Clustering in the Presence of Noise

by

Nika Haghtalab

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2013

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Clustering, which is partitioning data into groups of similar objects, has a wide range of applications. In many cases unstructured data makes up a significant part of the input. Attempting to cluster such part of the data, which can be referred to as noise, can disturb the clustering on the remaining domain points. Despite the practical need for a framework of clustering that allows a portion of the data to remain unclustered, little research has been done so far in that direction. In this thesis, we take a step towards addressing the issue of clustering in the presence of noise in two parts. First, we develop a platform for clustering that has a cluster devoted to the "noise" points. Second, we examine the problem of "robustness" of clustering algorithms to the addition of noise.

In the first part, we develop a formal framework for clustering that has a designated noise cluster. We formalize intuitively desirable input-output properties of clustering algorithms that have a noise cluster. We review some previously known algorithms, introduce new algorithms for this setting, and examine them with respect to the introduced properties.

In the second part, we address the problem of robustness of clustering algorithms to the addition of unstructured data. We propose a simple and efficient method to turn any centroid-based clustering algorithm into a noise robust one that has a noise cluster. We discuss several rigorous measures of robustness and prove performance guarantees for our method with respect to these measures under the assumption that the noise-free data satisfies some niceness properties and the noise satisfies some mildness properties. We also prove that more straightforward ways of adding robustness to clustering algorithms fail to achieve the above mentioned guarantees.

Acknowledgments

First and foremost, I would like to thank my supervisor, Professor Shai Ben-David, for his support and guidance throughout my Master's studies. I would also like to thank my committee members, Professors Daniel Brown and Daniel Lizotte, for their time and remarks. Thanks go out to Margareta Ackerman for many fruitful discussions.

I would like to thank Erik Louie for proofreading this thesis at various stages. I would also like to thank my friends and lab-mates for their support, suggestions, and for creating a great and lively environment at work.

Last but not least, I would like to thank my parents, Flora and Naser, and my sister Ayda for their unconditional love and support throughout the years.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Clustering, partitioning the data into groups of similar objects, has many applications in image analysis, information retrieval, market research, and city planning. It is often the case that data sets that one wishes to cluster contain, in addition to groups of similar objects, a significant subset that is unstructured. For example,

- Consider an image processing task with the goal of finding objects in a given image. Pixels in an image represent distinct objects that are placed on an unstructured background set. A suitable clustering method should find objects (represented by clusters) and discard the background set along with other noise points and outliers.

- Consider the use of clustering in market research for the purpose of finding target groups. Marketing strategies, usually, target significantly large groups of people with similar traits. A good clustering of the data should contain clusters that represent such large groups and ignore the rest of the data. The output of appropriate clustering methods should not be affected by the presence of a few small outlying groups of customers.

- Consider clustering traffic data for the purpose of city planning. Part of this data contains distinguished patterns of traffic, for instance the rush-hour commute to and from the business district, but a significant portion of this data is made from unstructured points, for example local traffic between residential areas. While significant patterns of traffic are important to city planning at large, for the most part, less structured data does not have the same effect.

Partitioning the whole data when there is a significant amount of noise can result in meaningless clusters. However, most common clustering algorithms produce a partition of the input set, regardless. Despite the practical need for a framework of clustering that addresses issues that arise in clustering in the presence of noise, little research has been done so far in that direction. In this thesis, we take a step towards addressing the issue of clustering in the presence of noise in two parts. In chapter 2, we develop a platform for clustering that has a cluster devoted to the "noise" points, such that the points in this cluster are not required to be similar. In chapter 3, we examine the problem of "robustness" of clustering algorithms to the addition of noise in the input data.

In the first part, we develop a formal framework for clustering with a cluster devoted to noise, called the "noise cluster". We define intuitive and desirable input-output properties of clustering algorithms that have a noise cluster. These properties address the richness of the range of the algorithms, their invariance properties with respect to various changes in data, and their computational feasibility compared to that of clustering algorithms without a noise cluster. We generalize some previously known algorithms that have a natural notion of a noise cluster. Moreover, we introduce two efficient algorithms that have a noise cluster. We examine these algorithms with respect to our properties.

In the second part, we address the issue of "robustness" of clustering algorithms to the addition of unstructured points. We introduce multiple rigorous measures of noise robustness. We propose a simple and efficient method to transform any centroid-based clustering algorithm to a noise-robust one that has a noise cluster. The degree of noise-robustness that is achieved by this transformation depends on a parameter that can be tuned based on users' needs. We prove performance guarantees for our method with respect to the robustness measures under the assumption that the noise-free data satisfies some niceness properties and the noise satisfies some mildness properties. We also prove that more straightforward ways adding robustness to clustering algorithms have inherent limitations and do not enjoy the same mentioned guarantees.

Chapter 2

Clustering with a Noise Cluster

2.1 Introduction

This chapter is devoted to developing a framework for clustering that allows a portion of the data to remain unclustered. The first contribution of this chapter is the introduction of a formalism for clustering that has a designated noise cluster. The points that belong to the noise cluster are not required to be similar. This formalism allows us to discuss desirable behaviour of clustering algorithms that produce a partial clustering of the data. The second contribution of this chapter is the introduction of properties that examine the input-output behaviour of clustering algorithms that possess a noise cluster. These properties address the richness of the range of potential clustering algorithms, their invariance with respect to various changes in the original data set, and their computational feasibility compared to that of clustering algorithms without a noise cluster.

Another contribution of this chapter is the introduction of clustering algorithms that have a noise cluster. We generalize two previously known clustering algorithms, *trimmed* algorithms by Cuesta-Albertos et al. [6] and DBScan by Ester et al. [11], which have a natural notion of a noise cluster. Moreover, we introduce two efficient clustering algorithms with a noise cluster. We prove that one of these algorithms is equivalent to a generalized non-fuzzy variation of an algorithm introduced by Dave [8]. Finally, we analyze these algorithms with respect to our properties. This analysis can be used to distinguish among different clustering algorithms that have a noise cluster. It can also guide the selection of clustering algorithms based on properties that one expects from specific clustering applications.

This chapter is organized as follows. In section 2.2, we provide a summary of related work. Section 2.3 introduces notations and definitions that are used in the rest of this chapter. Section 2.4 formalizes some intuitive and desirable properties of clustering algorithms that have a noise cluster and examines our algorithms with respect to them.

2.2 Related Work

Our approach for developing a theoretical framework for clustering with a noise cluster is related to two main research directions: First, developing a general theory for clustering (without a noise cluster). Second, developing algorithms that have a designated noise cluster.

Several directions have been taken in developing a theory of clustering without considering a noise cluster, e.g. [3, 17, 19, 22]. Puzicha et al. [22] investigate the class of clustering algorithms that arises from using a specific type of objective function. Jardine et al. [17] show that Single-Linkage is the only function that is consistent with a set of defined clustering axioms, while, Kleinberg [19] shows that no clustering algorithm satisfies a set of three intuitive axioms. Our approach is similar to the work of Ackerman et al. [3], which develops a property-based classification of clustering paradigms (without a noise cluster). That is, instead of using a set of "axioms" to define what should be called a clustering with a noise cluster, we introduce properties that explain the behaviour of clustering algorithms that have a noise cluster. The satisfaction of these properties can vary between different clustering paradigms and may be used to categorize clustering algorithms.

Several clustering methods that have a noise cluster have been suggested in the past [6, 8, 11]. Dave [8] introduces the concept of a "noise cluster" in a fuzzy centroid-based setting by defining a noise-prototype that is equidistant from all the domain points. Cuesta-Albertos et al. [6] proposes the use of *trimming*: searching for a subset of the input data of a predetermined size whose removal leads to the maximum improvement of the clustering quality (or objective function). A "density-based" approach for clustering noisy data is introduced by Ester et al. [11]. It assigns points from the sparse regions of the domain to the noise cluster. In this work, we generalize the above algorithms, introduce two new algorithms, and show that one of them is equivalent to a non-fuzzy variation of Dave's algorithm. Moreover, we examine the input-output properties of these algorithms with respect to a set of intuitive properties.

Discussing the details of previous work requires the definition of few notations, hence, it is delayed to the relevant sections.

2.3 Preliminaries

In this section we develop notions and definitions that are used in the remainder of this chapter. In many applications, there is a natural notion of weighted input, where every data point is associated with a real valued weight. The weight of a point, which is a measure of its significance, plays an important role in deciding how it should be clustered. For example, in the field of market research, different customers carry different levels of importance. We might prefer to target customers with higher income or loyalty. This can be easily done in the weighted setting by assigning weights that correspond to the significance of the individuals. In another example, consider the problem of placing fire stations such that most regions of a city can be accessed quickly. Providing quick firefighting service to certain sites may be more important than to others. In the weighted setting, we can easily prioritize certain landmarks over others by assigning higher weights to them. In this chapter, we use a setting for clustering weighted data that was introduced by Ackerman et al. [1].

2.3.1 Background

For a finite set $\mathcal{X}$ and integer $k \geq 1$, a *k-clustering with a noise cluster* of $\mathcal{X}$ is an ordered pair $(\{C_1, \ldots, C_k\}, \Phi)$, such that $\{C_1, \ldots, C_k, \Phi\}$ is a *partition* of $\mathcal{X}$. The set Φ represents the noise cluster and $\mathcal{C} = \{C_1, \ldots, C_k\}$ represents the set of (traditional) clusters. A *general clustering with a noise cluster* of $\mathcal{X}$ is a k-clustering with a noise cluster of $\mathcal{X}$ for an arbitrary $1 \leq k < |\mathcal{X}|$.

Let d denote a *distance* function over $\mathcal{X}$ that is *non-negative, symmetric* and for all $x \in \mathcal{X}$, $d(x,x) = 0$. For any $\alpha > 0$ and distance function d_1, the distance function $d_2 = \alpha d_1$ is defined such that for all $x, y \in \mathcal{X}$, $d_2(x,y) = \alpha d_1(x,y)$. Let w be a positive *weight* function defined over

$\mathcal{X}$. For any $\alpha > 0$ and a weight function w_1, the weight function $w_2 = \alpha w_1$ is defined such that for all $x \in \mathcal{X}$, $w_2(x) = \alpha w_1(x)$. With a slight abuse of notation, we denote the total weight of a set $\mathcal{X}$ by $w(\mathcal{X}) = \sum\limits_{x \in \mathcal{X}} w(x)$.

A *k-clustering algorithm with a noise cluster* is a computable function $\mathcal{A}$ that takes as input a set $\mathcal{X}$, a distance function d, a (possibly constant) weight function w, and returns a k-clustering with a noise cluster of $\mathcal{X}$. Similarly, a *general clustering algorithm with a noise cluster* is a function $\mathcal{A}$ that takes as input a set $\mathcal{X}$, a distance function d, a weight function w, and returns a general clustering with a noise cluster of $\mathcal{X}$. We use $\mathcal{A}(\mathcal{X}, d, w) = (\mathcal{C}, \Phi)$, to refer to the resulting clustering.

In the remainder of this chapter, whenever it is clear from the context, we use *clustering* to refer to a clustering with a noise cluster, and *clustering algorithm* to refer to a clustering algorithm with a noise cluster.

2.3.2 Clustering Algorithms

In this section, we introduce several algorithms for clustering weighted input. An important aspect of our formulation is that these algorithms return the same clustering in the weighted setting as they would produce in a setting that, instead of using weights, includes multiple copies of a point. Given a data set $\mathcal{X}$ with distance d, elements $x, y \in \mathcal{X}$ are considered duplicates if $d(x, y) = 0$ and for all $z \in \mathcal{X}$, $d(x, z) = d(y, z)$. To transform unweighted input to weighted input we remove the duplicate points and associate every remaining point with a weight equal to the number of its copies in the original data. In the same way, zero weight for a point can be represented by removing the point from the domain set. For this reason, we simply restrict our attention to positive weight functions, though not necessarily integral ones.

Throughout this chapter, let $g : \mathbb{R}^+ \to \mathbb{R}^+$ be any continuous, increasing, and unbounded function. A function $f : \mathbb{R}^+ \to \mathbb{R}^+$ is *homogeneous* with *degree* $p \geq 1$ if for any $x \in \mathbb{R}^+$ and $\alpha \geq 0$, $f(\alpha x) = \alpha^p f(x)$. In the remainder of this chapter, we say that g is homogeneous, if it is in addition to being continuous, increasing, and unbounded (the standard requirements) is also homogeneous. Note that many common functions satisfy these properties, e.g. $g(x) = x^2$.

Consider an input set $\mathcal{X}$, drawn from a given space E, along with distance d and weights w.

The (k, g)-*centroid clustering algorithm* (without a noise cluster) is a clustering algorithm that minimizes the function

$$\Lambda^g_{d,w}(\{C_1, \ldots, C_k\}) = \min_{\mu_1,\ldots,\mu_k \in E} \sum_{i\in[k]} \sum_{x\in C_i} w(x) \cdot g(d(x, \mu_i))$$

We refer to μ_i as the *center* of cluster C_i and we define $\mu(x) = \arg\min_{\mu_i\in\{\mu_1,\ldots,\mu_k\}} d(x, \mu_i)$. With a slight abuse of notation we can also define the (k, g)-centroid algorithm as the algorithm that chooses centers $\mu_1, \ldots, \mu_k$ that minimize

$$\Lambda^g_{\mathcal{X},d,w}(\mu_1, \ldots, \mu_k) = \sum_{x\in\mathcal{X}} w(x) \cdot g(d(y, \mu(x)))$$

In the remainder of this section, we introduce four clustering algorithms that have a noise cluster, three of which are based on the (k, g)-centroid clustering algorithm, and one is a general clustering algorithm.

Definition 2.1 ((k, g)-δ-truncated)**.** *The* (k, g)-δ-truncated *algorithm is a* k*-clustering algorithm that defines* $d'(x, y) = \min\{d(x, y), \delta\}$*, minimizes the objective function*

$$\Lambda^g_{\mathcal{X},d',w}(\mu_1, \ldots, \mu_k)$$

and returns $(\{C_1, \ldots, C_k\}, \Phi)$*, such that for* $j \in [k]$*,* $C_j = \{x \in \mathcal{X} | \, j = \arg\min_i d(x, \mu_i)$ *and* $d(x, \mu_j) \leq \delta\}$ *and* $\Phi = \{x \in \mathcal{X} | \, \min_i d(x, \mu_i) > \delta\}$*.*

Dave [8] defines the *noise prototype* by introducing a point that is equidistant from all points in E. The clustering is then obtained by performing a fuzzy $(k + 1)$-means algorithm with one center fixed as the noise prototype. In the next definition, we provide a generalization of the non-fuzzy variation of Dave's algorithm for any centroid-based algorithm. Furthermore, we show that the class of algorithms produced in such way is equivalent to the class of (k, g)-δ-truncated algorithms.

Definition 2.2 ((k, g)-δ-centroid)**.** *Let* μ^* *be defined such that for all* $y \in E$*,* $d(y, \mu^*) = \delta$*. The* (k, g)-δ-centroid *algorithm minimizes the objective function*

$$\Lambda^g_{\mathcal{X},d,w}(\mu_1, \ldots, \mu_k, \mu^*)$$

and returns $(\{C_1, \ldots, C_k\}, C_{k+1})$*, such that if we define* $\mu_{k+1} = \mu^*$*, then for* $j \in [k + 1]$*,* $C_j = \{x \in \mathcal{X} | \, j = \arg\min_i d(x, \mu_i)\}$*.*

Theorem 2.1. *Clustering $(\mathcal{C}, \Phi)$ is an optimal (k, g)-δ-centroid clustering of $\mathcal{X}$ if and only if $(\mathcal{C}, \Phi)$ is an optimal (k, g)-δ-truncated clustering of $\mathcal{X}$.*

Proof. Let $\mu^* = \mu_{k+1}$ be defined as the noise prototype (equidistant from all points in E) and for all $x, y \in E$, $d'(x, y) = \min\{\delta, d(x, y)\}$. We show that a (k, g)-δ-centroid clustering with centers $\mu_1, \ldots, \mu_k, \mu^*$ and a (k, g)-δ-truncated clustering with centers $\mu_1, \ldots, \mu_k$ have the same objective value.

$$\begin{aligned}
\Lambda^g_{\mathcal{X},d',w,E}(\mu_1, \ldots, \mu_k) &= \sum_{x \in \mathcal{X}} w(x) g \left(\min_{i \in [k]} \{\min\{\delta, d(x, \mu_i)\}\} \right) \\
&= \sum_{x \in \mathcal{X}} w(x) g \left(\min_{i \in [k]} \{\min\{d(x, \mu_{k+1}), d(x, \mu_i)\}\} \right) \\
&= \sum_{x \in \mathcal{X}} w(x) g \left(\min_{i \in [k+1]} d(x, \mu_i) \right) \\
&= \Lambda^g_{\mathcal{X},d,w,E \cup \{\mu^*\}}(\mu_1, \ldots, \mu_k, \mu^*)
\end{aligned}$$

Moreover, $\mu_1, \ldots, \mu_k, \mu^*$ induce the same (k, g)-δ-centroid clustering as the (k, g)-δ-truncated clustering induced by $\mu_1, \ldots, \mu_k$. Therefore, $(\mathcal{C}, \Phi)$ is an optimal (k, g)-δ-centroid clustering if and only if it is an optimal (k, g)-δ-truncated clustering. □

Note that for an optimal (k, g)-δ-centroid clustering, $(\mathcal{C}, \Phi)$, the cost of this solution is equal to the cost of the optimal (k, g)-centroid clustering of $\mathcal{C}$ added with a constant cost of $g(\delta)$ for each point in Φ. With a slight abuse of notation and using Theorem 2.1, we use the following notation to refer to the cost of (k, g)-δ-truncated clustering $(\mathcal{C}, \Phi)$.

$$\Lambda^g_{d',w}(\mathcal{C}, \Phi) = \Lambda^g_{d,w}(\mathcal{C}) + w(\Phi) g(\delta) \tag{2.1}$$

In the next algorithm, we use a "naive" approach for determining the noise using a centroid-based clustering algorithm. The following algorithm first finds a (k, g)-centroid clustering of the data, and then declares any point farther than δ from its corresponding cluster center to be noise.

Definition 2.3 ((k, g)-δ-naive-truncated)**.** *The* (k, g)-δ-naive-truncated *algorithm is a k-clustering algorithm that minimizes the objective function $\Lambda^g_{\mathcal{X},d,w}(\mu_1, \ldots, \mu_k)$ and returns the k-clustering $(\{C_1, \ldots, C_k\}, \Phi)$, such that for all $j \in [k]$, $C_j = \{x \in \mathcal{X} |\, j = \arg\min_i d(x, \mu_i)$ and $d(x, \mu_j) \leq \delta\}$ and $\Phi = \{x \in \mathcal{X} |\, \min_i d(x, \mu_i) > \delta\}$.*

Cuesta-Albertos et al. [6] suggest a *trimming* procedure for clustering noisy data. This approach searches for a subset of the input data (of a predetermined size) whose removal leads to the maximum improvement of the value of the objective function. Here, we define this category of algorithms in the weighted setting.

Definition 2.4 ((k,g)-η-trimmed)**.** *The* (k,g)-η-trimmed *algorithm is a* k*-clustering algorithm that minimizes the objective function*

$$\min_{\mathcal{X}'\subseteq\mathcal{X}:\, w(\mathcal{X}')\geq(1-\eta)w(\mathcal{X})} \quad \min_{\mu_1,\dots,\mu_k} \Lambda^g_{\mathcal{X}',d,w}(\mu_1,\dots,\mu_k)$$

and returns $(\{C_1,\dots,C_k\},\Phi)$*, such that for all* $j\in[k]$*,* $C_j=\{x\in\mathcal{X}' \,|\, j=\arg\min_i d(x,\mu_i)\}$ *and* $\Phi=\mathcal{X}\setminus\mathcal{X}'$*.*

In the (k,g)-η-trimmed algorithm, parameter η bounds the size of the noise cluster. On the other hand, in the (k,g)-δ-truncated algorithm, parameter δ determines the radius of a clusters, and consequently, affects the size of the noise cluster. The following theorem implies that if the size of the noise clusters in the (k,g)-η-trimmed and (k,g)-δ-truncated optimal clusterings are the same, then they are equivalent.

Theorem 2.2. *For any* $\mathcal{X}$*,* d*, and* w*, let* $\mathcal{A}$ *be the* (k,g)*-*δ*-truncated algorithm and* $\mathcal{A}(\mathcal{X},d,w)=(\mathcal{C},\Phi)$*. For* $\eta=\frac{w(\Phi)}{w(\mathcal{X})}$*, let* $\mathcal{A}'$ *be the* (k,g)*-*η*-trimmed algorithm. Then,* $\mathcal{A}'(\mathcal{X},d,w)=(\mathcal{C},\Phi)$*.*

Proof. Let $\mathcal{A}'(\mathcal{X},d,w)=(\mathcal{C}',\Phi')$ such that $w(\Phi')\leq\eta w(\mathcal{X})$.

$$\begin{aligned}\Lambda^g_{d,w}(\mathcal{C}) &= \Lambda^g_{d',w}(\mathcal{C},\Phi)-w(\Phi)g(\delta)\\ &\leq \Lambda^g_{d',w}(\mathcal{C}',\Phi')-\eta w(\mathcal{X})g(\delta)\\ &\leq \Lambda_{d,w}(\mathcal{C}')+w(\Phi')g(\delta)-\eta w(\mathcal{X})g(\delta)\\ &\leq \Lambda_{d,w}(\mathcal{C}')\end{aligned}$$

Moreover, $w(\Phi)\leq\eta w(\mathcal{X})$, therefore, $\mathcal{A}'(\mathcal{X},d,w)=(\mathcal{C},\Phi)$. □

Ester et al. [11] suggest using a density-based clustering for noisy data. Their algorithm, called DBScan, clusters data points that are close to dense regions of the input, and declares any point that is not clustered as noise. Here, we define a variation of DBScan for clustering weighted data.

The ϵ-neighbourhood of a point $x \in \mathcal{X}$ is denoted by $N_\epsilon(x) = \{y \in \mathcal{X} | \, d(x,y) \leq \epsilon\}$. A point x is density-reachable from a point y with respect to ϵ and $minPts$ if there is a chain of points $y = p_0, \ldots, p_n = x$, such that for all $i \in [n]$, $p_i \in N_\epsilon(p_{i-1})$, and for all $i < n$, $w(N_\epsilon(p_i)) \geq minPts$. Two points $x, y \in \mathcal{X}$ are density-connected with respect to ϵ and $minPts$ if there is a point $z \in \mathcal{X}$, such x and y are density-reachable from z with respect to ϵ and $minPts$.

Definition 2.5 ($(\epsilon, minPts)$-DBScan)**.** *The* $(\epsilon, minPts)$-DBScan *algorithm is a general clustering algorithm that takes $\mathcal{X}$, d, and w, and returns $(\{C_1, \ldots, C_k\}, \Phi)$, such that for every x, if $w(N_\epsilon(x)) \geq minPts$, then there exists C_j, such that $x \in C_j$. Furthermore, for all C_i*

1. $w(C_i) \geq minPts$
2. *for all $x \in C_i$, if y is density-reachable from x, then $y \in C_i$.*
3. *for all $x, y \in C_i$, x and y are density-connected.*

and $\Phi = \mathcal{X} \setminus \bigcup_{i \in [k]} C_i$.

Note that a clustering algorithm can have multiple ideal outcomes. Therefore, $\mathcal{A}(\mathcal{X}, d, w) = (\mathcal{C}, \Phi)$ is used to indicated that $(\mathcal{C}, \Phi)$ is one possible outcome of $\mathcal{A}(\mathcal{X}, d, w)$, not necessarily the only one.

2.4 Properties of Clustering Algorithms with a Noise Cluster

The problem of clustering with a designated noise cluster is developed around an intuitive goal of capturing all the noise from the input within the noise cluster, while partitioning the rest of the data into representative clusters. However, an important question to ask is, *what is a noise cluster?* Different clustering applications employ very different clustering algorithms and there is no single clustering algorithm that is suitable for all applications. Consequently, what should be found in a noise cluster varies based on clustering needs and applications. Therefore, it is

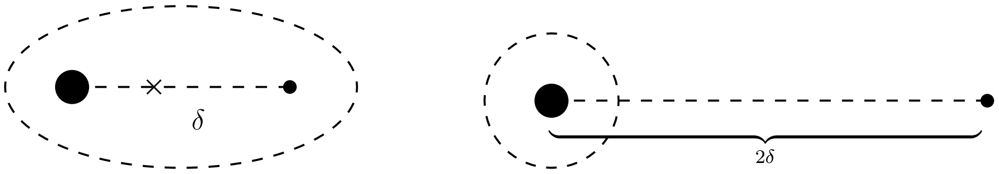
	Weight-Scalability	Distance-Scalability	Cluster-Weight-Scalability	Noise-Weight-Scalability	Noise-Removal-Invariance	Noise-Scatter-Invariance	Noise-Dispersion-Invariance	Cluster-Richness	Noise-Richness	Efficiency
(k, g)-δ-truncated	✓	×	✓	✓	✓	✓	✓	✓	×	✓
(k, g)-δ-naive-truncated	✓	×	×	×	×	×[1]	×[1]	✓	×	✓
(k, g)-η-trimmed	✓	✓[1]	×	×	×	✓	✓	✓	✓	×
$(\epsilon, minPts)$-DBScan	×	×	×	✓	✓	✓	✓	✓	✓	✓

Table 2.1: Illustrating the properties that are satisfied by some common clustering algorithms.

unlikely that an axiomatic platform would be universally appropriate for defining what a noise cluster and a clustering algorithm with such a cluster should be. Instead, we formalize some intuitively desirable input-output properties of clustering algorithms that have a noise cluster, and compare our algorithms with respect to these properties. This comparison, on one hand, distinguishes between different clustering algorithms that have a noise cluster, on the other hand, can guide the selection of clustering algorithms based on properties that are desired in specific clustering applications.

In this section, we propose properties that examine different aspects of clustering algorithms that have a noise cluster, including, their response to changes in the original data (see Sections 2.4.1, 2.4.2, and 2.4.3), richness of their clustering range (see Section 2.4.4) and their efficiency (see Section 2.4.5). We examine the algorithms with respect to these properties (see Table 2.1). Statements and proofs of positive results are included in Section 2.4.6, while the negative results are demonstrated by counter examples after introducing each property.

[1] g is homogeneous



(a) Original Data

(b) Change in the distance

Figure 2.1: (k, g)-δ-truncated and (k, g)-δ-naive-truncated are not distance scalable

2.4.1 Properties Pertaining to Scaling the Input

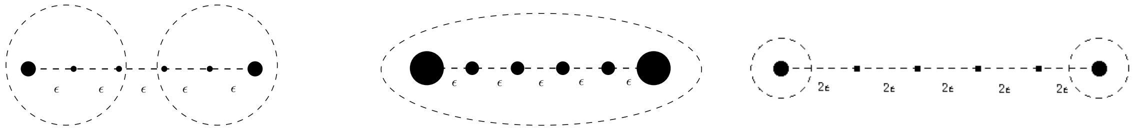
In this section, we introduce properties that address an invariance in the output of clustering algorithms when there is a change in the scale of distance or weight measurements. Uniform scaling of the distance or weight functions does not change the relative positions or significance of the input points. Therefore, it may be desirable that the outcome of an algorithm would remain invariant when such changes are applied to the data.

Definition 2.6 (Weight-Scalability). *A clustering algorithm $\mathcal{A}$ satisfies* weight-scalability *if for any $\mathcal{X}$, d, w, and $\alpha > 0$, $\mathcal{A}(\mathcal{X}, d, \alpha w) = \mathcal{A}(\mathcal{X}, d, w)$.*

Definition 2.7 (Distance-Scalability). *A clustering algorithm $\mathcal{A}$ satisfies* distance-scalability*, if for any $\mathcal{X}$, d, w, and $\alpha > 0$, $\mathcal{A}(\mathcal{X}, \alpha d, w) = \mathcal{A}(\mathcal{X}, d, w)$.*

These scalability properties can be viewed as stating that the clustering algorithms should not have a built-in unit of weight or distance. Therefore, algorithms that use parameters to specify a scale do not satisfy these properties. For example, the (k, g)-δ-truncated and (k, g)-δ-naive-truncated algorithms use δ to determine the distance beyond which a point is considered an outlier, hence, they do not satisfy distance-scalability. Similarly, $(\epsilon, minPts)$-DBScan uses ϵ and $minPts$ as measures of distance and weight, to determine dense regions of the data, hence, it does not satisfy distance-scalability or weight-scalability. The following example demonstrates the lack of scalability in the (k, g)-δ-truncated and (k, g)-δ-naive-truncated algorithm.

Example 1. *For $k = 1$, let $\mathcal{A}$ represent the (k, g)-δ-truncated or (k, g)-δ-naive-truncated algorithm. Let $\mathcal{X} = \{x, y\}$, such that $d(x, y) = \delta$ and $w(x) > w(y)$ (as shown in Figure 2.1a), then $\mathcal{A}(\mathcal{X}, d, w) = (\{\{x, y\}\}, \emptyset)$. For $\alpha = 2$, let μ represent the center of $\{x, y\}$ using*



(a) Original Clustering (b) Scaling the weight function (c) Scaling the distance function

Figure 2.2: $(\epsilon, minPts)$-DBScan does not satisfy weight-scalability, cluster-weight-scalability, or distance-scalability. The clusters are shown by dotted circles and the noise cluster includes any point that does not belong to a cluster.

distance measure αd. Since $w(x) > w(y)$ and g is an increasing function, $d(x, \mu) < d(y, \mu)$, hence $d(y, \mu) > \delta$. Therefore, $\mathcal{A}(\mathcal{X}, \alpha d, w) = (\{\{x\}\}, \{y\})$ (see Figure 2.1b). Hence, the (k, g)-δ-truncated and (k, g)-δ-naive-truncated algorithms do not satisfy distance-scalability.

Using k well-separated copies of the above structure, we can generalize Example 1 to hold for any number of clusters, $k \geq 1$. We now show the lack of scalability in $(\epsilon, minPts)$-DBScan.

Example 2. *For any ϵ and $minPts$, let $\mathcal{A}$ be $(\epsilon, minPts)$-DBScan algorithm. Let $\mathcal{X} = \{x_1, \ldots, x_6\}$ such that $w(x_1) = w(x_6) = minPts$, for $2 \leq i \leq 5$, $w(x_i) = minPts/4$, and for $i \leq 5$, $d(x_i, x_{i+1}) = \epsilon$. As shown in Figure 2.2a, $\mathcal{A}(\mathcal{X}, d, w) = (\{\{x_1, x_2, x_3\}\{x_4, x_5, x_6\}\}, \emptyset)$. For any $\alpha > 2$ and $i \leq 6$, $\alpha w(N_\epsilon(x_i)) > minPts$, so all the points are in one cluster, i.e. $\mathcal{A}(\mathcal{X}, d, \alpha w) = (\{\{x_1, \ldots, x_6\}\}, \emptyset)$ (see Figure 2.2b). Therefore, $(\epsilon, minPts)$-DBScan does not satisfy weight-scalability. Similarly, for any $\alpha > 1$, $\alpha(d(x_i, x_{i+1})) > \epsilon$, hence, $w(N_\epsilon(x_i)) < minPts$, for $2 \leq i \leq 5$. Therefore, $\mathcal{A}(\mathcal{X}, \alpha d, w) = (\{\{x_1\}, \{x_6\}\}, \{x_2, \ldots, x_5\})$ (see Figure 2.2c) and $(\epsilon, minPts)$-DBScan does not satisfy distance-scalability.*

On the other hand, Theorem 2.3, 2.4, and 2.5, respectively, show that the (k, g)-δ-truncated, (k, g)-δ-naive-truncated, and (k, g)-η-trimmed algorithms satisfy weight-scalability. Furthermore, Theorem 2.6 shows that the (k, g)-η-trimmed algorithm satisfies distance-scalability when g is a homogeneous function.

2.4.2 Properties Pertaining to the Significance of the Noise

In this section, we introduce properties that address an invariance in the output of clustering algorithms when there is a decrease in the intensity of the noise. Decreasing the weight of the points in the noise cluster, removing the noise points, or increasing the weight of the clustered data decreases the relative intensity of the noise. Therefore, it may be desirable that the outcome of a clustering algorithm with a noise cluster would remain invariant when such changes are applied to the data.

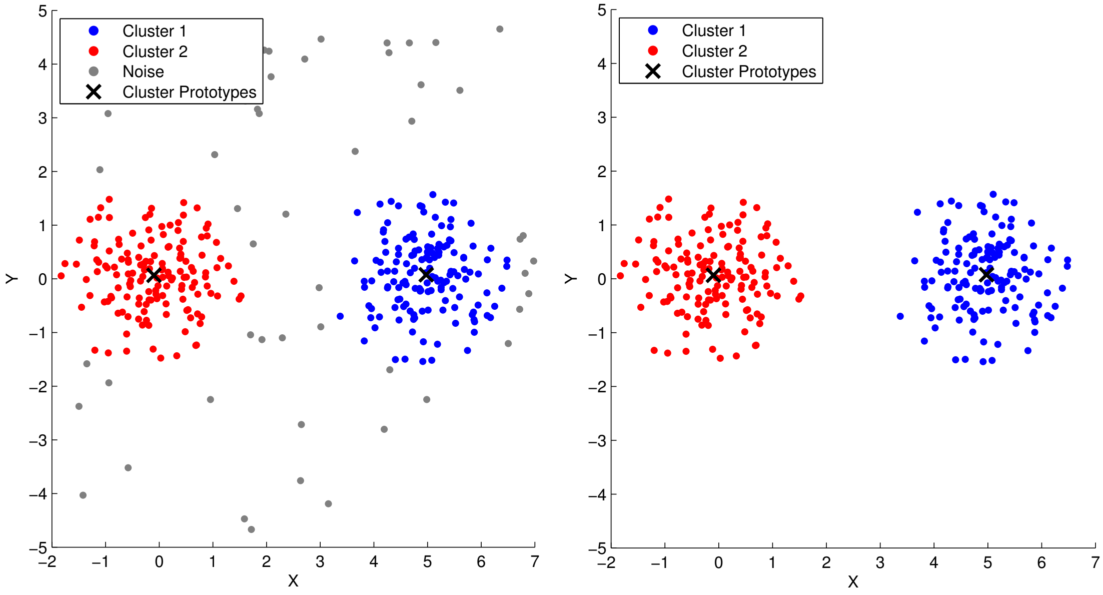
Definition 2.8 (Noise-Weight-Scalability). *Let $\mathcal{A}$ be any clustering algorithm and for any $\mathcal{X}$, d, and w_1, let $\mathcal{A}(\mathcal{X}, d, w_1) = (\mathcal{C}, \Phi)$. $\mathcal{A}$ satisfies* noise-weight-scalability *if for all $0 < \alpha \leq 1$ and w_2, such that $w_2(x) = \alpha w_1(x)$ for $x \in \Phi$ and $w_2(x) = w_1(x)$ otherwise, $\mathcal{A}(\mathcal{X}, d, w_2) = \mathcal{A}(\mathcal{X}, d, w_1)$.*

Definition 2.9 (Cluster-Weight-Scalability). *Let $\mathcal{A}$ be any clustering algorithm and for any $\mathcal{X}$, d, and w_1, let $\mathcal{A}(\mathcal{X}, d, w_1) = (\mathcal{C}, \Phi)$. $\mathcal{A}$ satisfies* cluster-weight-scalability *if for all $\alpha \geq 1$ and w_2, such that $w_2(x) = w_1(x)$ for $x \in \Phi$ and $w_2(x) = \alpha w_1(x)$ otherwise, $\mathcal{A}(\mathcal{X}, d, w_2) = \mathcal{A}(\mathcal{X}, d, w_1)$.*

In a clustering algorithm that does not have a built-in unit of weight, scaling down the weight of a noise cluster by α has the same effect as scaling up the weight of the clustered data by $\frac{1}{\alpha}$. Therefore, in such a clustering algorithm, noise-weight-scalability and cluster-weight-scalability are equivalent. The relation between these properties is shown in the following Lemma.

Lemma 2.1. *A weight-scalable clustering algorithm $\mathcal{A}$ satisfies cluster-weight-scalability if and only if it satisfies noise-weight-scalability.*

Proof. Assume $\mathcal{A}$ satisfies weight-scalability, cluster-weight-scalability, and for $\mathcal{X}$, d, and w_1, let $\mathcal{A}(\mathcal{X}, d, w_1) = (\mathcal{C}_1, \Phi_1)$. For any $\alpha \leq 1$, let $w_2 = \alpha w_1$ and $\mathcal{A}(\mathcal{X}, d, w_2) = (\mathcal{C}_2, \Phi_2)$. Since $\mathcal{A}$ satisfies weight-scalability, $(\mathcal{C}_2, \Phi_2) = (\mathcal{C}_1, \Phi_1)$. Let $w_3(x) = w_2(x)$ for $x \in \Phi_2$ and $w_3(x) = \frac{1}{\alpha} w_2(x)$ otherwise, and let $\mathcal{A}(\mathcal{X}, d, w_3) = (\mathcal{C}_3, \Phi_3)$. Since, $\mathcal{A}$ satisfies cluster-weight-scalability, $(\mathcal{C}_3, \Phi_3) = (\mathcal{C}_2, \Phi_2) = (\mathcal{C}_1, \Phi_1)$. Hence, for $w_3(x) = \alpha w_1(x)$ for $x \in \Phi$, and $w_3(x) = w_1(x)$ otherwise, $\mathcal{A}(\mathcal{X}, d, w_3) = (\mathcal{C}_1, \Phi_1)$. Therefore, $\mathcal{A}$ satisfies noise-weight-scalability.



(a) Original Clustering

(b) Clustering after removing the noise.

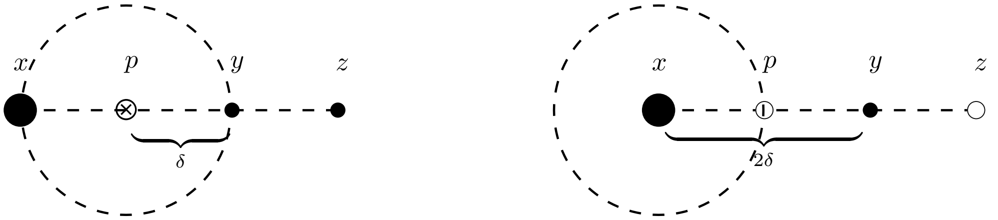
Figure 2.3: An example demonstrating noise-removal-invariance of (k, g)-δ-truncated

Assume $\mathcal{A}$ satisfies weight-scalability, noise-weight-scalability, and $\mathcal{A}(\mathcal{X}, d, w_1) = (\mathcal{C}_1, \Phi_1)$. For any $\alpha \geq 1$, let $w_2 = \alpha w_1$ and $\mathcal{A}(\mathcal{X}, d, w_2) = (\mathcal{C}_2, \Phi_2)$. Since $\mathcal{A}$ satisfies weight-scalability, $(\mathcal{C}_2, \Phi_2) = (\mathcal{C}_1, \Phi_1)$. Let $w_3(x) = \frac{1}{\alpha} w_2(x)$ for $x \in \Phi_2$ and $w_3(x) = w_2(x)$ otherwise. Since $\mathcal{A}$ satisfies noise-weight-scalability, $(\mathcal{C}_3, \Phi_3) = (\mathcal{C}_2, \Phi_2) = (\mathcal{C}_1, \Phi_1)$. Hence, for $w_3(x) = w_1(x)$ for $x \in \Phi_1$, and $w_3(x) = \alpha w_1(x)$ otherwise, $\mathcal{A}(\mathcal{X}, d, w_3) = (\mathcal{C}_1, \Phi_1)$. Therefore, $\mathcal{A}$ satisfies cluster-weight-scalability. $\square$

Definition 2.10 (Noise-Removal-Invariance)**.** *Let $\mathcal{A}$ be any clustering algorithm and for any $\mathcal{X}$, d, and w, let $\mathcal{A}(\mathcal{X}, d, w) = (\mathcal{C}, \Phi)$. $\mathcal{A}$ satisfies* noise-removal-invariance *if $\mathcal{A}(\mathcal{X} \setminus \Phi, d, w) = (\mathcal{C}, \emptyset)$.*

Removing points from a set has the same effect as reducing their weights to zero. Therefore, noise-removal-invariance can be viewed as an extreme case of noise-weight-removal.

Theorem 2.7 shows that the (k, g)-δ-truncated algorithm satisfies cluster-weight-scalability and noise-weight-scalability. Theorem 2.8 shows that the (k, g)-δ-truncated algorithm possesses



(a) Original Clustering

(b) Clustering after the noise is removed

Figure 2.4: (k, g)-δ-naive-truncated does not satify noise-removal-invariance

a much stronger property than noise-removal-invariance: Removing any subset of the noise cluster does not change the clustering on the rest of the data. Therefore, Corollary 2.1 shows that the (k, g)-δ-truncated algorithm satisfies noise-removal-invariance. On the other hand, the following examples show that the (k, g)-δ-naive-truncated algorithm does not satisfy noise-weight-scalability, cluster-weight-scalability, or noise-removal-invariance, when g is homogeneous.

Example 3. *For any δ and any homogeneous function g, let $\mathcal{A}$ be the $(1, g)$-δ-naive-truncated algorithm. Let $E = \{x, p, y, z\}$, $x = 0$, $p = \delta$, $y = 2\delta$, $z = 3\delta$, and distance measure d be the absolute value of the difference between the elements of E. Let r be the degree of homogeneity of g, i.e. $g(\alpha x) = \alpha^r g(x)$. Let $\mathcal{X} = \{x, y, z\}$ and $w(x) = 2^r$ and $w(y) = w(z) = 1$ (see Figure 2.4). Then the cost of different clusterings are as follows:*

$$w(x)g(\delta) + w(y)g(\delta) + w(z)g(2\delta) < w(x)g(2\delta) + g(\delta)$$
$$\Lambda^g_{\mathcal{X},d,w}(p) < \Lambda^g_{\mathcal{X},d,w}(y)$$
$$w(x)g(\delta) + w(y)g(\delta) + w(z)g(2\delta) < w(y)g(2\delta) + w(z)g(3\delta)$$
$$\Lambda^g_{\mathcal{X},d,w}(p) < \Lambda^g_{\mathcal{X},d,w}(x)$$

Therefore, $\mathcal{A}(\mathcal{X}, d, w) = (\{\{x, y\}\}, \{z\})$. Let $\alpha < \frac{1}{3^r - 2^r}$, and $w'(z) = \alpha w(z)$, $w'(x) = w(x)$, and $w'(y) = w(y)$. However,

$$w'(x)g(\delta) + w'(y)g(\delta) + w'(z)g(2\delta) > w'(y)g(2\delta) + w'(z)g(3\delta)$$
$$\Lambda^g_{\mathcal{X},d,w'}(p) > \Lambda^g_{\mathcal{X},d,w'}(x)$$

Therefore, $\mathcal{A}(\mathcal{X}, d, w') = (\{\{x\}\}, \{y, z\})$. Hence $\mathcal{A}$ does not satisfy noise-weight-scalability. $\mathcal{A}$ satisfies weight-scalability, using Lemma 2.1, $\mathcal{A}$ does not satisfy cluster-weight-scalability.

Example 4. *Let $\mathcal{X}$, d, and w be defined as in Example 3. Then, $\mathcal{A}(\mathcal{X}, d, w) = (\{\{x, y\}\}, \{z\})$. However,*

$$\begin{aligned} 2^r g(\delta) &< 2^r g(\delta) + g(\delta) \\ w(y)g(2\delta) &< w(x)g(\delta) + w(y)g(\delta) \\ \Lambda^g_{\mathcal{X}\setminus\{z\},d,w}(x) &< \Lambda^g_{\mathcal{X}\setminus\{z\},d,w}(p) \end{aligned}$$

So, $\mathcal{A}(\mathcal{X} \setminus \{z\}, d, w) = (\{\{x\}\}, \{y\})$. Hence, $\mathcal{A}$ does not satisfy noise-removal-invariance.

Using k well-separated copies of the above structures, we can generalize Examples 3 and 4 to hold for any k clusters, for $k \geq 1$.

Similarly, the (k, g)-η-trimmed algorithm does not satisfy cluster-weight-scalability, noise-weight-scalability, or noise-removal-invariance. The (k, g)-η-trimmed algorithm finds a clustering of a fixed signal-to-noise ratio. However, removing the noise points from a clustering changes this ratio, as a result, it changes the clustering. The next examples display the lack of cluster-weight-scalability, noise-weight-scalability and noise-removal-invariance in the (k, g)-η-trimmed algorithm for any $\eta > 0$.

Example 5. *For any g, $\eta > 0$ and an arbitrary $0 < \alpha \leq 1$, let $\mathcal{A}$ be the $(1, g)$-η-trimmed clustering. Let $\mathcal{X} = E$ be any set of $\frac{1}{\eta(1-\eta)+\alpha\eta^2-\alpha\eta}$ unique points in $\mathbb{R}^2$ and let d be the Euclidean distance between them. For any $x \in \mathcal{X}$, let $w(x) = 1$ and $w(\mathcal{X}) = \frac{1}{\eta(1-\eta)+\alpha\eta^2-\alpha\eta}$. Assume that $\mathcal{A}(\mathcal{X}, d, w) = (\{\mathcal{X}'\}, \mathcal{X} \setminus \mathcal{X}')$. Let $w'(x) = w(x)$ for $x \in \mathcal{X}'$, and $w'(x) = \alpha w(x)$ otherwise. For every $x \in \mathcal{X}'$,*

$$\frac{w'(x) + w'(\mathcal{X} \setminus \mathcal{X}')}{w'(\mathcal{X})} \leq \frac{w(x) + \alpha\eta w(\mathcal{X})}{(1-\eta)w(\mathcal{X}) + \alpha\eta w(\mathcal{X})} \leq \eta$$

Therefore, there exists $x \in \mathcal{X}'$, such that $\mathcal{A}(\mathcal{X}, d, w) = (\{\mathcal{X}' \setminus \{x\}\}, \{x\} \cup \mathcal{X} \setminus \mathcal{X}')$. Hence $\mathcal{A}$ does not satisfy noise-weight-scalability. Since, $\mathcal{A}$ satisfies weight-scalability, using Lemma 2.1, $\mathcal{A}$ does not satisfy cluster-weight-scalability.

Example 6. *Let $\mathcal{X}$, d, w, be as in Example 5, $|\mathcal{X}| = \frac{1}{\eta(1-\eta)}$, and $\mathcal{A}(\mathcal{X}, d, w) = (\{\mathcal{X}'\}, \mathcal{X} \setminus \mathcal{X}')$. Then for every $x \in \mathcal{X}'$,*

$$\frac{w(x)}{w(\mathcal{X}')} \leq \frac{w(x)}{(1-\eta)w(\mathcal{X})} \leq \eta$$

Therefore, there exists $x \in \mathcal{X}'$ that would be clustered as noise in $\mathcal{A}(\mathcal{X}', d, w)$. Hence $\mathcal{A}$ does not satisfy noise-removal-invariance.

Using k well-separated copies of the above structures, we can generalize Examples 5 and 6 to hold for any number of clusters, $k \geq 1$.

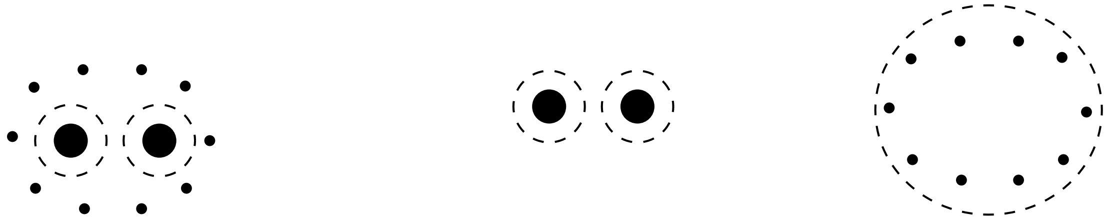
Corollaries 2.2 and 2.3 respectively show that $(\epsilon, minPts)$-DBScan satisfies noise-weight-scalability and noise-removal-invariance. On the other hand, Example 2 shows that for any ϵ and $minPts$, $(\epsilon, minPts)$-DBScan does not satisfy cluster-weight-scalability.

2.4.3 Properties Pertaining to the Distance of the Noise

In this section, we introduce properties that address an invariance in the output of clustering algorithms when the noise points are scattered in the space. Noise is often a structure-less set of data points. Moving the noise points farther from each other and the clustered data reduces the significance of any existing patterns in the noise cluster, hence, it decreases the effect of the noise points on the clustering. Therefore, it may be desirable that the outcome of a clustering algorithm with a noise cluster would remain invariant when such changes are applied to the noise.

Definition 2.11 (Noise-Scatter-Invariance)**.** *Let $\mathcal{A}$ be any clustering algorithm and for any $\mathcal{X}$ in space E, d_1, and w, let $\mathcal{A}(\mathcal{X}, d_1, w) = (\mathcal{C}, \Phi)$. $\mathcal{A}$ satisfies* noise-scatter-invariance *if for any d_2, such that $d_2(x, y) = d_1(x, y)$ for $x, y \in E \setminus \Phi$, and $d_2(x, y) \geq d_1(x, y)$ otherwise, $\mathcal{A}(\mathcal{X}, d_2, w) = \mathcal{A}(\mathcal{X}, d_1, w)$.*

Noise-scatter-invariance considers a setting where the distance between any two non-noise points remain the same, but other distance measurements can be increased in any manner. While (k, g)-δ-truncated, (k, g)-η-trimmed, and $(\epsilon, minPts)$-DBScan satisfy noise-scatter-invariance (see Theorems 2.10, 2.11, and 2.12, respectively), noise-scatter-invariance has some counter-intuitive consequences. For example, if the noise cluster is moved farther from the clustered data, while the distance between the noise points remain unchanged, the noise points will be well-separated from the clustered data and perhaps should be considered as a proper cluster (see Figure 2.5). To avoid this situation, we define a weaker property, *noise-dispersion-invariance*, which suggests that the outcome of clustering algorithms should remain unchanged when the distance measure involving every noise point is scaled uniformly.



(a) Original Clustering (b) Moving the noise cluster creates well-separated clusters

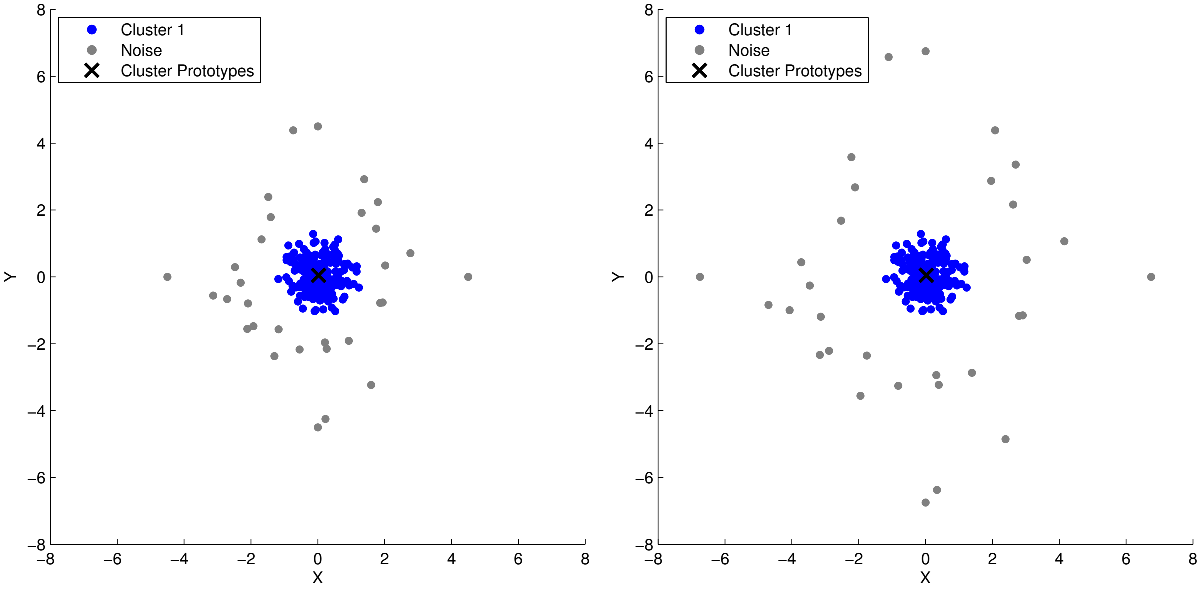
Figure 2.5: Noise-scatter-invariance is not a suitable criteria for evaluating clustering algorithms that have a noise cluster. The dotted circles demonstrate the clusters and the noise cluster is made of points that do not belong to any clusters.

Definition 2.12 (Noise-Dispersion-Invariance)**.** *Let $\mathcal{A}$ be any clustering algorithm and for any $\mathcal{X}$ in space E, d_1, and w, let $\mathcal{A}(\mathcal{X}, d_1, w) = (\mathcal{C}, \Phi)$. $\mathcal{A}$ satisfies* noise-dispersion-invariance *if for any $\alpha \geq 1$ and d_2, such that $d_2(x, y) = d_1(x, y)$ for $x, y \in E \setminus \Phi$, and $d_2(x, y) = \alpha d_1(x, y)$ otherwise, $\mathcal{A}(\mathcal{X}, d_2, w) = \mathcal{A}(\mathcal{X}, d_1, w)$.*

Since noise-dispersion-invariance is implied by noise-scatter-invariance, (k, g)-δ-truncated, (k, g)-η-trimmed, and $(\epsilon, minPts)$-DBScan satisfy noise-dispersion-invariance. On the other hand, the (k, g)-δ-naive-truncated algorithm does not satisfy noise-dispersion-invariance. The centers of the optimal (k, g)-δ-naive-truncated clustering are calculated before the noise points are identified. Therefore, the position of noise points affects the structure of the clustering. The following example displays the lack of noise-dispersion-invariance for any (k, g)-δ-truncated algorithm, when g is homogeneous. Note that many of the commonly used function are homogeneous, e.g. $g(x) = x$ and $g(x) = x^2$.

Example 7. *For any δ and any homogeneous function g, let $\mathcal{A}$ be the $(1, g)$-δ-naive-truncated algorithm. Let $E = \{x, p, q, y\}$, $x = 0$, $p = \delta$, $q = 2\delta$, $y = 3\delta$, and distance measure d be the absolute value of the difference between the elements of E. Let r be the degree of homogeneity of g. Let $\mathcal{X} = \{x, y\}$ and let $3^r - 2^r > \frac{w(x)}{w(y)}$.*

$$w(x)g(\delta) + w(y)g(2\delta) < w(x)g(2\delta) + w(y)g(\delta)$$
$$\Lambda^g_{\mathcal{X},d_1,w}(p) < \Lambda^g_{\mathcal{X},d_1,w}(q)$$



(a) The original clustering

(b) Noise cluster is scattered

Figure 2.6: An example demonstrating noise-scatter-invariance of (k, g)-δ-truncated

$$\begin{aligned} w(x)g(\delta) + w(y)g(2\delta) &< w(y)g(3\delta) \\ \Lambda^g_{\mathcal{X},d_1,w}(p) &< \Lambda^g_{\mathcal{X},d_1,w}(x) \end{aligned}$$

Therefore, $\mathcal{A}(\mathcal{X}, d_1, w) = (\{\{x\}\}, \{y\})$*. Let* $\alpha > \sqrt[r]{\frac{w(x)}{w(y)}}$*. Then*

$$\begin{aligned} w(x)g(\delta) + w(y)g(\alpha \cdot 2\delta) &> w(x)g(2\delta) + w(y)g(\alpha \cdot \delta) \\ \Lambda^g_{\mathcal{X},d_2,w}(p) &> \Lambda^g_{\mathcal{X},d_2,w}(q) \\ w(y)g(\alpha \cdot 3\delta) &> w(x)g(2\delta) + w(y)g(\alpha \cdot \delta) \\ \Lambda^g_{\mathcal{X},d_2,w}(x) &> \Lambda^g_{\mathcal{X},d_2,w}(q) \end{aligned}$$

Therefore, $\mathcal{A}(\mathcal{X}, d_2, w) = (\{\{\}\}, \{x, y\})$*. Hence* $\mathcal{A}$ *does not satisfy noise-dispersion-invariance.*

Using k well-separated copies of the above structure, we can generalize Example 7 to hold for any number of clusters, $k \geq 1$.

2.4.4 Properties Pertaining to the Range of an Algorithm

In this section, we propose properties that examine the richness of the range of clustering algorithms. Let $\mathcal{A}$ be a clustering algorithm that ignores its input and returns a constant clustering. Then $\mathcal{A}$ satisfies all the scalability and invariance properties discussed in the previous sections. However, the range of $\mathcal{A}$ is not rich enough to represent the structure of any data set. To avoid this problem, we expect the range of a desirable clustering algorithm to include clusterings where each point is clustered as noise and as part of a (traditional) cluster. We introduce two properties, *cluster-richness* and *noise-richness*. In the former, given $\mathcal{X}$ and d, for any point x^*, we examine the existence of a weight function that results in x^* being clustered. In the latter, we examine the existence of a weight function that leads to x^* being assigned to the noise cluster.

Definition 2.13 (Cluster-Richness)**.** *A clustering algorithm* $\mathcal{A}$ *satisfies* cluster-richness *if for any* $\mathcal{X}$ *and* d*, and any* $x \in \mathcal{X}$*, there exists a weight function* w*, such that* $0 < w(x) < w(\mathcal{X})$ *and if* $\mathcal{A}(\mathcal{X}, d, w) = (\mathcal{C}, \Phi)$*, then* $x \in \bigcup \mathcal{C}$*.*

Theorems 2.13, 2.14, 2.15, and 2.16, show that all the clustering algorithms used in this work satisfy cluster-richness.

Definition 2.14 (Noise-Richness). *A clustering algorithm $\mathcal{A}$ satisfies* noise-richness *if for any $\mathcal{X}$ and d, and any $x \in \mathcal{X}$, there exists a weight function w, such that $0 < w(x) < w(\mathcal{X})$ and if $\mathcal{A}(\mathcal{X}, d, w) = (\mathcal{C}, \Phi)$, then $x \in \Phi$.*

Theorems 2.17 and 2.18 show that the (k, g)-η-trimmed and $(\epsilon, minPts)$-DBScan algorithms satisfy noise-richness. On the other hand, the (k, g)-δ-truncated and (k, g)-δ-naive-truncated algorithms do not satisfy noise-richness. This is due to the fact that in these algorithms the distance between noise points and potential cluster centers is bounded below by δ. Therefore, in a space where all points are close to potential cluster centers, the (k, g)-δ-truncated and (k, g)-δ-naive-truncated algorithms can not find any noise. Example 8 demonstrates this lack of noise-richness in the (k, g)-δ-truncated and (k, g)-δ-naive-truncated algorithms.

Example 8. *For any k, g, and δ, let $\mathcal{A}$ be the (k, g)-δ-truncated or (k, g)-δ-naive-truncated algorithm. Let $\mathcal{X}$ and d be such that $diam(\mathcal{X}) < \delta$. Since, the center of any cluster is within its convex hull, for any center of the optimal clustering, μ, and for any $x \in \mathcal{X}$, $d(\mu, x) \leq diam(\mathcal{X}) < \delta$. Hence, x is not noise. Therefore, $\mathcal{A}$ does not satisfy noise-richness.*

2.4.5 Computational Feasibility of Clustering Algorithms

An important aspect of a clustering algorithm with a noise cluster is its computational feasibility for clustering large amount of data. However, even in the setting of clustering without a noise cluster, many objective-based clustering costs are NP-hard to optimize, e.g. k-means [7]. We are interested in drawing a comparison between the clustering algorithms with a noise cluster and comparing their efficiency to that of clustering algorithms without a noise cluster. Hence, examining their absolute computational complexity does not provide an informative comparison. Here, we examine the efficiency of clustering algorithms assuming we have access to an oracle that computes a (k, g)-centroid clustering.

Given $\mathcal{X}$, d, and w, the (k, g)-δ-truncated algorithm optimizes the value of $\Lambda^g_{\mathcal{X},d',w}$. This is equivalent to optimizing the (k, g)-centroid objective function for input $\mathcal{X}$, d', and w, where

$d'(x, y) = \min\{\delta, d(x, y)\}$. Similarly, the (k, g)-δ-naive-truncated algorithm for $\mathcal{X}$, d, and w makes a single call to the (k, g)-centroid oracle with input $\mathcal{X}$, d, and w, and then removes points that are farther than δ from their corresponding centers. Therefore, the (k, g)-δ-truncated and (k, g)-δ-naive-truncated algorithms each make a single call to the (k, g)-centroid oracle, so they are as efficient as our oracle. On the other hand, the (k, g)-η-trimmed algorithm optimizes the (k, g)-centroid cost over all $\mathcal{X}' \subseteq \mathcal{X}$, such that $w(\mathcal{X}) \geq (1 - \eta)w(\mathcal{X}')$. The brute-force approach for minimizing this objective function makes one call for each such $\mathcal{X}'$, making a total of $O(|\mathcal{X}|^{(1-\eta)|\mathcal{X}|})$ calls to our oracle. We are not aware of any algorithm that optimizes this objective function using a number of calls to our oracle that is polynomial with respect to $|\mathcal{X}|$. Therefore, we consider (k, g)-η-trimmed to be computationally infeasible compared to our oracle.

DBScan is not an objective-based algorithm, therefore, we can not use the above efficiency model. In settings where region queries (for example computing an ϵ-neighbourhood) are done quickly, $(\epsilon, minPts)$-DBScan is efficient. Moreover, experiments on real and synthetic data have demonstrated the efficiency of $(\epsilon, minPts)$-DBScan for clustering large amount of data [11].

2.4.6 Proofs of Results Pertaining to the Properties

Lemma 2.2. *For any $\mathcal{X}$, distance measure d, and weight function w, $\alpha > 0$, and $\mu_1, \ldots, \mu_k$,*

$$\Lambda^g_{\mathcal{X},d,\alpha w}(\mu_1, \ldots, \mu_k) = \alpha\Lambda^g_{\mathcal{X},d,w}(\mu_1, \ldots, \mu_k)$$

Proof. Using the definition of $\Lambda^g_{\mathcal{X},d,\alpha w}(\mu_1, \ldots, \mu_k)$,

$$\begin{aligned}\Lambda^g_{\mathcal{X},d,\alpha w}(\mu_1, \ldots, \mu_k) &= \sum_{x\in\mathcal{X}} \alpha w(x) \min_{i\in[k]}\{g(d(x, \mu_i))\} \\ &= \alpha \sum_{x\in\mathcal{X}} w(x) \min_{i\in[k]}\{g(d(x, \mu_i))\} \\ &= \alpha\Lambda^g_{\mathcal{X},d,w}(\mu_1, \ldots, \mu_k)\end{aligned}$$

□

Theorem 2.3. *For any k, function g, and δ, the (k, g)-δ-truncated algorithm satisfies weight-scalability.*

Proof. Let $\mathcal{A}$ be the (k, g)-δ-truncated algorithm and $\mathcal{A}(\mathcal{X}, d, \alpha w) = (\mathcal{C}, \Phi)$. Let $\mu_1^*, \ldots, \mu_k^*$ be the centers of $(\mathcal{C}, \Phi)$. For any $\mu_1', \ldots, \mu_k'$, we have

$$\begin{aligned} \Lambda^g_{\mathcal{X},d,\alpha w}(\mu_1^*, \ldots, \mu_k^*) &= \alpha \Lambda^g_{\mathcal{X},d',w}(\mu_1^*, \ldots, \mu_k^*) && (2.2) \\ &\leq \alpha \Lambda^g_{\mathcal{X},d',w}(\mu_1', \ldots, \mu_k') \\ &\leq \Lambda^g_{\mathcal{X},d',\alpha w}(\mu_1', \ldots, \mu_k') && (2.3) \end{aligned}$$

where, Equation 2.2 and 2.3 hold using Lemma 2.2 with distance function d'. So, $\mathcal{A}(\mathcal{X}, d, \alpha w) = (\mathcal{C}, \Phi)$. Therefore, $\mathcal{A}$ satisfies weight-scalability. □

Theorem 2.4. *For any k, function g, and δ, the (k, g)-δ-naive-truncated algorithm satisfies weight-scalability.*

Proof. Let $\mathcal{A}$ be the (k, g)-δ-naive-truncated algorithm, and $\mathcal{A}(\mathcal{X}, d, \alpha w) = (\mathcal{C}, \Phi)$ with centers $\mu_1^*, \ldots, \mu_k^*$. For any $\mu_1', \ldots, \mu_k'$, we have

$$\begin{aligned} \Lambda^g_{\mathcal{X},d,\alpha w}(\mu_1^*, \ldots, \mu_k^*) &= \alpha \Lambda^g_{\mathcal{X},d,w}(\mu_1^*, \ldots, \mu_k^*) \\ &\leq \alpha \Lambda^g_{\mathcal{X},d,w}(\mu_1', \ldots, \mu_k') \\ &\leq \Lambda^g_{\mathcal{X},d,\alpha w}(\mu_1', \ldots, \mu_k') \end{aligned}$$

Hence, $\mathcal{A}(\mathcal{X}, d, \alpha w) = (\mathcal{C}, \Phi)$. Therefore, $\mathcal{A}$ satisfies weight-scalability. □

Theorem 2.5. *For any k, objective function g, and $\eta \leq 1$, the (k, g)-η-trimmed algorithm satisfies weight-scalability.*

Proof. Let $\mathcal{A}$ be the (k, g)-η-trimmed algorithm. For any $\mathcal{X}$ and w_1, let $\mathcal{X}^* \subseteq \mathcal{X}$ and $\mu_1^*, \ldots, \mu_k^*$ be defined by the optimal clustering of $\mathcal{A}(\mathcal{X}, d, w_1) = (\mathcal{C}, \Phi)$. In other words,

$$(\mathcal{X}^*, (\mu_1^*, \ldots, \mu_k^*) : \underset{\substack{\mathcal{X}' \subseteq \mathcal{X} : w_1(\mathcal{X}') \geq (1-\eta) w_1(\mathcal{X}) \\ \mu_1, \ldots, \mu_k \in E}}{\arg\min} \Lambda^g_{\mathcal{X}',d,w_1}(\mu_1, \ldots, \mu_k)$$

For any $\alpha > 0$, let $w_2 = \alpha w_1$. For any $\mathcal{X}' \subseteq \mathcal{X}$, if $w_2(\mathcal{X}') \geq (1 - \eta) w_2(\mathcal{X})$, then $w_1(\mathcal{X}') \geq$

$(1-\eta)w_1(\mathcal{X})$. Hence, for any $\mathcal{X}' \subseteq \mathcal{X}$, such that $w_2(\mathcal{X}') \geq (1-\eta)w_2(\mathcal{X})$, and any $\mu_1', \ldots, \mu_k'$.

$$\begin{aligned}
\Lambda^g_{\mathcal{X}^*,d,w_2}(\mu_1^*, \ldots, \mu_k^*) &= \alpha\Lambda^g_{\mathcal{X}^*,d,w_1}(\mu_1^*, \ldots, \mu_k^*) \\
&\leq \alpha\Lambda^g_{\mathcal{X}',d,w_1}(\mu_1', \ldots, \mu_k') \\
&\leq \alpha\Lambda^g_{\mathcal{X}',d,\frac{1}{\alpha}w_2}(\mu_1', \ldots, \mu_k') \\
&\leq \Lambda^g_{\mathcal{X}',d,w_2}(\mu_1', \ldots, \mu_k')
\end{aligned}$$

Therefore, $\mathcal{A}(\mathcal{X}, d, w_2) = (\mathcal{C}, \Phi)$. So $\mathcal{A}$ satisfies weight-scalability. □

Theorem 2.6. *For any k, homogeneous function g, and $\eta \leq 1$, the (k, g)-η-trimmed algorithm satisfies distance-scalability.*

Proof. Let p be the degree of the homogeneous function g, i.e. $g(\alpha x) = a^p g(x)$. Let $\mathcal{A}$ be the (k, g)-η-trimmed algorithm. For any $\mathcal{X}$, d, and w, let $\mathcal{X}^* \subseteq \mathcal{X}$ and $\mu_1^*, \ldots, \mu_k^*$ be the centers of the optimal clustering of $\mathcal{A}(\mathcal{X}, d, w) = (\mathcal{C}, \Phi)$. For any $\mathcal{X}' \subseteq \mathcal{X}$, such that $w(\mathcal{X}') \geq (1-\eta)w(\mathcal{X})$, any $\mu_1', \ldots, \mu_k'$, and $\alpha > 0$,

$$\begin{aligned}
\Lambda^g_{\mathcal{X}^*,\alpha d,w}(\mu_1^*, \ldots, \mu_k^*) &= \sum_{x\in\mathcal{X}} w(x) \min_{i\in[k]}\{g(\alpha d(x, \mu_i^*))\} \\
&= \alpha^p \sum_{x\in\mathcal{X}} w(x) \min_{i\in[k]}\{g(d(x, \mu_i^*))\} \\
&= \alpha^p \sum_{x\in\mathcal{X}} w(x) \min_{i\in[k]}\{g(d(x, \mu_i'))\} \\
&\leq \sum_{x\in\mathcal{X}} w(x) \min_{i\in[k]}\{g(\alpha d(x, \mu_i'))\} \\
&\leq \Lambda^g_{\mathcal{X},\alpha d,w}(\mu_1', \ldots, \mu_k')
\end{aligned}$$

Therefore, $\mathcal{A}(\mathcal{X}, \alpha d, w) = (\mathcal{C}, \Phi)$. So $\mathcal{A}$ satisfies distance-scalability. □

Theorem 2.7. *For any k, function g, and δ, the (k, g)-δ-truncated algorithm satisfies cluster-weight-scalability and noise-weight-scalability.*

Proof. Let $\mathcal{A}$ be the (k, g)-δ-truncated algorithm. For any $\mathcal{X}$, d, and w_1, let $\mathcal{A}(\mathcal{X}, d, w_1) = (\mathcal{C}, \Phi)$. For any $0 < \alpha \leq 1$, let $w_2(x) = \alpha w_1(x)$ if $x \in \Phi$, and $w_2(x) = w_1(x)$, otherwise. For any clustering $(\mathcal{C}', \Phi')$, let $\mu'(x)$ be the closest center of $\mathcal{C}$ to x.

$$\begin{align}
\Lambda^g_{d',w_2}(\mathcal{C}', \Phi') \qquad &= \sum_{x\in\mathcal{X}\setminus\Phi} w_2(x)g(d'(x,\mu'(x))) + \sum_{x\in\Phi} w_2(x)g(d'(x,\mu'(x))) \notag\\
&= \sum_{x\in\mathcal{X}\setminus\Phi} w_1(x)g(d'(x,\mu'(x))) + \sum_{x\in\Phi} \alpha w_1(x)g(d'(x,\mu'(x))) \notag\\
&\quad + \Lambda^g_{d',w_2}(\mathcal{C},\Phi) - \Lambda^g_{d',w_1}(\mathcal{C},\emptyset) - \alpha w_1(\Phi)g(\delta) \tag{2.4}\\
&= \sum_{x\in\mathcal{X}\setminus\Phi} w_1(x)g(d'(x,\mu'(x))) - \Lambda^g_{d',w_2}(\mathcal{C},\emptyset) \notag\\
&\quad + \alpha\left(\sum_{x\in\Phi} w_1(x)g(d'(x,\mu'(x))) - w_1(\Phi)g(\delta)\right) \tag{2.5}\\
&\quad + \Lambda^g_{d',w_2}(\mathcal{C},\Phi) \notag\\
&\geq \sum_{x\in\mathcal{X}\setminus\Phi} w_1(x)g(d'(x,\mu'(x))) - \Lambda^g_{d',w_1}(\mathcal{C},\emptyset) \notag\\
&\quad + \sum_{x\in\Phi} w_1(x)g(d'(x,\mu'(x))) - w_1(\Phi)g(\delta) + \Lambda^g_{d',w_2}(\mathcal{C},\Phi) \notag\\
&\geq \left(\sum_{x\in\mathcal{X}} w_1(x)g(d'(x,\mu'(x))) - \Lambda^g_{d',w_1}(\mathcal{C},\Phi)\right) + \Lambda^g_{d',w_2}(\mathcal{C},\Phi) \tag{2.6}\\
&\geq \Lambda^g_{d',w_2}(\mathcal{C},\Phi) \tag{2.7}
\end{align}$$

Where Equation 2.4 holds from the definition of Λ^g_{d',w_2} (see Equation 2.1). Since, $d'(x,y) \leq \delta$ for all $x, y \in \mathcal{X}$, the bracketed value in Equation 2.5 is non-positive, therefore, the inequality is obtained through division by $0 < \alpha \leq 1$. Equation 2.6 holds by the optimality of $(\mathcal{C},\Phi)$ for $\mathcal{A}(\mathcal{X}, d, w_1)$. Equation 2.7 shows that $\mathcal{A}(\mathcal{X}, d, w_2) = (\mathcal{C},\Phi)$. Therefore, $\mathcal{A}$ satisfies noise-weight-scalability. Since $\mathcal{A}$ satisfies weight-scalability (see Theorem 2.3), Lemma 2.1 shows that $\mathcal{A}$ also satisfies cluster-weight-scalability. □

Theorem 2.8. *For any k, function g, and δ, let $\mathcal{A}$ be the (k,g)-δ-truncated algorithm and $\mathcal{A}(\mathcal{X}, d, w) = (\mathcal{C}, \Phi)$. For any $\bar{\Phi} \subseteq \Phi$, $\mathcal{A}(\mathcal{X} \setminus \bar{\Phi}, d, w) = (\mathcal{C}, \Phi \setminus \bar{\Phi})$.*

Proof. Assume on the contrary that $\mathcal{A}(\mathcal{X}\backslash\bar{\Phi}, d, w) = (\mathcal{C}', \Phi')$ and $\Lambda^g_{d',w}(\mathcal{C}', \Phi') < \Lambda^g_{d',w}(\mathcal{C}, \Phi\backslash\bar{\Phi})$.

$$\begin{aligned}\Lambda^g_{d',w}(\mathcal{C}', \Phi' \cup \bar{\Phi}) &= \Lambda^g_{d',w}(\mathcal{C}', \Phi') + g(\delta)w(\bar{\Phi}) \\ &< \Lambda^g_{d',w}(\mathcal{C}, \Phi \setminus \bar{\Phi}) + g(\delta)w(\bar{\Phi}) \\ &< \Lambda^g_{d',w}(\mathcal{C}, \Phi)\end{aligned}$$

This forms a contradiction. Therefore, $\mathcal{A}(\mathcal{X} \setminus \bar{\Phi}, d, w) = (\mathcal{C}, \Phi \setminus \bar{\Phi})$. □

Corollary 2.1. *For any k, function g, and δ, the (k, g)-δ-truncated algorithm satisfies noise-removal-invariance.*

Proof. Directly from Theorem 2.8 for $\bar{\Phi} = \Phi$. □

Theorem 2.9. *For any ϵ and $minPts$, let $\mathcal{A}$ be $(\epsilon, minPts)$-DBScan and for any $\mathcal{X}$, d, and w_1, let $\mathcal{A}(\mathcal{X}, d, w_1) = (\mathcal{C}, \Phi)$. For any $0 \leq \alpha \leq 1$ and w_2, such that $w_2(x) = \alpha w_1(x)$ if $x \in \Phi$, and $w_2(x) = w_1(x)$, otherwise, and for any $x, y \in \mathcal{X}$, x is density-reachable from y in $\mathcal{X}$ with weights w_1 if and only if it is density-reachable from y in $\mathcal{X}$ with weights w_2.*

Proof. Assume x is density-reachable from y in $\mathcal{X}$ with weights w_1 with respect to ϵ and $minPts$. There is a path $y = p_0, p_1, \ldots, p_n = x$, such that for all $i < n$, $w_1(N_\epsilon(p_i)) \geq minPts$ and for all $i > 0$, $p_i \in N_\epsilon(p_{i-1})$. For any p_i such that $w_1(N_\epsilon(p_i)) \geq minPts$, $N_\epsilon(p_i) \subseteq \bigcup\mathcal{C}$, so for all $q \in N_\epsilon(p_i)$, $w_2(q) = w_1(q)$. So, for all $i > 0$, $w_2(N_\epsilon(p_i)) \geq minPts$. Therefore, x is density-reachable from y in $\mathcal{X}$ with weights w_2 with respect to ϵ and $minPts$.

For all $x \in \mathcal{X}$, $w_2(x) \leq w_1(x)$, therefore, $w_2(N_\epsilon(x)) \leq w_1(N_\epsilon(x))$. Assume x is density-reachable from y in $\mathcal{X}$ with weights w_2 with respect to ϵ and $minPts$. There is a path $y = p_0, p_1, \ldots, p_n = x$, such that for all $i < n$, $w_1(N_\epsilon(p_i)) \geq w_2(N_\epsilon(p_i)) \geq minPts$ and for all $i > 0$, $p_i \in N_\epsilon(p_{i-1})$. Therefore, x is density-reachable from y in $\mathcal{X}$ with weights w_1 with respect to ϵ and $minPts$. □

Corollary 2.2. *For any ϵ and $minPts$, $(\epsilon, minPts)$-DBScan satisfies noise-weight-scalability.*

Proof. For any $\mathcal{X}$, d, and w_1, let $\mathcal{A}(\mathcal{X}, d, w_1) = (\mathcal{C}, \Phi)$. For any $\alpha \leq 1$ and w_2, such that $w_2(x) = \alpha w_1(x)$ if $x \in \Phi$, and $w_2(x) = w_1(x)$ otherwise, by Theorem 2.9, for any $x, y \in \mathcal{X}$, x is density-reachable from y in $\mathcal{X}$ and w_1 if and only if it is density-reachable from y in $\mathcal{X}$ and w_2. Therefore, $\mathcal{A}(\mathcal{X}, d, w_1) = \mathcal{A}(\mathcal{X}, d, w_2)$. □

Corollary 2.3. *For any ϵ and $minPts$, $(\epsilon, minPts)$-DBScan satisfies noise-removal-invariance.*

Proof. Similar to Corollary 2.2 with $\alpha = 0$. □

Lemma 2.3. *For any k, function g, set $\mathcal{X}$, distance functions d_1 and d_2, and weight function w, such that for all $x, y \in \mathcal{X}$, $d_1(x, y) \le d_2(x, y)$, and any $\mu_1, \dots, \mu_k$,*

$$\Lambda^g_{\mathcal{X},d_1,w}(\mu_1, \dots, \mu_k) \le \Lambda^g_{\mathcal{X},d_2,w}(\mu_1, \dots, \mu_k)$$

Proof. Using the definition of $\Lambda^g_{\mathcal{X},d_1,w}(\mu_1, \dots, \mu_k)$,

$$\begin{aligned}\Lambda^g_{\mathcal{X},d_1,w}(\mu_1, \dots, \mu_k) &= \sum_{x\in\mathcal{X}} w(x) \min_{i\in[k]} g(d_1(x, \mu_i)) \\ &\le \sum_{x\in\mathcal{X}} w(x) \min_{i\in[k]} g(d_2(x, \mu_i)) \\ &\le \Lambda^g_{\mathcal{X},d_2,w}(\mu_1, \dots, \mu_k)\end{aligned}$$

□

Theorem 2.10. *For any k, function g, and δ, the (k, g)-δ-truncated algorithm satisfies noise-scatter-invariance.*

Proof. Let $\mathcal{A}$ be the (k, g)-truncated algorithm and for $\mathcal{X} \subseteq E$, d_1, and w, let $\mathcal{A}(\mathcal{X}, d_1, w) = (\mathcal{C}, \Phi)$ with centers $\mu^*_1, \dots, \mu^*_k$. Let d_2 be such that $d_2(x, y) = d_1(x, y)$ if $x, y \in E \setminus \Phi$, and $d_2(x, y) \ge d_1(x, y)$ otherwise. For any $\mu'_1, \dots, \mu'_k$,

$$\begin{aligned}\Lambda^g_{\mathcal{X},d'_2,w}(\mu^*_1, \dots, \mu^*_k) &= \sum_{x\notin\Phi} w(x) \min_{i\in[k]} g(d'_2(x, \mu^*_i)) + w(\Phi)g(\delta) \\ &= \sum_{x\notin\Phi} w(x) \min_{i\in[k]} g(d'_1(x, \mu^*_i)) + w(\Phi)g(\delta) \\ &\le \Lambda^g_{\mathcal{X},d'_1,w}(\mu'_1, \dots, \mu'_k) \\ &\le \Lambda^g_{\mathcal{X},d'_2,w}(\mu'_1, \dots, \mu'_k)\end{aligned}$$

Where the last inequality holds by Lemma 2.3. Hence, $\mathcal{A}(\mathcal{X}, d_2, w) = (\mathcal{C}, \Phi)$. Therefore, $\mathcal{A}$ satisfies noise-scatter-invariance. □

Theorem 2.11. *For any k, function g, and $\eta \leq 1$, the (k, g)-η-trimmed algorithm satisfies noise-scatter-invariance.*

Proof. Let $\mathcal{A}$ be the (k, g)-η-trimmed algorithm. For any $\mathcal{X} \subseteq E$, w, and d, let $\mathcal{X}^* \subseteq \mathcal{X}$ and $\mu_1^*, \ldots, \mu_k^*$ be defined by the optimal clustering of $\mathcal{A}(\mathcal{X}, d_1, w) = (\mathcal{C}, \Phi)$. In other words,

$$(\mathcal{X}^*, (\mu_1^*, \ldots, \mu_k^*)) = \underset{\substack{\mathcal{X}' \subseteq \mathcal{X}: w_1(\mathcal{X}') \geq (1-\eta) w_1(\mathcal{X}) \\ \mu_1, \ldots, \mu_k \in E}}{\arg\min} \Lambda^g_{\mathcal{X}', d, w_1}(\mu_1, \ldots, \mu_k)$$

Let d_2 be such that $d_2(x, y) = d_1(x, y)$ if $x, y \in \mathcal{X}^*$, and $d_2(x, y) \geq d_1(x, y)$ otherwise. For any $\mu'_1, \ldots, \mu'_k$ and $\mathcal{X}'$, such that $w(\mathcal{X}') \geq (1 - \eta) w(\mathcal{X})$,

$$\begin{aligned}
\Lambda^g_{\mathcal{X}^*, d_2, w}(\mu_1^*, \ldots, \mu_k^*) &= \sum_{x \in \mathcal{X}^*} w(x) \min_{i \in [k]} g(d_2(x, \mu_i^*)) \\
&= \sum_{x \in \mathcal{X}^*} w(x) \min_{i \in [k]} g(d_1(x, \mu_i^*)) \\
&\leq \Lambda^g_{\mathcal{X}^*, d_1, w}(\mu_1^*, \ldots, \mu_k^*) \\
&\leq \Lambda^g_{\mathcal{X}', d_1, w}(\mu'_1, \ldots, \mu'_k) \\
&\leq \Lambda^g_{\mathcal{X}', d_2, w}(\mu'_1, \ldots, \mu'_k)
\end{aligned}$$

Where the last inequality holds by Lemma 2.3. Hence, $\mathcal{A}(\mathcal{X}, d_2, w) = (\mathcal{C}, \Phi)$. Therefore, $\mathcal{A}$ satisfies noise-scatter-invariance. □

Theorem 2.12. *For any ϵ and $minPts$, $(\epsilon, minPts)$-DBScan satisfies noise-scatter-invariance.*

Proof. Let $\mathcal{A}$ be $(\epsilon, minPts)$-DBScan and $\mathcal{A}(\mathcal{X}, d_1, w) = (\mathcal{C}, \Phi)$. Let d_2 be any function such that $d_2(x, y) \geq d_1(x, y)$ for $x, y \in \Phi$, and $d_2(x, y) = d_1(x, y)$, otherwise. We will show that for any $x, y \in \mathcal{X}$, x is density-reachable from y in $\mathcal{X}$, d_1, and w, if and only if it is density-reachable from y in $\mathcal{X}$, d_2, and w.

Let $N^1_\epsilon(x)$ and $N^2_\epsilon(x)$ denote the ϵ-neighbourhood of x in $(\mathcal{X}, d_1, w)$ and $(\mathcal{X}, d_2, w)$, respectively. Assume x is density-reachable from y in $(\mathcal{X}, d_1, w)$ with respect to ϵ and $minPts$. There is a path $y = p_0, p_1, \ldots, p_n = x$, such that for all $i < n$, $w(N^1_\epsilon(p_i)) \geq minPts$ and for all $i > 0$, $p_i \in N^2_\epsilon(p_{i-1})$. For any p_i such that $w(N^1_\epsilon(p_i)) \geq minPts$, $N^1_\epsilon(p_i) \subseteq \bigcup \mathcal{C}$, hence, for all

$i > 0$, $w(N^2_\epsilon(p_i)) = w(N^1_\epsilon(p_i)) \geq minPts$. Therefore, x is density-reachable from y using d_2 with respect to ϵ and $minPts$.

For all $x \in \mathcal{X}$, $d_2(x, y) \geq d_1(x, y)$, therefore, $N^2_\epsilon(x) \subseteq N^1_\epsilon(x)$. Assume x is density-reachable from y in $\mathcal{X}$, d_2, and w with respect to ϵ and $minPts$. There is a path $y = p_0, p_1, \ldots, p_n = x$, such that for all $i < n$, $w(N^1_\epsilon(p_i)) \geq w(N^2_\epsilon(p_i)) \geq minPts$ and for all $i > 0$, $p_i \in N^2_\epsilon(p_{i-1}) \subseteq N^1_\epsilon(p_{i-1})$. Therefore, x is density-reachable from y in $\mathcal{X}$, d and w with respect to ϵ and $minPts$.

Since for any $x, y \in \mathcal{X}$ the density-reachability does not change, the clustering stays the same. Therefore, $\mathcal{A}$ satisfies noise-scatter-invariance. □

Theorem 2.13. *For any $k > 1$, function g, and δ, the (k, g)-δ-truncated algorithm satisfies cluster-richness.*

Proof. For any k, function g, and δ, let $\mathcal{A}$ be the (k, g)-δ-truncated algorithm. Choose an arbitrary $\epsilon < \frac{1}{2}$, and for a given $x^* \in \mathcal{X}$ and d, choose w such that $w(x^*) = \frac{1}{2} + \epsilon$ and for $x \neq x^*$, $w(x) = \frac{1/2-\epsilon}{|X|-1}$. Let $\mathcal{A}(\mathcal{X}, d, w) = (\mathcal{C}, \Phi)$ and let $(\mathcal{C}', \Phi')$ be any clustering with one center on x^*. Assume on the contrary that $x^* \in \Phi$.

$$\Lambda^g_{\mathcal{X},d',w}(\mathcal{C}, \Phi) \geq w(x^*)g(\delta) > w(\mathcal{X} \setminus \{x^*\})g(\delta) \geq \Lambda^g_{\mathcal{X},d',w}(\mathcal{C}', \Phi')$$

Contradiction, hence, $x^* \in \bigcup \mathcal{C}$ and $\mathcal{A}$ satisfies cluster-richness. □

Theorem 2.14. *For any k, function g, and δ, the (k, g)-δ-naive-truncated algorithm satisfies cluster-richness.*

Proof. For any k, function g, and δ, let $\mathcal{A}$ be the (k, g)-δ-naive-truncated algorithm. For a given $x^* \in \mathcal{X}$ and d, choose w such that $w(x^*) = \frac{\sum_{x \neq x^*} g(d(x,x^*))}{g(\delta)} + 1$, and $w(x) = 1$, for $x \neq x^*$. Let $\mathcal{A}(\mathcal{X}, d, w) = (\mathcal{C}, \Phi)$ and let $(\mathcal{C}', \Phi')$ be any clustering with one center on x^*. Assume on the contrary that $x^* \in \Phi$.

$$\begin{aligned}
\Lambda^g_{d,w}(\mathcal{C}, \Phi) &\geq w(x^*)g(\delta) \\
&> \sum_{x \neq x^*} g(d(x, x^*)) \\
&> \Lambda^g_{\mathcal{X},d,w}(x^*, \mu_2, \ldots, \mu_k) \\
&> \Lambda^g_{d,w}(\mathcal{C}', \Phi')
\end{aligned}$$

Contradiction, hence, $x^* \in \bigcup \mathcal{C}$ and $\mathcal{A}$ satisfies cluster-richness. $\square$

Theorem 2.15. *For any k, function g, and $\eta < 1$, the (k, g)-η-trimmed algorithm satisfies cluster-richness.*

Proof. For any k, function g, and η, let $\mathcal{A}$ be the (k, g)-η-trimmed algorithm. Given $x^* \in \mathcal{X}$, choose w such that $w(x^*) = \frac{\eta(|\mathcal{X}|-1)}{1-\eta} + 1$ and $w(x) = 1$ for $x \neq x^*$. Since $w(x^*) > \eta w(\mathcal{X})$, x^* can not be clustered as noise. Therefore, $\mathcal{A}$ satisfies cluster-richness. $\square$

Theorem 2.16. *For any ϵ and $minPts$, $(\epsilon, minPts)$-DBScan satisfies cluster-richness.*

Proof. For any ϵ and $minPts$, let $\mathcal{A}$ be $(\epsilon, minPts)$-DBScan. Given $x^* \in \mathcal{X}$, choose w such that $w(x^*) = minPts$. Since $w(N_\epsilon(x^*)) \geq minPts$, x^* is clustered. Therefore, $\mathcal{A}$ satisfies cluster-richness. $\square$

Theorem 2.17. *For any k, function g, and $\eta > 0$, the (k, g)-η-trimmed algorithm satisfies noise-richness.*

Proof. For any k, g, and $\eta < 1$, let $\mathcal{A}$ be the (k, g)-η-trimmed algorithm. Given $x^* \in \mathcal{X}$ and distance d, if $|\mathcal{X}| > 2$, then choose an arbitrary $\epsilon < \eta$ and choose w such that $w(x^*) = \eta - \epsilon$, $w(x') = 1 - \eta$ for an arbitrary $x' \in \mathcal{X}$, and $w(x) = \frac{\epsilon}{|\mathcal{X}|-2}$ otherwise. If $|\mathcal{X}| = 2$, let $w(x^*) = \eta$ and $w(x') = 1-\eta$. Note that such x' exists because $1 \leq k < |\mathcal{X}|$. Then $\Lambda^g_{\{x'\},d,w}(x', \ldots, \mu_k) = 0$ and $w(\{x'\}) \geq (1 - \eta)w(\mathcal{X})$. Hence x^* is clustered as noise and $\mathcal{A}$ satisfies noise-richness. $\square$

Theorem 2.18. *For any ϵ and $minPts > 0$, $(\epsilon, minPts)$-DBScan satisfies noise-richness.*

Proof. For any ϵ and $minPts > 0$, let $\mathcal{A}$ be $(\epsilon, minPts)$-DBScan. For any $x \in \mathcal{X}$, let $w(x) = \frac{minPts}{|\mathcal{X}|+1}$. Since $w(\mathcal{X}) < minPts$, $w(N_\epsilon(x)) < minPts$ for all $x \in \mathcal{X}$. Hence, $\mathcal{A}(\mathcal{X}, d, w) = (\{\}, \mathcal{X})$. Therefore, $\mathcal{A}$ satisfies noise-richness. $\square$

2.5 Conclusions

In this chapter we developed a formalism for clustering that has a designated noise cluster. We proposed properties that evaluate the input-output behaviour of clustering algorithms that have a noise cluster. These properties addressed the richness of the range of clustering algorithms, their invariance with respect to various changes in the original data set, and their computational feasibility compared to that of clustering algorithms without a noise cluster.

We presented four clustering algorithms with a noise cluster. We extended and generalized the definition of trimmed algorithms, introduced by Cuesta-Albertos et al. [6], and DBScan, introduced by Ester et al. [11]. Furthermore, we introduced two new algorithms, (k, g)-δ-truncated and (k, g)-δ-naive-truncated, and showed that the former is equivalent to a generalized non-fuzzy variation of an algorithm introduced by Dave [8].

We examined the above mentioned algorithms with respect to our proposed properties. Our analysis showed that (k, g)-δ-truncated and $(\epsilon, minPts)$-DBScan, on top of being efficient, possess the most number of desirable properties. On the other hand, the (k, g)-η-trimmed algorithm not only lacks many desirable properties, but is also not efficient. We also observed that the set of desirable properties satisfied by (k, g)-δ-naive-truncated is a strict subset of those satisfied by (k, g)-δ-truncated.

Chapter 3

Adding Robustness to Centroid-Based Algorithms

3.1 Introduction

Presence of significant amount of unstructured data tends to disrupt clustering algorithms and make it difficult to detect the cluster structure of the remaining domain points. This problem is commonly referred to as the issue of *noise robustness*. The first question in this context is whether there are useful clustering algorithms that are noise robust. Short reflection reveals that the noise robustness of an algorithm is closely related to its sensitivity to the input data. As an extreme example, it is easy to achieve noise robustness by ignoring the input data and always returning a fixed output. For clustering algorithms, Ackerman et al. [4] provide some formal trade-offs between these two desired properties. Roughly stated, their results (for example, their Theorem 4.3) show that no algorithm can be both noise robust and responsive to cluster structure in the data (in the language of Ackerman et al. [4] these properties are called *robustness to oligarchies* and *separability-detecting*). However, those results consider applying an algorithm with a fixed number-of-clusters parameter. This chapter addresses the possibility of overcoming those pessimistic results by allowing clustering algorithms to add to the set of clusters they output one or more clusters, serving as "noise collectors".

An important aspect of clustering, which distinguishes it from major other learning tasks, like classification prediction, is the wide variability of input-output behavior among common clustering algorithms. In fact, clustering can be viewed as an umbrella term for a wide range of different sub-tasks. Different clustering applications employ very different clustering algorithms and there is no single clustering algorithm that is suitable for all clustering applications. Consequently, solutions to fundamental clustering challenges, like the trade-off between sensitivity to the input and noise-robustness, should be modular in the sense of being applicable to a variety of clustering algorithms. In this chapter, we propose such a modular solution. We consider a method to transform any centroid-based clustering algorithm to a noise-robust one without sacrificing much of its ability to detect clear cluster structures. The degree of noise-robustness that such transformations achieve depends on a parameter that can be tuned by the user, depending upon the level and properties of the noise expected in the input data. We refer to the methods that add robustness to clustering algorithms as "robustifying methods".

Another critical feature of clustering algorithms is their computational complexity. Clustering is often applied to very large data sets and, therefore, the scalability of proposed clustering tools is a crucial factor. Regrettably, some natural solutions to the noise-robustness challenge are inherently computationally inefficient. This is the case, for example, with the Trimmed-k-means paradigm [6, 13] (as discussed in Chapter 2) that proposes to find the "least structured" η fraction of an input data set and discard it before applying a clustering algorithms (see Section 3.2 for more details). In contrast, our proposed paradigm employs a simple and efficiently implementable transformation of the input data, after which users can apply their favourite clustering algorithm. As mentioned above, the degree of noise-robustness achieved by this procedure is determined by a user-tunable parameter.

Yet another contribution of this chapter is the introduction of rigorous measures of noise-robustness. We consider three aspects of noise robustness for centroid-based clustering algorithms; the degree by which noise can affect the location of the centers of the clusters (or the archetypal cluster representatives), the effect of noise on the cost of the clustering solution (or, the value of the clustering objective function) and the similarity between the clustering of the un-noised data, to its clustering in the presence of the noise. More concretely, we consider a scenario in which the input data set $\mathcal{X}$ consists of two parts, the original input $\mathcal{I}$ and an added noise set $\mathcal{X} \setminus \mathcal{I}$ (the identity of which is not known to the clustering algorithm), and two clustering algorithms, the original one, $\mathcal{A}$, and its "robustified" transformation $R_p(\mathcal{A})$ (where p is a noise-level parameter set by the user). We compare the clustering $\mathcal{A}(\mathcal{I})$, to the clustering induced on $\mathcal{I}$ when the algorithm $R_p(\mathcal{A})$ is applied to $\mathcal{X}$, in terms of the three aspects mentioned above.

Our results consider to what extent clusterability of $\mathcal{I}$ and mildness properties of $\mathcal{X} \setminus \mathcal{I}$ (in terms of the size and/or diameter of this set, relative to that of $\mathcal{I}$) affect the above mentioned similarity measures between the clusterings $\mathcal{A}(\mathcal{I})$ and $R_p(\mathcal{A})(\mathcal{X})$ restricted to $\mathcal{I}$. We compare the behavior, in that respect, of our proposed robustifying paradigm with the behavior of the simple transformation in which $R_p(\mathcal{A})$ is the original algorithm $\mathcal{A}$ with an increased number of output clusters. We prove that our proposed paradigm has indeed better noise-robustness performance (with respect to those measures).

This chapter is organized as follows. In section 3.2, we provide a summary of related work. Section 3.3 introduces notations and definitions that are used in the rest of this chapter. Section 3.4 introduces two clustering paradigms and discusses how they relate to existing algorithms. In section 3.5, we define new measures of robustness and examine some previous measures of robustness and results pertaining to them. In section 3.6, we prove guaranteed robustness for our paradigm and show that more straightforward robustifying paradigms do not enjoy the same guarantees.

3.2 Related Work

Previous work on the robustness of clustering methods have focused on two directions. First, developing measures of robustness and examining the performance of traditional clustering algorithms based on those measures. Secondly, developing clustering algorithms that are robust to noise and outliers.

Various measures of robustness have been developed for examining the robustness characteristics of clustering algorithms to noise [10, 14, 15]. These measures have been used to demonstrate the lack of robustness of some traditional algorithms, when the number of clusters is fixed [4, 15]. That is, they consider the scenario in which a clustering algorithm is used with the same number-of-clusters parameter for both the clean input and the input after the addition of noisy points. We believe that a noise-handling version of the algorithm may be allowed to allocate more clusters (so as to accommodate the added noisy data points) and introduce (and analyze) noise-robustness measures that allow such flexibility. In fact, we show that using that added flexibility of our proposed noise robustifying paradigm, we can overcome some of the limitations that are shown to be inevitable as long as one does not allow extra noise-accommodating clusters.

Several methods have been suggested for clustering a potentially noisy data set [6, 9, 11]. One interesting work is the development of the concept of a "noise cluster" in a fuzzy setting by Dave [8, 9], which we have discussed extensively in Chapter 2. In this work, we introduce a novel paradigm for "robustifying" any centroid-based clustering algorithm. We show that our paradigm generalizes a non-fuzzy variation of the algorithm introduced by Dave [8]. In addition, we prove noise robustness guarantees for our proposed paradigm that were not proven in any of the earlier works we are aware of.

Some of the earlier work on noise-robustness of clustering algorithms propose the use of *trimming*. That is, searching for a subset of the input data of size determined by a pre-chosen fraction of the input size whose removal leads to the maximum improvement of the clustering quality (or objective function) [6, 13, 12]. However, these methods are approximately optimized by efficient heuristics that have no performance guarantees. In this work, we avoid this issue by developing a paradigm that is of comparable computational complexity to the centroid-based clustering algorithms, upon which they are based.

Discussing the details of previous work requires the definition of few notations, hence, it is delayed to the relevant sections.

3.3 Preliminaries

In this section, we develop notions of clustering for unweighted input. Our definitions and results can be extended to the weighted setting (as presented in Chapter 2). However, using weighted input provides little benefit for our examination of robustness at the cost of unnecessarily complicating our approach. Therefore, we restrict our work to clustering unweighted data and reprise some definitions from the past chapter in the unweighted setting.

For a set $\mathcal{X}$ and integer $k \geq 1$, a *k-clustering* of $\mathcal{X}$ is a partition $\mathcal{C} = \{C_1, \ldots, C_k\}$ of $\mathcal{X}$ into k disjoint sets. For a clustering $\mathcal{C}$ of $\mathcal{X}$ and points $x, y \in \mathcal{X}$, we say $x \sim_{\mathcal{C}} y$, if x and y

are in the same cluster, otherwise $x \not\sim_{\mathcal{C}} y$. For sets $\mathcal{X}$ and $\mathcal{I}$ such that $\mathcal{I} \subseteq \mathcal{X}$, and a clustering $\mathcal{C} = \{C_1, \ldots, C_k\}$ of $\mathcal{X}$, we denote the *restriction of* $\mathcal{C}$ *to* $\mathcal{I}$ by $\mathcal{C}|\mathcal{I} = \{C_1 \cap \mathcal{I}, \ldots, C_k \cap \mathcal{I}\}$.

For two clusterings $\mathcal{C}$ and $\mathcal{C}'$ of the set $\mathcal{X}$, we define the distance between them as $\Delta(C, C')$, the fraction of pairs of domain points which are in the same cluster under $\mathcal{C}$ but in different clusters under $\mathcal{C}'$ or vice-versa. Equivalently, $\Delta(C, C') = 1 - i_R(\mathcal{C}, \mathcal{C}')$, where i_R is the Rand index as defined by Rand [23]. The following theorem shows that Δ satisfies the triangle inequality.

Theorem 3.1. *For any clusterings C_1, C_2, and C_3 of $\mathcal{X}$, $\Delta(C_1, C_3) \leq \Delta(C_1, C_2) + \Delta(C_2, C_3)$*

Proof. Using the terminology of [20], let $N_{disagree}(C_1, C_2)$ represent the number of $\{x, y\}$ pairs that are clustered differently in $\mathcal{C}_1$ and $\mathcal{C}_2$ i.e. $x \sim_{C_1} y$ and $x \not\sim_{C_2} y$, or $x \not\sim_{C_1} y$ and $x \sim_{C_2} y$. Let $x, y \in \mathcal{X}$ be such that $x \sim_{C_1} y$ and $x \not\sim_{C_3} y$. Then the pair $\{x, y\}$ contributes to the value of $N_{disagree}(C_1, C_3)$. There are two cases:

1. $x \sim_{C_2} y$: $\{x, y\}$ contributes to $N_{disagree}(C_2, C_3)$.

2. $x \not\sim_{C_2} y$: $\{x, y\}$ contributes to $N_{disagree}(C_1, C_2)$.

Similarly, if $x \not\sim_{C_1} y$ and $x \sim_{C_3} y$, then $\{x, y\}$ contributes to one of $N_{disagree}(C_1, C_2)$ or $N_{disagree}(C_2, C_3)$. Therefore, $N_{disagree}(C_1, C_3) \leq N_{disagree}(C_1, C_2) + N_{disagree}(C_2, C_3)$.

$$\begin{aligned}
\Delta(C_1, \mathcal{C}_3) &= \frac{N_{disagree}(C_1, C_3)}{\binom{\mathcal{X}}{2}} \\
&\leq \frac{N_{disagree}(C_1, C_2)}{\binom{\mathcal{X}}{2}} + \frac{N_{disagree}(C_2, C_3)}{\binom{\mathcal{X}}{2}} \\
&\leq \Delta(C_1, C_2) + \Delta(C_2, C_3)
\end{aligned}$$

□

Let d be a symmetric distance function defined over $\mathcal{X}$ with $d(x, x) = 0$, satisfying the triangle inequality (unless otherwise stated). The *diameter* of $\mathcal{X}$, indicated by $\operatorname{diam}(\mathcal{X})$, is defined as the maximum distance between two elements of $\mathcal{X}$. For a clustering $\mathcal{C} = \{C_1, \ldots, C_k\}$, the diameter of $\mathcal{C}$ is defined as $\max_{C_i \in \mathcal{C}} \operatorname{diam}(C_i)$. The *radius* of $\mathcal{X}$ is shown by $\operatorname{rad}(\mathcal{X}) =$

$\min_{c\in\mathcal{X}} \max_{x\in\mathcal{X}} d(c,x)$. Clustering $\mathcal{C}$ is *σ-separable* for $\sigma \geq 1$, if $\min_{x\not\sim_{\mathcal{C}} y} d(x,y) > \sigma \cdot \max_{x\sim_{\mathcal{C}} y} d(x,y)$. Clustering $\mathcal{C}$ is *(ρ_1,ρ_2)-balanced* if for all $i \in [k]$, $\rho_1|\mathcal{X}| \leq |C_i| \leq \rho_2|\mathcal{X}|$. We use *$\rho$-balanced* to refer to a clustering that is $(0,\rho)$-balanced.

A *clustering algorithm* is a function $\mathcal{A}$ that takes as input $\mathcal{X}$ and d and returns a clustering $\mathcal{C}$ of $\mathcal{X}$. An *objective function* is a function that takes as input a clustering and outputs a non-negative cost associated with it. An *objective-based clustering algorithm* is an algorithm that produces a clustering that minimizes a specified objective function.

Consider an input set $\mathcal{X}$ drawn from a given space E with distance function d. Throughout this chapter, let $g : \mathbb{R}^+ \to \mathbb{R}^+$ be any continuous, increasing, and unbounded function. The *(k,g)-centroid algorithm* is an objective-based clustering algorithm with objective function

$$\Lambda^g_{E,d}(\{C_1,\ldots,C_k\}) = \min_{\mu_1,\ldots,\mu_k\in E} \sum_{i\in[k]} \sum_{x\in C_i} g(d(x,\mu_i))$$

We refer to μ_i as the *center* of cluster C_i and we define $\mu(x) = \arg\min_{\mu_i\in\{\mu_1,\ldots,\mu_k\}} d(x,\mu_i)$. With a slight abuse of notation we can also define the (k,g)-centroid algorithm as the algorithm that chooses centers $\mu_1,\ldots,\mu_k$ that minimize

$$\Lambda^g_{\mathcal{X},E,d}(\mu_1,\ldots,\mu_k) = \sum_{x\in\mathcal{X}} g(d(x,\mu(x)))$$

We remove $\mathcal{X}$ and E from the notation whenever they are clear from the context. Two of the commonly used (k,g)-centroid algorithms are *k-median* and *k-means*, which are obtained by setting $g(x) = x$ and $g(x) = x^2$, respectively.

3.4 Robustifying Centroid-Based Algorithms

We define parameterized robustifying paradigms that transform any clustering algorithm to an algorithm that is more robust to noise to the extent determined by a predefined parameter. Parameters play an important role in defining the limit to which an algorithm should be robustified. Unsuitable values of these parameters can result in algorithms that are not responsive to the structure of the data or are extremely unrobust to the addition of noise. In this section, we define two robustifying paradigms, one of which is equivalent to a generalization of an existing algorithm, and discuss the choice of parameters.

3.4.1 Parameterized Robustifying Paradigms

A *robustifying parameter*, p, denotes the degree to which an algorithm should be robustified to noise; For example, the number of extra clusters that can be used or a notion of distance beyond which a point is considered an outlier. A *robustifying paradigm*, $R_p(\cdot)$, is a function that takes a clustering algorithm $\mathcal{A}$ and returns a robustified clustering algorithm $R_p(\mathcal{A})$ based on the robustifying parameter p. We refer to $\mathcal{A}$ as the *ground* clustering algorithm of $R_p(\mathcal{A})$.

Since noise, unstructured data, and outlying structures are heterogeneous with respect to the existing data, outliers and noise groups can be considered as separate clusters. Therefore, some statisticians simply recommend increasing the number of clusters when dealing with noisy data [12]. The next paradigm captures robustification as used in this practice.

Definition 3.1 (p-Increased Paradigm). *The* p-Increased Paradigm *is a robustifying paradigm,* $RI_p(\cdot)$*, that takes as input a* (k, g)*-centroid algorithm and returns a* $(k+p, g)$*-centroid algorithm.*

The next paradigm is parameterized by the distance after which a point should be considered an outlier. To define this paradigm, we first introduce a class of algorithms. Given a space E and distance function d, *the* δ*-truncated* distance function corresponding to d is the function d' such that $d'(x, y) = \min\{\delta, d(x, y)\}$ for $x, y \in E$. The (k, g)-δ-truncated algorithm is an objective based algorithm that, given $\mathcal{X} \subseteq E$, first optimizes the function $\Lambda^g_{\mathcal{X},d'}(\mu'_1, \ldots, \mu'_k)$. For $j \in [k]$, let $C'_j = \{x \in \mathcal{X} | j = \arg\min_i d(x, \mu'_i)$ and $d(x, \mu'_j) < \delta\}$ and $C'_{k+1} = \{x \in \mathcal{X} | \min d(x, \mu'_i) \geq \delta\}$. Then the (k, g)-δ-truncated algorithm returns the $(k + 1)$-clustering $\mathcal{C}' = \{C'_1, \ldots, C'_{k+1}\}$. We refer to μ'_i as the center of C'_i for $i \leq k$. With a slight abuse of notation, we define $\mu'(x) = \min_{i\in[k]} d(x, \mu'_i)$.

Definition 3.2 (δ-Truncated Paradigm). *The* δ-Truncated Paradigm *is a robustifying paradigm,* $RT_\delta(\cdot)$*, that takes as input a* (k, g)*-centroid algorithm and returns a* (k, g)*-*δ*-truncated algorithm.*

In the next definition, we provide a generalization of the non-fuzzy variation of Dave's algorithm [8] for any centroid-based algorithm.

Definition 3.3. *Let μ^* be defined such that for all $y \in E$, $d(y, \mu^*) = \delta$. The generalized (k, g)-δ-centroid algorithm is an objective-based algorithm with the following objective function*

$$\Lambda^g_{\mathcal{X}, E \cup \{\mu^*\}, d}(\mu_1, \ldots, \mu_k, \mu^*)$$

We refer to (k, g)-δ-centroid as δ-k-*median* and δ-k-*means* when $g(x) = x$ and $g(x) = x^2$, respectively. As shown in Theorem 2.1, the class of algorithms produced by the δ-Truncated paradigm is equivalent to the class of (k, g)-δ-centroid algorithms. Therefore, the cost of a $\{C_1, \ldots, C_k, C_{k+1}\}$ clustering, where C_{k+1} is the cluster associated with the noise prototype μ^*, is indicated by

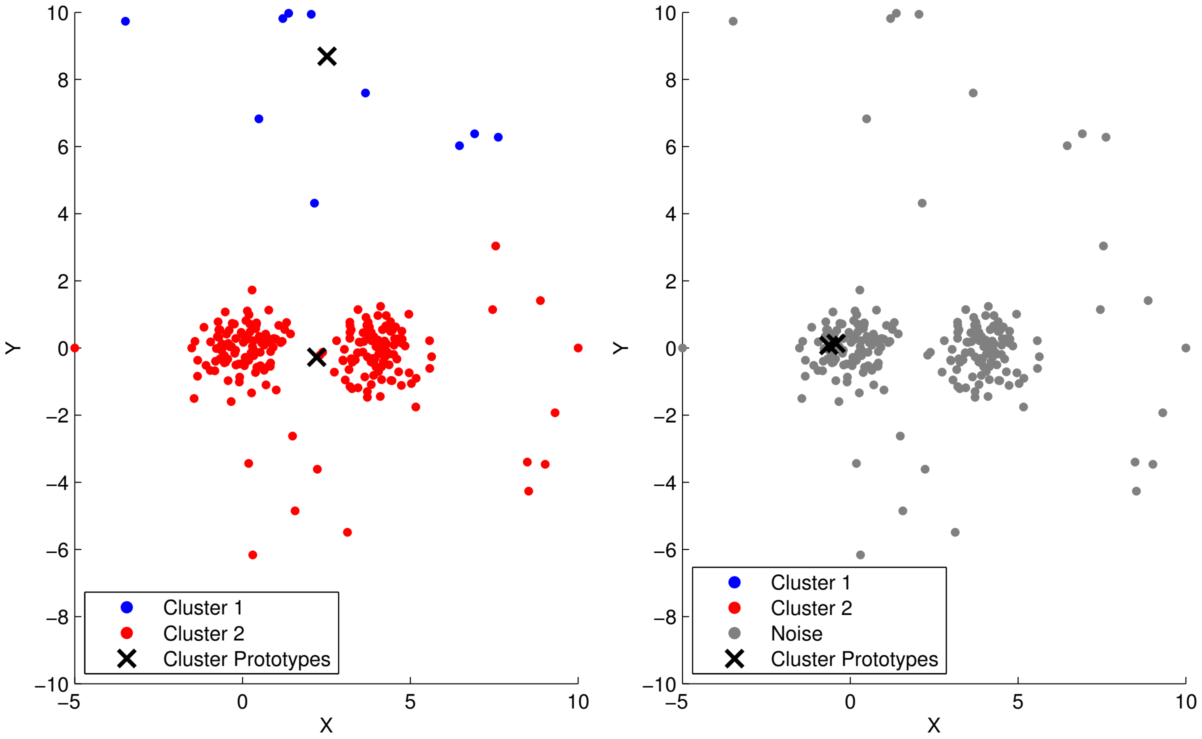
$$\Lambda^g_{d'}(\{C_1, \ldots, C_k, C_{k+1}\}) = \Lambda^g_d(\{C_1, \ldots C_k\}) + |C_{k+1}| \cdot g(\delta) \tag{3.1}$$

3.4.2 Effects of Parameters

In this section we discuss the choice of robustifying parameters for the δ-Truncated and p-Increased paradigms.

In the δ-truncated paradigm, parameter δ quantifies a measure of distance beyond which a point is considered an outlier (or a distance above which the measurements are considered unreliable). As shown in Equation 3.1, $g(\delta)$ plays the role of a constant penalty for any point that is not clustered using the ground clustering. This penalty affects the size and structure of the noise cluster. Figure 3.1 demonstrates the performance of $R^t_\delta(\mathcal{A})$ for two extreme values of δ. The value of $\delta = \infty$ imposes an infinitely large penalty for any unclustered point, resulting in a (k, g)-centroid algorithm that does not enjoy the robustness the δ-Truncated paradigm can offer (see Figure 3.1a). On the other hand, $\delta = 0$ imposes no penalty for leaving a point unclustered, resulting in an algorithm that clusters every point as noise (see Figure 3.1b). It is clear that neither of these parameters result in an appropriate clustering of the given data. Figure 3.2 demonstrates the clustering resulted by using an appropriate parameter.

The range of appropriate values of δ depends on statistical parameters that are determined by the structure of the data sets. Dave [8] proposes an iterative scheme to set the value of δ based



(a) (k, g)-δ-truncated for $\delta = \infty$

(b) (k, g)-δ-truncated for $\delta = 0$

Figure 3.1: Effects of parameters on (k, g)-δ-truncated for $k = 2$ and $g(x) = x^2$

on the average inter-cluster distance at each iteration of the Lloyd algorithm. The generalization of this method for an arbitrary function g is as follows:

$$\delta = g^{-1}\left(\lambda \frac{\sum_{y\in\mathcal{X}} g(d(y,\mu(y)))}{|\mathcal{X}|k}\right)$$

where λ is a multiplier that can be determined using other statistical measures. In our work, we avoid computing several statistical measures for each data set by specifying a range of appropriate parameters for our results.

For the p-Increased paradigm, a good choice for the total number of clusters $(k+p)$ can be estimated using several methods [21, 24]. Large values of p result in clusterings where individual points form their own clusters. These clustering algorithms overfit and do not produce a clustering that is representative of the underlying structure of the data. On the other hand, $p = 0$ results in algorithms that do not take advantage of the potential robustness of the p-Increased paradigm and simply return the (k, g)-centroid clustering. In section 3.6, we show some inherent limitations of p-Increased paradigm for a large ranges of p.

3.5 Measures of Robustness

In this section, we introduce a general approach for defining measures of robustness to noise. We discuss the suitability of this approach and define three rigorous measures of robustness. We review existing measures of robustness and their implications and discuss the possibility of overcoming previous negative results in our approach.

In the following, $\mathcal{A}$ denotes any clustering algorithm, $R_p(\cdot)$ denotes a robustifying paradigm with parameter p, and $(\mathcal{X}, d)$ denotes some domain space with a distance measure. Given $\mathcal{I} \subseteq \mathcal{X}$ and a centroid-based clustering, $\mathcal{A}(\mathcal{I})$, we use $\mathcal{A}'(\mathcal{X})$ to denote the clustering of $\mathcal{X}$ by $R_p(\mathcal{A})$, and for any $x \in \mathcal{X}$, we use $\mu(x)$ to denote the center of the $\mathcal{A}(\mathcal{I})$ cluster to which x belongs, and we use $\mu'(x)$ to denote the center of the $\mathcal{A}'(\mathcal{X})$ cluster to which x belongs. $\mathcal{I}$ is said to be robust to $\mathcal{X} \setminus \mathcal{I}$ with respect to the $R_p(\mathcal{A})$ algorithm if certain properties of $\mathcal{A}(\mathcal{I})$ are preserved in $\mathcal{A}'(\mathcal{X})$.

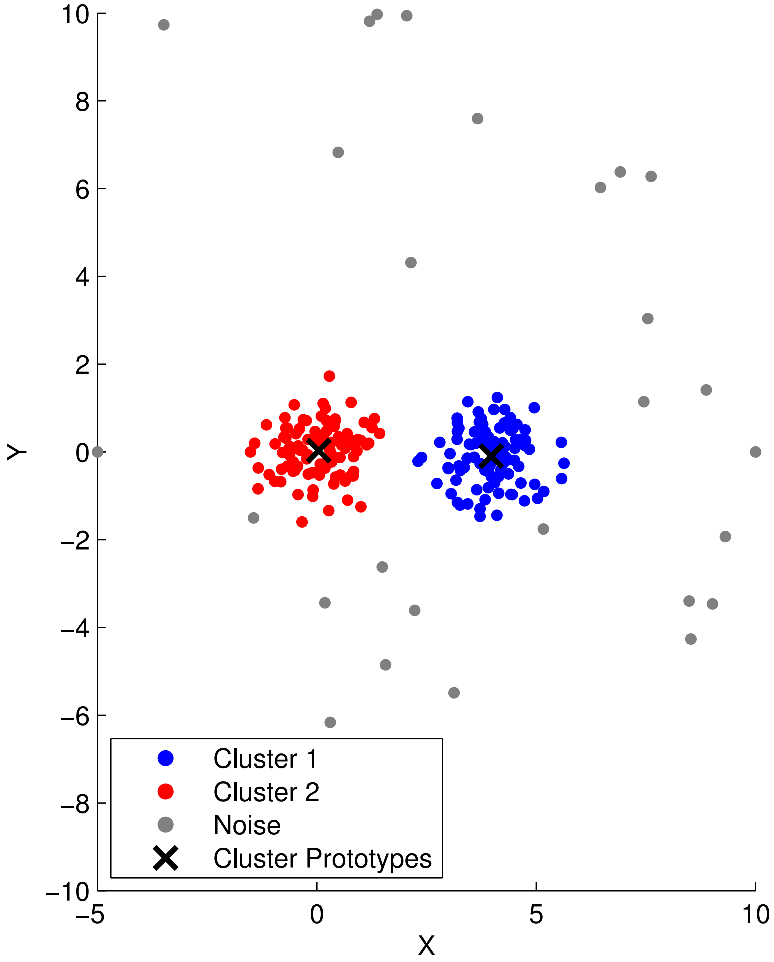


Figure 3.2: A good choice of parameter $\delta = 2.0$, for δ-k-means on this input data.

Definition 3.4 (α-distance-robust)**.** *A subset $\mathcal{I} \subseteq \mathcal{X}$ is α-distance-robust (to $\mathcal{X} \setminus \mathcal{I}$) with respect to $\mathcal{A}'$ if for all $y \in \mathcal{I}$,*

$$d(y, \mu'(y)) \leq d(y, \mu(y)) + \alpha$$

α-distance-robustness measures how much the position of the cluster prototypes are affected by the noisy data. Large changes in the position of the cluster centers leads to large changes in the structure of the clustering. Roughly speaking, $\mathcal{I} \subseteq \mathcal{X}$ is clustered similarly in $\mathcal{A}'(\mathcal{X})$ and $\mathcal{A}(\mathcal{I})$ if $\mathcal{I}$ is α-distance-robust to $\mathcal{X} \setminus \mathcal{I}$ with respect to $\mathcal{A}'$ for small values of α.

Definition 3.5 (β-cost-robust)**.** *Given an objective (cost) function* cost*, $\mathcal{I} \subseteq \mathcal{X}$ is* β-cost-robust *(to $\mathcal{X} \setminus \mathcal{I}$) with respect to $\mathcal{A}'$ for* cost*, if there exists $C_1^*, \ldots, C_j^* \in \mathcal{A}'(\mathcal{X})$, such that*

- $|\bigcup_{i=1}^{j} C_i^*| \leq |\mathcal{X} \setminus \mathcal{I}|$
- $\mathrm{cost}(\mathcal{A}'(\mathcal{X}) \setminus \bigcup_{i=1}^{j} C_i^*) \leq \mathrm{cost}(\mathcal{A}(\mathcal{I})) + \beta$

β-cost-robustness measures the degree to which cost of a clustering is affected by the presence of noise when we allow a few clusters with a few points to act as “garbage collectors”, i.e. have no effect on the cost. In the extreme case, if $\mathcal{A}'(\mathcal{X}) = \mathcal{A}(\mathcal{I}) \cup \{X \setminus \mathcal{I}\}$, then $\mathcal{I}$ is 0-cost-robust to $\mathcal{A}'$.

Definition 3.6 (γ-robust)**.** *$\mathcal{I} \subseteq \mathcal{X}$ is γ-robust (to $\mathcal{X} \setminus \mathcal{I}$) with respect to algorithm $\mathcal{A}'$, if*

$$\Delta(\mathcal{A}'(\mathcal{X})|\mathcal{I}, \mathcal{A}(\mathcal{I})) \leq \gamma$$

γ-robustness measures the degree to which noise affects the structure of a clustering. For example, $\mathcal{I}$ is 0-robust with respect to $\mathcal{A}'$, if the clustering of points in $\mathcal{I}$ is not changed after adding noise, i.e. for all $x, y \in \mathcal{I}$, $x \sim_{\mathcal{A}(\mathcal{I})} y$, if and only if $x \sim_{\mathcal{A}'(\mathcal{X})} y$.

Lemma 3.1. *For any $\mathcal{I} \subseteq \mathcal{X}$, $\mathcal{I}$ is 0-robust to $\mathcal{A}'$ if and only if for every non-empty $C'_j \in \mathcal{A}'(\mathcal{X})$ there is a unique non-empty $C_i \in \mathcal{A}(\mathcal{I})$ such that $C_i \subseteq C'_j$.*

Proof. If $\Delta(\mathcal{A}'(\mathcal{X})|\mathcal{I}, \mathcal{A}(\mathcal{I})) = 0$, then for every $x, y \in \mathcal{I}$, $x \sim_{\mathcal{A}'(\mathcal{X})} y$ if and only if $x \sim_{\mathcal{A}(\mathcal{I})} y$. So, for every $C'_j \in \mathcal{A}(\mathcal{X})$ there is a unique $C_i \in \mathcal{A}(\mathcal{I})$, such that $\mathcal{C}_i \subseteq C'_j$. Conversely, if for every $C'_j \in \mathcal{A}'(\mathcal{X})$, there is a unique $C_i \in \mathcal{A}(\mathcal{I})$ such that $C_i \subseteq C'_j$, then for every $x, y \in \mathcal{X}$ and any $\mathcal{C}_i \in \mathcal{A}(\mathcal{I})$, $x \sim_{C_i} y$ if and only if $x \sim_{C'_j} y$. So, $\Delta(\mathcal{A}'(\mathcal{X})|\mathcal{I}, \mathcal{A}(\mathcal{I})) = 0$. □

Lemma 3.2. *Given a (k, g)-centroid algorithm $\mathcal{A}$ and parameter δ, let $\mathcal{A}' = R^t_\delta(\mathcal{A})$. For any $\mathcal{I} \subseteq \mathcal{X}$, such that for all $y \in \mathcal{I}$, $d(y, \mu'(y)) \leq \delta$, $\mathcal{I}$ is $g(\delta)|\mathcal{X} \setminus \mathcal{I}|$-cost-robust to $\mathcal{X} \setminus \mathcal{I}$ with respect to $R^t_\delta(\mathcal{A})$ for the Λ^g_d (cost) function.*

Proof. If $d(y, \mu'(y)) \leq \delta$, then y is not in the noise cluster. Therefore, there is a cluster $C^* \in \mathcal{C}$, such that $C^* \subseteq \mathcal{X} \setminus \mathcal{I}$, and for any $C' \neq C^*$,

$$\Lambda^g_d(C') \leq \Lambda^g_d(C' \cap \mathcal{I}) + |C' \setminus \mathcal{I}| \cdot g(\delta)$$

Therefore, $\mathcal{I}$ is $g(\delta)|\mathcal{X} \setminus \mathcal{I}|$-cost-robust to $|\mathcal{X} \setminus \mathcal{I}|$ with respect to $R_\delta(\mathcal{A})$ for the Λ^g_d cost function. □

Comparison with Previous Work

In previous work, robustness to the addition of noisy data has been measured mainly by comparing the output of an algorithm with a fixed number of clusters before and after adding noise and outliers. One of the examples of these results is the work of Ackerman et al. [4], which shows that algorithms that are responsive to the structure of the data are not noise-robust. More precisely, let k-clustering algorithm $\mathcal{A}$ be *σ-separability-detecting* for $\sigma \geq 1$ with respect to k, if for all $\mathcal{I}$, if there exists an σ-separable k-clustering $\mathcal{C}$ of $\mathcal{I}$, then $\mathcal{A}(\mathcal{I}) = \mathcal{C}$. Ackerman et.al. show that for any σ-separability-detecting algorithm and any ρ, there is a ρ-balanced σ-separable set $\mathcal{I}$ that is not robust to an oligarchy set of size as small as k. These pessimistic results only hold when the number of clusters are fixed. By using robustness measures that accommodate a change in the number of clusters, we allow the possibility of overcoming these negative results

3.6 Results

In this section, we compare to what extent the well-clusterability of $\mathcal{I}$ and mildness properties of $\mathcal{X} \setminus \mathcal{I}$ affect the robustness with respect to the p-Increased and δ-Truncated paradigms. We derive upper bounds on the robustness (lower bounds on the unrobustness) of the p-Increased paradigm and provide guaranteed lower bounds on robustness (upper bounds on the unrobustness) of

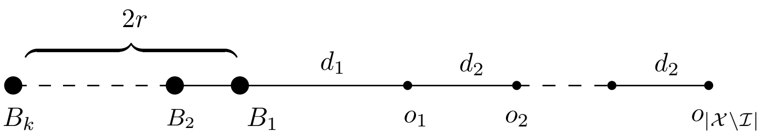


Figure 3.3: Structure of a data set that is not robust w.r.t $RI_p(\mathcal{A})$.

the δ-Truncated paradigm. Through this comparison, we show that p-Increased algorithms are inherently less robust than δ-Truncated algorithms.

3.6.1 Robustness fails for the p-Increased Paradigm

In this subsection, we prove inherent limitations of the p-Increased paradigm, which allows a centroid-based algorithm to use extra clusters. More specifically, Theorems 3.2, 3.3 and 3.4 show that for any desired level of robustness and signal-to-noise ratio, there exists $\mathcal{I} \subseteq \mathcal{X}$ with the desired signal-to-noise ratio and a well-clusterable underlying pattern, but it is not robust to $\mathcal{X} \setminus \mathcal{I}$ with respect to the Increase-p paradigm as long as $p < |\mathcal{X} \setminus \mathcal{I}|$.

Theorem 3.2. *Let $\mathcal{A}$ be the k-means algorithm. For any $r > 0$, any desired level of robustness $\alpha > 0$, and any signal-to-noise ratio $\lambda > 0$, there exists $\mathcal{X}$ and $\mathcal{I} \subseteq \mathcal{X}$, such that $\frac{|\mathcal{I}|}{|\mathcal{X}\setminus\mathcal{I}|} \geq \lambda$, $\operatorname{rad}(\mathcal{I}) = r$ and $\mathcal{I}$ can be covered with k balls, each of radius 0, but for any $p < |\mathcal{X} \setminus \mathcal{I}|$, $\mathcal{I}$ is neither α-distance-robust nor $g(\alpha)(|\mathcal{I}| - |\mathcal{X} \setminus \mathcal{I}|)$-cost-robust to $\mathcal{X} \setminus \mathcal{I}$ with respect to $RI_p(\mathcal{A})$ for cost function Λ_d^g, where $g(x) = x^2$.*

Proof. Let $d_1 = (\alpha + 2r)(\frac{\lambda}{\lambda+1}|\mathcal{X}| + 1)$ and $d_2 = 2(d_1 + 2r) + 1$. For $i \in [k]$, let B_i denote a set with radius 0, such that $|B_i| \geq \frac{\lambda}{k(\lambda+1)}|\mathcal{X}|$. Let $B_1, \ldots, B_k$ be evenly placed on a line of length $2r$. For, $i \in \left[\lfloor\frac{|\mathcal{X}|}{\lambda+1}\rfloor\right]$, let o_i be a point on the line that connects $B_1, \ldots, B_k$, such that $d(o_1, B_1) = d_1$ and $d(o_i, o_{i+1}) = d_2$ (see Figure 3.3). Let $\mathcal{I} = \bigcup_{i\in[k]} B_i$ and $\mathcal{X} = \mathcal{I} \cup \{o_1, \ldots, o_{|\mathcal{X}|/(\lambda+1)}\}$. Note that $\mathcal{X}$ and $\mathcal{I}$ are chosen such that $\frac{|\mathcal{I}|}{|\mathcal{X}\setminus\mathcal{I}|} \geq \lambda$.

Let $\mathcal{A}' = RI_p(\mathcal{A})$, $\mathcal{A}'(\mathcal{X}) = \{C_1, \ldots, C_{k+p}\}$, μ_i denote the center of C_i, and $\mu(x)$ denote the closest center to x. Assume (in the hope of finding a contradiction) that for all $i \in [k+p]$, $C_i \subseteq \mathcal{I}$ or $C_i \subseteq \mathcal{X} \setminus \mathcal{I}$. Without loss of generality let $C_1 \subseteq \mathcal{I}$, then $d(o_1, \mu_1) \leq d_1 + 2r$.

Moreover, for any $C_i \subseteq \mathcal{X} \setminus \mathcal{I}$, such that $|C_i| \geq 2$, if o_j is the left-most or the right-most point of C_i, $d(o_j, \mu_i) \geq d_2/2 > d(o_1, \mu_1)$. Without loss of generality, assume $o_1 \in C_2$. There are two cases:

1. $C_2 = \{o_1, o_j, \dots\}$: Then the cost of clustering $\mathcal{C}' = \{C_1 \cup \{o_1\}, C_2 \setminus \{o_1\}, \dots, C_{p+k}\}$ is lower than the cost of $\mathcal{C}$.

2. $C_2 = \{o_1\}$: Let $C_3 \subseteq \mathcal{X} \setminus \mathcal{I}$ be any cluster of size at least 2, and let o_i be its left-most point (such a cluster exists since $p < |\mathcal{X} \setminus \mathcal{I}|$). The cost of clustering $\mathcal{C}' = \{C_1 \cup \{o_1\}, \{o_i\}, C_3 \setminus \{o_i\}, \dots, C_{p+k}\}$ is lower than the cost of $\mathcal{C}$.

Hence, $\mathcal{C}$ is not an optimal clustering. Therefore, for any optimal $RI_p(\mathcal{A})$ clustering, there exists a cluster C_i such that $\{o_j, y\} \subseteq C_i$ for some $y \in \mathcal{I}$. Then, $d(y, \mu_i) \geq \frac{d_1}{|\mathcal{I}|+1} > \alpha + 2r$. Hence, $\mathcal{I}$ is not α-distance-robust to $\mathcal{X} \setminus \mathcal{I}$ with respect to $RI_p(\mathcal{A})$.

Since, there exists $y \in \mathcal{I}$, such that $d(y, \mu(y)) > \alpha + 2r$, for all $y' \in \mathcal{I}$, $d(y', \mu(y')) > \alpha$, so $\mathcal{I}$ is not $g(\alpha)(|\mathcal{I}| - |\mathcal{X} \setminus \mathcal{I}|)$-cost-robust to $\mathcal{X} \setminus \mathcal{I}$ with respect to $RI_p(\mathcal{A})$. □

Theorem 3.3. *Let $\mathcal{A}$ be the k-means algorithm. For any $r > 0$ and any signal-to-noise ratio $\lambda > 0$, there exists $\mathcal{X}$ and $\mathcal{I} \subseteq \mathcal{X}$, such that $\frac{|\mathcal{I}|}{|\mathcal{X} \setminus \mathcal{I}|} \geq \lambda$, $\mathcal{I}$ has radius r and can be covered with k balls of radius 0, but for any $p < |\mathcal{X} \setminus \mathcal{I}|$, $\mathcal{I}$ is not $(1 - \frac{1}{k})$-robust to $\mathcal{X} \setminus \mathcal{I}$ with respect to $RI_p(\mathcal{A})$.*

Proof. We repeat the construction from Theorem 3.2 and Figure 3.3 with $d_1 = 4r(\frac{\lambda}{\lambda+1}|\mathcal{X}| + 1)$ and $d_2 = 2(d_1 + 2r) + 1$. Note that clusters in any centroid-based clustering are convex. For any $y \in \mathcal{I}$, $d(y, \mu(y')) > 2r$, hence, the center of the cluster containing any B_i is to the right of $\mathcal{I}$ (see figure 3.3). Therefore, $B_1, \dots, B_k$ are all in one cluster of $R_p^i(\mathcal{A})(\mathcal{X})$. Each B_i forms a unique cluster in $\mathcal{A}(\mathcal{I})$. Therefore,

$$\begin{aligned}
\Delta(\mathcal{A}(\mathcal{I}), \mathcal{A}'(\mathcal{X})|\mathcal{I}) &\geq 1 - \frac{\sum_{i \in [k]} \binom{|B_i|}{2}}{\binom{|\mathcal{I}|}{2}} \\
&\geq 1 - \frac{k\binom{\frac{|\mathcal{I}|}{k}}{2}}{\binom{|\mathcal{I}|}{2}} \\
&\geq 1 - \frac{1}{k}
\end{aligned}$$

$\mathcal{I}$ is not $(1-\frac{1}{k})$-robust to $\mathcal{X}\setminus\mathcal{I}$ with respect to $R^i_p(\mathcal{A})$. □

Theorem 3.4. *Let $\mathcal{A}$ be the k-means clustering algorithm. For any desired level of robustness $\alpha > 0$ and any signal-to-noise ratio $\lambda > 0$, there exists $\mathcal{X}$ and $\mathcal{I} \subseteq \mathcal{X}$, such that $\frac{|\mathcal{I}|}{|\mathcal{X}\setminus\mathcal{I}|} \geq \lambda$ and $\mathcal{I}$ has radius 0, but for any $p < |\mathcal{X}\setminus\mathcal{I}| - k$, $\mathcal{I}$ is not α-distance-robust or $g(\alpha)(|\mathcal{I}| - |\mathcal{X}\setminus\mathcal{I}|)$-cost-robust to $\mathcal{X}\setminus\mathcal{I}$ with respect to $RI_p(\mathcal{A})$ for cost function Λ^g_d, where $g(x) = x^2$.*

Proof. The proof is similar to the proof of Theorem 3.2 with $r = 0$. □

Corollaries 3.1 and 3.2 demonstrate the limitations of p-Increased algorithms even when the the space is bounded. In these corollaries, the degree of robustness (or lack thereof) is restricted by a function of the values that indicate the clusterability of $\mathcal{I}$ and the size of the noise cluster.

Corollary 3.1. *Let $\mathcal{A}$ be the k-means algorithm. For any λ, k and ν, such that $\nu < \frac{1}{k}$, there exists $\mathcal{I} \subseteq \mathcal{X}$, such that $\frac{|\mathcal{I}|}{|\mathcal{X}\setminus\mathcal{I}|} \geq \lambda$, $\mathcal{X}$ has diameter 1, $\mathcal{I}$ can be covered by k clusters of radius zero whose centers are at least ν away from each other, but for any $p < |\mathcal{X}\setminus\mathcal{I}|$ and any $\alpha \leq \frac{1-k\nu}{2|\mathcal{X}|(|\mathcal{I}|+1)}$, $\mathcal{I}$ is not α-distance-robust or $(1-\frac{1}{k})$-robust, or $(\alpha-\nu k)^2(|\mathcal{I}|-|\mathcal{X}\setminus\mathcal{I}|)$-cost-robust with respect to $RI_p(\mathcal{A})$ for cost function Λ^g_d, where $g(x) = x^2$.*

Corollary 3.2. *Let $\mathcal{A}$ be the k-means algorithm. For any $\lambda > 0$, there exists $\mathcal{I} \subseteq \mathcal{X}$, such that $\frac{|\mathcal{I}|}{|\mathcal{X}\setminus\mathcal{I}|} \geq \lambda$, $\mathcal{X}$ has diameter 1, $\mathcal{I}$ has radius 0, but for any $p < |\mathcal{X}\setminus\mathcal{I}| - k$ and any $\alpha \leq \frac{1}{2(|\mathcal{I}|+1)|\mathcal{X}|}$, $\mathcal{I}$ is not α-distance-robust or $\alpha^2(|\mathcal{I}|-|\mathcal{X}\setminus\mathcal{I}|)$-cost-robust with respect to $R^i_p(\mathcal{A})$ for cost function Λ^g_d, where $g(x) = x^2$.*

3.6.2 Robustness of the δ-Truncated paradigm

In this section, we show guaranteed robustness results for the δ-Truncated paradigm. We prove robustness, distance-robustness, and cost-robustness based on several types of underlying structures of $\mathcal{I}$ and mildness properties of $\mathcal{X}\setminus\mathcal{I}$:

The Radius of the Ball Covering $\mathcal{I}$

The following results guarantee distance-robustness and cost-robustness for the δ-Truncated paradigm. Theorem 3.4 derives values of δ that render $\mathcal{I}$ robust to $\mathcal{X}\setminus\mathcal{I}$ with respect to a

δ-Truncated algorithm, based on the radius of $\mathcal{I}$ and the signal-to-noise ratio of $\mathcal{X}$, i.e. $\frac{|\mathcal{I}|}{|\mathcal{X}\setminus\mathcal{I}|}$. Corollary 3.3 demonstrates the implications of Theorem 3.5 on the robust variants of two common clustering algorithms, k-means and k-median.

Theorem 3.5. *For all k and g, let $\mathcal{A}$ be the (k,g)-centroid algorithm. For all $\mathcal{I} \subseteq \mathcal{X}$, such that $\mathcal{I}$ has radius r and for any $\delta \in \left[4r, g^{-1}(\frac{|\mathcal{I}|}{|\mathcal{X}\setminus\mathcal{I}|}(g(2r)-g(r)))\right)$, $\mathcal{I}$ is $4r$-distance-robust and $g(\delta)|\mathcal{X}\setminus\mathcal{I}|$-cost-robust to $\mathcal{X}\setminus\mathcal{I}$ with respect to $R^t_\delta(\mathcal{A})$.*

Proof. Let $\mathcal{A}' = RT_\delta(\mathcal{A})$ and let $\mathcal{A}'(\mathcal{X}) = \mathcal{C}'$. For all $x \in \mathcal{X}$, let $\mu'(x)$ denote the closest center of $\mathcal{C}'$ to x. Assume on the contrary that there exists $y \in \mathcal{I}$ such that $d(y,\mu'(y)) > 4r$, then for any $y' \in \mathcal{I}$, $d(y',\mu'(y')) > 2r$. Therefore, $\Lambda^g_{d'}(\mathcal{C}') \geq |\mathcal{I}| \cdot g(2r)$. For any clustering $\mathcal{C}''$ that has a center at the center of the r-ball that covers $\mathcal{I}$, $\Lambda^g_{d'}(\mathcal{C}'') \leq |\mathcal{I}|\cdot g(r) + |\mathcal{X}\setminus\mathcal{I}|\cdot g(\delta)$. By the choice of δ, $\Lambda^g_{d'}(\mathcal{C}'') < \Lambda^g_{d'}(\mathcal{C}')$, so $\mathcal{C}'$ is not optimal. Hence, $\mathcal{I}$ is $4r$-distance-robust to $\mathcal{X}\setminus\mathcal{I}$ with respect to $RT_\delta(\mathcal{A})$. Using Lemma 3.2, $\mathcal{I}$ is also $g(\delta)|\mathcal{X}\setminus\mathcal{I}|$-cost-robust to $\mathcal{X}\setminus\mathcal{I}$ with respect to $RT_\delta(\mathcal{A})$ for cost function Λ^g_d. □

Corollary 3.3. *For all $\mathcal{I} \subseteq \mathcal{X}$, such that $\mathcal{I}$ has radius r, and for any $\delta \in \left[4r, r\sqrt{3\frac{|\mathcal{I}|}{|\mathcal{X}\setminus\mathcal{I}|}}\right)$, $\mathcal{I}$ is $4r$-distance robust and $\delta^2|\mathcal{X}\setminus\mathcal{I}|$-cost-robust to δ-k-means for the k-means cost. Similarly, for $\delta \in \left[4r, r\frac{|\mathcal{I}|}{|\mathcal{X}\setminus\mathcal{I}|}\right)$, $\mathcal{I}$ is $4r$-distance robust and $\delta|\mathcal{X}\setminus\mathcal{I}|$-cost-robust to δ-k-median for the k-median cost.*

The Underlying Structure of $\mathcal{I}$

The following results guarantee robustness, distance-robustness and cost-robustness for the δ-Truncated paradigm. Lemmas 3.3 and 3.4 examine the output of the (k,g)-centroid and (k,g)-δ-truncated algorithms when $\mathcal{I}$ has a well-clusterable underling pattern. Theorem 3.6 derives values of δ that render $\mathcal{I}$ robust to $\mathcal{X}\setminus\mathcal{I}$ with respect to a δ-Truncated algorithm based on the signal-to-noise ratio of $\mathcal{X}$ and the separability and balancedness of the underlying pattern of $\mathcal{I}$.

Lemma 3.3. *For all k and g, let $\mathcal{A}$ be the (k,g)-centroid algorithm. For any $\mathcal{I}$, such that there exists $\mathcal{B}$ that is a (ρ_1,ρ_2)-balanced set of k balls of radius r with centers at least $\nu > 4r + 2g^{-1}(\frac{\rho_1+\rho_2}{\rho_1}g(r))$ apart and $\mathcal{B}$ covers $\mathcal{I}$, then $\mathcal{A}(\mathcal{I}) = \mathcal{B}$.*

Proof. Let $\mathcal{B} = \{B_1, \ldots, B_k\}$ and for $i \in [k]$ let b_i represent the center of B_i and D_i represent a ball of radius $\frac{\nu}{2} - r$ centered at b_i. Let $\mathcal{A}(\mathcal{X}) = \mathcal{C}$ with centers $\mu_1, \ldots, \mu_k$. Let $\mathcal{D}_1 = \{D_i | \, D_i \text{ does not cover any } \mu_j\}$ and $\mathcal{D}_2 = \{D_i | \, D_i \text{ covers more than one } \mu_j\}$. Since $D_1, \ldots, D_k$ are pairwise disjoint, $|\mathcal{D}_1| \geq |\mathcal{D}_2|$. Assume in search of a contradiction that $\mathcal{D}_1 \neq \emptyset$. For any $D_i \in \mathcal{D}_1$, for all $y \in D_i$, $d(y, \mu(y)) \geq \frac{\nu}{2} - 2r$. Consider the following set of $\mu''_1, \ldots, \mu''_k$: If D_j includes exactly one center, μ_i, then let $\mu''_j = \mu_i$, otherwise $\mu''_j = b_j$.

$$\begin{aligned}
\Lambda^g_{\mathcal{I},d}(\mu''_1, \ldots, \mu''_k) &\leq \Lambda^g_{\mathcal{I},d}(\mu_1, \ldots, \mu_k) + \sum_{D_i \in \mathcal{D}_1} \sum_{y \in B_i} [g(d(y, \mu''(y))) - g(d(y, \mu(y)))] \\
&\quad + \sum_{D_i \in \mathcal{D}_2} \sum_{y \in B_i} [g(d(y, \mu''(y))) - g(d(y, \mu(y)))] \\
&\leq \Lambda^g_{\mathcal{I},d}(\mu_1, \ldots, \mu_k) + \sum_{D_i \in \mathcal{D}_1} |B_i| \left(g(r) - g(\frac{\nu}{2} - 2r) \right) + \sum_{D_i \in \mathcal{D}_2} |B_i| g(r) \\
&\leq \Lambda^g_{\mathcal{I},d}(\mu_1, \ldots, \mu_k) + \rho_1 |\mathcal{D}_1||\mathcal{I}| \left(g(r) - g(\frac{\nu}{2} - 2r) \right) + \rho_2 |\mathcal{D}_2||\mathcal{I}| g(r) \\
&\leq \Lambda^g_{\mathcal{I},d}(\mu_1, \ldots, \mu_k) + |\mathcal{D}_1||\mathcal{I}| \left((\rho_1 + \rho_2) g(r) - \rho_1 g(\frac{\nu}{2} - 2r) \right) \\
&< \Lambda^g_{\mathcal{I},d}(\mu_1, \ldots, \mu_k)
\end{aligned}$$

This forms a contradiction, so without loss of generality every D_i covers a center μ_i. For $i \neq j$ and for all $y \in B_i$, $d(y, \mu_i) \leq \frac{\nu}{2} < d(y, \mu_j)$. Therefore, $\mathcal{A}(\mathcal{I}) = \mathcal{B}$. □

Lemma 3.4. *For all k and g, let $\mathcal{A}$ be the (k, g)-centroid algorithm. For any $\mathcal{I} \subseteq \mathcal{X}$, such that there exists $\mathcal{B}$ that is a (ρ_1, ρ_2)-balanced set of k balls, each of radius r, and centers that are at least $\nu > 4r + 2g^{-1}(\frac{\rho_1 + \rho_2}{\rho_1} g(r))$ apart, and $\mathcal{B}$ covers $\mathcal{I}$, for any $\delta \in \left[\frac{\nu}{2}, g^{-1} \left(\frac{|\mathcal{I}|}{|\mathcal{X} \setminus \mathcal{I}|} (\rho_1 g(\frac{\nu}{2} - 2r) - (\rho_1 + \rho_2) g(r)) \right) \right)$, if $\mathcal{A}' = R^t_\delta(\mathcal{A})$,*

- *$\mathcal{A}'(\mathcal{X}) | \mathcal{I} = \{B_1, \ldots, B_k, \emptyset\}$*
- *For all $y \in \mathcal{I}$, $d(y, \mu'(y)) \leq min\{\nu/2, g^{-1} \left(g(\frac{\nu}{2} - 2r) - \frac{\rho_2}{\rho_1} g(r) \right) + 2r\}$.*

Proof. We use a similar approach as in Lemma 3.3. Let $\mathcal{B} = \{B_1, \ldots, B_k\}$ and for $i \in [k]$ let b_i represent the center of B_i and D_i represent a ball of radius $\frac{\nu}{2} - r$ centered at b_i. Let $\mathcal{A}'(\mathcal{X}) = \mathcal{C}'$ with centers, $\mu'_1, \ldots, \mu'_k$ that minimize $\Lambda^g_{d'}$. Let $\mathcal{D}_1 = \{D_i | D_i \text{ does not cover any } \mu'_j\}$ and $\mathcal{D}_2 = \{D_i | D_i \text{ covers more than one } \mu'_j\}$. Since $D_1, \ldots, D_k$ are pairwise disjoint, $|\mathcal{D}_1| \geq |\mathcal{D}_2|$. Assume in search of a contradiction that $\mathcal{D}_1 \neq \emptyset$. For any $D_i \in \mathcal{D}_1$, for all $y \in D_i$, $d(y, \mu'(y)) \geq \frac{\nu}{2} - 2r$. Consider the following set of $\mu''_1, \ldots, \mu''_k$: If D_j includes exactly one center, μ'_i, then let $\mu''_j = \mu'_i$, otherwise $\mu''_j = b_j$.

$$\begin{aligned}
\Lambda^g_{\mathcal{X},d'}(\mu''_1, \ldots, \mu''_k) \leq & \Lambda^g_{\mathcal{X},d'}(\mu'_1, \ldots, \mu'_k) + \sum_{D_i \in \mathcal{D}_1} \sum_{y \in B_i} [g(d'(y, \mu''(y))) - g(d'(y, \mu'(y)))] \\
& + \sum_{D_i \in \mathcal{D}_2} \sum_{y \in B_i} [g(d'(y, \mu''(y))) - g(d'(y, \mu'(y)))] + \sum_{y \in \mathcal{X} \setminus \mathcal{I}} g(d'(y, \mu''(y))) \\
\leq & \Lambda^g_{\mathcal{X},d'}(\mu'_1, \ldots, \mu'_k) + |\mathcal{D}_1||\mathcal{I}|\rho_1 \left(g(r) - g(\frac{\nu}{2} - 2r) \right) + |\mathcal{D}_2||\mathcal{I}|\rho_2 g(r) + |\mathcal{X} \setminus \mathcal{I}| g(\delta) \\
\leq & \Lambda^g_{\mathcal{X},d'}(\mu'_1, \ldots, \mu'_k) + |\mathcal{D}_1||\mathcal{I}| \left((\rho_1 + \rho_2) g(r) - \rho_1 g(\frac{\nu}{2} - 2r) \right) + |\mathcal{X} \setminus \mathcal{I}| g(\delta) \\
< & \Lambda^g_{\mathcal{X},d'}(\mu'_1, \ldots, \mu'_k)
\end{aligned}$$

This forms a contradiction, so without loss of generality let every D_i cover a center μ'_i. For $i \neq j$ and for all $y \in B_i$, $d(y, \mu'_i) \leq \frac{\nu}{2} < d(y, \mu'_j)$. Therefore, $\mathcal{A}'(\mathcal{X})|\mathcal{I} = \{B_1, \ldots, B_k, \emptyset\}$.

For every $C'_i \in \mathcal{C}'$, $|B_i| \cdot \min_{y \in B_i} g(d(y, \mu'_i)) \leq |B_i| g(r) + |C_i \setminus \mathcal{I}| g(\delta)$. Therefore, there exists $y \in B_i$, such that

$$\begin{aligned}
g(d(y, \mu'_i)) &\leq g(r) + \frac{|\mathcal{X} \setminus \mathcal{I}|}{|B_i|} g(\delta) \\
&\leq g(r) + \frac{|\mathcal{I}|}{|B_i|} \left(\rho_1 g(\frac{\nu}{2} - 2r) - (\rho_1 + \rho_2) g(r) \right) \\
&\leq g(\frac{\nu}{2} - 2r) - \frac{\rho_2}{\rho_1} g(r)
\end{aligned}$$

Hence, for all $y \in \mathcal{I}$, $d(y, \mu'(y)) \leq min\{\nu/2, g^{-1} \left(g(\frac{\nu}{2} - 2r) - \frac{\rho_2}{\rho_1} g(r) \right) + 2r\}$. □

The next theorem guarantees robustness for the δ-Truncated paradigm based on a measure of clusterability of $\mathcal{I}$. As an example, one of the implications of this theorem is as follows: For any $\mathcal{I} \subseteq \mathcal{X}$, $\mathcal{I}$ is 0-robust, $4r$-distance robust, and $\delta|\mathcal{X} \setminus \mathcal{I}|$-cost-robust to $\mathcal{X} \setminus \mathcal{I}$ with respect to δ-k-median for the k-median cost function, if $\mathcal{I}$ can be covered by a $(\frac{1}{2k}, \frac{3}{2k})$-balanced collection of k balls of radius r whose centers are atleast $14r$ apart and $\delta \in [7r, \frac{|\mathcal{I}|r}{2k|\mathcal{X} \setminus \mathcal{I}|})$.

Theorem 3.6. *For all k and g, let $\mathcal{A}$ be the (k, g)-centroid algorithm. For any $\mathcal{I} \subseteq \mathcal{X}$, such that $\mathcal{I}$ can be covered by a (ρ_1, ρ_2)-balanced set of k balls of radius r whose centers at least $\nu >$*

$4r + 2g^{-1}(\frac{\rho_1+\rho_2}{\rho_1}g(r))$ *apart, and for any* $\delta \in \left[\frac{\nu}{2}, g^{-1}\left(\frac{|\mathcal{I}|}{|\mathcal{X}\setminus\mathcal{I}|}(\rho_1 g(\frac{\nu}{2} - 2r) - (\rho_1 + \rho_2)g(r))\right)\right)$, $\mathcal{I}$ *is*

- $g(\delta)|\mathcal{X} \setminus \mathcal{I}|$*-cost-robust to* $\mathcal{X} \setminus \mathcal{I}$ *with respect to* $R^t_\delta(\mathcal{A})$ *for cost function* Λ^g_d*.*

- 0*-robust to* $\mathcal{X} \setminus \mathcal{I}$ *with respect to* $R^t_\delta(\mathcal{A})$*.*

- $min\{\nu/2, g^{-1}(g(\nu/2 - 2r) - \frac{\rho_2}{\rho_1}g(r)) + 2r\}$*-distance-robust with respect to* $R^t_\delta(\mathcal{A})$*.*

Proof. Lemma 3.4 shows that for every $y \in \mathcal{I}$, $d(y, \mu'(y)) \leq \delta$. Using Lemma 3.2, $\mathcal{I}$ is $g(\delta)|\mathcal{X} \setminus \mathcal{I}|$-cost-robust to $\mathcal{X} \setminus \mathcal{I}$ with respect to $R^t_\delta(\mathcal{A})$ for cost function Λ^g_d.

Let $B_1, \ldots, B_k$, be the described set of balls that cover $\mathcal{I}$. Lemma 3.4 and Lemma 3.3 show that $\mathcal{A}'(\mathcal{X})|\mathcal{I} = \{B_1, \ldots, B_k, \emptyset\}$ and $\mathcal{A}(\mathcal{I}) = \{B_1, \ldots, B_k\}$. Therefore, $\mathcal{I}$ is 0-robust to $\mathcal{X} \setminus \mathcal{I}$ with respect to $R^t_\delta(\mathcal{A})$.

Lemma 3.4 shows that for any $y \in \mathcal{I}$, $d(y, \mu'(y)) \leq min\{\nu/2, g^{-1}\left(g(\frac{\nu}{2} - 2r) - \frac{\rho_2}{\rho_1}g(r)\right) + 2r\}$. Therefore, $\mathcal{I}$ is $d(y, \mu'(y)) \leq min\{\nu/2, g^{-1}\left(g(\frac{\nu}{2} - 2r) - \frac{\rho_2}{\rho_1}g(r)\right) + 2r\}$-distance-robust to $\mathcal{X} \setminus \mathcal{I}$ with respect to $R^t_\delta(\mathcal{A})$ □

The Underlying Structure of $\mathcal{I}$ and Convexity of g

The following results guarantee a level of robustness for the δ-Truncated paradigm based the value of δ, the internal structure of $\mathcal{I}$, and the signal-to-noise ratio of $\mathcal{X}$. Lemmas 3.7 and 3.8 examine the output of the (k, g)-centroid and (k, g)-δ-Truncated algorithms when g is convex and $\mathcal{I}$ has a σ-separable, ρ-balanced underlying clustering of diameter s. Theorem 3.7 proves a lower-bound on the robustness (an upper-bound on the value of γ-robustness) of $\mathcal{I}$ to $\mathcal{X} \setminus \mathcal{I}$ with respect to (k, g)-δ-Truncated algorithms when g is convex and $\mathcal{I}$ has the mentioned underlying clustering.

In this section, we assume that function g in addition to being continuous, increasing, and unbounded (which are the standard requirement) is also convex. We call such functions, simply, as *convex*. The following lemma shows an important property of these functions that helps us with bounding the cost of a clustering.

Lemma 3.5. *For any $x, y \in \mathcal{X}$, a metric distance function d, and a convex function g,*

$$g(d(x,c)) + g(d(y,c)) \geq 2g\left(\frac{d(x,y)}{2}\right)$$

Proof. In the following, the first inequality holds by the convexity of g and the second inequality holds by the fact that g is increasing and d satisfies the triangle inequality.

$$g(d(x,c)) + g(d(y,c)) \geq 2g\left(\frac{d(x,c)+d(y,c)}{2}\right) \geq 2g\left(\frac{d(x,y)}{2}\right)$$

□

Lemma 3.6. *[4, Lemma 5.3] Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be two clusterings of $\mathcal{Y}$, where $\mathcal{C}_1$ is ρ-balanced and has k clusters. If $\Delta(\mathcal{C}_1, \mathcal{C}_2) \geq \gamma$, then the number of disjoint pairs $\{x, y\} \subseteq \mathcal{Y}$ such that $x \not\sim_{\mathcal{C}_1} y$ and $x \sim_{\mathcal{C}_2} y$ is at least $\frac{1}{2}(\gamma - k\rho^2)|\mathcal{Y}|$.*

Lemma 3.7. *For any k and convex function g, let $\mathcal{A}$ be the (k, g)-centroid algorithm. Let $\mathcal{I}$ have an σ-separable, ρ-balanced clustering of diameter s, namely $\mathcal{B}$. Then,*

$$\Delta(\mathcal{A}(\mathcal{I}), \mathcal{B}) \leq \frac{g(s)}{g(\sigma s/2)} + k\rho^2$$

Proof. Let $\mathcal{A}(\mathcal{I}) = \mathcal{C}$ with centers $\mu_1, \ldots, \mu_k$. Let $\Delta(\mathcal{B}, \mathcal{C}) = \gamma$ and assume, in search of a contradiction, that $\gamma > \frac{g(s)}{g(\sigma s/2)} + k\rho^2$. For any $\{x, y\} \in \mathcal{I}$ such that $x \not\sim_{\mathcal{B}} y$ but $x \sim_{\mathcal{C}} y$, using Lemma 3.5, $g(d(x, \mu_i)) + g(d(y, \mu_i)) \geq 2g(\frac{\sigma s}{2})$. Lemma 3.6, shows that there are at least $\frac{1}{2}(\gamma - k\rho^2)|\mathcal{I}|$ many such disjoint pairs. Therefore,

$$\begin{aligned}\Lambda_d^g(\mathcal{C}) &\geq \frac{1}{2}(\gamma - k\rho^2)|\mathcal{I}|2g(\sigma s/2)\\ &> g(s)|\mathcal{I}|\\ &> \Lambda_d^g(\mathcal{B})\end{aligned}$$

This forms a contradiction. Therefore, $\Delta(\mathcal{A}(\mathcal{I}), \mathcal{B}) \leq \frac{g(s)}{g(\sigma s/2)} + k\rho^2$. □

Lemma 3.8. *For any k and convex function g, let $\mathcal{A}$ be the (k, g)-centroid algorithm. For all $\mathcal{I} \subseteq \mathcal{X}$ that has a σ-separable, ρ-balanced k-clustering of diameter s, namely $\mathcal{B}$ and any $\delta > \frac{\sigma s}{2}$, if $\mathcal{A}'$ is $R_\delta^t(\mathcal{A})$,*

$$\Delta(\mathcal{B}, \mathcal{A}'(\mathcal{X})|\mathcal{I}) \leq \frac{\frac{|\mathcal{X}\setminus\mathcal{I}|}{|\mathcal{I}|}g(\delta) + g(s)}{g(\sigma s/2)} + k\rho^2$$

Proof. Let $\mathcal{A}'(\mathcal{X}) = \mathcal{C}'$ with centers $\mu'_1, \ldots, \mu'_k$. Let $\Delta(\mathcal{B}, \mathcal{C}') = \gamma$ and assume on the contrary that $\gamma > \frac{\frac{|\mathcal{X} \setminus \mathcal{I}|}{|\mathcal{I}|} g(\delta) + g(s)}{g(\sigma s/2)} + k\rho^2$. Using Lemma 3.5, for any $\{x, y\} \in \mathcal{I}$ such that $x \not\sim_{\mathcal{B}} y$ but $x \sim_{\mathcal{C}'} y$, $g(d'(x, \mu'_i)) + g(d'(y, \mu'_i)) \geq \min\{2g(\delta), 2g(\frac{\sigma s}{2})\} \geq 2g(\frac{\sigma s}{2})$. Lemma 3.6 shows that there are at least $\frac{1}{2}(\gamma - k\rho^2)|\mathcal{I}|$ many such disjoint pairs. Therefore,

$$\begin{aligned}
\Lambda^g_{d'}(\mathcal{C}) &\geq g(\frac{\sigma s}{2})(\gamma - k\rho^2)|\mathcal{I}| \\
&> g(\frac{\sigma s}{2}) \frac{\frac{|\mathcal{X} \setminus \mathcal{I}|}{|\mathcal{I}|} g(\delta) + g(s)}{g(\sigma s/2)} |\mathcal{I}| \\
&> g(s)|\mathcal{I}| + |\mathcal{X} \setminus \mathcal{I}| g(\delta) \\
&> \Lambda^g_{d'}(\mathcal{B} \cup \{\mathcal{X} \setminus \mathcal{I}\})
\end{aligned}$$

Contradiction. Therefore, $\Delta(\mathcal{B}, \mathcal{A}'(\mathcal{X})|\mathcal{I}) \leq \frac{\frac{|\mathcal{X} \setminus \mathcal{I}|}{|\mathcal{I}|} g(\delta) + g(s)}{g(\sigma s/2)} + k\rho^2$. □

Theorem 3.7. *For any k and convex function g, let $\mathcal{A}$ be the (k, g)-centroid algorithm. For all $\mathcal{I} \subseteq \mathcal{X}$ that has an σ-separable, ρ-balanced clustering of diameter s, and for any $\delta > \sigma s/2$, $\mathcal{I}$ is γ-robust to $\mathcal{X} \setminus \mathcal{I}$ with respect to $R^t_\delta(\mathcal{A})$, for*

$$\gamma \leq \frac{\left(\frac{|\mathcal{X} \setminus \mathcal{I}|}{|\mathcal{I}|} + 1\right) g(\delta) + 2g(s)}{g(\sigma s/2)} + 2k\rho^2$$

Proof. Let $\mathcal{B}$ be the σ-separable, ρ-balanced, k-clustering of diameter s that covers $\mathcal{I}$, and let $\gamma' = \frac{\frac{|\mathcal{X} \setminus \mathcal{I}|}{|\mathcal{I}|} g(\delta) + g(s)}{g(\sigma s/2)} + k\rho^2$, and $\gamma'' = \frac{g(\delta) + g(s)}{g(\sigma s/2)} + k\rho^2$. Lemmas 3.7 and 3.8 respectively show that $\Delta(\mathcal{B}, \mathcal{A}(\mathcal{I})) \leq \gamma'$ and $\Delta(\mathcal{B}, \mathcal{A}'(\mathcal{X})|\mathcal{I}) \leq \gamma''$. Using Theorem 3.1, $\Delta(\mathcal{A}(\mathcal{I}), \mathcal{A}'(\mathcal{X})|\mathcal{I}) \leq \gamma' + \gamma'' \leq \gamma$. □

3.6.3 Robustness of δ-Truncated vs. p-Increased

Here, we will further demonstrate the limitations of p-Increased algorithms for clustering noisy data compared to the strengths of δ-Truncated algorithms.

Example 9. *Let $\mathcal{A}$ denote the k-means algorithm. According to Corollary 3.2, there exists a set $\mathcal{X}$ and $\mathcal{I} \subseteq \mathcal{X}$, such that $\mathcal{X}$ has diameter 1, $|\mathcal{X}| = n$, $|\mathcal{I}| = 0.9n$, $\mathcal{I}$ has radius 0, but for any*

$p < 0.1n - k$, $\mathcal{I}$ is not $5/n^2$-distance robust to $\mathcal{X} \setminus \mathcal{I}$ with respect to $RI_p(\mathcal{A})$. Since $\mathcal{I}$ has radius 0, we can cover it by a ball of any radius. Let us choose a ball of radius $1/4n^2$. According to Corollary 3.3, for $\delta \in [\frac{1}{n^2}, \frac{3\sqrt{3}}{4n^2})$, $\mathcal{I}$ is $1/n^2$-distance-robust to $\mathcal{X} \setminus \mathcal{I}$ with respect to $RT_\delta(\mathcal{A})$.

3.7 Conclusions

In this chapter, we examined the problem of robustness of clustering algorithms to the addition of unstructured data points (that we termed "noise"). We proposed to transform any given centroid-based clustering algorithm to an efficient "noise-robust" one that has an additional cluster used as a "garbage collector". We introduced rigorous notions of robustness that capture different aspects of robustness that may be desirable for such algorithms. We proved that our algorithmic paradigm indeed guarantees desirable noise robustness, and showed that the simple strategy of just applying the ground clustering algorithms with extra clusters (to accommodate such noisy data) does not enjoy similar performance.

Chapter 4

Concluding Remarks

In this thesis, we discussed the issue of clustering in the presence of noise in two parts. In the first part, we developed a framework for clustering with a noise cluster. In the second part, we examined the robustness of clustering algorithms with respect to the addition of the unstructured data.

In the first part, we developed a formalism for clustering with a noise cluster. We introduced some input-output properties of clustering algorithms that have a noise cluster. Our properties addressed intuitive and desirable behaviour of clustering algorithms that have a noise cluster with respect to changes in the input, their computational complexity, and the richness of their clustering range. We generalized two existing algorithms with a noise cluster, (k, g)-η-trimmed and $(\epsilon, minPts)$-DBScan, and introduced two new algorithms with a noise cluster, (k, g)-δ-truncated and (k, g)-δ-naive-truncated. We examined these algorithms with respect to the mentioned properties. Our analysis showed that (k, g)-δ-truncated and $(\epsilon, minPts)$-DBScan, on top of being efficient, possess most desirable properties. On the other hand, the (k, g)-η-trimmed algorithm not only lacks many desirable properties, but also is not efficient compared to the (k, g)-centroid algorithm. We also observed that the set of desirable properties satisfied by (k, g)-δ-naive-truncated is a strict subset of those satisfied by (k, g)-δ-truncated.

In the second part, we addressed the problem of noise robustness of clustering algorithms to the addition of unstructured data. We defined three rigorous measures of robustness. We also introduced the δ-Truncated robustifying paradigm that transforms any centroid-based algorithm to a noise-robust one that has a noise cluster. We proved guaranteed robustness, with respect to all three measures of robustness, for the δ-Truncated paradigm when the un-noised data satisfies some niceness properties and the noise satisfies some mildness properties. On the other hand, we showed that the p-Increased robustifying paradigm, which allows centroid-based clustering algorithms to use a few more clusters, does not enjoy similar robustness guarantees.

References

[1] Margareta Ackerman, Shai Ben-David, Simina Branzei, and David Loker. Weighted clustering. In *Proceedings of the 26th Association for the Advancement of Artificial Intelligence*, 2012.

[2] Margareta Ackerman, Shai Ben-David, and David Loker. Characterization of linkage-based clustering. In *Proceedings of the 23rd International Conference on Learning Theory*, 2010.

[3] Margareta Ackerman, Shai Ben-David, and David Loker. Towards property-based classification of clustering paradigms. In *Advances in Neural Information Processing Systems*, pages 10–18, 2010.

[4] Margareta Ackerman, Shai Ben-David, David Loker, and Sivan Sabato. Clustering oligarchies. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics*, 2013.

[5] Pankaj K. Agarwal and Cecilia Magdalena Procopiuc. Exact and approximation algorithms for clustering. *Algorithmica*, 33(2):201–226, 2002.

[6] J. A. Cuesta-Albertos, Alfonso Gordaliza, and C. Matrán. Trimmed k-means: An attempt to robustify quantizers. *The Annals of Statistics*, 25(2):553–576, 1997.

[7] Sanjoy Dasgupta. The hardness of k-means clustering. Technical Report CS2008-0916, Department of Computer Science and Engineering, University of California, San Diego, 2008.

[8] Rajesh N. Dave. Characterization and detection of noise in clustering. *Pattern Recognition Letters*, 12(11):657–664, 1991.

[9] Rajesh N. Dave. Robust fuzzy clustering algorithms. In *Proceedings of the 2nd IEEE International Conference on Fuzzy Systems*, pages 1281–1286, 1993.

[10] David L. Donoho. Breakdown properties of multivariate location estimators. Technical report, Harvard University, 1982.

[11] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of Knowledge Discovery in Databases*, volume 96, pages 226–231, 1996.

[12] Luis Ángel García-Escudero and Alfonso Gordaliza. Robustness properties of k means and trimmed k means. *Journal of the American Statistical Association*, 94(447):956–969, 1999.

[13] Luis Ángel García-Escudero, Alfonso Gordaliza, Carlos Matrán, and Agustin Mayo-Iscar. A general trimming approach to robust cluster analysis. *The Annals of Statistics*, pages 1324–1345, 2008.

[14] Frank R. Hampel. A general qualitative definition of robustness. *The Annals of Mathematical Statistics*, 42(6):1887–1896, 1971.

[15] Christian Hennig. Dissolution point and isolation robustness: robustness criteria for general cluster analysis methods. *Journal of Multivariate Analysis*, 99(6):1154–1176, 2008.

[16] Anil K. Jain and Richard C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.

[17] Nicholas Jardine and Robin Sibson. Mathematical taxonomy. *London etc.: John Wiley*, 1971.

[18] Jean-Michel Jolion and Azriel Rosenfeld. Cluster detection in background noise. *Pattern Recognition*, 22(5):603–607, 1989.

[19] Jon Kleinberg. An impossibility theorem for clustering. *Advances in Neural Information Processing Systems*, pages 463–470, 2003.

[20] Marina Meila. Comparing clusterings: an axiomatic view. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 577–584, 2005.

[21] Glenn W. Milligan and Martha C. Cooper. An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50(2):159–179, 1985.

[22] Jan Puzicha, Thomas Hofmann, and Joachim M. Buhmann. A theory of proximity based clustering: Structure detection by optimization. *Pattern Recognition*, 33(4):617–634, 2000.

[23] William M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

[24] Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.