Robustness Guarantees for Density Clustering

Heinrich Jiang
Google Research

Jennifer Jang
Uber

Ofir Nachum
Google Brain

Abstract

Despite the practical relevance of density-based clustering algorithms, there is little understanding in its statistical robustness properties under possibly adversarial contamination of the input data. We show both robustness and consistency guarantees for a simple modification of the popular DBSCAN algorithm. We then give experimental results which suggest that this method may be relevant in practice.

1 INTRODUCTION

Density-based clustering has had a large practical impact on wide range of areas in machine learning and data mining. These methods typically proceed by estimating the underlying density of the data and then the clusters would be based on certain structures of the density. The most notable such examples include DBSCAN [14] which estimates the connected components of the level-set of the density, and Mean Shift [9, 10], which clusters based on the modes of the density. There are several key advantages of density-based clustering algorithms over the more common objective-based clustering algorithms such as k-means [21] and spectral clustering [23]: density-based clustering algorithms automatically determine the number of clusters and the clusters can be of arbitrary shape and relative position to each other. With modern datasets growing in both volume and complexity, non-parametric methods such as density-based clustering may play an important role as they can automatically adapt and discover structures in the data.

One important challenge in modern data analysis is that of robustness. For example, if there is possible adversarial corruption of the data, it would be desirable to have guarantees that the clustering will not be very sensitive to such corruption. Another example where having such guarantees can be beneficial is during the data curation process: the data sources may be more willing to release their data to the curator if it can be guaranteed for each source that their additional data will not change the final outcome by much. Robustness is also intimately related to privacy [13].

While the issue of robustness has been studied for many of the objective-based clustering procedures such as k-means (e.g. [1, 24, 31]) and spectral clustering (e.g. [4, 34]), there is surprisingly little known about the robustness of density-based clustering.

In this paper, we provide guarantees of robustness under possibly adversarial contamination of the input data for a simple modification of the popular DBSCAN algorithm [14], which we call *Robust DBSCAN*. Our results hold under two non-parametric assumptions: smoothness of the underlying density function (i.e. α-Hölder continuous) and curvature of the level-sets of the density (parameterized by β), where the level-set of density function f at level λ is defined as $\{x : f(x) \geq \lambda\}$. The first assumption is standard and the second assumption ensures that the density function decays sufficiently around level-set boundaries so that they are salient enough to be estimated with statistical guarantees. The level-sets play a critical role in our analysis because it has been shown that the clustering DBSCAN produces approximates the connected components of the density level-set at a particular density level depending on the hyperparameters and number of samples drawn from the underlying density [28, 16].

Our first guarantee shows that as long as the hyperparameters are chosen appropriately depending on the number of samples n, and the number of possibly adversarial added samples, ℓ, then with high probability, the number of clusters will not change when adding these samples and that furthermore, there exists a one-to-one correspondence between the clusters and obtained on the original dataset and the clusters obtained on the contaminated dataset such that the distances between the corresponding clusters is bounded.

Proceedings of the 22$^{\text{nd}}$ International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89. Copyright 2019 by the author(s).

We then show that if the goal is to estimate the clusters corresponding to a certain density level λ, then Robust DBSCAN (if tuned appropriately given knowledge of λ and ℓ) can recover these clusters with statistical guarantees. We further show that if we have E-fraction contamination (i.e. $\ell \approx E \cdot n$) with E sufficiently small depending on the density, then as $n \to \infty$, the procedure is robust up to an error of $O\left(E^{\alpha/(\beta\cdot\alpha+\beta\cdot D)}\right)$, where D is the dimension of the data. To our knowledge, this is the first time such a guarantee has been established for a density-based clustering algorithm.

We then show on both simulated and real datasets that Robust DBSCAN is indeed stable to adversarial contaminations of the data. We design two types of adversarial contaminations: one proceeds by adding points to create the appearance of noisy clusters and the other proceeds by deleting points in an attempt to break apart clusters. We show that DBSCAN is sensitive to these contaminations while Robust DBSCAN can preserve the number of clusters as well as the clustering of the original uncontaminated points.

2 ALGORITHM

In this section, we give the forumulation for both DBSCAN and the Robust DBSCAN algorithm. We have n i.i.d. samples $X = \{x_1, ..., x_n\}$ drawn from a density f with compact support $\mathcal{X} \subseteq \mathbb{R}^D$. The DBSCAN algorithm has two hyperparameters k and ε. These hyperparmaters are often referred to as minPts and *bandwidth*, respectively. We next define core-points, which are essentially the sample points of high empirical density and are the points that end up belonging to some cluster by both DBSCAN and Robust DBSCAN. The definition is w.r.t. to the hyperparameter setting k and ε.

Definition 1 (Core-Point)**.** *$x \in X$ is a core-point if $|B(x,\varepsilon) \cap X| \geq k$, where $B(x,\varepsilon) := \{x' : |x - x'| \leq \varepsilon\}$.*

DBSCAN (Algorithm 1) then proceeds as follows: first it takes the sample points which are core-points and then constructs the ε-neighborhood graph out of them. The connected components of this graph are the clusters, and the remaining points (i.e. non-core-points) are unclustered and considered *noise-points*.

To provide more intuition, the first step of finding the core-points is equivalent to finding samples whose k-NN radius is at most ε, which means the k-NN density estimator (defined later) is above a certain fixed threshold. It has been shown that the k-NN density estimator converges to the true underlying density [12]. This implies the core-points approximate the level-set of the underlying density at some fixed density level. Then, taking the ε-neighborhood graph of the core-points groups nearby core-points together and it has been shown that the connected components of this graph approximate the connected components of the aforementioned level-set of the underlying density [5].

We note that the original DBSCAN algorithm of Ester et al. [14] is a bit different than the version of DBSCAN described here (Algorithm 1). The only difference is that in Ester et al. [14], non-core-points which are within an ε distance of a core-point are clustered with the corresponding core-point at the end (see e.g. [16]). Such points are often referred to as *border points*, which we leave unclustered here. This makes only a minor additive difference of ε in the theoretical guarantees (as shown in [16]) and does not significantly change the clustering from the original DBSCAN algorithm, so for simplicity, we use formulation of Algorithm 1. The modification to allow border points can be found in the Appendix.

Algorithm 1 DBSCAN

```
**Inputs**: $X$, $\varepsilon$, $k$
$H := \{x \in X : |B(x,\varepsilon) \cap X| \geq k\}$.
$G :=$ undirected graph with vertices $H$ and edge between $x, x' \in H$ if $|x - x'| \leq \varepsilon$.
**return** connected components of $G$.
```

Algorithm 2 Robust DBSCAN [19, 6]

```
**Input:** $X$, $\varepsilon$, $\widetilde{\varepsilon}$, $k$
$H := \{x \in X : |B(x,\varepsilon) \cap X| \geq k\}$.
$D := \text{DBSCAN}(X, \widetilde{\varepsilon}, k)$
$\mathcal{C} := \{C \cap H : C \in D\}$.
**return** $\mathcal{C}$.
```

The difference with Robust DBSCAN (Algorithm 2) is that instead of taking an ε-neighborhood graph, we use an $\widetilde{\varepsilon}$-neighborhood graph for some appropriately chosen $\widetilde{\varepsilon} > \varepsilon$. This encourages more connectivity which in turn allows us to give robustness guarantees under adversarial corruption of the data. In fact, this idea of modifying DBSCAN so that a different ε is used to choose core-points from the ε used to compute the neighborhood graph is not new. It has been studied in the context of cluster-tree estimation, known as pruning, (e.g. [19, 6]) and recent analyses of DBSCAN (e.g. [28, 16, 33]. However in all these cases, it was to establish theoretical results so that spurious clusters won't form near cluster boundaries due to variability from drawing i.i.d. samples. In this work, we show that this technique also gives robustness to adversarial contamination of the input data.

3 ROBUSTNESS

In this section, we show that if we add ℓ samples (possibly adversarially), then the number of clusters will not change and the clustering assignments are preserved (i.e. no new clusters, old clusters do not merge, etc) when using Algorithm 2.

3.1 Regularity Assumptions

We first require that the density has smoothness (i.e. α-Hölder continuous).

Assumption 1 (Smoothness)**.** *Let $0 < \alpha \leq 1$. There exists constant $C_\alpha > 0$ such that $|f(x) - f(x')| \leq C_\alpha \cdot |x - x'|^\alpha$ for all $x, x' \in \mathcal{X}$.*

We next define the (upper) level-set of a density function f.

Definition 2 (Level-set)**.** *The λ-level set of f is defined as $L_f(\lambda) := \{x \in \mathcal{X} : f(x) \geq \lambda\}$.*

The next assumption says that the level sets are smooth w.r.t. the level. We denote the ϵ-interior of A as $A \ominus \epsilon := \{x \in A, \inf_{y \in \partial A} d(x, y) \geq \epsilon\}$ (∂A is the boundary of A).

Assumption 2 (Curvature)**.** *There exists $C_\beta > 0$ and $\beta > 0$ such that the following holds. For any $0 < \lambda \leq \lambda' < ||f||_\infty$, we have $L_f(\lambda) \ominus (\iota(|\lambda' - \lambda|)) \subseteq L_f(\lambda')$ where $\iota(r) := C_\beta \cdot r^\beta$.*

This ensures that there is sufficient decay around level-set boundaries so that the level-sets are salient enough to be detected. A similar assumption appears in [17].

3.2 Supporting Results

We next define the k-NN density estimator which plays an important role about reasoning which points the procedure selects as part of a cluster.

Definition 3 (k-NN Density Estimator)**.** *Define the k-NN radius of $x \in \mathbb{R}^D$ as $r_k(x) := \inf\{r > 0 : |X \cap B(x, r)| \geq k\}$. Then the k-NN density estimator is:*

$$f_k(x) := \frac{k}{n \cdot v_D \cdot r_k(x)^D}.$$

where v_D is the volume of a unit ball in $\mathbb{R}^D$.

We give the following high-probability uniform rates of consistency for k-NN density estimation. This follows from Lemma 3 and 4 of [12] and we omit the proof.

Lemma 1 (k-NN density estimation rates)**.** *Let $0 < \delta < 1$. Suppose that f satisfies Assumption 1. Then the following holds for some constants C and C_l depending on f. Suppose k satisfies*

$$k \geq C_l \cdot \log(1/\delta)^2 \cdot \log n.$$

Then with probability at least $1 - \delta$,

$$\sup_{x \in \mathcal{X}} |f(x) - f_k(x)| \leq C \cdot \left(\frac{\log(1/\delta)\sqrt{\log n}}{\sqrt{k}} + \left(\frac{k}{n}\right)^{\alpha/D} \right).$$

3.3 Guarantees on Core-Points

We now show robustness guarantees on the core-points returned by Algorithm 2 (i.e. the samples that belong to some returned cluster). That is, when running algorithm on X vs running it on X with ℓ additional samples, then any new core-points that appear will be near the original core-points.

Theorem 1. *Suppose that Assumption 1 and 2 hold. There exists constants C_l and C depending on f such that the following holds. Let $0 < \delta < 1$ and k satisfy*

$$k \geq C_l \cdot \log(1/\delta)^2 \cdot \log n + \ell,$$

and $\widetilde{\varepsilon} \geq \varepsilon > 0$. Let $\widehat{C}$ and $\widehat{C}'$ be the core-points returned by Algorithm 2 when run on X and X', respectively. With probability at least $1 - \delta$,

$$\widehat{C}' \subseteq \widehat{C} \oplus \tilde{r},$$

where $\oplus$ denotes a tube around a set (i.e. $A \oplus r := \{x \in \mathcal{X} : \inf_{a \in A} |x - a| \leq r\}$), and

$$\tilde{r} := C \cdot \left(\left(\frac{\ell}{n\epsilon^D}\right)^{\frac{1}{\beta}} + \frac{\log(1/\delta)^{\frac{1}{\beta}} \cdot (\log n)^{\frac{1}{2\beta}}}{(k - \ell)^{\frac{1}{2\beta}}} + \left(\frac{k}{n}\right)^{\frac{\alpha}{\beta \cdot D}} \right).$$

Proof. It is clear from Algorithm 2 that

$$\widehat{C} = \{x \in X : |B(x, \varepsilon) \cap X| \geq k\}$$
$$\widehat{C}' = \{x \in X' : |B(x, \varepsilon) \cap X'| \geq k\}.$$

We note that x being a core-point (i.e. having at least k samples in its ϵ-neighborhood) is equivalent to its k-NN radius being at most ε. It suffices to show that

$$\inf_{x \in \mathcal{X} \setminus (\widehat{C} \oplus \tilde{r})} r_{k-\ell}(x) > \varepsilon. \tag{1}$$

where $r_k(x)$ is the k-NN radius of any point x. This is because when inserting ℓ points, we can only decrease the k-NN distance of any point up to its $(k - \ell)$-NN distance. Thus, if we can show that the above holds, then it will imply that $\widehat{C}' \subseteq \widehat{C} \oplus \tilde{r}$.

By Lemma 1 the following holds for some constant $C_1 > 0$ depending on f. For any $x \in \mathcal{X}$, if

$$f(x) \geq \frac{k}{n v_D \varepsilon^d} + \frac{C_1 \log(1/\delta)\sqrt{\log n}}{\sqrt{k}} + C_1 \left(\frac{k}{n}\right)^{\alpha/D}, \tag{2}$$

then $f_k(x) \geq \frac{k}{n v_D \varepsilon^d}$, or equivalently, $r_k(x) \leq \varepsilon$ implying $x \in \widehat{C}$. Thus, letting A_k be the quantity on the RHS of (2), we have

$$L_f(A_k) \subseteq \widehat{C}.$$

Hence, to show (1), it suffices to show that

$$\inf_{x \in \mathcal{X} \setminus (L_f(A_k) \oplus \tilde{r})} r_{k-\ell}(x) > \varepsilon. \tag{3}$$

Now, by Assumption 2, we have $f(x) \geq A_k - C_\beta \cdot \widetilde{r}^\beta$ implies that $x \in L_f(A_k) \oplus \widetilde{r}$. It now follows that to (3), it is enough to show that

$$\inf_{x \in \mathcal{X} \setminus L_f(A_k - C_\beta \tilde{r}^\beta)} r_{k-\ell}(x) > \varepsilon. \tag{4}$$

This can be re-written as

$$\sup_{x \in \mathcal{X} \setminus L_f(A_k - C_\beta \tilde{r}^\beta)} f_{k-\ell}(x) < \frac{k-\ell}{n \cdot v_D \cdot \varepsilon^D}.$$

By the k-NN density estimation bounds of Lemma 1, we have the following for some constant $C_2 > 0$ depending on f.

$$\begin{aligned} &\sup_{x \in \mathcal{X} \setminus L_f(A_k - C_\beta \tilde{r}^\beta)} f_{k-\ell}(x) \\ &\leq A_k - C_\beta \tilde{r}^\beta + \frac{C_2 \log(1/\delta)\sqrt{\log n}}{\sqrt{k-\ell}} + C_2 \left(\frac{k}{n}\right)^{\alpha/D}. \end{aligned}$$

It thus suffices to take

$$\begin{aligned} C_\beta \tilde{r}^\beta \geq & \frac{\ell}{n \cdot v_D \cdot \epsilon^D} + \frac{(C_1 + C_2)\log(1/\delta)\sqrt{\log n}}{\sqrt{k-\ell}} \\ & + (C_1 + C_2)\left(\frac{k}{n}\right)^{\alpha/D}, \end{aligned}$$

which holds for a sufficiently large choice of C, as desired. □

3.4 Guarantees on Clusters

The previous result of Theorem 1 bounds the distance of between the new core-points that appear after adding ℓ samples and the original core-points. We now show that the original clustering is preserved when adding ℓ samples (i.e. no new clusters appear, original clusters don't merge, if two points were in the same cluster in X, then they are also in the same cluster in X', etc). These two results combined together show that there will be a one-to-one mapping between the returned clusters by running Algorithm 2 on X and that of X', and that each cluster of the former is contained in the corresponding cluster of the latter, and the distance between the two corresponding sets (clusters) is small.

Theorem 2. *Suppose that the conditions of Theorem 1 hold. Let $\mathcal{C}$, $\mathcal{C}'$ be output of Algorithm 2 on X and X', respectively and define the minimum inter-cluster distance of the returned clusters $R := \min_{C_1, C_2 \in \mathcal{C}, C_1 \neq C_2} \min_{x_1 \in C_1, x_2 \in C_2} d(x_1, x_2)$. If additionally, the following holds:*

$$\widetilde{r} \leq \widetilde{\varepsilon} \leq \frac{1}{2}R - \widetilde{r},$$

then $|\mathcal{C}| = |\mathcal{C}'|$ (i.e. the number of clusters does not change) and there exists a one-to-one mapping of the clusters $\sigma : \mathcal{C} \to \mathcal{C}'$ such that $C \subseteq \sigma(C)$ for all $C \in \mathcal{C}$ (i.e. original clusters are preserved).

Proof. Note that any point appearing a cluster of $\mathcal{C}$ will also appear in some cluster of $\mathcal{C}'$. By Theorem 1, we have that any newly appearing points in $\mathcal{C}'$ will be at a distance of at most $\widetilde{r}$ from a point appearing originally in $\mathcal{C}$. If $\widetilde{\varepsilon} \geq \widetilde{r}$, then such new points will become reconnected to a cluster in $\mathcal{C}$. Finally, since $\widetilde{\varepsilon} \leq \frac{1}{2}R - \widetilde{r}$, this means that no two distinct clusters appearing in $\mathcal{C}$ will become merged in $\mathcal{C}'$. □

Remark 1. *It may be the case that $\widetilde{r}$ is not small enough compared to R in the above result. We note that as $k/n \to 0$, $\ell/k \to 0$, and we choose $\epsilon \approx (k/n)^{1/D}$, then $\widetilde{r}$ will go to 0. We will discuss when and why these hyperparameter choices make sense in the next section. We will also discuss what happens when $\ell = E \cdot n$ for some $0 < E < 1$ (i.e. constant E-fraction of contaminated samples).*

4 RATES OF CONSISTENCY

We now discuss the statistical consistency of Algorithm 2 when the desired density level λ in which to estimate the clusters is known (Theorem 3). We then give a culminating result about the consistency of Algorithm 2 when E-fraction of the sample are subjective to possibly adversarial contamination.

4.1 No contamination

The following result says that that Algorithm 2, when tuned appropriately to estimate the connected components of the λ-level set of the density given, can recover these connected components with finite-sample guarantees under the Hausdorff metric. Moreover, the hyperparameter settings only requires knowledge of λ and nothing else about the underlying unknown density f except a finite sample drawn from it, and the results hold for n sufficiently large depending on f. This result follows from the same arguments made to prove Theorem 1 and 2 of [16]. We give a proof sketch which discusses the differences.

Theorem 3 (Rates of consistency)**.** *Suppose that Assumption 1 and 2 hold. Let $0 < \delta < 1$, $0 < \lambda < ||f||_\infty$. There exists $C_l, C, R_\lambda > 0$ depending on f such that the following holds with probability $1 - \delta$ for n sufficiently large depending on f. Suppose that Algorithm 2 has the following settings:*

$$
\begin{aligned}
&k \geq C_l \cdot \log(1/\delta)^2 \cdot \log n, \\
&\varepsilon = \left(\frac{k}{n \cdot v_D \cdot (\lambda - \lambda \cdot 16 \log(2/\delta)\sqrt{\log n}/\sqrt{k})} \right)^{1/D}, \\
&\varepsilon \cdot \left(\frac{1 - 16\log(2/\delta)\sqrt{\log n}/\sqrt{k})}{1 - 16\log(2/\delta)\sqrt{\log n}/\sqrt[3]{k})} \right)^{1/D} \leq \widetilde{\varepsilon} < R_\lambda.
\end{aligned}
$$

Let $\mathcal{C}$ be the connected components of $L_f(\lambda)$ and $\widehat{\mathcal{C}}$ be the output of Algorithm 2 on X. Then we have $|\mathcal{C}| = |\widehat{\mathcal{C}}|$ and there exists a one-to-one mapping $\sigma : \mathcal{C} \to \widehat{\mathcal{C}}$ such that for all $C \in \mathcal{C}$, we have

$$
d_H(C, \sigma(C)) \leq C \cdot \left(\frac{\log(1/\delta)^{1/\beta} (\log n)^{1/2\beta}}{k^{1/2\beta}} + \left(\frac{k}{n}\right)^{\frac{\alpha}{\beta \cdot D}} \right),
$$

where d_H is Hausdorff distance between two sets defined as $d_H(A, B) := \max\{\sup_{a \in A} \inf_{b \in B} |a - b|, \sup_{b \in B} \inf_{a \in A} |a - b|\}$.

Proof Sketch. Theorem 3 follows from Theorem 1 and 2 of [16] but with two differences: in the previous work, $\alpha = \beta$ and the k-NN density estimation bound used was less general than Lemma 1, in that it had an upper bound on k so that the first term of the bound (of order $\log(1/\delta)\sqrt{\log n}/\sqrt{k}$) dominated the second (of order $(k/n)^{\alpha/D}$) to avoid explicitly writing out the latter term. Here, we don't enforce such an upper bound on k since we require k to grow as fast as n in a later result. The proof for Theorem 3 can be obtained by following the same steps of the proofs of Theorem 1 and 2 of [16] but using the density estimation bound of Lemma 1 and explicitly writing out both terms instead of enforcing an upper bound on k, and treating α and β separately. □

Remark 2. *The $\sqrt[3]{k}$ appearing in the inequality for $\widetilde{\varepsilon}$ is a theoretical artifact to ensure $\widetilde{\varepsilon}$ is of higher order than ε.*

4.2 E-fraction contamination

We next consider the setting where $\ell = \lfloor E \cdot n \rfloor$, that is, the number of points the adversary can add can grow linearly with the original dataset size. In such a situation, it is clear that we cannot in general consistently recover the clusters of the original density because the adversary can augment the dataset to look like a density which has sufficiently different clusters. Here, we give a result about how robust Algorithm 2 is to such setting of ℓ when trying to recover the clusters corresponding to the connected components of the λ-level set where λ is known. The goal is similar to before: ensure that the original clusters are preserved, that no new clusters arise, and that the clusters don't change much.

To provide some intuition of why Algorithm 2 can have robustness here, a key idea is that we can set k higher than ℓ. Otherwise, an adversary can simply place a clump of k points in some ball of radius smaller than ε somewhere arbitrary far away from the other clusters, and the algorithm will believe that there is at least a core-point (since there will be a sample with at most k points in its ε-neighborhood) and thus a new cluster will form. If k is sufficiently higher than ℓ, then it will be more difficult for the adversary to create a new core-point which is far away from other core-points.

Corollary 1 (Rates of consistency under E-contamination)**.** *Suppose that Assumption 1 and 2 hold. Let $\mathcal{C}$ be the connected components of $L_f(\lambda)$ and $\widehat{\mathcal{C}}$ be the output of Algorithm 2 on X'. There exists constants C_λ, C depending on f and λ such that the following holds. Suppose that $\ell = \lfloor E \cdot n \rfloor$. Choosing $k = \lfloor \bar{E} \cdot n \rfloor$ with $\bar{E} > E$, then for n sufficiently large depending on f, λ, δ, E and $\bar{E}$, and taking*

$$
\begin{aligned}
&\varepsilon := \left(\bar{E} \cdot v_D \cdot \lambda \cdot (1 - 16\log(2/\delta)\sqrt{\log n}/\sqrt{\bar{E} \cdot n}) \right)^{1/D}, \\
&\widetilde{\varepsilon} \geq \left(\bar{E} \cdot v_D \cdot \lambda \cdot (1 - 16\log(2/\delta)\sqrt{\log n}/\sqrt[3]{\bar{E} \cdot n}) \right)^{1/D}, \\
&\widetilde{r} \leq \widetilde{\varepsilon} \leq \frac{1}{2} C_\lambda - \widetilde{r}, \text{ where } \tilde{r} := C\left((E/\bar{E})^{1/\beta} + \bar{E}^{\alpha/(\beta \cdot D)} \right).
\end{aligned}
$$

then with probability at least $1 - \delta$, we have $|\mathcal{C}| = |\widehat{\mathcal{C}}|$ and there exists a one-to-one mapping $\sigma : \mathcal{C} \to \widehat{\mathcal{C}}$ such that for all $C \in \mathcal{C}$, we have

$$
d_H(C, \sigma(C)) \leq C \cdot \left((E/\bar{E})^{1/\beta} + \bar{E}^{\alpha/(\beta \cdot D)} \right).
$$

In particular, if we choose $\bar{E} = E^{D/(D+\alpha)}$ (i.e. $k = \lfloor E^{D/(D+\alpha)} \cdot n \rfloor$), then we attain the following rate:

$$
d_H(C, \sigma(C)) \leq O\left(E^{\alpha/(\beta \cdot \alpha + \beta \cdot D)} \right).
$$

Proof. The consistency result of Theorem 3 bounds the the distance between the clustering of X and the true clusters, then the robustness results Theorem 1 and 2 bounds the distance between the clustering of X' and that of X and the result follows by a triangle inequality. □

5 SIMULATIONS

We evaluate Robust DBSCAN in adversarial settings on the common benchmarks "Circles" and "Moons" as

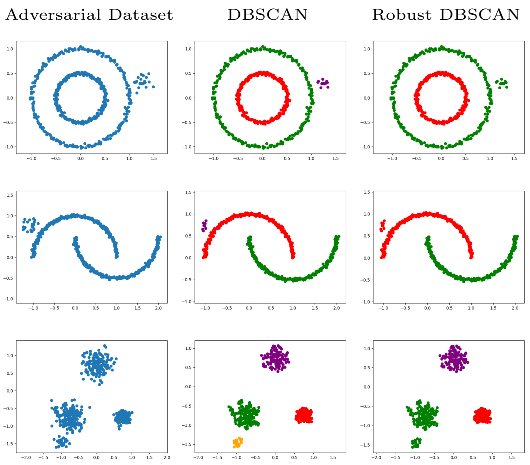

Figure 1: Adversarial augmentation applied to datasets and the resulting clusterings of DBSCAN and Robust DBSCAN.

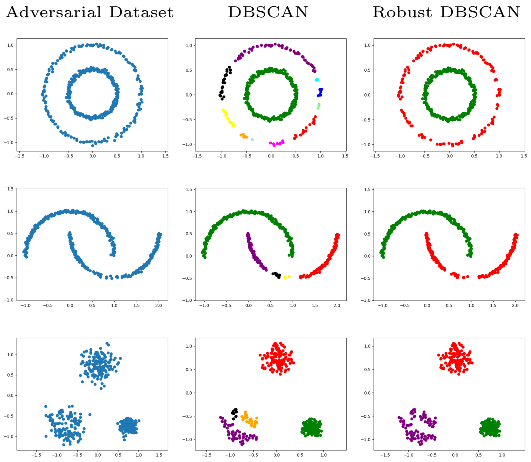

Figure 2: Adversarial deletion applied to datasets and the resulting clusterings of DBSCAN and Robust DBSCAN.

well as a mixture of three Gaussians. For each dataset, we evaluate DBSCAN and Robust DBSCAN on an adversarial version of the dataset via either point augmentation or point deletion. To adversarially augment the data, we compute its convex hull, and then add a number of points sampled from a Gaussian positioned slightly outside this hull. In this way, the adversary may add a small number of points which cause a clustering algorithm to produce an additional cluster beyond the desired, true clustering. To adversarially delete points, we randomly choose two points in the same cluster, and then compute the minimum vertex cut on the neighborhood graph. That is, we find the minimum number of points to remove from the clustering so that the two chosen points are not reachable from each other using hops between datapoints within an ϵ-ball [22, 11]. In this way, the adversary may remove a small number of points which cause a single cluster to be clustered as two distinct clusters.

Results are presented in Figures 1 and 2. We find that DBSCAN is vulnerable to the adversarial dataset perturbations we described. Even though the adversarial perturbations are slight, DBSCAN yields clusterings which are substantially distinct from the true data clustering. On the other hand, Robust DBSCAN is able to successfully preserve the true clusterings, despite the adversarial augmentations or deletions.

6 EXPERIMENTS

We test the robustness properties of Robust DBSCAN against DBSCAN on 9 benchmark UCI datasets [20]. We use the same adversarial contamination methods as used for the simulations. For each dataset, the setting of k and ε were the same for DBSCAN and Robust DBSCAN and we chose $\tilde{\varepsilon} = 1.2 \cdot \varepsilon$ for Robust DBSCAN. We then choose k and ε for each dataset so that DBSCAN and Robust DBSCAN gave the same clustering on the initial dataset.

We stress that the purpose of these experiments is not necessarily to show that our algorithms provides accurate clusterings, but rather show how stable they are to adversarial contamination of the data. As clustering is an unsupervised method, oftentimes the ground truth is not well defined or not available in practice and there is still no agreement in the appropriate notion of clusters [2]. Thus, our real-data experiments will simply focus on robustness in the context of the *density-based* notion of clusters. As such, the number of clusters found by our algorithms may not always match the number of labels in the dataset.

The adversarial augmentation tries to create the appearance of new clusters and the adversarial deletion tries to break apart clusters. Hence, in either situation, an increase in the number of clusters found after adversarial perturbation indicates that the algorithm was not robust. Thus, observing how much the number of returned clusters increases under adversarial perturbation is a reasonable way to test the robustness of the algorithm.

In Figure 3 we test the robustness of our algorithms under adversarial augmentation as we increase the number of additions to the dataset. Since the augmentation is randomized, we ran it multiple times and showed the average across 100 runs. We see that number of cluster increases considerably slower for Robust DBSCAN than DBSCAN thus showing that Robust DBSCAN is stable to additive perturbations.

In Figure 4 we test robustness under adversarial dele-

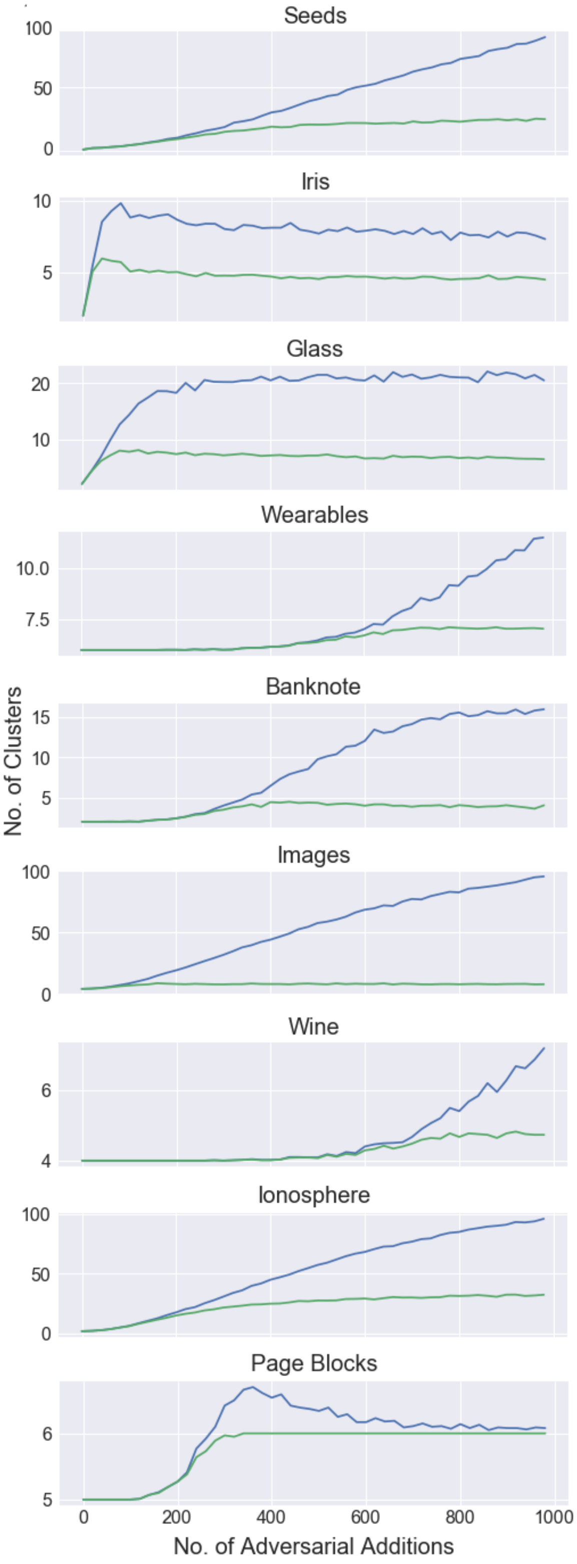

Figure 3: **Adversarial augmentation**. We plot the number of clusters returned by DBSCAN (in blue) and Robust DBSCAN (in green) as the number of adversarial additions increases. We see that Robust DBSCAN is better able to maintain the original number of clusters.

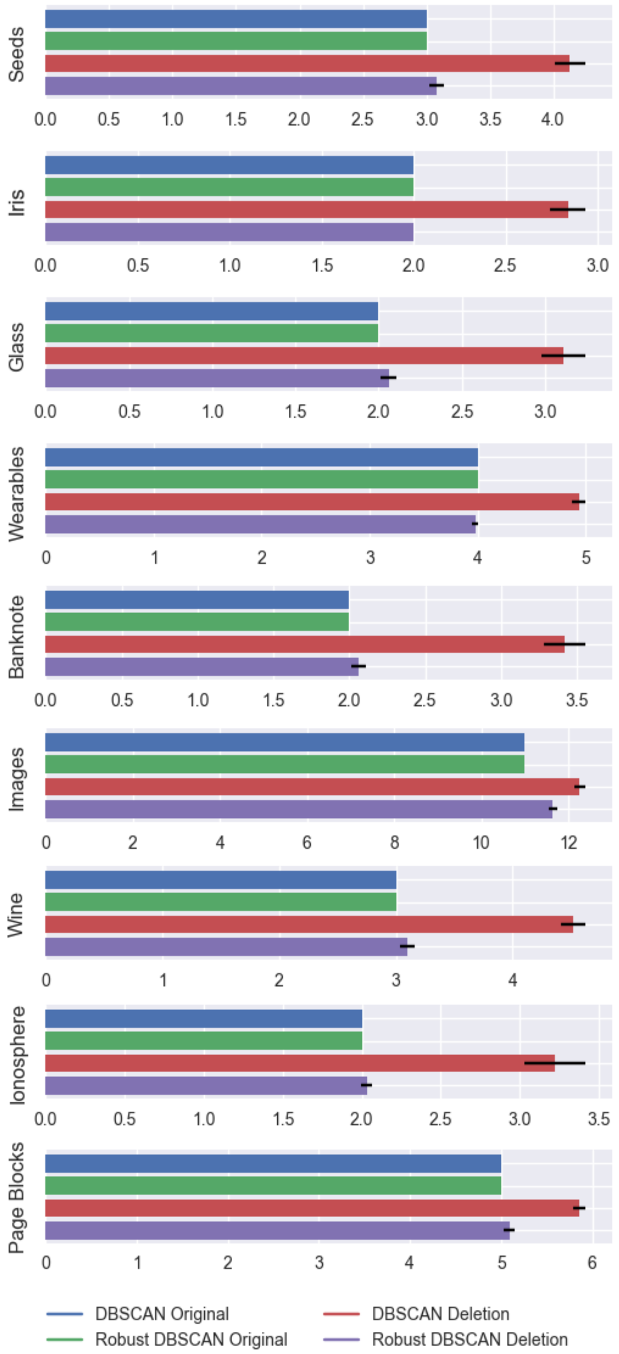

Figure 4: **Adversarial deletion**. We show the number of clusters before and after adversarial deletion averaged over multiple runs for both DBSCAN and Robust DBSCAN. We see that the number of clusters is stable for Robust DBSCAN under adversarial deletion while for DBSCAN the number increases substantially.

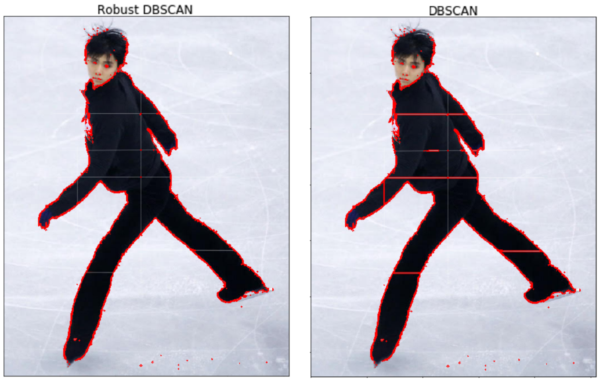

Figure 5: **Image Segmentation**. We apply density clustering on the pixels of an image to produce a segmentation, where each pixel is represented as a 5-dimensional point consisting of position and RGB color channel and the final clusters are the segments. We use an image that has been corrupted with vertical and horizontal white lines. We show that, compared to DBSCAN, with the same hyperparameters $k = 10$ and $\epsilon = 10$, Robust DBSCAN is able to recover the segments without letting the corrupted pixels affect the result while the corruption causes DBSCAN to oversegment. This suggests that our robust algorithm is versatile and can be applied to damaged images.

tion. For both algorithms, we show the number of clusters found on the original and perturbed dataset. Since the perturbation is randomized, we show the average over 100 runs and also show the standard error bar. We see that for adversarial deletion, Robust DBSCAN is able to maintain the same number of clusters while the number of clusters DBSCAN finds increases considerably.

7 RELATED WORKS

Density level-set estimation has a long history. [15] first formulated the notion of clusters as the connected components of level-sets. [32] then provided consistent procedures as well as lower bounds for estimating the level-sets. There have since been a number of analyses (see e.g. [30, 18, 3, 25, 26, 7, 8]). [27] provided the first result under the Hausdorff metric, which is a strong notion of consistency since it provides a uniform guarantee on the samples. However, such approaches which attained the strongest consistency results were largely unimplementable and thus had little practical value. It was only recently shown that DBSCAN [14], an algorithm that has already shown to be of high practical value, was able to attain the strongest consistency results [28, 16, 33, 29]. This observation was largely due to the fact that DBSCAN works in similar way to procedures which estimate the cluster-tree, which is the hierarchical nesting structure of the clusters at varying density levels (see e.g. [5, 6]), and much of the techniques developed in this line of work was applicable in studying DBSCAN. The results in this paper borrow some of these previous results and are also under the strong notions of consistency (i.e. finite-sample rates under Hausdorff metric).

As mentioned earlier, the Robust DBSCAN idea has appeared before in cluster-tree estimation as a way to *prune* the neighborhood graph [19, 6] as well as previous analyses of DBSCAN [28, 16]. In all such cases, it was to ensure that the clustering was robust to the variability of the data near the boundary of the clusters. That is, at cluster boundaries, samples are sensitive in whether they are classified as core-points or not, and without increasing the connectivity of the neighborhood graph, small spurious clusters may arise in these areas.

However, previous analyses analyze consistency under no contamination. In this work, we analyze robustness in the face of possibly adversarial contamination of the data. Some key differences from previous works include the following. First, we give precise requirements for parameter settings in terms of the number of contaminated samples. Second, with the contamination of the data, the sample drawn is *biased* and previous instantiations of the procedure will in general not be robust since the clusters of corresponding to the biased distribution can possibly be very different from that of the true distribution. For example, previous works such as [16] assume that $k/n \to 0$ and/or $\widehat{\varepsilon} \to 0$, which as shown earlier will fail with sufficient contamination of the data. Lastly, we provide a novel robustness rate in terms of E, the fraction of contaminated data.

8 CONCLUSION

Density clustering has played a key role in data analysis but little is understood about its statistical robustness properties under possibly adversarial contamination, an important challenge in modern data analysis. In this paper, we provided the first theoretical and experimental analysis of robustness for a simple modification of DBSCAN showing that robustness can be obtained for density clustering.

Acknowledgements

We thank the anonymous reviewers for their helpful suggestions to improve the paper.

References

[1] LNF Ana and Anil K Jain. Robust data clustering. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–II. IEEE, 2003.

[2] Shai Ben-David. Clustering-what both theoreticians and practitioners are doing wrong. *arXiv preprint arXiv:1805.08838*, 2018.

[3] Benoı^t Cadre. Kernel estimation of density level sets. *Journal of multivariate analysis*, 97(4):999–1023, 2006.

[4] Hong Chang and Dit-Yan Yeung. Robust path-based spectral clustering. *Pattern Recognition*, 41 (1):191–203, 2008.

[5] Kamalika Chaudhuri and Sanjoy Dasgupta. Rates of convergence for the cluster tree. In *Advances in Neural Information Processing Systems*, pages 343–351, 2010.

[6] Kamalika Chaudhuri, Sanjoy Dasgupta, Samory Kpotufe, and Ulrike von Luxburg. Consistent procedures for cluster tree estimation and pruning. *IEEE Transactions on Information Theory*, 60(12): 7900–7912, 2014.

[7] Yen-Chi Chen, Christopher R Genovese, Larry Wasserman, et al. A comprehensive approach to mode clustering. *Electronic Journal of Statistics*, 10(1):210–241, 2016.

[8] Yen-Chi Chen, Christopher R Genovese, and Larry Wasserman. Density level sets: Asymptotics, inference, and visualization. *Journal of the American Statistical Association*, pages 1–13, 2017.

[9] Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE transactions on pattern analysis and machine intelligence*, 17(8):790–799, 1995.

[10] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):603–619, 2002.

[11] George Bernard Dantzig and DR Fulkerson. On the max flow min cut theorem of networks. 1955.

[12] Sanjoy Dasgupta and Samory Kpotufe. Optimal rates for k-nn density and mode estimation. In *Advances in Neural Information Processing Systems*, pages 2555–2563, 2014.

[13] Cynthia Dwork and Jing Lei. Differential privacy and robust statistics. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 371–380. ACM, 2009.

[14] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[15] John A Hartigan. *Clustering algorithms*, volume 209. Wiley New York, 1975.

[16] Heinrich Jiang. Density level set estimation on manifolds with dbscan. In *International Conference on Machine Learning*, pages 1684–1693, 2017.

[17] Heinrich Jiang. On the consistency of quick shift. In *Neural Information Processing Systems (NIPS)*, 2017.

[18] Jussi Klemelä. Visualization of multivariate density estimates with level set trees. *Journal of Computational and Graphical Statistics*, 13(3):599–620, 2004.

[19] Samory Kpotufe and Ulrike von Luxburg. Pruning nearest neighbor cluster trees. *arXiv preprint arXiv:1105.0540*, 2011.

[20] M. Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`.

[21] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2): 129–137, 1982.

[22] Karl Menger. Zur allgemeinen kurventheorie. *Fundamenta Mathematicae*, 10(1):96–115, 1927.

[23] Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pages 849–856, 2002.

[24] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 75–84. ACM, 2007.

[25] Philippe Rigollet, Régis Vert, et al. Optimal rates for plug-in estimators of density level sets. *Bernoulli*, 15(4):1154–1178, 2009.

[26] Alessandro Rinaldo and Larry Wasserman. Generalized density clustering. *The Annals of Statistics*, pages 2678–2722, 2010.

[27] Aarti Singh, Clayton Scott, Robert Nowak, et al. Adaptive hausdorff estimation of density level sets. *The Annals of Statistics*, 37(5B):2760–2782, 2009.

[28] Bharath Sriperumbudur and Ingo Steinwart. Consistency and rates for clustering with dbscan. In *Artificial Intelligence and Statistics*, pages 1090–1098, 2012.

[29] Ingo Steinwart, Bharath K Sriperumbudur, and Philipp Thomann. Adaptive clustering using kernel density estimators. *arXiv preprint arXiv:1708.05254*, 2017.

[30] Werner Stuetzle. Estimating the cluster tree of a density by analyzing the minimal spanning tree of a sample. *Journal of classification*, 20(1):025–047, 2003.

[31] Dong Su, Jianneng Cao, Ninghui Li, Elisa Bertino, and Hongxia Jin. Differentially private k-means clustering. In *Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy*, pages 26–37. ACM, 2016.

[32] Alexandre B Tsybakov et al. On nonparametric estimation of density level sets. *The Annals of Statistics*, 25(3):948–969, 1997.

[33] Daren Wang, Xinyang Lu, and Alessandro Rinaldo. Optimal rates for cluster tree estimation using kernel density estimators. *arXiv preprint arXiv:1706.03113*, 2017.

[34] Yue Wang, Xintao Wu, and Leting Wu. Differential privacy preserving spectral graph analysis. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 329–340. Springer, 2013.