# LINSCAN - A Linearity Based Clustering Algorithm

**Andrew Dennehy**
Computational and Applied Mathematics
University of Chicago
adennehy@uchicago.edu

**Xiaoyu Zou**
Institute of Geophysics and Planetary Physics
Scripps Institution of Oceanography
University of California San Diego
x3zou@ucsd.edu

**Shabnam J. Semnani**
Department of Structural Engineering
University of California San Diego
ssemnani@ucsd.edu

**Yuri Fialko**
Institute of Geophysics and Planetary Physics
Scripps Institution of Oceanography
University of California San Diego
yfialko@ucsd.edu

**Alexander Cloninger**
Department of Mathematics and Halicioğlu Data Science Institute
University of California San Diego
acloninger@ucsd.edu

## Abstract

DBSCAN and OPTICS are powerful algorithms for identifying clusters of points in domains where few assumptions can be made about the structure of the data. In this paper, we leverage these strengths and introduce a new algorithm, LINSCAN, designed to seek lineated clusters that are difficult to find and isolate with existing methods. In particular, by embedding points as normal distributions approximating their local neighborhoods and leveraging a distance function derived from the Kullback Leibler Divergence, LINSCAN can detect and distinguish lineated clusters that are spatially close but have orthogonal covariances. We demonstrate how LINSCAN can be applied to seismic data to identify active faults, including intersecting faults, and determine their orientation. Finally, we discuss the properties a generalization of DBSCAN and OPTICS must have in order to retain the stability benefits of these algorithms.

## 1 Introduction

Many existing clustering algorithms require some prior knowledge of the dataset and are limited in the possible shapes they can identify. For example, both K-Means Clustering and Gaussian Mixture Model (GMM) Expectation Maximization require a prior estimate of the number of clusters existing in the dataset and struggle to distinguish clusters that are not linearly separable.

In contrast, DBSCAN and OPTICS iteratively generate clusters by leveraging a heuristic for the local behavior of clustered points. In particular, the designers equated clusters to connected regions of high density (Ester et al., 1996). Thus, by identifying points whose local neighborhoods are highly dense, even with little prior knowledge about the local geometry of the data, one can iteratively grow clusters from those points. The number of clusters then comes naturally from the geometry of the data itself, rather than being a parameter.

In this paper, we seek to leverage this characterization of clusters using a clustering metric other than Euclidean distance. In particular, we propose an algorithm that can distinguish between multiple

Proceedings of the 1st Conference on Topology, Algebra, and Geometry in Data Science(TAG-DS 2025).

quasi-linear clusters that may be closely spaced but have nearly orthogonal covariances. This is motivated in particular by the need to identify and map seismically active faults given a catalog of precisely located earthquakes, an important problem in geophysics (Fialko, 2021; Zou et al., 2023; Shelly et al., 2023). In addition, the potential of the algorithm is not limited to geophysics, but it may also help identify the linear spatial patterns of other natural features such as soil and airborne pollution, and man-made directional patterns including roads and hiking trials (Barden, 1963; Isaaks and Srivastava, 1989; Mai et al., 2018).

An application of the algorithm described in this paper can be found in Zou et al. (2023), wherein it was used to identify slip faults at many scales to investigate the distribution of dihedral angles of conjugate faults in the Anza-Borrego Shear Zone.

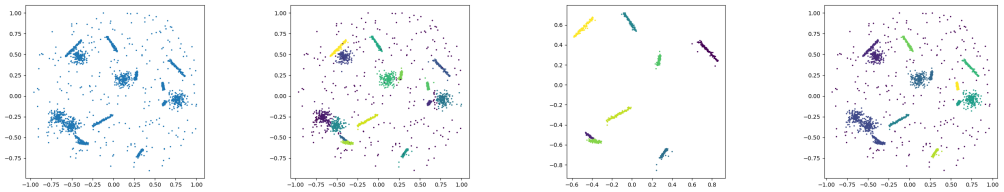
### 1.1 Motivating Problem

We wish to isolate quasi-linear clusters (QLCs) in point clouds and distinguish clusters that are geometrically close, or possibly overlap, but have different orientations. Quasi linear clusters are a cluster of points where: 1) each point is within $\epsilon$ of some other point in the cluster, 2) the total cluster has a nearly singular covariance matrix. This problem arises, for example, in geophysics, when one attempts to identify active seismogenic faults based on epicentral locations of microearthquakes (Cochran et al., 2020; Fialko, 2021; Shelly et al., 2023). Although faults are three-dimensional quasi-planar surfaces, in appropriate projections they appear as linear features, so that the associated locations of micro-earthquakes can be recognized as quasi-linear features after accounting for noise.

As an example of this task, consider a synthetic data set shown in Figure 1. The data set includes QLCs, some of which intersect each other (e.g., see around coordinate (-.4,-.6)), as well as irregularly shaped clusters and "background noise." Note how LINSCAN is capable of separating even spatially dense clusters into their component QLCs, in contrast to DBSCAN. 1c shows the results after an optional post-processing step we discuss at the start of section 5.

Figure 1: Synthetic Data

(a) Data (b) LINSCAN Results (c) Filtered Results (d) DBSCAN Results



### 1.2 Contributions

a. We design an algorithm that can be used to identify quasi-linear clusters in a point cloud without losing the stability guarantees of well-established clustering algorithms like DBSCAN and OPTICS.

b. We compare our framework to ADCN (Mai et al., 2016), a previous attempt at applying DBSCAN to a similar task, and discuss how the design of ADCN leads to the shape and number of clusters being sensitive to changes in the order of the points. This is in contrast to LINSCAN, which is invariant to the ordering of the points for clustering.

c. We prove that while our distance measure is not a metric, it satisfies positivity and symmetry on the space of Gaussian distributions (see Lemma 3.1), and a slightly relaxed form of the triangle inequality (see Theorem 3.2). These results combine to mean that clusters in this metric are stable under the order of the points and are spatially dense.

### 1.3 Notation

Here we summarize the notation that will be used throughout the rest of this paper. For finite $E \subseteq \mathbb{R}^d$, we let $\mu_E$ and $\Sigma_E$ be the sample mean and covariance of $E$. For positive definite $A$, $\|x\|_A := \sqrt{x^T A x}$ is the elliptic norm defined by $A$. Finally, a QLC is a set $S$ satisfying:

a. $\forall x \in S, \exists y \in S \setminus \{x\}$ such that $\|x - y\| < \epsilon$ for some small $\epsilon$,

b. the covariance $\Sigma_S$ satisfies $\tau < \mathrm{cond}_2(\Sigma_S) := \frac{\sigma_{\max}(\Sigma_S)}{\sigma_{\min}(\Sigma_S)}$ for some large $\tau$.

## 2 Background: DBSCAN and OPTICS

### 2.1 DBSCAN

The main principle behind DBSCAN is that clusters are equivalent to connected regions of high density. Thus, the most natural way to identify clusters is to search for points whose local neighborhoods contain a high density of points from the dataset and inductively grow clusters from those points.

In what follows, assume $X = \{x_1, ..., x_m\} \subseteq \mathbb{R}^d$ is a point cloud and let $\epsilon > 0$ and $\mathrm{minPts} \in \mathbb{N}$ be two parameters. We say $x \in X$ is a **core point** if $\#(B_\epsilon(x) \cap X) > \mathrm{minPts}$, where $B_\epsilon(x)$ is the ball of radius $\epsilon$ about $x$.

Then, for two points $p$ and $q$, we say $q$ is **core reachable** from $p$ if there exist core points $p_1, ..., p_n$ such that $p_{k+1} \in B_\epsilon(p_k)$ for all $k \in \{0, ..., n-1\}$, $p \in B_\epsilon(p_0)$, and $q \in B_\epsilon(p_n)$.

As a result, core reachability is an equivalence relation. DBSCAN then defines clusters to simply be equivalence classes under this relation, with clusters containing fewer than $\mathrm{minPts}$ points being labeled as noise. Algorithm 1 in the supplemental documents provides a pseudocode description of how this is done.

DBSCAN satisfies a few important properties. First, because core reachability is independent of the order of the points, DBSCAN is invariant under permutations of the point cloud. Furthermore, we do not need to specify the number of clusters beforehand, and all of the operations are highly efficient so long as one can efficiently calculate $B_\epsilon(x) \cap X$.

### 2.2 OPTICS

OPTICS acts as a generalization of DBSCAN, improving its robustness on datasets with regions of various densities and partially abstracting away the $\epsilon$ parameter (Ankerst et al., 1999). The most popular and effective implementation of OPTICS takes in three parameters: $\epsilon$, $\mathrm{minPts}$, and $\xi$, although $\epsilon$ is optional and only serves to shorten the run-time of the algorithm.

For $p \in X$, let $R_\delta(p) := X \cap B_\delta(p)$ for $\delta > 0$. We let the **core distance** $d_{\mathrm{core}}(p)$ be the minimum $\delta$ such that $R_\delta(p)$ contains $\mathrm{minPts}$ points. Alternatively, it is the minimum $\delta$ such that $p$ would be considered a core point if DBSCAN were to be performed using $\delta$ as $\epsilon$.

For $p, o \in X$, we define the **reachability distance** from $o$ to $p$ as $d_{\mathrm{reach}}(p|o) = \max\{d_{\mathrm{core}}(o), \|p - o\|\}$. The reachability distance describes the minimum $\epsilon$ such that $o$ is considered a core point and $p$ is contained in an $\epsilon$-neighborhood of $o$. Note that this can be infinite if $d_{\mathrm{core}}(o) = \infty$. OPTICS proceeds to develop a priority queue using a process described in the supplemental document in Algorithms 2 and 3.

While OPTICS is slightly slower than DBSCAN, it abstracts away one of the parameters, replacing it with one less tied to the geometry of $X$. Furthermore, it is far more robust to datasets with regions of varying density.

### 2.3 Related Work

The choice to use Euclidean distance with DBSCAN/OPTICS is arbitrary. The stability of the algorithm only depends on the fact that the distance function is symmetric and non-negative. Importantly, the function does not need to satisfy the triangle inequality (e.g., Khamsi and Kirk, 2011, p. 8), which allows us to work with non-metrics.

**Anisotropic DBSCAN:** The idea of extending DBSCAN/OPTICS to domains where we seek linearity is not entirely new. Previously, an algorithm called ADCN was developed to solve this problem by redefining the search neighborhoods from circles to ellipses whose eccentricity reflects the local covariance of the point (Mai et al., 2016). In practice, ADCN performs as well as DBSCAN in many tasks and performs better in cases where clusters are locally linear in otherwise highly noisy datasets.

However, ADCN is not well-suited for our task in particular because it does not provide the desired separation of adjacent or intersecting QLCs. On the contrary, it can produce artifacts around the intersection areas, say for a T-shaped intersection as in Figure 3. Furthermore, the point selection process in ADCN is non-symmetric, meaning that in certain cases the clustering behavior may be unstable to permutations of the points. Figures 3d and 3e show two runs of ADCN on the same dataset with the same parameters but with the dataset in a different order. Note how sensitive the behavior of the algorithm is to the order of the points. Our proposed algorithm performs more stably, as demonstrated below.

**Anisotropic Kernels and Spectral Clustering:** There are a large number of kernel method algorithms that use anisotropic kernels and local Mahalanobis distances to define similarity, see for example Wang et al. (2007); Talmon and Coifman (2013); Arias-Castro et al. (2017); Lahav et al. (2019); Cheng et al. (2020); Peterfreund et al. (2020). In practice, these can capture a similar notion of local similarity to our proposed approach and have been used for spectral clustering. For example, Arias-Castro et al. (2017) considers a similar problem to ours in clustering data that arises from intersecting manifolds.

However, regardless of the kernel similarity, spectral clustering and k-means (or another clustering algorithm) in the latent space fail in our noisy setting, where most of the points do not belong to any cluster. This is because k-means and spectral clustering algorithms perform poorly for data sets that are not a union of well-separated clusters (either in the original space or feature space of the kernel) Little et al. (2020), which was the motivation for the initial development of DBSCAN. There exist DBSCAN-like spectral clustering algorithms that are robust to outliers by using path-based similarity Chang and Yeung (2008); Little et al. (2020), but these algorithms have no bias towards QLCs or other degenerate clusters. For these reasons, we do not include explicit comparions to these methods in this manuscript and restrict ourselves to DBSCAN/OPTICS based algorithms.

# 3 New Algorithm: LINSCAN

## 3.1 The Embedding and Distance

LINSCAN seeks to keep the advantages of DBSCAN while being applicable to the task of distinguishing QLCs. To do this, we embed data points into $\mathbb{P}(\mathbb{R}^d)$, the space of probability measures on $\mathbb{R}^d$, and then cluster the data using a notion of distance between distributions.

LINSCAN has 3 required parameters $\mathrm{minPts}$, $\mathrm{eccPts}$, and $\xi$ and one optional parameter $\epsilon$. $\mathrm{minPts}$, $\xi$, and $\epsilon$ are identical to the corresponding parameters in OPTICS, but $\mathrm{eccPts}$ is a parameter specific to LINSCAN which determines how we form the distributions we use for clustering. Letting $R^m(x)$ be the $m$-nearest neighbors to $x$ in $X$, we define a mapping

$$x \in X \mapsto \mathcal{N}\left(\mu_{R^{\mathrm{eccPts}}(x)}, \frac{\Sigma_{R^{\mathrm{eccPts}}(x)}}{\left\|\Sigma_{R^{\mathrm{eccPts}}(x)}\right\|_2}\right)$$

Thus, we embed each point in the dataset as the normal distribution best approximating its $\mathrm{eccPts}$-nearest neighbors, which allows us to cluster the points based on the local covariance of the data. Note that we rescale the covariance matrix to have maximal eigenvalue 1.

To perform clustering in this space, we define a distance function as

$$\begin{aligned} D(P,Q) = \frac{1}{2}\left\|\Sigma_Q^{-1/2}\Sigma_P\Sigma_Q^{-1/2} - I\right\|_F + \frac{1}{2}\left\|\Sigma_P^{-1/2}\Sigma_Q\Sigma_P^{-1/2} - I\right\|_F \\ + \frac{1}{\sqrt{2}}\|\mu_P - \mu_Q\|_{\Sigma_Q^{-1}} + \frac{1}{\sqrt{2}}\|\mu_P - \mu_Q\|_{\Sigma_P^{-1}} \end{aligned}$$

where $P = \mathcal{N}(\mu_P, \Sigma_P)$ and $Q = \mathcal{N}(\mu_Q, \Sigma_Q)$ for positive definite $\Sigma_P$ and $\Sigma_Q$. Note that this function is symmetric and $D(P,Q) = 0$ if and only if $P = Q$. Although $D$ does not satisfy the triangle inequality and is thus not a metric, later we will discuss an approximate form of the triangle inequality that $D$ does satisfy (see Theorem 3.2). Note that by choosing to normalize the covariances as above, we have

$$D(P,Q) \geq \sqrt{2}\|\mu_P - \mu_Q\|_2 \tag{1}$$

Thus, points can be efficiently disqualified from consideration without having to calculate the more expensive matrix terms if the means are sufficiently far apart, which can be used to improve the run-time of the algorithm. This is, in particular, how we utilize $\epsilon$, as this means we can filter out pairs points using standard spatial methods (KD-Trees, etc.) in Euclidean space to filter out points that are sufficiently far apart without having to compute their distance in our distance measure.

Once the points have been embedded as distributions, we run OPTICS on $\mathcal{P} = \{P_i\}_{i=1}^m$ with Euclidean distance replaced by $D(\cdot, \cdot)$, and cluster $X$ based on the results. The full process is described in Algorithm 4 (see supplemental document).

### 3.2 Motivation of Distance Measure

We recall that on a probability space $\mathcal{X}$, the Kullback-Leibler Divergence between two Gaussians $P = \mathcal{N}(\mu_P, \Sigma_P)$ and $Q = \mathcal{N}(\mu_Q, \Sigma_Q)$ satisfies

$$KL(P|Q) = \frac{1}{2} \log \frac{|\Sigma_Q|}{|\Sigma_P|} + \frac{1}{2} \operatorname{tr}(\Sigma_Q^{-1}\Sigma_P - I) + \frac{1}{2}(\mu_P - \mu_Q)^T \Sigma_Q^{-1}(\mu_P - \mu_Q)$$

One can show (see supplemental document) that if $\left\| \Sigma_Q^{-1/2}\Sigma_P\Sigma_Q^{-1/2} - I \right\|_F < 1$, then

$$\begin{aligned} KL(P|Q) = \frac{1}{4}\left\| \Sigma_Q^{-1/2}\Sigma_P\Sigma_Q^{-1/2} - I \right\|_F^2 + \frac{1}{2}(\mu_P - \mu_Q)^T \Sigma_Q^{-1}(\mu_P - \mu_Q) \\ + o\left( \operatorname{tr}\left( \left( \Sigma_Q^{-1/2}\Sigma_P\Sigma_Q^{-1/2} - I \right)^3 \right) \right) \end{aligned}$$

So, we can define an approximation of $KL(P|Q)$ by

$$M(P|Q) = \frac{1}{4}\left\| \Sigma_Q^{-1/2}\Sigma_P\Sigma_Q^{-1/2} - I \right\|_F^2 + \frac{1}{2}(\mu_P - \mu_Q)^T \Sigma_Q^{-1}(\mu_P - \mu_Q)$$

This motivates the symmetric distance function $D(P, Q)$, which takes term-wise square roots of $M(P|Q)$ and $M(Q|P)$ to more closely approximate a metric.

We note that other metrics, in particular Wasserstein-2 distance, also have a closed form between Gaussians. While this is a metric, the distance between the means and covariances are independent, whereas $D$ incorporates the Mahalanobis distance and penalizes differences in mean more heavily in directions orthogonal to the local linearity of the point. Furthermore, the Wasserstein-2 distance scales polynomial in the magnitude of the eigenvalues of the covariance matrices as the angles diverge, whereas $D$ penalizes orthogonal covariance inversely to the size of the minimum eigenvalues for high eccentricity clusters. This ensures that two points with large deviations in covariance direction will be far apart in $D$, even if spatially close to one another, and thus these points will not fall into the same cluster.

### 3.3 Behavior of Distance Measure

Our distance measure is not a metric. However, in the case of Gaussians, it satisfies the properties of symmetry and separation of points in general, and, as we will show in Theorem 3.2, it satisfies a relaxed form of the triangle inequality.

**Lemma 3.1.** *Let $P = \mathcal{N}(\mu_P, \Sigma_P)$ and $Q = \mathcal{N}(\mu_Q, \Sigma_Q)$ be Gaussians. Then,*

*a. $D$ is symmetric, meaning*

$$D(P, Q) = D(Q, P)$$

*b. $D(P, Q) = 0$ iff $P = Q$ (in particular $D(P, P) = 0$)*

The proof of this lemma is in the appendix. While $D$ does not satisfy the full triangle inequality, one can show that it satisfies a slightly relaxed version. We utilize the matrix commutator $[\cdot, \cdot] : \mathbb{R}^{d\times d} \times \mathbb{R}^{d\times d} \to \mathbb{R}^{d\times d}$, which measures the degree to which two matrices commute via $[A, B] := AB - BA$.

**Theorem 3.2.** *Let $\epsilon > 0$. If $D(P,Q), D(Q,K) \leq \epsilon$, then*

$$D(P,K) \leq D(P,Q) + D(Q,K) + \sqrt{2}\epsilon + \sqrt{2}\epsilon\sqrt{1+\epsilon} + \epsilon^2 + E(P,Q,K),$$

*where $E(P,Q,K) = 0$ if $\Sigma_P$, $\Sigma_Q$, and $\Sigma_K$ commute and otherwise has a (loose) bound of*

$$\begin{aligned} E(P,Q,K) \leq\ & C_{Q,K}\left\|\left[\Sigma_P, \Sigma_Q^{-1/2}\right]\right\|_F + C_{P,Q}\left\|\left[\Sigma_K, \Sigma_Q^{-1/2}\right]\right\|_F \\ & + C'_{Q,K}\left\|\left[\Sigma_K^{-1/2}, \Sigma_Q^{-1/2}\Sigma_P\Sigma_Q^{-1/2}\right]\right\|_F + C'_{P,Q}\left\|\left[\Sigma_P^{-1/2}, \Sigma_Q^{-1/2}\Sigma_K\Sigma_Q^{-1/2}\right]\right\|_F, \end{aligned}$$

*and each constant $C_{i,j}$ depends on ratios of eigenvalues of $\Sigma_i$ and $\Sigma_j$ for $i, j \in \{P, Q, R\}$.*

The proof relies on a significant number of inequalities and is provided in the appendix. The proof proceeds by separating the first two terms of $D(P,K)$ from the last two and showing that each pair individually satisfies the triangle inequality with small additive errors.

Importantly, this shows that for small values of $\epsilon$, $D$ behaves approximately like a metric, which allows us to bound the diameter of any cluster in terms of $\epsilon$ and the number of steps between points in the cluster. This ensures that points whose local neighborhoods are nearly orthogonal are not clustered together. Compare this to the best results proven previously for the approximate triangle inequality of the unmodified KL-Divergence between Gaussians in Zhang et al. (2021), which was of exponential order.

## 4 Numerical Results

Experiments with synthetic data sets revealed that some clusters identified by LINSCAN may not appear as sufficiently "linear" upon visual inspection (e.g., due to high scatter of data points). Therefore we introduce an additional quality check whereby we compute the covariance matrix of each cluster. In the case of $\mathbb{R}^2$, we set a minimal threshold $\tau$ on the ratio of the minimum eigenvalue to the maximum eigenvalue of the covariance matrix and remove the groups that do not meet this threshold. For fair comparison, we also apply this filtering step to other clustering algorithms we test.

### 4.1 Runtime Comparisons

One possible issue with working with a custom distance measure is the cost of calculating all possible distances. In figure 2 we plot the cost of calculating all pairwise distances for datasets of various sizes as $\mathrm{eccPts}$ varies, as well as on a system with a GPU to accelerate the distance computation and one without. We can see that even for large amounts of points, the runtime for calculating both distance measures across all pairs of points is less than a second on average. For comparison, the runtime of the actual clustering algorithm is on the order of 20 seconds on our machine, so although the runtime is higher for LINSCAN's distance measure, that cost is dwarfed by the core clustering algorithm.

On top of this, if further speedups are required, we can use out-of-the box spatial indexing methods. Using 1, we can lower bound the distance between two distributions by the distance between their means, which means that we can perform an efficient initial step where we filter out pairs whose means are sufficiently far apart before calculating our distance measure on the remaining pairs.

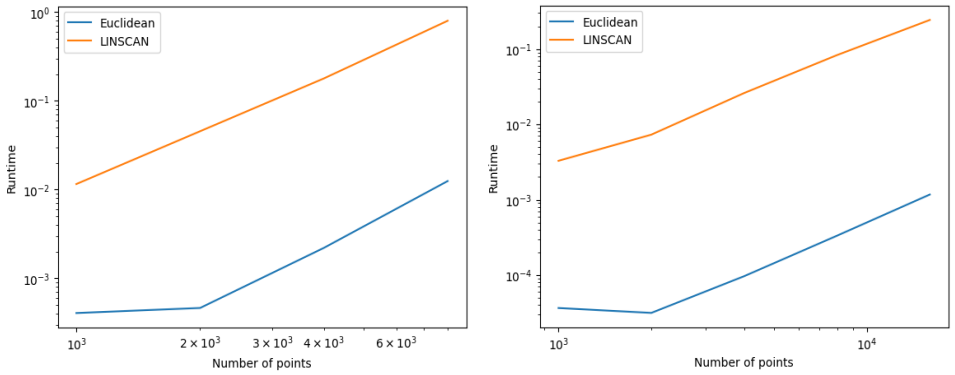
### 4.2 Example Datasets

Figure 3 shows an example of synthetic data with two QLCs intersecting at a high angle. Note that unlike DBSCAN, LINSCAN is able to separate the two QLCs. Further, we can see the dependence on point order in ADCN, as the clustering behavior is sensitive to initialization, in contrast to LINSCAN which is fully deterministic and independent of the ordering of points.

Figure 1b shows the results of applying LINSCAN to the same data as in Figure 1 and Figure 1c shows the results of labeling clusters with spectral ratio greater than $\frac{1}{2}$ as noise (and removing noise points for clarity).

Figure 4 shows the results of applying LINSCAN to real data representing earthquake epicenters in Southern California (Fialko and Jin, 2021). Not only does LINSCAN identify QLCs and remove the "diffuse" background seismicity, but it is also able to identify the clusters at multiple distinct scales

Figure 2: Distance Runtime




(a) Runtime without GPU Acceleration

(b) Runtime with GPU Acceleration

by varying $\mathrm{minPts}$. If we try to do the same thing with OPTICS we get multiple clusters, but we fail to form specifically linear clusters. For further examples of the use of LINSCAN on real data, we refer the reader to Zou et al. (2023), in particular figure 1.

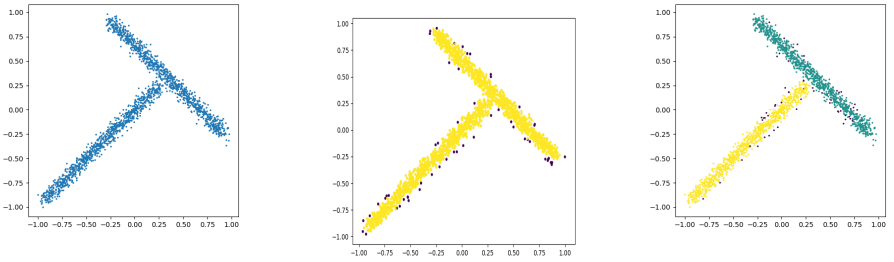
We don't contrast against ADCN on the real data, as getting a representative picture of its performance on a dataset of this size requires applying the algorithm many times from different initializations, due to the dependence of ADCN on the order of the points.

Figure 3: Crossing Lines

(a) Data

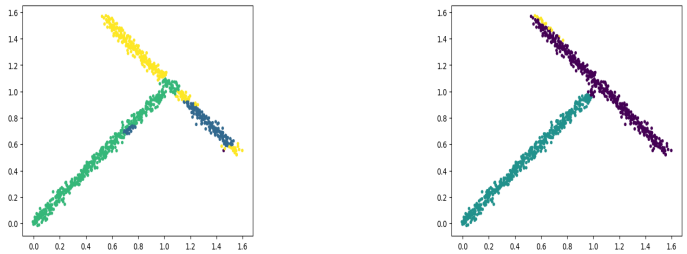
(b) DBSCAN results

(c) LINSCAN Results



(d) ADCN Result

(e) ADCN Result, Different Initialization

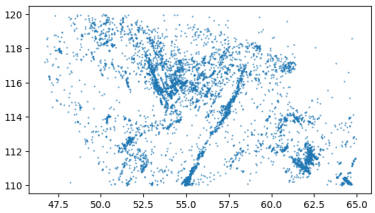


### 4.3 Measuring Performance

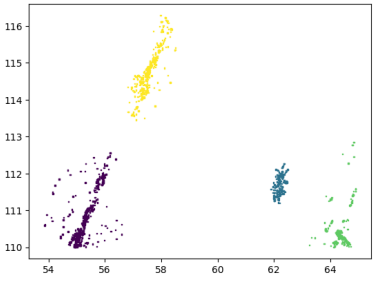
To quantitatively evaluate the algorithm performance, we conducted several tests on synthetic, labeled data. Due to the highly specific nature of the problem we are interested in solving, we are unable to use existing benchmark datasets for clustering algorithms. Each of our synthetic data examples consist of 10 linear clusters, 5 isotropic clusters, and 10 pairs of linear clusters intersecting at angles uniformly distributed in the range $[.1\pi, .9\pi]$ and separated them from one another in space. An example of one synthetic dataset is given in Figure 5a.

Figure 4: Real Data

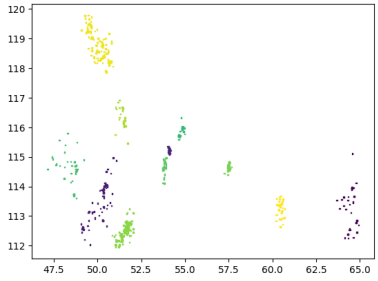
(a) Data



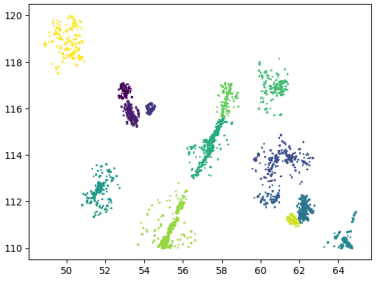
(b) Large Scale LINSCAN with Noise Removed



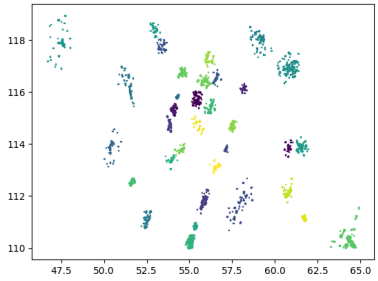
(c) Small Scale LINSCAN with Noise Removed



(d) Large Scale OPTICS with Noise Removed



(e) Small Scale OPTICS with Noise Removed



To score the performance, we use the Adjusted Rand Index from Hubert and Arabie (1985). We include a definition of this in the appendix.

## 5 Experimental results

In our synthetic experiments, we perform hyperparameter optimization of both LINSCAN and OPTICS (for comparison) on 10 synthetic datasets using a Tree-structured Parzen Estimator (Bergstra et al. (2011)) for 500 trials, applying our spectral filtering to both LINSCAN and OPTICS. We then report the test accuracy of both algorithms on 40 synthetic datasets. We report the means and 95% confidence intervals for both the validation and test data in 5.

| Algorithm | OPTICS | LINSCAN |
|---|---|---|
| Validation ARI | $0.34 \pm 0.09$ | $0.60 \pm 0.15$ |
| Testing ARI | $0.37 \pm 0.20$ | $0.51 \pm 0.14$ |

In particular, even though the parameter space for LINSCAN is much larger than OPTICS (optimizing $\mathrm{minPts}$, $\mathrm{eccPts}$, $\xi$, and $\tau$ compared to just $\mathrm{minPts}$, $\epsilon$, and $\tau$), LINSCAN performed better on both the validation data and the testing data and generalized as well as or better than OPTICS. A sample of the performance of LINSCAN and OPTICS on generated data is given in Figure 5. For ease of visualization, the noise points are filtered out.

We also remark that although many clusters which are far in space have similar colors, this is a product of the choice of colormap for plotting. All of the final clusters that both LINSCAN and OPTICS return are spatially connected.

**Remark 5.1** (Best practice for choosing hyperparameters)**.** *Here we describe informally the observed behavior of LINSCAN when the more unintuitive parameters are varied, for the benefit of would-be practitioners.*

*a.* $\xi$*: We observed that varying* $\xi$ *from the default value of* $0.05$ *from the original OPTICS paper was almost never beneficial to the final performance.*

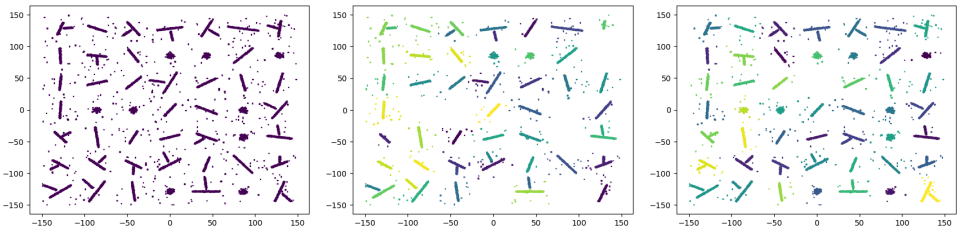
*b.* $\epsilon$*: While theoretically there is a benefit to choosing* $\epsilon$ *small to reduce the computational overhead of the distance function, the computational cost of the distance function was dwarfed by the cost of the clustering in practice. Thus, it is safe to disregard this parameter (or equivalently set it equal to* $\infty$*) outside of the largest of datasets.*

*c.* $\mathrm{eccPts}$*: Varying this parameter will adjust the scale at which QLCs are discovered, as in figure 4. For a given dataset of interest, the authors recommend visually tuning this value on a local subset of points to ensure that the extracted clusters correspond to features of the correct scale.*

*d.* $\mathrm{minPts}$*: Raising the value of this parameter makes clusters less likely, while reducing the number of noise points which appeared in the final clusters. In practice this was often the hardest parameter to tune.*

Figure 5: Generated Data

(a) Dataset (b) LINSCAN Results (c) OPTICS Results



## 6 Conclusion

We present a method for detecting linear clusters in noisy data. This is done using a novel distance measure, motivated by KL-divergence between small data-driven Gaussian representations of the points, inside of the OPTICS algorithm. We also prove that our distance measure has more regular local behavior than the standard symmetrized KL Divergence. This approach significantly outperforms the DBSCAN family of algorithms that do not have a priori bias towards lineated clusters. Finally, we have shown our approach is shown to be effective in detecting linear slip faults in seismic data and are currently exploring additional applications of our algorithm in other domains.


## Acknowledgements

The work was supported by a UCSD Chancellor's Interdisciplinary Collaboratories Grant. AD was also supported by the UCSD Undergrad Summer Research Award. AC was supported by NSF DMS 2012266. YF was supported by grants from NSF (EAR- 1841273) and NASA (80NSSC22K0506).


## References


Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M., 2019. Optuna: A next-generation hyperparameter optimization framework, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

Ankerst, M., Breunig, M.M., Kriegel, H.P., Sander, J., 1999. Optics: Ordering points to identify the clustering structure. SIGMOD Rec. 28, 49–60.

Arias-Castro, E., Lerman, G., Zhang, T., 2017. Spectral clustering based on local pca. Journal of Machine Learning Research 18, 1–57.

Barden, L., 1963. Stresses and displacements in a cross-anisotropic soil. Geotechnique 13, 198–210.

Bergstra, J., Bardenet, R., Bengio, Y., Kégl, B., 2011. Algorithms for hyper-parameter optimization, in: Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., Weinberger, K. (Eds.), Advances in Neural Information Processing Systems, Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2011/file/86e8f7ab32cfd12577bc2619bc635690-Paper.pdf.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M.J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., Zhang, Q., 2018. JAX: composable transformations of Python+NumPy programs. URL: http://github.com/jax-ml/jax.

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., Varoquaux, G., 2013. API design for machine learning software: experiences from the scikit-learn project, in: ECML PKDD Workshop: Languages for Data Mining and Machine Learning, pp. 108–122.

Chang, H., Yeung, D.Y., 2008. Robust path-based spectral clustering. Pattern Recognition 41, 191–203.

Cheng, X., Cloninger, A., Coifman, R.R., 2020. Two-sample statistics based on anisotropic kernels. Information and Inference: A Journal of the IMA 9, 677–719.

Cochran, E.S., Skoumal, R.J., McPhillips, D., Ross, Z.E., Keranen, K.M., 2020. Activation of optimally and unfavourably oriented faults in a uniform local stress field during the 2011 Prague, Oklahoma, sequence. Geophys. J. Int. 222, 153–168.

Ester, M., Kriegel, H.P., Sander, J., Xu, X., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise, in: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, AAAI Press. p. 226–231.

Fialko, Y., 2021. Estimation of absolute stress in the hypocentral region of the 2019 Ridgecrest, California, earthquakes. J. Geophys. Res. 126, e2021JB022000.

Fialko, Y., Jin, Z., 2021. Simple shear origin of the cross-faults ruptured in the 2019 Ridgecrest earthquake sequence. Nature Geoscience 14, 513–518.

Harris, C.R., Millman, K.J., van der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M.H., Brett, M., Haldane, A., del Río, J.F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., Oliphant, T.E., 2020. Array programming with NumPy. Nature 585, 357–362. URL: https://doi.org/10.1038/s41586-020-2649-2, doi:10.1038/s41586-020-2649-2.

Hubert, L., Arabie, P., 1985. Comparing partitions. Journal of Classification 2, 193–218.

Hunter, J.D., 2007. Matplotlib: A 2d graphics environment. Computing in Science & Engineering 9, 90–95. doi:10.1109/MCSE.2007.55.

Isaaks, E.H., Srivastava, M., 1989. Applied geostatistics .

Khamsi, M.A., Kirk, W.A., 2011. An introduction to metric spaces and fixed point theory. 304 pp., John Wiley & Sons.

Lahav, A., Talmon, R., Kluger, Y., 2019. Mahalanobis distance informed by clustering. Information and Inference: A Journal of the IMA 8, 377–406.

Little, A., Maggioni, M., Murphy, J.M., 2020. Path-based spectral clustering: Guarantees, robustness to outliers, and fast algorithms. Journal of machine learning research 21, 1–66.

Mai, G., Janowicz, K., Hu, Y., Gao, S., 2016. Adcn: An anisotropic density-based clustering algorithm, in: Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Association for Computing Machinery, New York, NY, USA. doi:10.1145/2996913.2996940.

Mai, G., Janowicz, K., Hu, Y., Gao, S., 2018. Adcn: An anisotropic density-based clustering algorithm for discovering spatial point patterns with noise. Transactions in GIS 22, 348–369.

Peterfreund, E., Lindenbaum, O., Dietrich, F., Bertalan, T., Gavish, M., Kevrekidis, I.G., Coifman, R.R., 2020. Local conformal autoencoder for standardized data coordinates. Proceedings of the National Academy of Sciences 117, 30918–30927.

Shelly, D.R., Skoumal, R.J., Hardebeck, J.L., 2023. Fracture-mesh faulting in the swarm-like 2020 Maacama sequence revealed by high-precision earthquake detection, location, and focal mechanisms. Geophys. Res. Lett. 50, e2022GL101233.

Talmon, R., Coifman, R.R., 2013. Empirical intrinsic geometry for nonlinear modeling and time series filtering. Proceedings of the National Academy of Sciences 110, 12535–12540.

Wang, D., Yeung, D.S., Tsang, E.C., 2007. Weighted mahalanobis distance kernels for support vector machines. IEEE Transactions on Neural Networks 18, 1453–1462.

Zhang, Y., Liu, W., Chen, Z., Li, K., Wang, J., 2021. On the properties of Kullback-Leibler divergence between Gaussians.

Zou, X., Fialko, Y., Dennehy, A., Cloninger, A., Semnani, S., 2023. High-angle active conjugate faults in the Anza-Borrego shear zone, Southern California. Geophys. Res. Lett. 50, e2023GL105783.

## A Expansion of KL-Divergence Between Gaussians

First, $\log|A|$ is the logarithm of the product of the eigenvalues of $A$, which is the same as the sum of the logarithms of the eigenvalues. Therefore,

$$\log|A| = \operatorname{tr}(\log(A))$$

where $\log(A)$ is the matrix logarithm, which exists and is unique for any positive definite matrix $A$. In particular, if $A = Q\Lambda Q^T$ for orthogonal $Q$ and diagonal $\Lambda \succ 0$,

$$\log A = Q\log(\Lambda)Q^T$$

where $\log\Lambda$ is the diagonal matrix given by applying the logarithm entrywise to each diagonal entry. Given this,

$$\begin{aligned}\log\frac{|\Sigma_Q|}{|\Sigma_P|} &= \log|\Sigma_Q| - \log|\Sigma_P| \\ &= \operatorname{tr}\left(\log(\Sigma_Q) - \log(\Sigma_P)\right)\end{aligned}$$

Next, for any positive definite matrices $A$ and $B$,

$$\begin{aligned}\operatorname{tr}\left(\log(AB)\right) &= \operatorname{tr}(\log(A)) + \operatorname{tr}(\log(B)) \\ \log(A^{-1}) &= -\log(A)\end{aligned}$$

Furthermore, if $\|A - I\| < 1$ for a submultiplicative norm $\|\cdot\|$, then the sum

$$\sum_{n=1}^{\infty}(-1)^{k+1}\frac{(A-I)^k}{k}$$

converges to $\log(A)$. Combining all of this, if $\left\|\Sigma_Q^{-1/2}\Sigma_P\Sigma_Q^{-1/2} - I\right\| < 1$ then

$$
\begin{aligned}
&\operatorname{tr}\left(\log(\Sigma_Q) - \log(\Sigma_P)\right) \\
&= -\operatorname{tr}\left(\log\left(\Sigma_Q^{-1/2}\right) + \log\left(\Sigma_P\right) + \log\left(\Sigma_Q^{-1/2}\right)\right) \\
&= -\operatorname{tr}\left(\log\left(\Sigma_Q^{-1/2}\Sigma_P\Sigma_Q^{-1/2}\right)\right) \\
&= -\operatorname{tr}\left(\sum_{k=1}^{\infty}(-1)^{k+1}\frac{\left(\Sigma_Q^{-1/2}\Sigma_P\Sigma_Q^{-1/2} - I\right)^k}{k}\right) \\
&= -\sum_{k=1}^{\infty}(-1)^{k+1}\frac{\operatorname{tr}\left(\left(\Sigma_Q^{-1/2}\Sigma_P\Sigma_Q^{-1/2} - I\right)^k\right)}{k} \\
&= -\operatorname{tr}\left(\Sigma_Q^{-1/2}\Sigma_P\Sigma_Q^{-1/2} - I\right) + \frac{1}{2}\operatorname{tr}\left(\left(\Sigma_Q^{-1/2}\Sigma_P\Sigma_Q^{-1/2} - I\right)^2\right) + o\left(\operatorname{tr}\left(\left(\Sigma_q^{-1/2}\Sigma_P\Sigma_Q^{-1/2} - I\right)^3\right)\right) \\
&= -\operatorname{tr}\left(\Sigma_Q^{-1/2}\Sigma_P\Sigma_Q^{-1/2} - I\right) + \frac{1}{2}\left\|\Sigma_Q^{-1/2}\Sigma_P\Sigma_Q^{-1/2} - I\right\|_F^2 + o\left(\operatorname{tr}\left(\left(\Sigma_Q^{-1/2}\Sigma_P\Sigma_Q^{-1/2} - I\right)^3\right)\right)
\end{aligned}
$$

where in the last line we used the fact that $\Sigma_Q^{-1/2}\Sigma_P\Sigma_Q^{-1/2} - I$ is symmetric and for any symmetric matrix $A$

$$\operatorname{tr}(A^2) = \operatorname{tr}(A^T A) = \|A\|_F^2$$

Next, note that

$$\operatorname{tr}(\Sigma_Q^{-1}\Sigma_P - I) = \operatorname{tr}(\Sigma_Q^{-1/2}\Sigma_P\Sigma_Q^{-1/2} - I)$$

So, combined with the prior derivations,

$$
\begin{aligned}
&\frac{1}{2}\log\frac{|\Sigma_Q|}{|\Sigma_P|} + \frac{1}{2}\operatorname{tr}(\Sigma_Q^{-1}\Sigma_P - I) \\
&= \frac{1}{4}\left\|\Sigma_Q^{-1/2}\Sigma_P\Sigma_Q^{-1/2} - I\right\|_F^2 + o\left(\operatorname{tr}\left(\left(\Sigma_Q^{-1/2}\Sigma_P\Sigma_Q^{-1/2} - I\right)^3\right)\right)
\end{aligned}
$$

from which the rest of the approximation follows.

## B Proof of Lemma 3.1

**Lemma.** *Let $P = \mathcal{N}(\mu_P, \Sigma_P)$ and $Q = \mathcal{N}(\mu_Q, \Sigma_Q)$ be Gaussians. Then,*

*a. $D$ is symmetric, meaning*

$$D(P, Q) = D(Q, P)$$

*b. $D(P, Q) = 0$ iff $P = Q$ (in particular $D(P, P) = 0$)*

*Proof.* Proof:

a. Trivial

b. Note that by the definition of $D$,

$$D(P, Q) = 0 \iff \|\mu_P - \mu_Q\|_{\Sigma_Q^{-1}} = \|\mu_P - \mu_Q\|_{\Sigma_P^{-1}} = 0 \text{ and } \Sigma_Q^{-1/2}\Sigma_P\Sigma_Q^{-1/2} = \Sigma_P^{-1/2}\Sigma_Q\Sigma_P^{-1/2} = I$$

Since we assume all of the covariance matrices are invertible, the first equalities hold iff $\mu_P = \mu_Q$. Similarly, the second equalities hold iff $\Sigma_P = \Sigma_Q$. Hence, $D(P, Q) = 0$ iff $P = Q$

□

## C Proof of Relaxed Triangle Inequality

**Theorem.** *Let $\epsilon > 0$. If $D(P,Q), D(Q,K) \leq \epsilon$, then*

$$D(P,K) \leq D(P,Q) + D(Q,K) + \sqrt{2}\epsilon + \sqrt{2}\epsilon\sqrt{1+\epsilon} + \epsilon^2 + E(P,Q,K),$$

*where $E(P,Q,K) = 0$ if $\Sigma_P$, $\Sigma_Q$, and $\Sigma_K$ commute and otherwise has a (loose) bound of*

$$\begin{aligned} E(P,Q,K) \leq\ & C_{Q,K}\left\|\left[\Sigma_P, \Sigma_Q^{-1/2}\right]\right\|_F + C_{P,Q}\left\|\left[\Sigma_K, \Sigma_Q^{-1/2}\right]\right\|_F \\ & + C'_{Q,K}\left\|\left[\Sigma_K^{-1/2}, \Sigma_Q^{-1/2}\Sigma_P\Sigma_Q^{-1/2}\right]\right\|_F + C'_{P,Q}\left\|\left[\Sigma_P^{-1/2}, \Sigma_Q^{-1/2}\Sigma_K\Sigma_Q^{-1/2}\right]\right\|_F, \end{aligned}$$

*and each constant $C_{i,j}$ depends on ratios of eigenvalues of $\Sigma_i$ and $\Sigma_j$ for $i,j \in \{P,Q,R\}$.*

We recall that

$$\begin{aligned} D(P,Q) = & \frac{1}{2}\left\|\Sigma_Q^{-1/2}\Sigma_P\Sigma_Q^{-1/2} - I\right\|_F + \frac{1}{2}\left\|\Sigma_P^{-1/2}\Sigma_Q\Sigma_P^{-1/2} - I\right\|_F \\ & + \frac{1}{\sqrt{2}}\|\mu_P - \mu_Q\|_{\Sigma_Q^{-1}} + \frac{1}{\sqrt{2}}\|\mu_P - \mu_Q\|_{\Sigma_P^{-1}} \end{aligned}$$

These terms are all nonnegative, so if $D(P,Q) \leq \epsilon$ then each term is at most $\epsilon$. To show the relaxed triangle inequality, we define

$$D_1(P,Q) := \left\|\Sigma_Q^{-1/2}\Sigma_P\Sigma_Q^{-1/2} - I\right\|_F + \left\|\Sigma_P^{-1/2}\Sigma_Q\Sigma_P^{-1/2} - I\right\|_F$$

and

$$D_2(P,Q) := \|\mu_P - \mu_Q\|_{\Sigma_Q^{-1}} + \|\mu_P - \mu_Q\|_{\Sigma_P^{-1}}$$

so that

$$D(P,Q) = \frac{1}{2}D_1(P,Q) + \frac{1}{\sqrt{2}}D_2(P,Q)$$

Then,

$$\begin{aligned} D_2(P,K) &= \|\mu_P - \mu_K\|_{\Sigma_K^{-1}} + \|\mu_P - \mu_K\|_{\Sigma_P^{-1}} \\ &\leq \|\mu_P - \mu_Q\|_{\Sigma_K^{-1}} + \|\mu_Q - \mu_K\|_{\Sigma_K^{-1}} + \|\mu_P - \mu_Q\|_{\Sigma_P^{-1}} + \|\mu_Q - \mu_K\|_{\Sigma_P^{-1}} \\ &= D_2(P,Q) + D_2(Q,K) + \|\mu_P - \mu_Q\|_{\Sigma_K^{-1}} - \|\mu_P - \mu_Q\|_{\Sigma_Q^{-1}} + \|\mu_Q - \mu_K\|_{\Sigma_P^{-1}} - \|\mu_Q - \mu_K\|_{\Sigma_Q^{-1}} \end{aligned}$$

Note that

$$\begin{aligned} \|\mu_P - \mu_Q\|_{\Sigma_K^{-1}} - \|\mu_P - \mu_Q\|_{\Sigma_Q^{-1}} &= \left\|\Sigma_K^{-1/2}(\mu_P - \mu_Q)\right\|_2 - \left\|\Sigma_Q^{-1/2}(\mu_P - \mu_Q)\right\|_2 \\ &\leq \left\|\Sigma_K^{-1/2}(\mu_P - \mu_Q) - \Sigma_Q^{-1/2}(\mu_P - \mu_Q)\right\|_2 \\ &= \left\|\left(\Sigma_K^{-1/2} - \Sigma_Q^{-1/2}\right)(\mu_P - \mu_Q)\right\|_2 \\ &= \left\|\left(\Sigma_K^{-1/2}\Sigma_Q^{1/2} - I\right)\Sigma_Q^{-1/2}(\mu_P - \mu_Q)\right\|_2 \\ &\leq \left\|\Sigma_K^{-1/2}\Sigma_Q^{1/2} - I\right\|_2\left\|\Sigma_Q^{-1/2}(\mu_P - \mu_Q)\right\|_2 \\ &= \left\|\Sigma_K^{-1/2}\Sigma_Q^{1/2} - I\right\|_2\|\mu_P - \mu_Q\|_{\Sigma_Q^{-1}} \\ &\leq \left\|\Sigma_K^{-1/2}\Sigma_Q^{1/2} - I\right\|_2\epsilon \end{aligned}$$

Now, note that $\left\|\Sigma_K^{-1/2}\Sigma_Q^{1/2} - I\right\|_2$ is the square root of the maximal eigenvalue of

$$(\Sigma_K^{-1/2}\Sigma_Q^{1/2} - I)^T(\Sigma_K^{-1/2}\Sigma_Q^{1/2} - I)$$

Therefore,

$$\begin{aligned}
\left\|\Sigma_K^{-1/2}\Sigma_Q^{1/2} - I\right\|_2^2 &= \left\|(\Sigma_K^{-1/2}\Sigma_Q^{1/2} - I)^T(\Sigma_K^{-1/2}\Sigma_Q^{1/2} - I)\right\|_2 \\
&= \left\|\Sigma_K^{-1/2}\Sigma_Q\Sigma_K^{-1/2} - \Sigma_K^{-1/2}\Sigma_Q^{1/2} - \Sigma_Q^{1/2}\Sigma_K^{-1/2} + I\right\|_2 \\
&\leq \left\|\Sigma_K^{-1/2}\Sigma_Q\Sigma_K^{-1/2} - I\right\|_2 + \left\|2I - \Sigma_K^{-1/2}\Sigma_Q^{1/2} - \Sigma_Q^{1/2}\Sigma_K^{-1/2}\right\|_2 \\
&\leq \left\|\Sigma_K^{-1/2}\Sigma_Q\Sigma_K^{-1/2} - I\right\|_2 + \left\|I - \Sigma_K^{-1/2}\Sigma_Q^{1/2}\right\|_2 + \left\|I - \Sigma_Q^{1/2}\Sigma_K^{-1/2}\right\|_2 \\
&= \left\|\Sigma_K^{-1/2}\Sigma_Q\Sigma_K^{-1/2} - I\right\|_2 + 2\left\|I - \Sigma_K^{-1/2}\Sigma_Q^{1/2}\right\|_2 \\
&= \left\|\Sigma_K^{-1/2}\Sigma_Q\Sigma_K^{-1/2} - I\right\|_2 + 2\left\|\Sigma_K^{-1/2}\Sigma_Q^{1/2} - I\right\|_2
\end{aligned}$$

Solving this for $\left\|\Sigma_K^{-1/2}\Sigma_Q^{1/2} - I\right\|_2$, we get

$$\left\|\Sigma_K^{-1/2}\Sigma_Q^{1/2} - I\right\|_2 \leq 1+\sqrt{1 + \left\|\Sigma_K^{-1/2}\Sigma_Q\Sigma_K^{-1/2} - I\right\|_2} \leq 1+\sqrt{1 + \left\|\Sigma_K^{-1/2}\Sigma_Q\Sigma_K^{-1/2} - I\right\|_F} \leq 1+\sqrt{1+\epsilon}$$

So,

$$\|\mu_P - \mu_Q\|_{\Sigma_K^{-1}} - \|\mu_P - \mu_Q\|_{\Sigma_Q^{-1}} \leq \left\|\Sigma_K^{-1/2}\Sigma_Q^{1/2} - I\right\|_2 \epsilon \leq \epsilon + \epsilon\sqrt{1+\epsilon}$$

A similar statement holds for $\|\mu_Q - \mu_K\|_{\Sigma_P^{-1}} - \|\mu_Q - \mu_K\|_{\Sigma_Q^{-1}}$, so

$$D_2(P,K) \leq D_2(P,Q) + D_2(Q,K) + 2\epsilon + 2\epsilon\sqrt{1+\epsilon}$$

Next,

$$\begin{aligned}
&\left\|\Sigma_P^{-1/2}\Sigma_K\Sigma_P^{-1/2} - I\right\|_F - \left\|\Sigma_Q^{-1/2}\Sigma_K\Sigma_Q^{-1/2} - I\right\|_F - \left\|\Sigma_P^{-1/2}\Sigma_Q\Sigma_P^{-1/2} - I\right\|_F \\
&\leq \left\|\Sigma_P^{-1/2}\Sigma_K\Sigma_P^{-1/2} - \Sigma_Q^{-1/2}\Sigma_K\Sigma_Q^{-1/2}\right\|_F - \left\|\Sigma_P^{-1/2}\Sigma_Q\Sigma_P^{-1/2} - I\right\|_F \\
&\leq \left\|\Sigma_P^{-1/2}\Sigma_K\Sigma_P^{-1/2} - \Sigma_Q^{-1/2}\Sigma_K\Sigma_Q^{-1/2} - \Sigma_P^{-1/2}\Sigma_Q\Sigma_P^{-1/2} + I\right\|_F \\
&= \left\|\left(I - \Sigma_Q^{-1/2}\Sigma_P\Sigma_Q^{-1/2}\right)\left(I - \Sigma_K^{-1/2}\Sigma_Q\Sigma_K^{-1/2}\right) + \Sigma_K^{-1/2}\Sigma_P\Sigma_K^{-1/2} - \Sigma_Q^{-1/2}\Sigma_P\Sigma_Q^{-1/2}\Sigma_K^{-1/2}\Sigma_Q\Sigma_K^{-1/2}\right\|_F \\
&\leq \left\|I - \Sigma_Q^{-1/2}\Sigma_P\Sigma_Q^{-1/2}\right\|_F\left\|I - \Sigma_K^{-1/2}\Sigma_Q\Sigma_K^{-1/2}\right\|_F + \left\|\Sigma_K^{-1/2}\Sigma_P\Sigma_K^{-1/2} - \Sigma_Q^{-1/2}\Sigma_P\Sigma_Q^{-1/2}\Sigma_K^{-1/2}\Sigma_Q\Sigma_K^{-1/2}\right\|_F \\
&\leq \epsilon^2 + \left\|\Sigma_K^{-1/2}\Sigma_P\Sigma_K^{-1/2} - \Sigma_Q^{-1/2}\Sigma_P\Sigma_Q^{-1/2}\Sigma_K^{-1/2}\Sigma_Q\Sigma_K^{-1/2}\right\|_F
\end{aligned}$$

A similar argument shows

$$\begin{aligned}
&\left\|\Sigma_K^{-1/2}\Sigma_P\Sigma_K^{-1/2} - I\right\|_F - \left\|\Sigma_Q^{-1/2}\Sigma_P\Sigma_Q^{-1/2} - I\right\|_F - \left\|\Sigma_K^{-1/2}\Sigma_Q\Sigma_K^{-1/2} - I\right\|_F \\
&\leq \epsilon^2 + \left\|\Sigma_P^{-1/2}\Sigma_K\Sigma_P^{-1/2} - \Sigma_Q^{-1/2}\Sigma_K\Sigma_Q^{-1/2}\Sigma_P^{-1/2}\Sigma_Q\Sigma_P^{-1/2}\right\|_F
\end{aligned}$$

Combining these,

$$\begin{aligned}
D_1(P,K) \leq\ & D_1(P,Q) + D_1(Q,K) + 2\epsilon^2 \\
&+ \left\|\Sigma_K^{-1/2}\Sigma_P\Sigma_K^{-1/2} - \Sigma_Q^{-1/2}\Sigma_P\Sigma_Q^{-1/2}\Sigma_K^{-1/2}\Sigma_Q\Sigma_K^{-1/2}\right\|_F \\
&+ \left\|\Sigma_P^{-1/2}\Sigma_K\Sigma_P^{-1/2} - \Sigma_Q^{-1/2}\Sigma_K\Sigma_Q^{-1/2}\Sigma_P^{-1/2}\Sigma_Q\Sigma_P^{-1/2}\right\|_F
\end{aligned}$$

If $[A, B] = AB - BA$ is the commutator of $A$ and $B$,

$$\begin{aligned}
&\left\|\Sigma_K^{-1/2}\Sigma_P\Sigma_K^{-1/2} - \Sigma_Q^{-1/2}\Sigma_P\Sigma_Q^{-1/2}\Sigma_K^{-1/2}\Sigma_Q\Sigma_K^{-1/2}\right\|_F \\
&\leq \left\|\Sigma_K^{-1/2}\Sigma_P\Sigma_K^{-1/2} - \Sigma_K^{-1/2}\Sigma_Q^{-1/2}\Sigma_P\Sigma_Q^{-1/2}\Sigma_Q\Sigma_K^{-1/2}\right\|_F \\
&\quad + \left\|\Sigma_K^{-1/2}\Sigma_Q^{-1/2}\Sigma_P\Sigma_Q^{-1/2}\Sigma_Q\Sigma_K^{-1/2} - \Sigma_Q^{-1/2}\Sigma_P\Sigma_Q^{-1/2}\Sigma_K^{-1/2}\Sigma_Q\Sigma_K^{-1/2}\right\|_F \\
&= \left\|\Sigma_K^{-1/2}\Sigma_P\Sigma_K^{-1/2} - \Sigma_K^{-1/2}\Sigma_Q^{-1/2}\Sigma_P\Sigma_Q^{-1/2}\Sigma_Q\Sigma_K^{-1/2}\right\|_F \\
&\quad + \left\|\left[\Sigma_K^{-1/2}, \Sigma_Q^{-1/2}\Sigma_P\Sigma_Q^{-1/2}\right]\Sigma_Q\Sigma_K^{-1/2}\right\|_F \\
&= \left\|\Sigma_K^{-1/2}\Sigma_P\Sigma_Q^{-1/2}\Sigma_Q^{-1/2}\Sigma_Q\Sigma_K^{-1/2} - \Sigma_K^{-1/2}\Sigma_Q^{-1/2}\Sigma_P\Sigma_Q^{-1/2}\Sigma_Q\Sigma_K^{-1/2}\right\|_F \\
&\quad + \left\|\left[\Sigma_K^{-1/2}, \Sigma_Q^{-1/2}\Sigma_P\Sigma_Q^{-1/2}\right]\Sigma_Q\Sigma_K^{-1/2}\right\|_F \\
&= \left\|\Sigma_K^{-1/2}\left[\Sigma_P, \Sigma_Q^{-1/2}\right]\Sigma_Q^{-1/2}\Sigma_Q\Sigma_K^{-1/2}\right\|_F + \left\|\left[\Sigma_K^{-1/2}, \Sigma_Q^{-1/2}\Sigma_P\Sigma_Q^{-1/2}\right]\Sigma_Q\Sigma_K^{-1/2}\right\|_F \\
&= \left\|\Sigma_K^{-1/2}\left[\Sigma_P, \Sigma_Q^{-1/2}\right]\Sigma_Q^{1/2}\Sigma_K^{-1/2}\right\|_F + \left\|\left[\Sigma_K^{-1/2}, \Sigma_Q^{-1/2}\Sigma_P\Sigma_Q^{-1/2}\right]\Sigma_Q\Sigma_K^{-1/2}\right\|_F
\end{aligned}$$

Similarly,

$$\begin{aligned}
&\left\|\Sigma_P^{-1/2}\Sigma_K\Sigma_P^{-1/2} - \Sigma_Q^{-1/2}\Sigma_K\Sigma_Q^{-1/2}\Sigma_P^{-1/2}\Sigma_Q\Sigma_P^{-1/2}\right\|_F \\
&\leq \left\|\Sigma_P^{-1/2}\left[\Sigma_K, \Sigma_Q^{-1/2}\right]\Sigma_Q^{1/2}\Sigma_P^{-1/2}\right\|_F + \left\|\left[\Sigma_P^{-1/2}, \Sigma_Q^{-1/2}\Sigma_K\Sigma_Q^{-1/2}\right]\Sigma_Q\Sigma_P^{-1/2}\right\|_F
\end{aligned}$$

So finally, if we let

$$\begin{aligned}
E(P, Q, K) := &\frac{1}{2}\left\|\Sigma_K^{-1/2}\left[\Sigma_P, \Sigma_Q^{-1/2}\right]\Sigma_Q^{1/2}\Sigma_K^{-1/2}\right\|_F + \frac{1}{2}\left\|\left[\Sigma_K^{-1/2}, \Sigma_Q^{-1/2}\Sigma_P\Sigma_Q^{-1/2}\right]\Sigma_Q\Sigma_K^{-1/2}\right\|_F \\
&+ \frac{1}{2}\left\|\Sigma_P^{-1/2}\left[\Sigma_K, \Sigma_Q^{-1/2}\right]\Sigma_Q^{1/2}\Sigma_P^{-1/2}\right\|_F + \frac{1}{2}\left\|\left[\Sigma_P^{-1/2}, \Sigma_Q^{-1/2}\Sigma_K\Sigma_Q^{-1/2}\right]\Sigma_Q\Sigma_P^{-1/2}\right\|_F
\end{aligned}$$

then the theorem follows.

$E(P, Q, K)$ satisfies slow growth behaviour in our context. If $\Sigma_P, \Sigma_Q, \Sigma_K$ are jointly diagonalizable, then clearly $E(P, Q, K) = 0$ since each commutator will be 0. Beyond this, we can trivially bound $E$ by

$$\begin{aligned}
E(P, Q, K) \leq &C_{Q,K}\left\|\left[\Sigma_P, \Sigma_Q^{-1/2}\right]\right\|_F + C'_{Q,K}\left\|\left[\Sigma_K^{-1/2}, \Sigma_Q^{-1/2}\Sigma_P\Sigma_Q^{-1/2}\right]\right\|_F \\
&+ C_{P,Q}\left\|\left[\Sigma_K, \Sigma_Q^{-1/2}\right]\right\|_F + C'_{P,Q}\left\|\left[\Sigma_P^{-1/2}, \Sigma_Q^{-1/2}\Sigma_K\Sigma_Q^{-1/2}\right]\right\|_F,
\end{aligned}$$

and each constant $C_{i,j}$ depends on ratios of eigenvalues of $i, j \in \{P, Q, R\}$.

# D Algorithms

**Algorithm 1** DBSCAN

**Input:** Data $X = \{x_1, ..., x_m\}$, $\epsilon > 0$, $\text{minPts} \in \mathbb{N}$
**Output:** Clusters $\{C_k\}$
$n \leftarrow 0$
$N \leftarrow \emptyset$
**while** $X \setminus (N \cup \bigcup_{k=0}^{n-1} C_k) \neq \emptyset$ **do**
  Pick $x \in X \setminus (N \cup \bigcup_{k=0}^{n-1} C_k)$
  **if** $\#R_\epsilon(x) < \text{minPts}$ **then**
    $N \leftarrow N \cup \{x\}$
  **else**
    $C_n \leftarrow \{x\}$
    $S \leftarrow R_\epsilon(x) \setminus (N \cup \{x\})$
    **while** $S \neq \emptyset$ **do**
      Pick $y \in S$
      **if** $\#R_\epsilon(y) < \text{minPts}$ **then**
        $N \leftarrow N \cup \{y\}$
        $S \leftarrow S \setminus \{y\}$
      **else**
        $C_n \leftarrow C_n \cup \{y\}$
        $S \leftarrow (S \cup R_\epsilon(y)) \setminus (N \cup C_n)$
      **end if**
    **end while**
    **if** $\#C_n < \text{minPts}$ **then**
      $N \leftarrow N \cup C_n$
      $C_n \leftarrow \emptyset$
    **else**
      $n \leftarrow n + 1$
    **end if**
  **end if**
**end while**

**Algorithm 2** OPTICS

**Input:** Data $X = \{x_1, ..., x_m\}$, $\epsilon > 0$, $\text{minPts} \in \mathbb{N}$, $n = 0$, $Q = \emptyset$
**Output:** Ordering $Q$, minimal reachability distances $d_{\text{min}} : X \to \mathbb{R}_{\geq 0}$
**for** $p \in X$ **do**
  $d_{\text{m}}(p) \leftarrow \infty$
**end for**
**for** $p \in X$ unprocessed **do**
  $N \leftarrow R_\epsilon(p)$
  Mark $p$ as processed
  $Q \leftarrow Q \cup \{p\}$
  **if** $d_{\text{core}}(p) \neq \infty$ **then**
    $S = \emptyset$
    $\text{update}(N, p, S, \epsilon, \text{minPts})$
    **for** $q \in S$ **do**
      $N' \leftarrow R_\epsilon(q)$
      Mark $q$ as processed
      $Q \leftarrow Q \cup q$
      **if** $d_{\text{core}}(q) \neq \infty$ **then**
        $\text{update}(N, p, S, \epsilon, \text{minPts})$
      **end if**
    **end for**
  **end if**
**end for**

**Algorithm 3** Update

**Input:** Neighborhood $N$, core point $p$, queue $S$, $\epsilon > 0$, $\text{minPts} \in \mathbb{N}$
**for** $o \in N$ **do**
  $d_{\text{new}} = \max\{d_{\text{core}}(p), \|p - o\|\}$
  **if** $d_{\min}(o) = \infty$ (Note this means $o \notin S$) **then**
    $d_{\min}(o) \leftarrow d_{\text{new}}$
    $S = S \cup \{o\}$
  **else**
    **if** $d_{\text{new}} < d_{\min}(o)$ **then**
      $d_{\min}(o) \leftarrow d_{\text{new}}$
      Reorganize $S$ to be in increasing order by value of $d_{\min}$
    **end if**
  **end if**
**end for**

**Algorithm 4** LINSCAN

**Input:** Data $X = \{x_1, ..., x_m\}$, $\epsilon > 0$, $\text{minPts} \in \mathbb{N}$, $n = 0$, $N = \emptyset$, $\text{eccPts} \in \mathbb{N}$
**Output:** Clusters $\{C_k\}$
$n \leftarrow 0$
$N \leftarrow \emptyset$
$\mathcal{P} \leftarrow \emptyset$
**for** $x \in X$ **do**
  $\mu \leftarrow \mu_{R^{\text{eccPts}}(x)}$
  $\Sigma \leftarrow \Sigma_{R^{\text{eccPts}}(x)}$
  $P \leftarrow \mathcal{N}(\mu, \Sigma)$
  $\mathcal{P} \leftarrow \mathcal{P} \cup \{P\}$
**end for**
$\{D_k\} \leftarrow \text{OPTICS}(\mathcal{P}, \epsilon, \text{minPts})$
**for** $k \in \{0, 1, ..., n\}$ **do**
  $C_k \leftarrow \{x_i \in X : P_i \in D_k\}$
**end for**

# E Rand Index and Adjusted Rand Index

**Definition E.1** (Rand Index)**.** *Let $X = \{x_1, ..., x_n\}$ and consider two partitions $\mathcal{C} = \{C_1, ..., C_m\}$ and $\mathcal{C}' = \{C'_1, ..., C'_n\}$ of $X$, i.e. $C_i \subseteq X$ and $C'_i \subseteq X$ for all $i$ and*

$$X = \bigcup_{i=1}^{m} C_i = \bigcup_{i=1}^{n} C'_i$$

*with*

$$C_i \cap C_j = C'_i \cap C'_j = \emptyset$$

*for all $i \neq j$. Let*

$$a := \#\left\{(x, y) \in X \times X : x \neq y, \exists i, j \text{ s.t. } x, y \in C_i, x, y \in C'_j\right\}$$

*and*

$$b := \#\left\{(x, y) \in X \times X : x \neq y, \exists i, j, k, l \text{ s.t. } i \neq j, k \neq l, x \in C_i, x \in C'_k, y \in C_j, y \in C'_l\right\}$$

*$a$ is the number of pairs of elements of $X$ such that both elements are in the same cluster in $\mathcal{C}$ and $\mathcal{C}'$ and $b$ is the number of pairs of elements of $X$ such that both elements are in different clusters in both $\mathcal{C}$ and $\mathcal{C}'$. Then, the Rand Index is given by*

$$R(X, \mathcal{C}, \mathcal{C}') = \frac{a + b}{\binom{n}{2}}$$

*So, $R(\mathcal{C}, \mathcal{C}')$ is the fraction of pairs of elements of $X$ such that $\mathcal{C}$ and $\mathcal{C}'$ both agree about whether the pair of elements lie in the same cluster or not. Note that $R$ is symmetric in $\mathcal{C}$ and $\mathcal{C}'$ and lies in*

*the interval* $[0, 1]$. *However, random partitions are not guaranteed to have near-zero pairwise Rand Index. To remedy this, we use the Adjusted Rand Index*

$$ARI(\mathcal{C}, \mathcal{C}') = \frac{R(\mathcal{C}, \mathcal{C}') - \mathbb{E}\left[R(\mathcal{C}, \mathcal{C}')\right]}{1 - \mathbb{E}\left[R(\mathcal{C}, \mathcal{C}')\right]}$$

*where the expectation is taken over random partitions of* $X$ *with the same number of clusters and number of elements in each cluster as* $\mathcal{C}$ *and* $\mathcal{C}'$. *Unlike the Rand Index, the Adjusted Rand Index may be negative, but it is a better measure of the similarity between two partitions as the Rand Index tends to be higher on average for finer partitions regardless of similarity.*

**Code availability**

The source codes are available for downloading at the link:
`https://github.com/aj111000/LINSCAN_Public`

We produced the code and images in this work using NumPy, JAX, Matplotlib, scikit-learn, and Optuna (Harris et al. (2020), Bradbury et al. (2018), Hunter (2007), Buitinck et al. (2013), Akiba et al. (2019)).