lambda, t) = \lambda g_c(t) + (1 - \lambda) g_v(t) \tag{4}$$

where the scalar $\lambda \in [0, 1]$ determines the relative weighting of the two objective functions. Here, $g_c$ denotes the categorizing objective function, hence $g_c \equiv f_1$, and $g_v$ the validating (non-categorizing) equally weighted sum-objective given by $g_v \equiv \sum_{i \in \{2,3\}; a} f_{i,a}$. Given the potentially long-tailed nature of the data distribution, all objectives $f_i$ are normalized, e.g., to the 95-percentile. Consequently, the nominal optimization problem over the set of candidate thresholds $\mathbb{T}$ and a fixed $\lambda$ is can be formulated as:

$$\underset{t \in \mathbb{T}}{\text{maximize}} \quad G(\lambda, t). \tag{5}$$

### 2.4 Robust Optimization

There is typically significant uncertainty about the value of $\lambda$. Often $\lambda$ is chosen ad hoc or its value is informed by experts, leaving the outcome sensitive to its choice and hence jeopardizing confidence in the results. We aim to address this issue and provide solutions that are robust to this uncertainty; in other words, the optimal choice of threshold should be as insensitive to $\lambda$ as possible. In this section, we provide an efficient algorithm for searching for such a robust threshold: i.e., the threshold $t^*$ that remains optimal over the largest range of $\lambda$. This corresponds to solving the following nested optimization problem:

$$\begin{aligned} \underset{t \in \mathbb{T}}{\text{maximize}} \quad & \lambda_{max,t} - \lambda_{min,t} \\ \text{subject to} \quad & \lambda_{max,t} = \underset{\lambda}{\text{argmax}}\ \lambda \\ & \qquad \text{subject to} \quad G(\lambda, t) \geq G(\lambda, s) \quad \forall s \in \mathbb{T} \\ & \lambda_{min,t} = \underset{\lambda}{\text{argmin}}\ \lambda \\ & \qquad \text{subject to} \quad G(\lambda, t) \geq G(\lambda, s) \quad \forall s \in \mathbb{T} \\ & \lambda_i \in [0, 1]. \end{aligned} \tag{6}$$

The structure of $G$ can be exploited to develop an efficient algorithmic solution to Problem (6) by formulating an auxiliary linear program. We see that $G(\lambda, t^*)$ is a convex piecewise-linear function in $\lambda$, where $t^*$ is the optimal value of $t$ for any given $\lambda$. By introducing an auxiliary variable $\hat{G}$, we can describe the set of thresholds optimal for some value of $\lambda$ as a polyhedron bounded by a set of linear constraints, where each threshold $t$ corresponds to a constraint of the form $\hat{G} \geq \lambda g_c(t) + (1 - \lambda) g_v(t)$ leading to $\hat{G} + (g_v(t) - g_c(t))\lambda \geq g_v(t)$. Then, for some threshold $t^*$ to be optimal for some value $\lambda^*$, the constraint corresponding to $t^*$ must be active at $\lambda^*$. The following theorem connects the optimal solution $t^*$ to the properties of the polyhedron.

**Theorem 1.** *Suppose we have a set of thresholds $t \in \mathbb{T}$ and their objective values $g_c(t)$ and $g_v(t)$.*

*Let $\mathbb{P} = \{x : x \in \langle \widehat{G}, \lambda \rangle \in \mathbb{R}^2 \mid \widehat{G} \geq \lambda g_c(t) + (1 - \lambda) g_v(t), \forall t \in \mathbb{T}\}$ be the polyhedron defined by these thresholds.*

*Then, $t^*$ is an optimal solution to Problem (5) for some value of $\lambda$, if and only if its corresponding constraint is active at the boundary of this polyhedron.*

*Proof.* Each direction is shown separately:

$\rightarrow$ Suppose we have a threshold $t^* = \text{argmax}_t G(\lambda^*, t)$ for some value $\lambda = \lambda^*$. Then, based on the formulation of $\mathbb{P}$, we know that $\widehat{G} \geq G(\lambda^*, t^*) \rightarrow \langle \widehat{G}, \lambda^* \rangle \in \mathbb{P}$. This is the constraint corresponding to $t^*$, and it follows that it must be active (i.e. equality).

$\leftarrow$ Suppose a constraint corresponding to threshold $t^*$ is active at the boundary of $\mathbb{P}$ for some fixed $\lambda^*$. Then, by the definition of $\mathbb{P}$, $\langle \widehat{G}, \lambda^* \rangle \in \mathbb{P} \rightarrow \widehat{G} \geq G(\lambda^*, t^*)$. Thus, $t^*$ must be optimal for $\lambda^*$. □

From this theorem, it follows that we can efficiently determine the set of thresholds $t^*$ such that for all $t$, $\exists \lambda^*, G(\lambda^*, t^*) \geq G(\lambda^*, t)$ by tracing the boundaries of $\mathbb{P}$, as illustrated in Figure 1. Furthermore, the distance in $\lambda$ between any two adjacent extreme points of the polyhedron determines the range of $\lambda$ for which the corresponding threshold is optimal.

**Corollary 1.** *Finding the range in $\lambda$ for which each threshold is optimal, is equivalent to finding the distance in $\lambda$ between two adjacent extreme points of $\mathbb{P}$.*

*Proof.* Since $\mathbb{P}$ is the epigraph of a convex, continuous, piecewise linear function of $\lambda$, each threshold $t$ must correspond to at most one set of values $\lambda_{min,t}$ and $\lambda_{max,t}$, defining the range of $\lambda$ values for which $t$ is optimal. Each of these $\lambda_{min}$ and $\lambda_{max}$ values, by definition, occur at the intersection of the constraint representing $t$ and a constraint representing some other threshold $t'$, which forms a basis for an extreme point. Thus, the size of the optimal $\lambda$ range for a threshold $t$ is equivalent to the distance in $\lambda$ between these two extreme points. □

Corollary 1 forms the basis for an algorithm to solve Problem (6). We need only to iterate through the extreme points of $\mathbb{P}$ and record at each step the constraints active at the extreme point. We then determine the distance in $\lambda$ between adjacent extreme points to obtain the solution. This intuition is shown in Figure 1.

We accomplish this by formulating an auxiliary linear program whose feasible set corresponds to $\mathbb{P}$, to which we can then apply the simplex method to traverse the extreme points. This has the additional benefit of modularity: any existing simplex implementation can be plugged into rMOC. Problem (7) serves this purpose: by setting an initial solution at $\lambda = 0$, applying the simplex method to this problem will traverse the extreme points in order of $\lambda$ from 0 to 1. At each iteration, we record the threshold whose corresponding slack variable enters the basis, as well as the $\lambda$ value of the extreme point.

$$\begin{aligned} \underset{\lambda, \widehat{G}}{\text{maximize}} \quad & \lambda \\ \text{subject to} \quad & \widehat{G} - \lambda(g_c(t) - g_v(t)) \geq g_v(t), \quad \forall t \in \mathbb{T} \\ & \lambda \in [0, 1]. \end{aligned} \tag{7}$$

At this point, we have the $\lambda$ range for which each threshold is optimal, and we simply take the maximum over this set to determine the solution to Problem (6): the threshold most robust to perturbations in $\lambda$. Algorithm 1 below details the method in full.

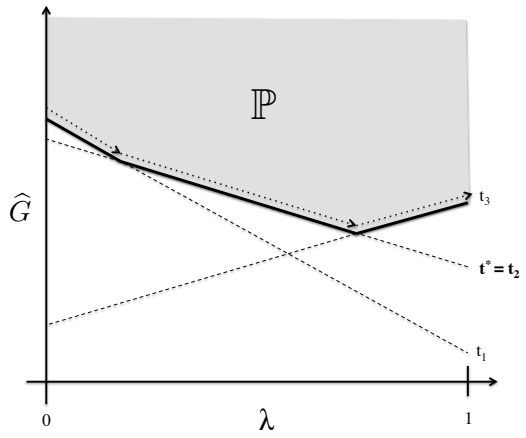


Fig. 1. Graphical representation of the polyhedron $\mathbb{P}$ in Theorem 1. Each threshold $t_i$ corresponds to a constraint, where the solid line denotes the optimal region for $t_i$ and the dashed line the non-optimal region. The dotted line shows the path of the rMOC algorithm.

**Algorithm 1** Simplex algorithm to search for optimal thresholds

**procedure** FINDINGTHRESHOLD ▷ Initialization
  extreme_lambdas $\leftarrow [0]$
  optimal_thresholds $\leftarrow []$ ▷ Determine the initial solution at $\lambda = 0$
  $x \leftarrow \text{argmax}_t \ g_v(t)$
  $N \leftarrow \{x\}$
  ▷ Iterate through extreme points of the polyhedron
  **while** $x$ is not optimal **do**
    Run one iteration of the simplex method on Problem (7); Let $\lambda_{\text{new}}$ be the value of the new solution, and $N_{\text{new}}$ the set of indices of the nonbasic columns in the solution.
    Append $\lambda_{\text{new}}$ to extreme_lambdas
    Append $N \setminus N_{\text{new}}$ to optimal_thresholds
    $N \leftarrow N_{\text{new}}$
  lambda_ranges $\leftarrow []$
  **for** $i$ in $length$(thresholds) **do**
    lambda_ranges $[i]$ $\leftarrow$ extreme_lambdas $[i+1]$ $-$ extreme_lambdas $[i]$
  $j \leftarrow \text{argmax}_i$ lambda_ranges $[i]$ **return** thresholds $[j]$

### 2.5 Computational Remarks

rMOC consists of two steps: (1) computing the value of the objective functions for each candidate threshold, and (2) optimizing over these values.

The complexity of the first step depends entirely on the user-supplied objective functions: given $m$ candidate thresholds and $n$ data points, where computing the objective functions for a single threshold takes $f(n)$ time, the time complexity of this first step is $O(m \cdot f(n))$. In the second step, the number of extreme points in our auxiliary LP in Equation 7 is bounded by the number of thresholds,

and thus requires at most $m$ iterations to complete. In combination, we find that rMOC has a total time complexity of $O(m \cdot f(n))$, largely driven by the first step. In the case study in Section 4, we achieve strong empirical results using a set of objective functions with complexity $O(n)$, leading to an overall complexity of $O(mn)$.

In practice, the first step is also trivially parallelizable (computing objective values for a single threshold requires no information from any other threshold) and thus scales fairly well. Additionally, most objectives exhibit some kind of macro-level trend across thresholds (e.g., in many situations for $t_1 < t_2$ the objectives are $f(t_1) < f(t_2)$ or $f(t_1) > f(t_2)$), suggesting that an approximate approach such as binning thresholds or performing a binary search across thresholds can be effective for particularly large problems.

Our implementation provides an option for parallelization to exploit multi-core and/or parallel architectures and solves the linear program using the IBM CPLEX solver.

Furthermore, the VC dimension, which predicts a probabilistic upper bound on the classification error of a model, is 3 for rMOC due to the linear nature of the clustering thresholds that it produces. This suits the goal of simplicity and computational efficiency for the algorithm: it estimates a single parameter, the threshold, at each level of clustering.

### 2.6 Summary

In summary, rMOC assesses potential clusters by evaluating a combination of objective functions describing properties of the cluster: stability, separation and anomaly detection. It efficiently determines the threshold that is most robust to uncertainty with regards to how to combine these objective functions. The algorithm is deterministic and has a worst-case time complexity of $O(m \cdot f(n))$, where $n$ is the number of data points, $m$ is the number of candidate thresholds, and $f(n)$ is the complexity of the user-supplied objective functions. The summarized steps of the algorithm are:

1) Determine a variable on which to cluster.
2) For each candidate threshold on this variable, calculate the stability, separation, and anomaly detection objective functions.
3) Use the simplex method to choose the threshold least sensitive to the weights of the linear superposition of the three objective functions.
4) If the depth of the hierarchy exceeds some predefined threshold, end the recursion. Otherwise, return to step 1 with each identified cluster as new input for deeper clustering.

## 3 Performance of the rMOC Method

We demonstrate the rMOC method on fivedifferent and commonly used benchmark datasets. To measure the performance, we compare rMOC to two other unsupervised clustering algorithms. Finally, since data can always be contaminated or incomplete, we probe the sensitivity of rMOC to such data uncertainties.

### 3.1 Comparison with Other Clustering Algorithms

Here, we compare the proposed rMOC algorithm with two of the most commonly used unsupervised clustering methods: Gaussian mixture models (GMM) and $K$-means clustering algorithms. We offer several use cases and examples that demonstrate the strengths and weaknesses of each model. For these experiments, we used the Mclust GMM implementation [16] and the variant of $K$-means [17], both of which have frequently used R implementations and represent a common application of these methods in the field.

We first compare the performance of our model on a dataset commonly used to demonstrate the effectiveness of clustering tools: the iris dataset, which is a widely known dataset in clustering literature and publicly available on the UCI Machine Learning Repository [18]. The dataset contains 50 instances with four features that are relevant to the classification of three species of the iris flower, which are to a large extent separable based on these four features. We take misassignment rate as our performance metric: the percentage of data points in the same predefined categories that the algorithm assigned to different clusters. Although not a perfect measure of cluster quality, if we take the predefined categories to represent a major portion of variation in the other features, then well-designed clusters are likely to correspond to a great extent to these predefined categories, as in the iris dataset.

Figure 2 displays a projection of the distribution of the data over two of the four features, namely Petal Length and Sepal Width. Each group of empty symbols represents a flower that was correctly clustered by the respective method. Filled symbols reflect the misassigned data, where the fill color reflects the cluster suggested by the method. All three methods correctly cluster all of the iris setosa data. In this projection, the clusters for iris versicolor and iris virginica have a sizable overlap with respect to all four features, two of which are shown in Figure 2. At the boundary, each of the three methods correctly assigns most of the data to the species cluster and has some misassigned data points.

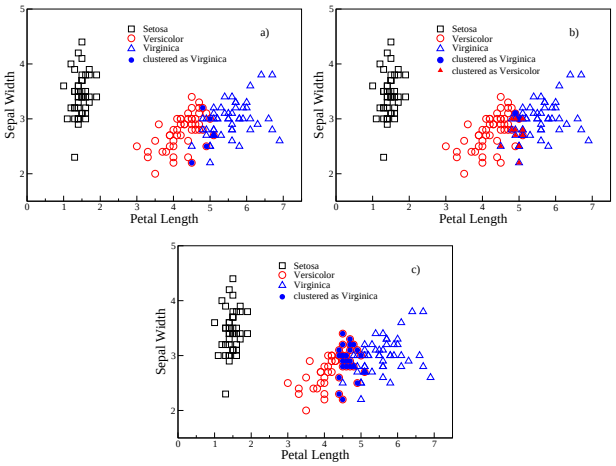


Fig. 2. Comparison of algorithms on iris dataset: (a) GMM, (b) $K$-means, (c) rMOC. The symbols represent the species: empty symbols for a correct assignment and filled otherwise (the color refers to incorrectly assigned species). Misassignment rates: GMM 3%, $K$-means 11%, rMOC 16%.

The performance of rMOC, as measured by misassignment rate, slightly underperforms both the GMM and $K$-means models; however, the results are still comparable. This result is due to the nature of the data: GMM and $K$-means are excellent tools for dealing with roughly spherical, evenly-sized clusters. Despite this fact, rMOC is comparable with these leading methods, and indeed determines very reasonable thresholds to divide the clusters, as shown in Figure 2. In many applications, however, the structure of the data cannot be measured using Euclidean distances or assumed follow a Gaussian distribution (or exhibit an elliptical distribution). We assessed the relative effectiveness of rMOC, GMM and $K$-means on four such real-world test datasets: blood donorship, banknote authentication, cancer survival, and vertebral disease [18]. Each dataset contains a number of features that are associated with binary predefined categories.

For the purposes of this analysis, all three clustering algorithms were forced to determine two clusters. The misassignment rate was determined by mapping clusters to predefined classes so as to minimize the misassignment rate. Additionally, for rMOC, we selected the categorizing variable to minimize misassignment rate. When comparing rMOC with GMM and $K$-means based on misassignment rate, rMOC outperforms both alternative methods on most datasets, achieving a lower mean misassignment rate, as well as less variance across bootstrap samples, as shown in Table 1. In fact, $K$-means only outperforms rMOC on the transfusion dataset; this, however, s not a statistically significant difference given the variance as estimated by the bootstrap, shown in Figure 3 and discussed in Section 3.2. This provides evidence of the effectiveness of rMOC in such non-idealized, real-world applications with non-elliptical cluster structures.

| Dataset | Size | Dimensions | GMM | $K$-means | rMOC |
|---|---|---|---|---|---|
| Transfusion | 748 | 4 | 41.2 % | **28.7 %** | 32.4 % |
| Banknote | 1372 | 4 | 45.0 % | 39.1 % | **24.9 %** |
| Cancer | 306 | 3 | 33.0 % | 45.3 % | **25.8 %** |
| Vertebra | 310 | 6 | 34.9 % | 34.7 % | **26.5 %** |

TABLE 1
Misassignment rates of methods applied to four datasets.

Beyond the data-structure independence of rMOC, its computational complexity is, in general, superior to GMM and $K$-means. Applying either of the latter reference methods requires solving an NP-hard problem, which in practice leads to the use of nondeterministic approximative methods (i.e. heuristics). In contrast, rMOC exhibits a polynomial time as long as the supplied objective functions have a polynomial time complexity, as discussed in Section 2.5. Additionally, rMOC processes data sequentially, reducing memory usage to a constant.

### 3.2 Sensitivity Comparison

As discussed, the outcome of $K$-means and GMM heavily depends on the data structure, on the definition of the norm used, and on the non-deterministic approach. Another key aspect that may jeopardize the quality of the results is when the input data are contaminated, either through faulty recording, processing or by their mere nature. Often, this uncertainty is considered as "noise", which is typically stochastic and can often be filtered using known distributional assumptions. However, in many real-world problems, the sources are deterministic and at times even adversarial, which cannot be modeled distributionally.

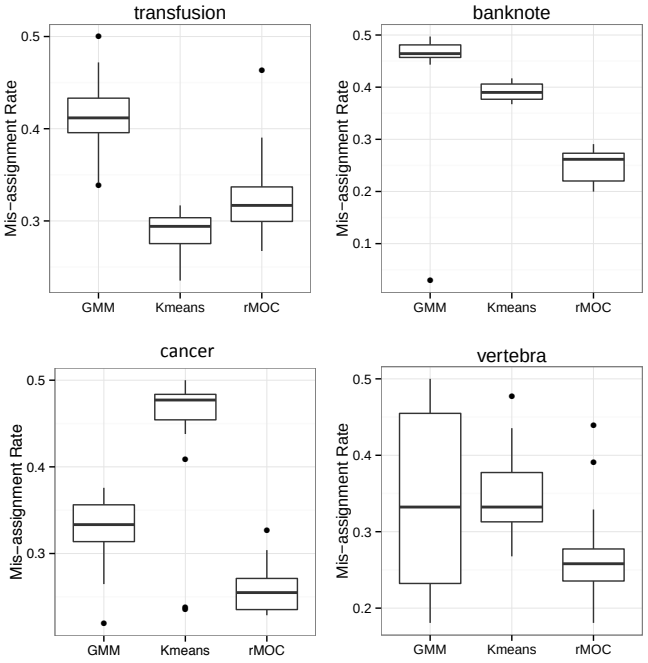


Fig. 3. Comparison of algorithms over four UCI datasets, showing the distribution of misassignment rate over 25 bootstrapped samples.

To probe their impact on the outcome of the compared clustering algorithms, bootstrapping was proposed to measure the robustness of the clusters and their estimators [19]. We applied a non-parametric bootstrap with 25 iterations to estimate the misassignment rate. Figure 3 exhibits the box plots of the bootstrapping of all four studied dataset when comparing $K$-means, GMM, and rMOC methods. Not only does rMOC typically outperform GMM and $K$-means in terms of misassignment rate, but it also does so quite consistently: the variance in misassignment rate across bootstrap runs is similar to or lower than $K$-means and GMM on all experimental datasets.

## 4 Case Study: nanoHUB Users

The National Science Foundation Center for Network for Computational Nanotechnology established a web-based platform, called nanoHUB, to accelerate the research and education in this domain [20]. This platform serves over 300,000 users across the world with scientific simulation tools, education materials, and a platform for exchange of data and know-how for a wide range of user expertise. Arguably, the most remarkable feature is a tool platform for remotely conducting state of the art scientific computation using just a web browser, allowing rapid deployment and dissemination of such (simulation) tools and efficient discovery without geographical or infrastructural limitations. A recent study of over 300 nanoHUB tools and their usage

over one academic year used heuristic methods and identified the undergraduate users as exhibiting the richest set of features [15]. Therefore, we focus on the tool usage data from undergraduate students and extend the dataset over three academic years (2009 through 2011).

### 4.1 Datasets and methodology

The dataset contains 8,646 users and 646 simulation tools. In direct coordination with the nanoHUB administration, we accessed the data via a remote MySQL database. Given our desire for a scalable approach over such a large dataset, features were determined based on data size.

To directly compute the size and volume without distributional assumptions, we took a measure theoretical ansatz to categorization. With this, a non-negative and real-valued number is assigned to a subset of data (for each user) without hypothesizing possible outcomes. The $p$-norm satisfies the conditions for a measure, namely the non-negativity, empty set, and the countable additivity properties, when $0 \leq p \leq \infty$ [21; 22]. For higher-dimensional data, nested $p$-norms allow for a down-projection to the one dimension of interest by differing the value of $p$. Specifically, we choose a combination of $p = 0$ and $p = \infty$ norms because they directly extract the size and volume metrics.

Given the three-dimensional nature of this data, the three possible arrangements of two nested zero and infinity norms yield the following three different metrics (two nested infinity norms is redundant).

Let $D_{ijk}$ be the length (in seconds) of user $i$'s $k^{\text{th}}$ job on tool $j$ and let $M(i)$ be the placeholder of a metric. We use the dimension-wise $p$-norm $\|\cdot\|_{p|_x}$ of the vector along the dimension $x$ to compute the metrics $M(i)$ through a nested arrangement of an outer and an inner norm as

$$M(i) = \left\| \|D_i\|_{p|_k} \right\|_{q|_j}, \quad p, q \in \{0, \infty\}. \tag{8}$$

The choices of $p$ and $q$ define metrics as following:

**Length**: $L_i$ is the maximum length of any job a given user has run during an academic year and is computed with $p = q = \infty$ in Eq. (8).

**Frequency**: $F_i$ is the extent to which a user has delved into any single tool, as measured by the maximum number of jobs a user has run repeatedly on any single tool divided by the time length (one academic year) and can be measured with $p = 0$ and $q = \infty$ in Eq. (8).

**Diversity**: $N_i$ is the diversity of a user's exposure to the website, as measured by the number of distinct tools a user has actually used over the corresponding timeframe. This can be measured by Eq. (8) with $p = q = 0$.

Note that the definition in Eq. (8) can be extended to higher dimensions with deeper nested arrangements. Also, other norms may probe different aspects of the data. However, the nature of the data needs to match the specific choice of other norms, making the results highly sensitive to the choice of these hyperparameters. This issue is prevalent in $K$-means approaches [10].

These features exhibit extremely long-tailed distributions, which would have an outsize impact on our objective functions. To remedy this, we truncate each feature to its $95^{\text{th}}$ percentile. The categorizing variable in our analysis is *frequency*.

### 4.2 Results

We achieved a number of interesting outcomes from applying our algorithm to this data, which we describe below. Results differed across demographic groups (each presented separately), but in all cases the algorithm determined clusters that exhibited unique "signatures" in their distributions. We first discuss results for undergraduate users, who exhibit the most visible clustering due to the organization of classes and the seasonality of the academic year; we then compare these results with those for faculty and graduate users.

#### 4.2.1 Undergraduate Users

For undergraduate users in all years, the algorithm determined one cluster whose distribution over *diversity* is roughly a decreasing exponential (we refer to them as *casual users*), while another cluster exhibits a mode at a diversity of 2 (we refer to them as *power users*), as shown in Figure 4.

Recall that our algorithm performs a univariate split on

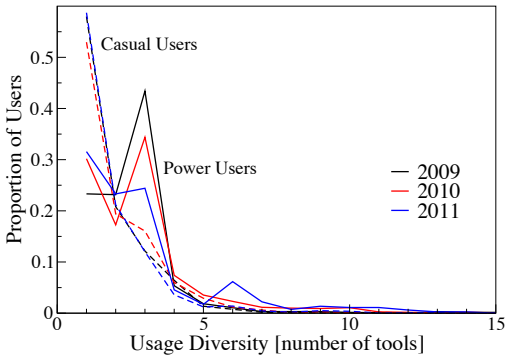


Fig. 4. The proportion of users, decaying with usage diversity, is clearly separated for power and casual users over different years.

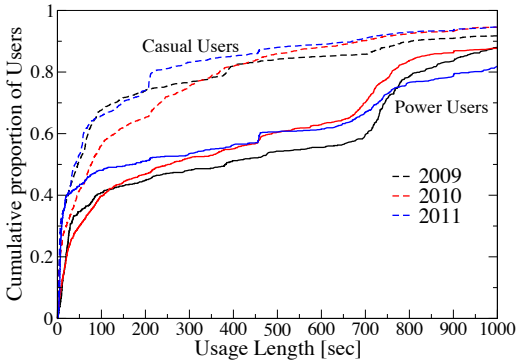


Fig. 5. The cumulative proportion of users, increasing with the usage length, is clearly separated for power and casual users.

*frequency*; in this case (and many others, as we will see), the algorithm exposes structural patterns in the noncategorizing features as well. Also note that despite the fact that the actual thresholds chosen are very different, the algorithm exposes the same underlying patterns.

We see a similarly interesting pattern when observing the distributions of the clusters over *length*, as illustrated

in Figure 5. The CDF of the entire population exhibits a number of sharp drops, or anomalies; after clustering, however, we find that only the power-users exhibit these anomalies in many years, while casual users exhibit a much smoother curve. This offers a very detailed picture of usage in the cluster; perhaps a large portion of users in the anomalous cluster have taken a particular course, where they all completed the same assignment, leading to the anomaly. Comparable education implications were discussed in [15]. The observation in Figure 5 offers strong evidence of the anomaly detection capabilities of rMOC, as well as an example of the insights to be gained from unsupervised clustering methods in general.


Fig. 6. Undergraduate users in the same cluster across different years use a very similar set of tools, as measured by cosine similarity, while there is a large difference across clusters.

Additionally, we found that features unrelated to those used in the algorithm also exhibit large differences across clusters. Of course, this results from the correlation between the clustering variable and these features; regardless, this provides further evidence of clustering's ability to uncover intrinsic patterns of variation in the data. One example is tool usage across groups: we created a vector of values $X_{ij}$ = Number of times a tool $i$ was used by any undergraduate user in cluster $j$, and calculated the cosine similarity between these vectors pairwise for every combination of cluster and year. The results, shown in Figure 6, demonstrate that casual clusters exhibit very high similarity with other casual clusters across years, as power clusters do with other power clusters across years; however, the similarity between power and casual users is very low.

### 4.2.2 *Faculty and Graduate Users*

Faculty and graduate users both exhibited interesting patterns as well, although these were not as pronounced as those of undergraduate users, as shown in Figures 7 and 8. Interesting to note is that the thresholds determined for faculty users were very similar for all three years, perhaps indicating that faculty exhibit more consistent behavior across years. Power-users for both groups also showed a spike in their *diversity* distributions similar to that of the undergraduate power-users.

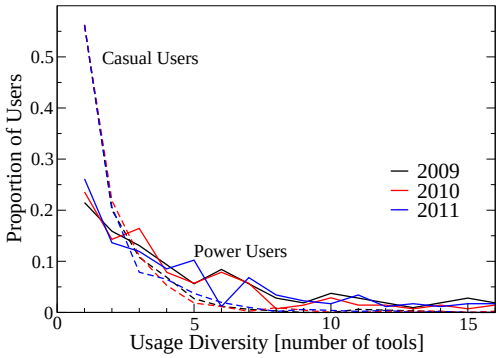


Fig. 7. The proportion of faculty decays with usage diversity and is separated for power and casual users.

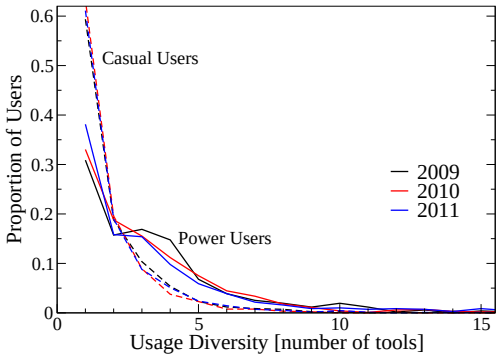


Fig. 8. The proportion of graduate students decays with diversity and is separated for power and casual users.

## 5 Discussion

In general, for the objective functions to be comparable, we must either normalize the input variables or normalize the objective function values. This is a common problem in multi-objective optimization, as well as in clustering approaches in general: the tendency of $K$-means to output spherical clusters, for example, means that the clusters are heavily dependent on the relative scaling of the input variables.

Although our application uses a single split, the algorithm is fundamentally a hierarchical approach, which follows naturally from applying the algorithm recursively. Future work can focus on exploring the hierarchical nature of the algorithm; outstanding tasks include determining heuristics to select a categorizing variable and to terminate the recursive splitting process. The number of hierarchical levels can be determined automatically through the comparison of the size of final clusters. If they equate to the pre-set minimum (or maximum) size of cluster, then the recursion terminates and yields the number of hierarchical levels.

We expect that pairing this approach with dimensionality reduction techniques (such as principal components analysis) will yield particularly powerful results. One consideration here is that the number of objectives to optimize increases linearly with the number of variables, which decreases the influence of each individual objective and could lead to selecting a threshold that simply achieves a weak compromise between all objectives. Dimensionality reduction offers a way to circumvent this limitation by reducing the number of objectives while retaining the structure of the data. Additionally, the axis-aligned nature of this algorithm

can be somewhat limiting; rotations or transformations of the feature space such as those performed in dimensionality reduction can extend the functionality of this algorithm.

Furthermore, in rich nature datasets, some features may be inherently correlated to other variables. As a result, additional characteristics can be detected without a direct analysis. Such an example occurs, when the weekly usage of nanoHUB tools exhibited a clear dispersion between power and casual users, even though no explicit time-series study was conducted. Similarly, tool popularity was not directly probed and yet a clear distinction between the two categories was observed.

## 6 Concluding Remarks

The proposed rMOC method represents a novel approach to unsupervised clustering, free of distance metrics and other common assumptions, and offers the additional flexibility of multiple objective functions. It achieves this by adapting the simplex method to select a clustering threshold whose optimality is robust to the relative weighting of these objectives.

Our approach exhibits several theoretical and empirical properties, including complexity, reliable detection of anomalies and the ability to identify common patterns from different datasets. For the practitioner, this algorithm works well out of the box thanks to its robust nature, and applies to a variety of datasets unsuitable for $K$-means and other common clustering techniques.

## Acknowledgment

We would like to acknowledge G. Klimeck for insightful discussions and providing the nanoHUB user data.

## References

[1] R. C. Tryon, *Cluster analysis: correlation profile and orthometric (factor) analysis for the isolation of unities in mind and personality*. Edwards Brother, Incorporated, lithoprinters and publishers, 1939.

[2] D. J. Hand, H. Mannila, and P. Smyth, *Principles of data mining*. MIT Press, 2001.

[3] C. S. Hilas and P. A. Mastorocostas, "An application of supervised and unsupervised learning approaches to telecommunications fraud detection," *Knowledge-Based Systems*, vol. 21, no. 7, pp. 721–726, 2008.

[4] G. Magoulas, V. Plagianakos, D. Tasoulis, and M. Vrahatis, "Tumor detection in colonoscopy using the unsupervised k-windows clustering algorithm and neural networks," in *Fourth European Symposium on Biomedical Engineering*, 2004.

[5] H. Xie, L. Zhang, J. Sun, and X. Yu, "Application of k-means clustering algorithms in news comments," in *E-Business and E-Government, 2010 Intern. Conference on*. IEEE, 2010, pp. 3759–3762.

[6] S. Lloyd, "Least squares quantization in pcm," *IEEE Trans. Inform. Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[7] P. Shrivastava and H. Gupta, "A review of density-based clustering in spatial data," *International Journal of Advanced Computer Research*, vol. 2, 2012.

[8] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," vol. 31, no. 3, pp. 264–323, 1999.

[9] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, 2007.

[10] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.

[11] M. Ehrgott, *Multicriteria optimization*. Springer Science & Business Media, 2006.

[12] A. A. Freitas, "A critical review of multi-objective optimization in data mining: a position paper," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 2, pp. 77–86, 2004.

[13] J. Handl and J. Knowles, "Multi-objective clustering and cluster validation," in *Multi-Objective Machine Learning*. Springer, 2006, pp. 21–47.

[14] H. Akaike, "Likelihood of a model and information criteria," *J. Econometrics*, vol. 16, no. 1, pp. 3–14, 1981.

[15] O. Nohadani, J. Dunn, and G. Klimeck, "Categorizing users of cloud services," *Service Science*, vol. 8, no. 1, pp. 59–70, 2016.

[16] C. Fraley and A. E. Raftery, "Model-based clustering, discriminant analysis, and density estimation," *Journal of the American statistical Association*, vol. 97, no. 458, pp. 611–631, 2002.

[17] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Applied Statistics*, pp. 100–108, 1979.

[18] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[19] C. Field, Z. Pang, and A. Welsh, "Bootstrapping robust estimates for clustered data," *J. Amer. Statist. Assoc.*, vol. 105, no. 492, pp. 1606–1616, 2010.

[20] M. Lundstrom and G. Klimeck, "The NCN: Science, Simulation, and Cyber Services," *IEEE Conference on Emerging Technologies - Nanoelectronics*, 2006.

[21] F. Riesz, "Sur les systèmes orthogonaux de fonctions," *CR Acad. Sci. Paris*, vol. 144, pp. 615–619, 1907.

[22] E. Fischer, "Sur la convergence en moyenne," *CR Acad. Sci. Paris*, vol. 144, pp. 1022–1024, 1907.

**Omid Nohadani** received a Diploma degree in physics from the University of Bonn, Germany and a Ph.D. in physics from the University of Southern California. He was a postdoctoral researcher at the Operations Research Center at the Massachusetts Institute of Technology and a research fellow at Harvard Medical School. He is an Associate Professor in the department of Industrial Engineering and Management Sciences at Northwestern University. His primary area of research is robust optimization with applications to machine learning, healthcare, and high-performance computing.

**Andrew T. Zheng** received a B.S. in Industrial Engineering and Management Sciences and an M.S. in Computer Science from Northwestern University. He currently works as a data scientist at Uber Technologies, Inc. in San Francisco, CA, applying machine learning and optimization techniques to urban transportation problems. His primary area of research is machine learning and optimization.