Master's Thesis

Topological and Practical Aspects of Data Separability in Complex High-Dimensional Data

Department of Statistics
Ludwig-Maximilians-Universität München

Jana Gauß

Munich, 2 November 2022

Submitted in partial fulfillment of the requirements for the degree of M. Sc.
Supervised by PD Dr. Fabian Scheipl & Dr. Moritz Herrmann

Abstract

This thesis investigates clustering and separability from a topological perspective, where a cluster is defined as a connected component of the underlying manifold. The question if the classes in a given data set correspond to such components is crucial for the evaluation of clustering algorithms using real-world data sets. A review of the existing literature shows that neither classification-based complexity measures nor cluster validity indices (CVIs) adequately incorporate the central aspects of separability from a topological perspective: between-class separation and within-class connectedness. A newly developed measure (DCSI) aims to quantify these two characteristics and can also be used as CVI, however it lacks robustness when it comes to multi-class data sets with overlapping classes. The experiments on synthetic data sets indicate that most measures highly correlate with the performance of DBSCAN. Of all 13 separability measures, DCSI has the highest correlation with maximum ARI on the original data sets as well as a very high correlation with ARI on UMAP and t-SNE embeddings. The clustering performance can be increased in many cases by manifold learning methods UMAP and t-SNE. UMAP typically yields more compact, separated clusters than t-SNE (which is also indicated by most separability measures) although the performance of DBSCAN is relatively similar on both embeddings. t-SNE on the other hand preserves more of the outer geometry when it comes to nested spheres for example. The results on some frequently used real-world data sets show that manifold learning can increase the separability of relatively well-separated components but is not able to separate classes with high overlap, e.g. some classes of the FMNIST data set. Such classes that might not correspond to connected components can be identified using separability measures.

Contents

List of Figures

List of Tables

1 Introduction

The goal of clustering is generally described as finding groups of similar objects in data (Hennig, 2015, Adolfsson et al., 2019). However, Ackerman et al. (2010) consider clustering as an ill defined problem, as there exists no unique definition of a "correct" clustering or "true" clusters. Analogous to the wide variety of possible desired characteristics that clusters can fulfill (e.g. see Hennig, 2015), there exist several algorithms for clustering using different approaches (Saxena et al., 2017, Jain et al., 1999).

While a probabilistic perspective of clustering is often limited to the assumption that the data is drawn from a mixture of distributions (Herrmann et al. (2022), see the reviews on clustering by Jain et al. (1999) and Saxena et al. (2017) for example), Niyogi et al. (2011) consider clustering as a "topological question": Clusters are the connected components of a manifold. High-dimensional data typically isn't uniformly distributed in $\mathbb{R}^D$ but concentrates around a lower dimensional manifold whose homology (especially its number of connected components) one tries to estimate (Niyogi et al., 2011). This perspective reveals the connection of clustering to topological data analysis (TDA), a recent field of statistics that uses topology and computational geometry to find structure in data (Chazal and Michel, 2021, Wasserman, 2018).

TDA is related to the task of manifold learning (often used as a synonym for nonlinear dimensionality reduction (Herrmann, 2022)), which aims to find low-dimensional representations of high-dimensional data (Cayton, 2005). A famous algorithm for manifold learning, UMAP (McInnes et al., 2018), has a strong mathematical foundation in geometry and algebraic topology and seems therefore well suited for the enhancement of clustering, as proposed by Herrmann et al. (2022): UMAP is used to infer the topological structure of the data set and the resulting embedding vectors are clustered, as the low-dimensional representation is optimized for separability in a certain sense (Herrmann et al., 2022).

Evaluation of clustering is often done using data sets with labels, e.g. real-world data sets from classification (Zimek and Vreeken, 2013, Hennig, 2015). This procedure is somewhat critical, as it is usually not known if the labels correspond to the type of structure a given algorithm can discover and if the given classes represent the characteristics desired in that specific situation (Zimek and Vreeken, 2013, Hennig, 2015). Schubert et al. (2017) suggest that we might use the "wrong" data sets for evaluation as the classes might not correspond to meaningful clusters. Furthermore, Herrmann et al. (2022) emphasize that it's important to distinguish the "probabilistic perspective" (mixture of distributions) from the topological view of clustering, where clusters don't overlap.

It is thus of great interest to be able to quantify to which extent the classes of a given data set form meaningful clusters, i.e. to measure the data set's separability. Existing separability measures rather focus on classification like the distance-based separability index proposed by Guan and Loew (2021) or the complexity measures in Ho and Basu (2002) and Lorena et al. (2019).

The aim of this thesis is to investigate how separability can be measured in the context of clustering (with a focus on the topological perspective of clustering, i.e. clusters as connected components) and how it can be increased using manifold learning methods that preserve (or emphasize) the topological structure of a data set.

This thesis is structured as follows: Chapter 2 provides the necessary theoretical background by introducing topological data analysis and persistent homology, manifold learning and two popular manifold learning algorithms, UMAP and t-SNE, as well as clustering both from a rather abstract perspective (what are "true" clusters and how can clustering be axiomatized) and an algorithmic one, with a focus on the density-based clustering algorithm DBSCAN and the combination of clustering and manifold learning.

In chapter 3, the existing notions of separability are reviewed and their suitability for clustering is examined. Several measures to quantify separability are presented, including complexity measures and cluster validity indices as well as a proposed method to quantify separability from a topological perspective and a newly developed separability index for density-based clustering.

Experiments were conducted both on synthetic and real-world data sets in order to investigate the behavior of the presented measures and their ability to quantify separability as well as the effects of manifold learning methods on separability and the clustering performance. The procedure and the data sets are described in chapter 4. The results of the experiments are presented in chapter 5. In chapter 6, the results and the limitations of this thesis are discussed and a conclusion is drawn.

2 Theoretical Background

2.1 Topological Data Analysis (TDA)

2.1.1 Overview & Introduction

Topological Data Analysis (TDA) is a collection of statistical methods to find structure in data using topological ideas (Wasserman, 2018). Often, the term TDA refers mainly to a method called *persistent homology*, but in a broader sense, it can also include methods like clustering, manifold estimation and nonlinear dimension reduction (Wasserman, 2018). TDA is a fast growing field that emerged from applied topology and computational geometry in the early 2000s (Chazal and Michel, 2021) and was established by Carlsson (2009) and Edelsbrunner and Harer (2010). Introductions to TDA are provided by Wasserman (2018) and Chazal and Michel (2021). Fasy et al. (2014a) give a short tutorial to the R package *TDA* (Fasy et al., 2022).

The main goal of TDA is to summarize and visualize complex data by inferring the topological and geometrical structure of the underlying space (Chazal and Michel, 2021, Wasserman, 2018). The data (represented as point clouds) is seen as a finite, possibly noisy sample from an unknown, potentially lower-dimensional set S. TDA aims to estimate the topological features of this set (Fasy et al., 2014a). In particular, one is interested in the *homology* of S, especially its number of connected components and k-dimensional holes. 0-dimensional holes are connected components, 1-dimensional holes are loops, 2-dimensional holes are cavities and so on. The number of k-dimensional holes β_k is called the k^{th} Betti number[1] (Wasserman, 2018).

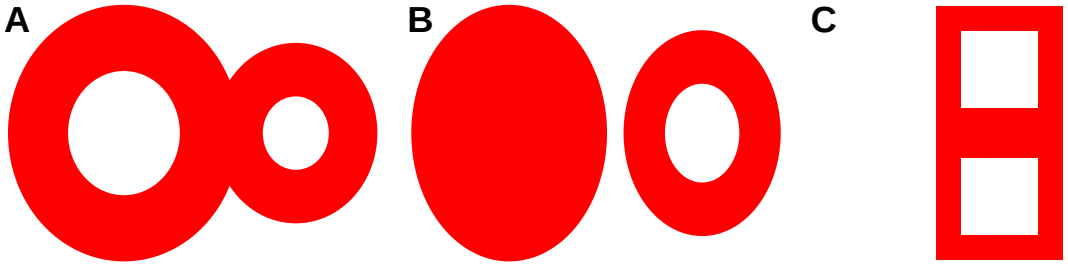

Figure 1: Examples homology and Betti numbers

Figure 1 shows three sets as examples for homology and Betti numbers. The set in **A** has one connected component and two (one-dimensional) holes, so $\beta_0 = 1, \beta_1 = 2$. The set in **B** has two connected components and one (one-dimensional) hole, so $\beta_0 = 2, \beta_1 = 1$ (an analogous example can be found in Wasserman (2018)). The set in **C** has the same homology as **A**, they are *homotopy equivalent* (Chazal and Michel, 2021). A 3-dimensional sphere (i.e. a 2-sphere, an empty ball) has $\beta_0 = 1, \beta_1 = 0, \beta_2 = 1$ (Wasserman, 2018).

Topological features are usually associated to continuous spaces, but the data - as a finite set of observations - is discrete (Chazal and Michel, 2021). In figure 2 **A**, the data $X_1, ..., X_n$ is sampled uniformly from a circle. In order to infer the topological structure of the support of the distribution, the union of balls of radius ε around X_i, $\bigcup_{i=1}^{n} B(X_i, \varepsilon)$ is considered (Wasserman, 2018). Figure 2 shows this union for different values of ε. For $\varepsilon = 0$, there are n (here $n = 50$) connected components (the disconnected balls, **A**). As ε increases, balls begin to overlap and some components merge (**B**). At some value of ε, only one component remains and a hole is born (**C**). Now, the union has the same homology as a circle. For a larger value of ε, the hole dies (**D**). This example describes the basic idea of *persistent homology*: for a nested family of sets, the evolution of their topological features (connected components and k-dimensional holes) is computed. The birth and death time of each feature is encoded in a so-called *persistence diagram* (Wasserman, 2018). In practice, one doesn't compute the topological features directly from the union of balls but from a *simplicial complex*, which has the same homology as the union. The

[1] Formally, homology uses group theory. The k-dimensional holes are represented by a vector space H_k. The dimension of this vector space (i.e. the rank of the k^{th} homology group (Wasserman, 2018)) is the k^{th} Betti number (Chazal and Michel, 2021).

advantage is that the homology of a simplicial complex can be computed using basic matrix operations (Wasserman, 2018). In the next two sections, the mathematical foundations and computational and statistical aspects of persistent homology are described in more detail.

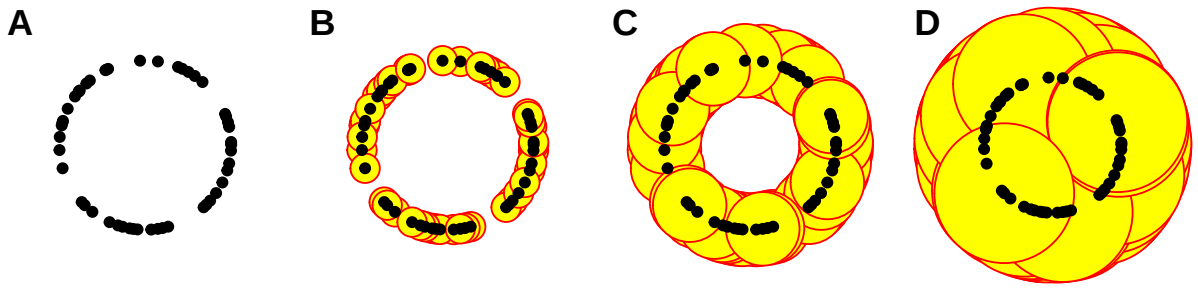

Figure 2: Idea of persistent homology

2.1.2 Persistent Homology

As mentioned above, topological features are associated to continuous spaces, whereas the input for TDA is a finite set, equipped with some notion of distance (e.g. a metric or a pairwise distance matrix) (Chazal and Michel, 2021). A natural way to discover some topological structure is to connect nearby points. This leads to neighboring graphs, but the idea in TDA is to go beyond connectivity and study topological features of higher dimensions like loops and voids (Chazal and Michel, 2021). This is achieved using *simplicial complexes*, which can be seen as higher dimensional generalizations of graphs. The following definitions can be found in Chazal and Michel (2021) (see Edelsbrunner and Harer (2010) for analogous definitions and more detailed examples):

Definition 2.1 (k-dimensional simplex)**.** Given a set $\mathbb{X} = \{x_0, ..., x_k\} \subset \mathbb{R}^d$ of k+1 affinely independent points, the *k-dimensional simplex* $\sigma = [x_0, ..., x_k]$ spanned by $\mathbb{X}$ is the convex hull of $\mathbb{X}$. The x_i are called the *vertices* of σ. The *faces* of σ are the simplices spanned by the subsets of $\mathbb{X}$.

So a 0-simplex is a point, a *vertex*; a 1-simplex is a line, an *edge*, and has two vertices as faces. A *triangle* is a 2-simples, it consists of three edges, and a 3-simplex is a *tetrahedron* and has four triangles as faces (Edelsbrunner and Harer, 2010).

Definition 2.2 (Geometric simplicial complex)**.** A *geometric simplicial complex* K in $\mathbb{R}^d$ is a collection of simplices such that any face of a simplex of K is a simplex of K and the intersection of any two simplices of K is either empty or a common face of both.

Definition 2.3 (Abstract simplicial complex)**.** Given a set V, an *abstract simplicial complex* with vertex set V is a set $\tilde{K}$ of finite subsets of V such that the elements of V belong to $\tilde{K}$ and for any $\sigma \in \tilde{K}$, any subset of σ belongs to $\tilde{K}$.

Any geometric simplicial complex K gives rise to an abstract simplicial complex $\tilde{K}$, and to any abstract simplicial complex $\tilde{K}$ one can associate a geometric simplicial complex K, which is called *geometric realization* of $\tilde{K}$ (Chazal and Michel, 2021, Edelsbrunner and Harer, 2010). $\tilde{K}$ can be seen as a topological space[2] and K as a geometric realization of its underlying combinatorial structure. This means that by building a simplicial complex on top of a data set, the topological features of the associated topological space can be inferred. At the same time, as combinatorial objects, simplicial complexes are well-suited for effective computations (Chazal and Michel, 2021).

[2] A *topological space* is a pair $(\mathbb{X}, \mathcal{U})$ where $\mathbb{X}$ is a point set and $\mathcal{U}$ is a *topology*, that is a collection of subsets of $\mathbb{X}$, called *open sets* such that
(i) $\mathbb{X} \in \mathcal{U}, \emptyset \in \mathcal{U}$, (ii) $U_1, U_2 \in \mathcal{U} \Rightarrow U_1 \cap U_2 \in \mathcal{U}$,
(iii) the (finite or infinite) union of sets in $\mathcal{U}$ is in $\mathcal{U}$.
A topology is a way to define which points are near, without specifying how near. It can be seen as an abstraction of Euclidean space in which neighborhoods are open balls around points (Edelsbrunner and Harer, 2010).

So an important step of persistent homology is to build simplicial complexes from data. The two most common ways to do this are *Vietoris-Rips complexes* and *Čech complexes* (Chazal and Michel, 2021). The following definitions can be found in Chazal and Michel (2021), Edelsbrunner and Harer (2010) and Wasserman (2018) ($\mathbb{X}$ is a set of points in a metric space (M, d) and $\alpha \geq 0$):

Definition 2.4 (Cech complex)**.** A *Čech complex* $\text{Cech}_\alpha(\mathbb{X})$ is the set of simplices $[x_0, ..., x_k] \subseteq \mathbb{X}$ such that the $k+1$ closed balls $B(x_i, \alpha)$ of radius α have a non-empty intersection, so $\text{Cech}_\alpha(\mathbb{X}) = \{\sigma \subseteq \mathbb{X} | \bigcap_{x \in \sigma} B(x, \alpha) \neq \emptyset\}$.

Definition 2.5 (Vietoris-Rips complex)**.** A *Vietoris-Rips complex* $\text{Rips}_\alpha(\mathbb{X})$ is the set of simplices $[x_0, ..., x_k] \subseteq \mathbb{X}$ such that $d(x_i, x_j) \leq \alpha$ for all $i, j \in \{0, ..., k\}$, so a simplex is included, if each pair of vertices is no more than α apart.

A Vietoris-Rips complex can be seen as an immediate extension of an α-neighboring graph (Chazal and Michel, 2021). Note that the complexes are related by $\text{Rips}_\alpha(\mathbb{X}) \subseteq \text{Cech}_\alpha(\mathbb{X}) \subseteq \text{Rips}_{2\alpha}(\mathbb{X})$ (Chazal and Michel, 2021).

An important result for TDA is, that the Čech complex $\text{Cech}_\alpha(\mathbb{X})$ is *homotopy equivalent* to the union of balls $\cup_{x \in \mathbb{X}} B(x, \alpha)$. This follows from the *Nerve theorem*, which states that under some conditions, a topological space X and the *nerve of a cover* are *homotopy equivalent* (Chazal and Michel, 2021).

Definition 2.6 (Cover)**.** A *cover* $\mathcal{U}$ of X is a family of sets U_i such that $X = \cup_{i \in \mathcal{I}} U_i$.

Definition 2.7 (Nerve)**.** The *nerve of* $\mathcal{U}$ is an abstract simplicial complex $C(\mathcal{U})$ whose vertices are the U_i and a simplex $\sigma = [U_{i_0}, ..., U_{i_k}]$ is in $C(\mathcal{U})$ if and only if $\bigcap_{j=0}^{k} U_{i_j} \neq \emptyset$.

Now the data $\mathbb{X}_n = \{x_1, .., x_n\}$ is assumed to be sampled i.i.d. according to some probability measure with compact support $\mathcal{M} \subset \mathbb{R}^d$. Under some conditions and for well chosen α, the union of balls $\cup_{x \in \mathbb{X}} B(x, \alpha)$ is homotopy equivalent to $\mathcal{M}$ (Niyogi et al., 2008, Chazal and Michel, 2021). For the case of noisy data, Niyogi et al. (2011) show that if some conditions are fulfilled, the homology of $\mathcal{M}$ can be recovered with high confidence by using a simplicial complex that isn't built directly on the (noisy) data but on a subset of the data. Note that the algorithm presented there only weakly depends on the ambient dimension of the data but rather on the dimension of $\mathcal{M}$ (Niyogi et al., 2008).

The results from the Nerve theorem provide an efficient way to compute the topological features using simplicial complexes. In practice, one uses Vietoris-Rips complexes instead of Čech complexes, as they are easier to compute (the computation Čech complexes requires the testing of emptiness of unions of balls (Chazal and Michel, 2021)) and approximate the homology of Čech complexes (Wasserman, 2018). What remains open is the choice of α, the radius of the balls. The idea of *persistent homology* is to study the topological features at different scale, i.e. for different values of α, and to encode the evolution of connected components and k-dimensional holes (their birth and death times) in a *persistence diagram* (Chazal and Michel, 2021). In practice, the homology is computed from a *filtration* (Chazal and Michel, 2021):

Definition 2.8 (Filtration of a simplicial complex)**.** A *filtration of a simplicial complex* K is a nested family of subcomplexes $(K_r)_{r \in T}$, $T \subseteq \mathbb{R}$, such that $\forall r, r' \in T : r \leq r' \Rightarrow K_r \subseteq K_{r'}$ and $K = \cup_{r \in T} K_r$.

Given a point cloud $\mathbb{X} \subset \mathbb{R}^d$, the families of Vietoris-Rips complexes $(\text{Rips}_\alpha(\mathbb{X}))_{\alpha \in \mathbb{R}}$ and Čech complexes $(\text{Cech}_\alpha(\mathbb{X}))_{\alpha \in \mathbb{R}}$ are filtrations. Because of the Nerve theorem, $(\text{Cech}_\alpha(\mathbb{X}))_{\alpha \in \mathbb{R}}$ encodes the homology of the family of unions of balls $\cup_{x \in \mathbb{X}} B(x, \alpha)$ as α goes from 0 to $+\infty$ (Chazal and Michel, 2021). As figure 2 shows, as α, the radius, increases, features appear and disappear. The birth and death time of each topological feature can be plotted as coordinates in $\mathbb{R}^2$. This is called *persistence diagram*. The longer the lifetime of a feature, the further it is from the diagonal. Figure 3 **B** shows a persistence diagram obtained from a Vietoris-Rips filtration of the data in **A** (the same data as in figure 2). For $\alpha = 0$, there are $n = 50$ connected components, so their birth time is 0 (x-axis, red points). As α increases, some components merge (they "die") and their death time is plotted on the y-axis. The rule in persistent homology is that if two components merge, the component that appeared most recently dies (Chazal and Michel, 2021). At some point, there is only one component left and a hole (blue triangle) is born. As α increases further, the hole dies and only the component remains.

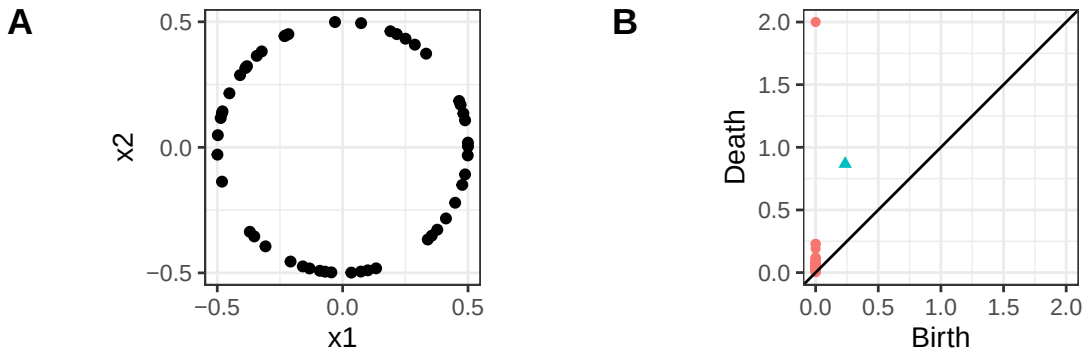

Figure 3: Persistence diagram of Vietoris-Rips filtration

In practice, persistence diagrams are often not built directly on a Čech or a Vietoris-Rips filtration but based on filtrations of a function:

Definition 2.9 (Sublevel set filtration)**.** Let K be a simplicial complex with vertex set V and $f : V \to \mathbb{R}$. f can be extended to all simplices of K: for any simplex $\sigma = [v_0, ..., v_k] \in K$, $f([v_0, ..., v_k]) = \max\{f(v_i) : i = 0, ..., k\}$. The *sublevel set filtration of* f is the familiy of subcomplexes $K_r = \{\sigma \in K : f(\sigma) \leq r\}$ (Chazal and Michel, 2021).

Upperlevel set filtrations can be defined similarly. A sublevel set filtration of f can also be written as $(F_r = f^{-1}((-\infty, r]))_{r\in\mathbb{R}}$. Given a compact set $X \subset \mathbb{R}^d$, the sublevel set filtration of the *distance function* is the filtration given by the union of growing balls $\cup_{x\in X} B(x, r)$, which is - thanks to the Nerve theorem - homotopy equivalent to the Čech filtration (Chazal and Michel, 2021). The *distance function* and its empirical version are defined like this:

Definition 2.10 (Distance function)**.** Given a set S, the *distance function* is defined to be $d_S(x) = \inf_{y\in S}||x - y||$. If a sample $X_1, ..., X_n$ is drawn from a distribution supported on S, the *empirical distance function* is $\hat{d}(x) = \min_{1\leq i\leq n}||x - X_i||$ (Wasserman, 2018).

Using the empirical distance function to compute the persistent homology of a data set is a common method, but unfortunately, this function isn't very robust to outliers and noise (Wasserman, 2018). There are several approaches to get more robust results. The distance function can be replaced by a density estimator for example (then, the upperlevel set filtration is computed and the birth times are after the death times) (Wasserman, 2018). One approach is to use a version of the distance function which is robust to noise, the *distance to a measure*, introduced by Chazal et al. (2011).

Definition 2.11 (Distance to a probability measure)**.** Given a parameter $0 \leq m < 1$ and a distribution P in $\mathbb{R}^d$, the distance to P can be defined as $\delta_{P,m} : x \in \mathbb{R}^d \mapsto \inf\{r > 0 : P(\bar{B}(x, r)) > m\}$ (Chazal et al., 2011).

For $m = 0$, this coincides with the usual distance function to the support of P (Chazal et al., 2011). This function can be seen as some sort of generalization of the distance function, but it lacks some features such a generalization should have (Chazal et al., 2011). Therefore, the *distance function to* P (distance to a measure, DTM) is defined as follows:

Definition 2.12 (Distance to a measure)**.** The *distance function to* P with parameter m_0 is defined by $d^2_{P,m_0} : x \in \mathbb{R}^d \mapsto \frac{1}{m_0}\int_0^{m_0} \delta_{P,m}(x)^2 \mathrm{d}m$ (Chazal et al., 2011).

This function can be seen as a smooth, probabilistic version of the distance function (Wasserman, 2018).

Definition 2.13 (Empirical distance to a measure)**.** The empirical version of the DTM for a given sample $X_1, ..., X_n$ is

$$\hat{d}^2_{m_0}(x) = \frac{1}{k} \sum_{X_i \in N_k(x)} ||X_i - x||^2 \text{ for } m_0 = \frac{k}{n}$$

where $N_k(x)$ is the set containing the k nearest neighbors of x among $X_1, ..., X_n$ (Chazal et al., 2011).

This is just the average squared distance to the k nearest neighbors (Wasserman, 2018). As mentioned above, one can also use upperlevel set filtrations of a density estimator instead of the distance function or DTM. One example is the k-nearest neighbor density estimator (knnDE) (Biau et al., 2011):

Definition 2.14 (k-nearest neighbor density estimator)**.** Let d be the dimension of the data $X_1, ..., X_n$, let $X_k(x)$ be the k-th nearest neighbor among $X_1, ..., X_n$ of a point $x \in \mathbb{R}^d$ and let V_d be the volume of the unit ball in $\mathbb{R}^d$. The k-nearest neighbor density estimator is defined as

$$f(x) = \frac{k}{nV_d||X_k(x) - x||^d}$$

Figure 4 shows the persistence diagrams of sublevel sets of the DTM (**B**) and the distance function (**C**) and upperlevel sets of knnDE (**D**). The data (**A**) was sampled uniformly from a circle ($n_{circle} = 200$) and some noise sampled uniformly from $[-0.5, 0.5]^2$ was added ($n_{noise} = 50$). The persistence diagrams show that DTM is less sensitive to noise, as it clearly depicts one prominent connected component and one hole, whereas the persistence diagram using the usual distance function shows several holes. The diagram of knnDE shows one hole, but it's less prominent than for DTM. See the examples in section 3.5.2 for further differences in behavior between these functions.

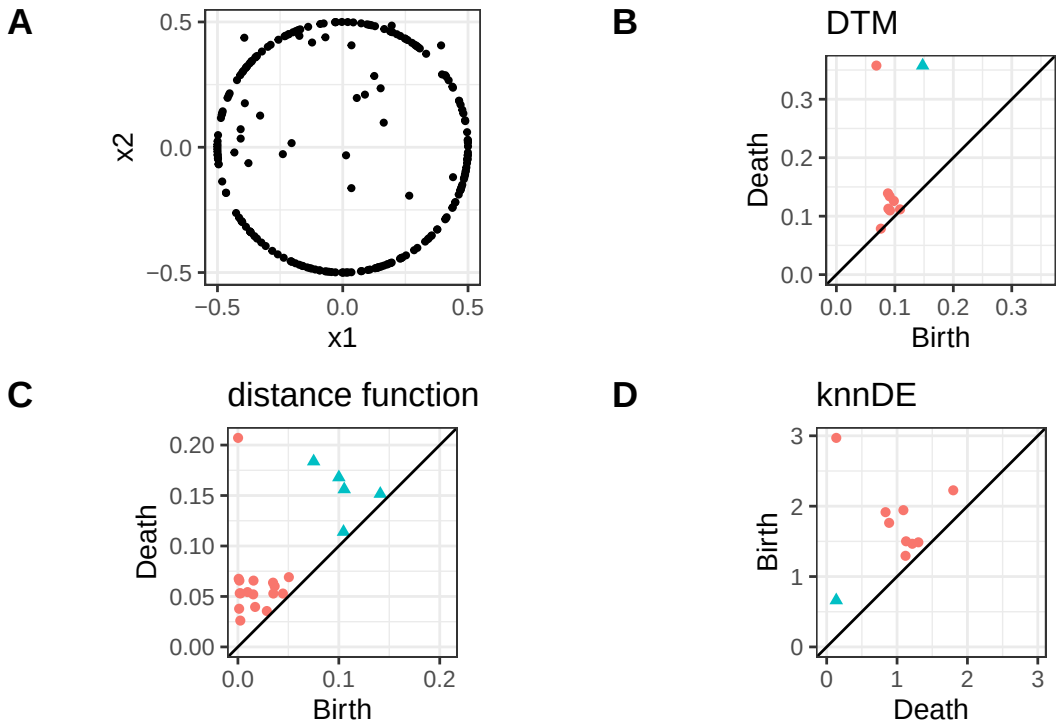

Figure 4: Persistence diagram using DTM, distance function and knnDE

2.1.3 Significance of Topological Features

Persistent homology itself doesn't take the randomness of the data into account. However, one would like to be able to differentiate between noise and significant topological features. Fasy et al. (2014b) and Chazal et al. (2014) provide a method to obtain confidence sets for persistence diagrams using bootstrap. The validity of the bootstrap is shown for the distance function and DTM (Chazal et al., 2014)[3]. In order

[3]No source was found that shows the validity for the k-nearest neighbor density estimator, so this section only considers the distance function and DTM.

to measure the variability of persistence diagrams, the *bottleneck distance* between two diagrams D_1 and D_2 can be defined as follows (Wasserman, 2018, Chazal et al., 2014):

Definition 2.15 (Bottleneck distance)**.** Given two persistence diagrams D_1, D_2, the *bottleneck distance* is $W_\infty(D_1, D_2) = \inf_\gamma \sup_{z \in D_1} ||z - \gamma(z)||_\infty$, where $\gamma : D_1 \mapsto D_2$ ranges over all bijections between D_1 and D_2.

Intuitively, one tries to match every point in D_1 to a point in D_2 and vice versa (bijection between D_1 and D_2) and measures the biggest distance of points of the best matching. Notice that D_1 and D_2 contain all pairs of points of the diagram as well as the diagonal, so points can also be matched to the diagonal (Chazal and Michel, 2021). Figure 5 is taken from Chazal and Michel (2021) (figure 12) and shows the best matching between two diagrams (red and blue) and the corresponding bottleneck distance (here d_B).

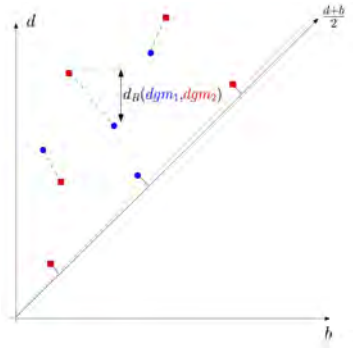

Figure 5: Perfect matching and bottleneck distance between two persistence diagrams (from Chazal and Michel (2021))

The bottleneck distance between two persistence diagrams D_1 and D_2 is bounded by

$$W_\infty(D_1, D_2) \leq ||\delta_1 - \delta_2||_\infty \tag{1}$$

where D_1 and D_2 are calculated from the distances functions or DTMs δ_1 and δ_2 (Chazal et al., 2014). This means that one can obtain approximate confidence sets for persistence diagrams from confidence bands for δ: Define c_α by

$$P(\sqrt{n}||\hat{\delta} - \delta||_\infty > c_\alpha) = \alpha \tag{2}$$

where δ is the true distance function/DTM and $\hat{\delta}$ is the estimated version (Chazal et al., 2014). A confidence band for δ is then given by

$$\left[\hat{\delta} - \frac{c_\alpha}{\sqrt{n}}, \hat{\delta} + \frac{c_\alpha}{\sqrt{n}}\right] \tag{3}$$

because $P(||\hat{\delta} - \delta||_\infty \leq \frac{c_\alpha}{\sqrt{n}}) = 1 - \alpha$ (Fasy et al., 2014a, Chazal et al., 2014). One can now determine if a feature in a persistence diagram is significant (Chazal et al., 2014):

Definition 2.16 (Significance persistence diagram)**.** A feature with birth and death time (u, v) is significant at level α if $|v - u| > 2\frac{c_\alpha}{\sqrt{n}}$ where c_α is defined by $P(\sqrt{n}||\hat{\delta} - \delta||_\infty > c_\alpha) = \alpha$ and δ is the true distance function/DTM and $\hat{\delta}$ is the estimated version used for the persistence diagram.

c_α can be estimated using bootstrap:

$$P(\sqrt{n}||\hat{\delta^*} - \hat{\delta}||_\infty > \hat{c}_\alpha) = \alpha \tag{4}$$

Then, $\hat{c}_\alpha$ is a consistent estimate of c_α (Chazal et al., 2014).
The following considerations from Chazal et al. (2014) might facilitate the understanding why the above definition holds: Let $\mathcal{D}$ be the set of persistence diagrams, let $D = D_\delta$ be the true diagram and $\hat{D} = D_{\hat{\delta}}$

the estimated one. Let $\mathcal{C}_n$ be the set of all diagrams with a bottleneck distance smaller than $\hat{c}_\alpha/\sqrt{n}$ from $\hat{D}$:

$$\mathcal{C}_n = \left\{ E \in \mathcal{D} : W_\infty(\hat{D}, E) \leq \frac{\hat{c}_\alpha}{\sqrt{n}} \right\} \tag{5}$$

Using equation (1) it holds

$$P(D \in \mathcal{C}_n) = P\left(W_\infty(\hat{D}, D) \leq \frac{\hat{c}_\alpha}{\sqrt{n}} \right) \geq P\left(||\hat{\delta} - \delta||_\infty \leq \frac{\hat{c}_\alpha}{\sqrt{n}} \right) \to 1 - \alpha \tag{6}$$

as $n \to \infty$. $|v - u| > 2\hat{c}_\alpha/\sqrt{n}$ holds if and only if the feature cannot be matched to the diagonal for any diagram in $\mathcal{C}_n$ (see figure 5 for a visualization of the bottleneck distance. A point x that is matched to a point d on the diagonal with $||x - d||_\infty = \hat{c}_\alpha/\sqrt{n}$ has a lifetime of $2\hat{c}_\alpha/\sqrt{n}$). As the diagonal corresponds to features with zero lifetime, a feature that can't be matched to the diagonal is significant at level α (Chazal et al., 2014).

Significant features can be visualized in the persistence diagram by drawing a confidence band of width $2\hat{c}_\alpha/\sqrt{n}$ around the diagonal. Features above the band are significant at level α (Chazal et al., 2014). Figure 6 shows a persistence diagram with a confidence band ($\alpha = 0.1$) obtained from 100 bootstrap replications. The persistence diagram is based on DTM and the data shown in figure 4 **A**. One connected component and the hole are significant. More details on the estimation via bootstrap and the calculation of p-values for components can be found in section 3.5.1.

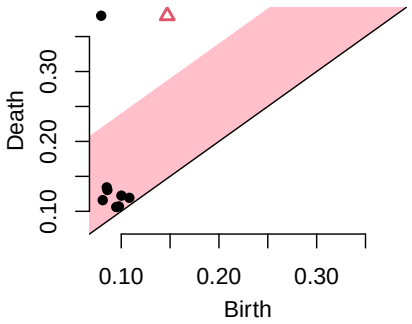

Figure 6: Persistence diagram with bootstrap confidence band

So why might persistent homology be relevant in the context of separability and clustering? In section 2.3.1.1 a topological view of clustering is presented: clustering is an example of TDA where one wants to find the connected components of the data. If a data set has two clearly separable clusters, the persistence diagram should indicate two significant connected components. So a persistence diagram could be used to decide if a data set consists of clusters and how separable they are. This approach will be presented in section 3.5.

2.2 Manifold Learning Methods

Manifold Learning is an approach to find a low-dimensional representation of high-dimensional data (Cayton, 2005). It is often used as a synonym for *nonlinear dimensionality reduction* (Herrmann, 2022). Manifold learning methods are based on the idea that the dimensionality of the data is often only artificially high, i.e. their intrinsic dimensionality is much lower than the observed one. More precisely, the data is assumed to lie on or near a low-dimensional, potentially nonlinear manifold that is embedded in a higher dimension. The goal is to "learn" the structure of this manifold and find a low-dimensional

embedding of the data that preserves this structure as close as possible (Cayton, 2005, Herrmann, 2022). Manifold learning algorithms are most commonly used for visualization of high-dimensional data in two or three dimensions and for pre-processing for machine learning tasks (Cayton, 2005).

The most popular method for *linear* dimensionality reduction is *Principal Component Analysis* (PCA) which assumes that the data lie on a linear subspace (Cayton, 2005). Its aim is to find a linear projection of the data that has maximum variance. This is identical to preserving the pairwise Euclidean distances as close as possible, which makes PCA a special form of *Multidimensional Scaling* (MDS) (a method that is non-linear in general) (Cayton, 2005). MDS takes a matrix of dissimilarities as input and tries to find an embedding such that the Euclidean distances of the embedding match the given dissimilarities. This is done by computing the spectral decomposition of a matrix calculated from the dissimilarity matrix (Cayton, 2005). MDS is equivalent to PCA if the dissimilarities in the high-dimensional space are defined to be the Euclidean distances (Herrmann, 2022). Another instance of MDS is *Isomap*. For Isomap, the dissimilarities in the high-dimensional space are given by the shortest path distances on a k-NN graph, i.e. one tries to approximate the geodesic distances on the manifold (Cayton, 2005, Herrmann, 2022). Two more recently developed methods - *UMAP* and *t-SNE* - are presented in the next sections. But first, a closer look is taken on the different perspectives underlying different manifold learning algorithms:

Often, manifold learning methods are divided into *local* and *global* methods: Local methods aim to preserve the local neighborhood structure whereas global methods focus on preserving the relative distances between all points, but this conceptualization isn't consistent (Herrmann, 2022). For example in the literature, sometimes Isomap is considered a global method and sometimes a local one. Moreover, it is rarely specified what *local* and *global* mean from a problem-driven perspective (Herrmann, 2022). Herrmann (2022) argues that one should rather differentiate between the *inner* geometry, the *outer* geometry and the *topology* of a data set. Which of these structures should be preserved depends on the application of the manifold learning method. The *inner* geometry is the intrinsic geometry of the manifold (i.e. geodesic distances) and can be preserved with methods like Isomap. This is most relevant for the standard notion of manifold learning where one wants to infer the structure of a single, connected manifold (Herrmann, 2022). The *outer* geometry on the other hand means the geometry inherited from the ambient space (i.e. the original data space), that is spatial positions and (Euclidean) distances. This is especially relevant for outlier detection (Herrmann, 2022). In both cases - when preserving the inner or outer geometry - an isometric mapping is required, i.e. a function that preserves distances (Herrmann, 2022). The *topological structure* is important for applications like clustering. Here, the main interest is not in preserving distances in the ambient space or on the manifold but in highlighting connected components (see section 2.3.1.1). In this case, a mapping that preserves topological features - a *homeomorphism* - is required (Herrmann, 2022). An example for such a method is the UMAP algorithm which is introduced in the next section. In section 2.3.3.3, the use of manifold learning methods as pre-processing for clustering (i.e. one doesn't apply the clustering algorithm on the original data but on an embedding) is described in more detail. The different behavior of several manifold learning methods and the different structures they preserve are demonstrated in the example below.

Figure 7 shows the so-called "Swiss-roll" data set, which is a popular example for manifold learning (see e.g. Cayton (2005) and Wasserman (2018)). The data ($n = 1000$) lie on a two-dimensional manifold but the ambient dimension is three. PCA, Isomap, t-SNE and UMAP are applied to this data[4] and the resulting two-dimensional embeddings are shown in figure 8. By estimating the geodesic distances, Isomap (**B**) is able to "roll out" the two-dimensional manifold and therefore perfectly detects its inner geometry. The PCA (**A**) projection is a rotation of the x- and y-coordinates while the z-dimension is left out and thereby preserves the spiral-like folding of the manifold. t-SNE (**C**) and UMAP (**D**) neither perfectly "unfold" the manifold nor capture the spiral-like structure. However, compared to PCA and Isomap, they are both more suited for indicating the presence of clusters in the data (McInnes et al., 2018, van der Maaten and Hinton, 2008), i.e. preserving the topological structure of the manifold (see sections 2.3.3.3 and 2.3.1.1). UMAP and t-SNE are described in more detail in the next sections.

[4]The nearest-neighbor parameter used for Isomap and UMAP and the perplexity parameter for t-SNE is 20. See the next sections for an explanation of the nearest-neighbor parameter for UMAP and the perplexity for t-SNE.

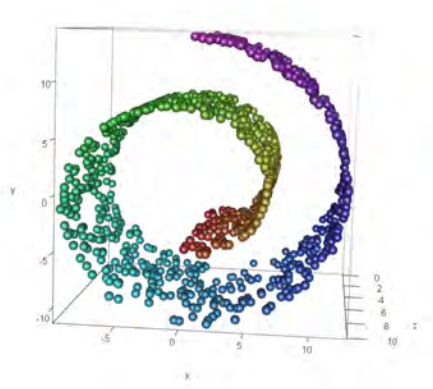

Figure 7: Swiss-roll data set

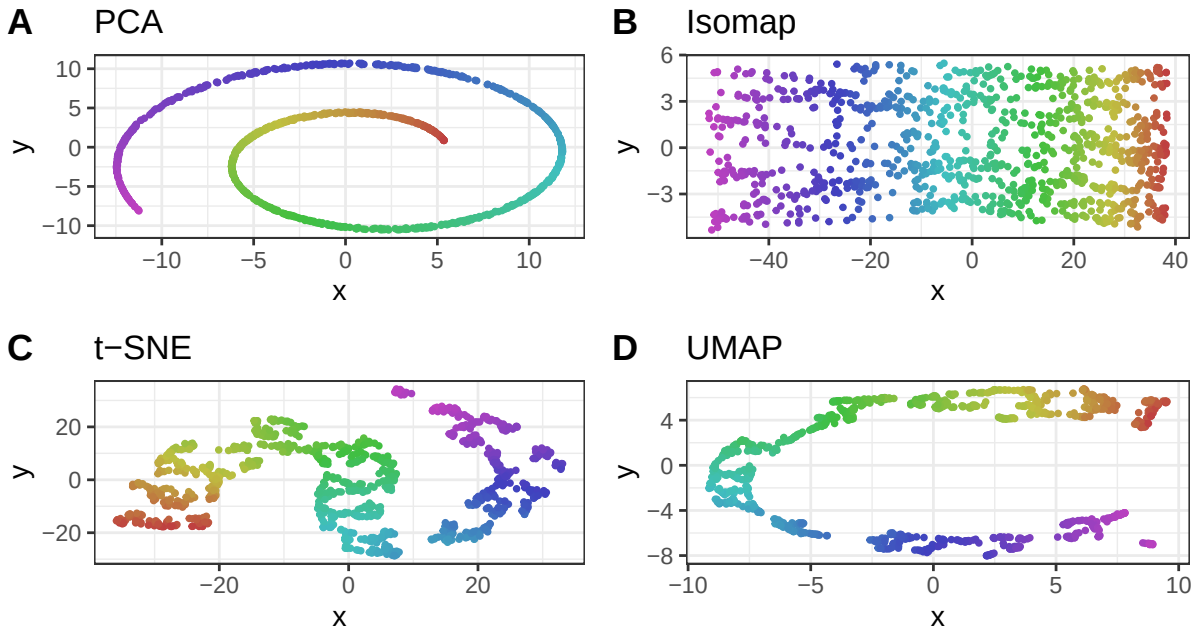

Figure 8: Examples of 2-dimensional embeddings of the Swiss-roll data

2.2.1 UMAP

Uniform Manifold Approximation and Projection (UMAP) is a manifold learning algorithm that - because of its foundation in Riemannian geometry and algebraic topology - has a strong connection with TDA (McInnes et al., 2018). Its main idea is to approximate the manifold underlying the (high-dimensional) data using *fuzzy simplicial sets*. In practice, one first constructs a particular weighted k-NN graph (*Graph construction step*). In the second step (*Graph layout step*), an optimization problem is solved in order to obtain a low-dimensional representation of the data, such that its fuzzy topological representation (the fuzzy simplicial set) is close to the one of the original data (McInnes et al., 2018). These two steps - constructing a k-NN graph and finding a low-dimensional representation that preserves some of the graph's characteristics - are the basic principle of many nonlinear dimensionality reduction algorithms, but in the case of UMAP, the choices for each step are mathematically justified (McInnes et al., 2018). The following description of the algorithm mainly concentrates on the computational aspects. The detailed theoretical background can be found in McInnes et al. (2018). Some more intuitive explanations are provided by the authors in an online software user guide (McInnes, 2018).

Graph Construction: Let $X = \{x_1, ..., x_n\}$ be some input data, $d(x_i, x_j)$ a metric on $X \times X$ and k a nearest neighbor parameter. $\{x_{i_1}, ..., x_{i_k}\}$ is the set of the k nearest neighbors of x_i under d. In the first phase of UMAP, the *Graph Construction*, a weighted (directed) k-NN graph $\bar{G}(V, E, w)$ is constructed. The vertices V are the x_i. The set E of directed edges is $E = \{(x_i, x_{i_j}) | 1 \leq j \leq k, 1 \leq i \leq n\}$. The weights w are defined as follows: For each x_i, ρ_i and σ_i are defined as

Definition 2.17 (Distance to nearest neighbor)**.** $\rho_i = \min\{d(x_i, x_{i_j}) | 1 \leq j \leq k, d(x_i, x_{i_j}) > 0\}$

Definition 2.18 (Normalisation factor)**.** σ_i is set such that $\sum_{j=1}^{k} \exp\left(\frac{-\max(0, d(x_i, x_{i_j}) - \rho_i)}{\sigma_i}\right) = \log_2(k)$ (McInnes et al., 2018)

The weight function w is given by

Definition 2.19 (Weight function)**.** $w(x_i, x_{i_j}) = \exp\left(\frac{-\max(0, d(x_i, x_{i_j}) - \rho_i)}{\sigma_i}\right)$ (McInnes et al., 2018)

The definition of ρ_i ensures that every x_i has at least one other point with edge weight 1 (its nearest neighbor) which is important as the mathematical foundation behind UMAP assumes that the underlying manifold is locally connected (McInnes et al., 2018). From a practical perspective, this ensures that no point is isolated from the others and improves the representation of high-dimensional data when other methods like t-SNE suffer from the curse of dimensionality (McInnes et al., 2018). There is another assumption behind the derivation of the UMAP algorithm: The data is uniformly distributed on the underlying manifold. This assumption may not be true for real-world data. By defining a local metric (by normalizing with σ_i), one ensures the validity of this assumption, as the data is approximately uniformly distributed with regard to that metric (McInnes et al., 2018).

So for every x_i, one has a metric space. This family of metric spaces should be "merged" into a single global structure that captures the topology of the underlying manifold. This is achieved using *fuzzy simplicial sets* (McInnes et al., 2018). Using the above definitions, each x_i induces a graph associated to its metric space. This graph is the 1-skeleton of a fuzzy simplicial set. Simplicial sets can be seen as a higher-dimensional generalizations of *directed* graphs (whereas simplicial complexes are generalizations of undirected graphs, see section 2.1.2). As the graph only contains vertices (1-simplices), it is a 1-skeleton. The set is fuzzy, because the membership of vertices to the set is no longer binary (true or false), but given by the membership strength $w(x_i, x_{i_j})$. This weight corresponds to the probability that a given edge exists (McInnes et al., 2018).

One now takes the (fuzzy) union of the family of fuzzy simplicial sets in order to obtain a single fuzzy simplicial set. This set captures the relevant topological structure of the underlying manifold (see McInnes et al. (2018) for mathematical details). In practice, the set of local graphs (represented as a single directed graph) is combined into a single undirected weighted graph. Let A be the (weighted) adjacency matrix of $\bar{G}$. The undirected weighted graph G that is used for the further steps of UMAP is defined by its symmetric adjacency matrix B:

Definition 2.20 (Adjacency matrix undirected graph)**.** $B = A + A^T - A \circ A^T$

where $\circ$ is the Hadamard (i.e. the pointwise) product (McInnes et al., 2018). The formula derives from the union of fuzzy sets: If A_{ij} is interpreted as the probability that the directed edge from x_i to x_j exists, then $B_{ij} = A_{ij} + A_{ji} - A_{ij} * A_{ji}$ is the probability that at least one of the two directed edges (from x_i to x_j or x_j to x_i) exists (McInnes et al., 2018).

Graph Layout: By G, one has now learned a fuzzy topological representation of the data. The goal of the second step of the UMAP algorithm is to find a low-dimensional representation that preserves the topological structure. This is achieved by choosing the low-dimensional representation $Y = \{y_1, ..., y_n\}$ such that the fuzzy simplicial set of Y closely matches the one of X (McInnes et al., 2018). In order to compare the embedding Y and the original data X, one uses cross entropy. Let $w(e)$ be the weight of edge e in the original data and $v(e)$ the edge weight in the lower-dimensional representation. The cost function for UMAP is then given by

Definition 2.21 (Cost function UMAP). $C = \sum_{e \in E} w(e) \log\left(\frac{w(e)}{v(e)}\right) + (1 - w(e)) \log\left(\frac{1 - w(e)}{1 - v(e)}\right)$ (McInnes et al., 2018)

The two terms represent attractive and repulsing forces: The first term ensures that for an edge with a large $w(e)$, i.e. an edge belonging to points that are close in the high-dimensional space, the distance of the corresponding points in the low-dimensional space will be small. When $w(e)$ is small or zero (distant points in the high-dimensional space), the second term ensures that $v(e)$ will be small as well, i.e. the embeddings of the corresponding points are placed far from each other. This optimization step increases the separability of clusters (Herrmann et al., 2022). The benefits of UMAP for clustering are described in more detail in section 2.3.3.3.

As G faithfully captures the topology of the original data, the weighted graph constructed from Y matches the topology as closely as possible. Y therefore provides a good low-dimensional representation that preserves the topological structure of the manifold underlying the high-dimensional data (McInnes et al., 2018). Here, the comparison of the fuzzy simplicial sets of X and Y is restricted to graphs, i.e. the 1-skeleton (a graph only contains 1-simplices). In theory, this could be extended to any dimension of simplices, which would capture the overall topological structure even more accurately. However, this is computationally very expensive and therefore not implemented (McInnes et al., 2018).

C is optimized with respect to the embedding Y using stochastic gradient descent (McInnes et al., 2018). The edge weights $v(e)$ of the lower-dimensional representation are not computed as the weights $w(e)$ for the original data. In order to be able to apply SGD for optimization, a differentiable approximation is needed. The edge weights v are given by

Definition 2.22 (Edge weights low-dim. representation). $v(y_i, y_j) = \left(1 + a||y_i - y_j||_2^{2b}\right)^{-1}$

where a and b are chosen by a non linear least squares fitting against the values that are approximated (McInnes et al., 2018). UMAP has four hyper-parameters: 1. k, the number of nearest neighbors, 2. the dimension of the embedding, 3. *min-dist*, the desired distance between close points in the embedding (mainly an aesthetic parameter relevant for visualization) and 4. the number of training epochs for SGD (McInnes et al., 2018). k determines the scale at which the manifold is approximated: A small k ensures that a detailed structure of the manifold is captured but can lead to a loss of the "big picture" as the manifold tends to break into many small components (McInnes et al., 2018).

Unlike other manifold learning methods that focus on the inner or outer geometry of the manifold, the goal of UMAP is to preserve the topological structure. Because of the definition of the weights in the k-NN graph, it doesn't isolate points and therefore doesn't retain the outer geometry, which makes UMAP not very suited for outlier detection (Herrmann, 2022). On the other hand, it emphasizes clusters (i.e. connected components of the manifold) by increasing their separation and decreasing within-cluster distances (Herrmann et al., 2022). How clustering can benefit from UMAP is described in more detail in section 2.3.3.3. In figure 8 and in appendix A examples of UMAP embeddings are shown.

2.2.2 t-SNE

Another popular manifold learning method is *t-Distributed Stochastic Neighbor Embedding* (t-SNE) (van der Maaten and Hinton, 2008), a modification of *Stochastic Neighbor Embedding* (SNE) (Hinton and Roweis, 2002). SNE aims to find a low-dimensional representation of high-dimensional data such that the probabilities that a (high-dimensional) point x_i would pick a point x_j as its neighbor are preserved (Hinton and Roweis, 2002). t-SNE tackles some problems of SNE by modifying the definition of the probability distribution over the potential neighbors and the cost function (van der Maaten and Hinton, 2008).

First, the concept of SNE is introduced in order to understand the changes t-SNE employs. The starting point of SNE is to convert the pairwise distance between two points x_i and x_j into a conditional probability $p_{j|i}$ that x_i would pick x_j as its neighbor. This probability is defined in such a way that the neighbors are picked in proportion to their density under a Gaussian centered at x_i (van der Maaten and Hinton, 2008). $p_{j|i}$ is therefore given by (van der Maaten and Hinton, 2008)

$$p_{j|i} = \frac{\exp(-||x_i - x_j||^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-||x_i - x_k||^2/2\sigma_i^2)} \tag{7}$$

A method to determine σ_i^2, the variance of the Gaussian centered at x_i, is presented later. For the low-dimensional representations y_i and y_j, the probability $q_{j|i}$ is defined similar, but the variance of the Gaussian is set to $\frac{1}{2}$, which leads to (van der Maaten and Hinton, 2008) [5]

$$q_{j|i} = \frac{\exp(-||y_i - y_j||^2)}{\sum_{k \neq i} \exp(-||y_i - y_k||^2)} \tag{8}$$

Now one wants to find a mapping $x_i \mapsto y_i$ for all i such that the mismatch between the probabilities $p_{j|i}$ and $q_{j|i}$ is minimized. A natural way to measure this is to use the Kullback-Leibler divergence. The cost function C for SNE is given by

$$C = \sum_i KL(P_i||Q_i) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}} \tag{9}$$

where P_i and Q_i represent the conditional probability distributions over all other points given x_i and its mapping y_i respectively (van der Maaten and Hinton, 2008). As the KL-divergence is not symmetric, there is a large cost if nearby points x_i, x_j (large $p_{j|i}$) are represented by distant points y_i, y_j, whereas the cost for representing widely separated points (small $p_{j|i}$) by close embeddings y_i, y_j is small. The cost function is optimized using gradient descent (van der Maaten and Hinton, 2008).

What remains is the choice of σ_i, the standard deviation of the Gaussian centered over x_i. The suggested method is to set σ_i such that the entropy of the distribution over the neighbors equals $\log_2(k)$ (i.e. $H(P_i) = \log_2(k)$). k is called *perplexity* and can be seen as the effective number of local neighbors (Hinton and Roweis, 2002) [6]. σ_i can thus be determined by

$$k = Perp(P_i) = 2^{H(P_i)} \tag{10}$$

where $H(P_i)$ is the Shannon entropy (van der Maaten and Hinton, 2008):

$$H(P_i) = -\sum_j p_{j|i} \log_2(p_{j|i}) \tag{11}$$

σ_i is found by a binary search. The perplexity parameter k is set by the user; typical values are between 5 and 50. σ_i increases monotonically with k (van der Maaten and Hinton, 2008).

t-SNE is a variation of SNE that is easier to optimize and reduces a tendency of SNE called the *crowding problem* (van der Maaten and Hinton, 2008). The first modification is to use a symmetrized version of the cost function which leads to a simpler gradient. Instead of conditional probabilities P_i and Q_i, one uses joint probability distributions P and Q:

Definition 2.23 (Cost function t-SNE). $C = KL(P||Q) = \sum_i \sum_j p_{ji} \log \frac{p_{ji}}{q_{ji}}$

This is called *symmetric SNE*, because for all i, j it holds $p_{ij} = p_{ji}$ and $q_{ij} = q_{ji}$ (van der Maaten and Hinton, 2008). The probabilities q_{ij} of the low-dimensional map are now defined as

Definition 2.24 (Probabilities low-dim. space, symmetric SNE). $q_{ij} = \frac{\exp(-||y_i - y_j||^2)}{\sum_{k \neq l} \exp(-||y_k - y_l||^2)}$ (van der Maaten and Hinton, 2008)

[5] Using the same variance (here $\frac{1}{2}$) for every data point in the low-dimensional mapping means that an embedding in the same dimensionality as the original data will differ from the original data (van der Maaten and Hinton, 2008).

[6] Note that for $k \in \mathbb{N}$ the entropy of a discrete uniform distribution with k outcomes is $\log_2(k)$. However, for SNE and t-SNE, k doesn't have to be a natural number.

The p_{ij} are *not* defined in the obvious way (by replacing y with x and using a variance parameter σ_i^2), because for an outlier x_i, this would result in extremely small values of p_{ij} for all j. The location of its low-dimensional representation y_i would have very little effect on the cost function and the position of y_i would thus not be well determined by the position of the other y_j. That's why p_{ij} is now set to be

Definition 2.25 (Probabilities high-dim. space, symmetric SNE and t-SNE)**.**
$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$, where $p_{j|i} = \frac{\exp(-||x_i - x_j||^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-||x_i - x_k||^2/2\sigma_i^2)}$ (same as for SNE)

which ensures that every x_i has an impact on the cost function (van der Maaten and Hinton, 2008). As for SNE, the σ_i are determined by k. The form of the resulting gradient is easier and thus faster to compute (see van der Maaten and Hinton (2008) for details).

SNE faces a problem called the *crowding problem*, which means that often a lot of points crowd in the center of the map and gaps between clusters aren't formed. The solution of van der Maaten and Hinton (2008) is to use a probability distribution with much heavier tails than a Gaussian for the low-dimensional map. This leads to the effect that moderate distances in the high-dimensional space can be modeled better (i.e. by a larger distance in the mapping) than with SNE (van der Maaten and Hinton, 2008). So while the p_{ij} remain unchanged (i.e. a Gaussian distribution is used to convert distances into probabilities), in the low-dimensional space, one now converts distances into probabilities using the heavy-tailed Student t-distribution with one of freedom:

Definition 2.26 (Probabilities low-dim. space t-SNE)**.** $q_{ij} = \frac{(1 + ||y_i - y_j||^2)^{-1}}{\sum_{k \neq l}(1 + ||y_k - y_l||^2)^{-1}}$ (van der Maaten and Hinton, 2008)

The result of this modification is that - compared to SNE - dissimilar points x_i, x_j that are modeled by a small distance in the low-dimensional space are pulled further apart. The gradient of the cost function for t-SNE and its derivation can be found in van der Maaten and Hinton (2008). Possible drawbacks of t-SNE are that the cost function is not convex and that it suffers from the *curse of intrinsic dimensionality* of the data, i.e. if the intrinsic dimensionality of the data is high and the underlying manifold is highly varying, t-SNE might fail as - by employing Euclidean distances - it implicitly makes an assumption of local linearity (van der Maaten and Hinton, 2008).

Similar to UMAP, t-SNE can be used as a pre-processing for clustering and enhance the clustering performance by increasing the separation of clusters. In section 2.3.3.3 this is described in more detail. Examples of t-SNE embeddings are shown in figure 8 and appendix A.

2.3 Clustering

2.3.1 What are "true" Clusters? Different Views of Clustering

One of the main goals of this thesis is to investigate how one can quantify the separability of a data set with regard to cluster analysis, i.e. measure the presence of clearly separable clusters. But what are "clusters" and what is a good or a correct clustering of a data set? The goal of clustering is generally described as finding inherent cluster structure in data, i.e. discovering meaningful groups of similar items (Adolfsson et al., 2019, Hennig, 2015). However, this is an ill defined problem as there is no definition of what a "correct" clustering is (Ackerman et al., 2010). Adolfsson et al. (2019) and Hennig (2015) state that one needs to distinguish between clustering with "constructive" or "realistic" aims (here, the focus lies on the latter): Constructive clustering means that one wants to cluster the data for pragmatic reasons, irrespective of the presence of a cluster structure. On the other hand, the goal of realistic clustering is to find "real" groups. But there is no definition of a "true" cluster. Often, when speaking about a "real" cluster structure, one thinks of the existence of an unobserved categorical variable whose values define the "true" clusters (Hennig, 2015). However, there is no guarantee that there is a unique categorical variable defining the clusters or that such a meaningful variable corresponds to structures in the given data (Hennig, 2015). Furthermore, the application determines which clusters are meaningful - i.e. which properties of clusters are desired - and therefore may even dictate if the given data are

clusterable[7] (Hennig, 2015, Adolfsson et al., 2019).

Specifying which characteristics of a cluster may be desirable can clarify what the underlying "cluster concept" is. Hennig (2015) provides a list of several properties that might be desired: The most commonly requested properties are probably (1) small within- and (2) large between-cluster dissimilarities. Most definitions of clustering include the notion of "groups of similar objects", see above and also Saxena et al. (2017) for example. However, differences in measuring these qualities lead to different resulting clusters: For example should the maximum or the average dissimilarity within a cluster be small? Concerning large dissimilarities between clusters, are big gaps between the clusters desired or rather that the cluster centroids are far from each other (Hennig, 2015)? Other desired characteristics might be that (3) the clusters fit well to some probability model (e.g. a mixture of Gaussians), that (4) they are well represented by their centroid or (5) have a certain shape, e.g. convex. A cluster could also correspond to (6) a connected area with high density (density-based clustering). One often requests that (7) all clusters have roughly the same size and that (8) the number of clusters is low. Some of these properties may conflict with others as a connected area of high density (6) might have an undesired shape (5) and maybe can't be represented by its centroid (4) (Hennig, 2015). In section 2.3.3 an overview on clustering algorithms is provided that presents some methods corresponding to these different desired characteristics.

In practice, most clustering algorithms implicitly define their own "truth", e.g. as the solution of an optimization problem (Hennig, 2015). This means that for a given problem, care must be taken when a clustering algorithm is chosen because often, the problem doesn't directly translate into a corresponding objective function or clustering method (Hennig, 2015, Ackerman et al., 2010). Another way to implicitly define "true" clusters is by optimizing an internal cluster validation index (see more in section 3.4) or by using external information i.e. class labels (Hennig, 2015). The latter is an artificial situation often used to compare clustering algorithms. However, it is hardly investigated to what extent the given classes from supervised classification problems correspond to clusters with characteristics desired in that specific situation (Hennig, 2015). Similarly, Zimek and Vreeken (2013) state that evaluating clustering on classification data might be a bad choice, as this doesn't allow for multiple clustering solutions to be true or worthwhile and only makes sense if the labels correspond to the type of structure an algorithm can discover. Hennig (2015) also criticizes the use of synthetic data sets with "true" classes without a formal definition, e.g. points distributed on a ball around the origin and points on a much wider circle, separated from the central ball (figure 9 **A**). Such data is used to illustrate that "k-means doesn't work"[8], however, the ball and the circle qualify as "true" clusters in some ways but not in others, as the biggest distances occur within a cluster and they have the same centroid (Hennig, 2015). In order to clarify on what view of clustering this thesis is based on, another perspective is now presented:

2.3.1.1 Topological View of Clustering Niyogi et al. (2011) develop a new view of clustering: a topological one. Clustering can be seen as a topological question about the data and their underlying probability distribution: Clusters are defined as connected components. Similar to manifold learning, the approach of Niyogi et al. (2011) to adequately model data in high-dimensional spaces is based on the idea that high-dimensional data isn't uniformly distributed on $\mathbb{R}^D$ (let D be the dimension of the data) but concentrates around a lower dimensional manifold $\mathcal{M}$. A suitable probability distribution whose support is all of $\mathbb{R}^D$ but that concentrates around $\mathcal{M}$ can be defined using a marginal distribution $P(x)$ supported on $\mathcal{M}$ and a conditional distribution $P(y|x)$ that models noise (e.g. Gaussian noise) in the normal direction (Niyogi et al., 2011). The connected components of $\mathcal{M}$ can be seen as clusters. A mixture of Gaussians with k components is a special case of this framework: $\mathcal{M}$ is zero-dimensional and consists of k points, i.e. k clusters. An important result of Niyogi et al. (2011) is that it is possible to learn the homology of the d-dimensional manifold $\mathcal{M}$ if the noise is sufficiently small, without encountering the curse of dimensionality even if the ambient dimension D is high. The consequence for clustering is that the difficulty of a clustering task does not primarily depend on the ambient dimension of the data (Herrmann et al., 2022).

[7] By clusterable, one means the presence of an inherent cluster structure (Adolfsson et al., 2019). *Measures of Clusterability* aim to quantify how "strong" this structure is (Ackerman and Ben-David, 2009). See more in section 3.3.

[8] The k-means algorithm is described in section 2.3.3.

In many settings, this view provides a clear definition of "true" clusters. However, in situations with overlapping clusters or clusters connected by a "bridge" of points, from a topological perspective, there is only one connected component. In this case, it might make more sense to take a "probabilistic perspective", i.e. assume that the data is sampled from a mixture of distribution. Herrmann et al. (2022) argue that it's important to distinguish such a situation (the domains of the clusters are connected) from "purely" topological tasks (clusters without overlap, i.e. disconnected domains). The focus of this thesis lies on the latter situation and the question how the separability of these components can be (1) adequately measured and (2) increased by manifold learning methods that aim to preserve topology.

Figure 9 shows that "true" clusters can be defined in different ways, depending on the view of clustering and the desired characteristics of clusters. The data in **A** clearly has two connected components, i.e. clusters from a topological perspective: The inner ball and the outer circle. However, if compact clusters that can be represented by their centroid are required, these two components don't qualify as meaningful clusters. **B** demonstrates that the topological perspective has to be distinguished from a perspective that allows for clusters to overlap, e.g. a probabilistic one. From a topological point of view, the data has a single component. However, the data is sampled from two overlapping disks, i.e. a mixture distribution with two components that might be seen as meaningful clusters too.

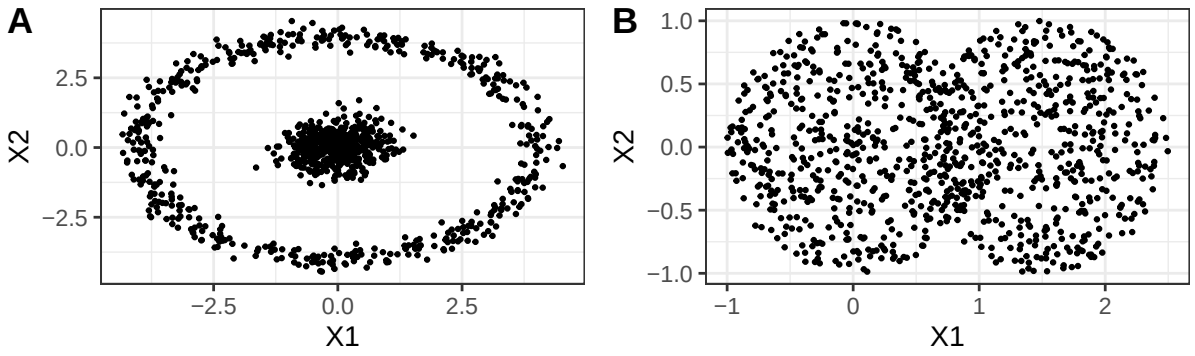

Figure 9: Examples of different views on clustering

2.3.2 Axiomatization of Clustering & Properties of Clustering Functions

In order to develop a unified framework for clustering and to formalize it, Kleinberg (2002) presents three axioms that a clustering function should fulfill. A clustering function is any function f that takes a symmetric distance function d over a set X of points as input and returns a partition of X, i.e. disjoint subsets C_i (called clusters) such that $\bigcup_i X_i = X$ (Kleinberg, 2002, Ackerman et al., 2010). The proposed three axioms are *scale-invariance*, *consistency* and *richness* (Kleinberg, 2002, Ackerman and Ben-David, 2009):

Definition 2.27 (Scale-invariance)**.** A clustering function f is *scale-invariant* if for any distance function d and any $\alpha > 0$, $f(d) = f(\alpha d)$, where αd is defined by $\forall x, y : \alpha d(x, y) = \alpha * d(x, y)$

Definition 2.28 (Consistency)**.** Given a clustering C over (X, d), a distance function d' is a *C-consistent variant* of d, if $d'(x, y) \leq d(x, y)$ for all x, y belonging to the same cluster of C and $d'(x, y) \geq d(x, y)$ for all x, y belonging to different clusters of C. A clustering function f is *consistent* if $f(d') = f(d)$ whenever d' is a *$f(d)$-consistent variant* of d

Definition 2.29 (Richness)**.** A clustering function f is *rich* if for each partition C of X, there exists a distance function d over X such that $f(d) = C$

Scale-invariance means that f must not be sensitive to a scaling of the distances. Consistency requires that if the within-cluster distances are decreased and the between-cluster distances are increased, the resulting clustering should be the same. Richness means that it should be possible to obtain any desired partition C if the distance function is modified. These three properties seem reasonable, but Kleinberg

(2002) shows that there exists no clustering function that simultaneously satisfies all three of them. Ackerman et al. (2010) present a quite simple proof of this result:

Let f be a clustering function satisfying all three axioms and let $X = \{x_1, ..., x_n\}$ be a set with three or more elements. Richness implies that there exist distance functions d_1 and d_2 such that $f(d_1) = \{\{x_1\}, ..., \{x_n\}\}$ (every point is a cluster on its own) and $f(d_2)$ is some different clustering. Let $r = \max\{d_1(x, y) : x, y, \in X\}$ and let c be such that for every $x \neq y$, $d'(x, y) = cd_2(x, y) \geq r$. As $d'(x, y) \geq d_1(x, y)$ for all x, y, consistency implies that $f(d') = f(d_1)$ (because the inter-cluster distances have been increased). However, by scale-invariance it holds $f(d') = f(d_2)$, which is a contradiction because $f(d_1)$ and $f(d_2)$ are different clusterings (Ackerman et al., 2010).

Ackerman and Ben-David (2009) argue that consistency is the weakness of these axioms: It requires that a change of the distances must not create any new contender for the best clustering. The impossibility result of Kleinberg (2002) is therefore not an "inherent feature of clustering" but rather an "artifact" of the formalism used by Kleinberg (2002; Ackerman and Ben-David, 2009). Consider a setting with three clusters for example. Now one cluster is moved far apart from the other two (this is a consistent version of the distance function, as only inter-cluster distances are increased). One might argue that the quality of the original clustering hasn't decreased. However, the quality of a 2-clustering, i.e. merging the two clusters whose distance has remained the same, has improved and might be considered a better clustering (see Ackerman and Ben-David (2009), section 3 for an analogous example and a figure). Ben-David and Ackerman (2008) propose that one should axiomatize clustering quality measures (measures that quantify the goodness of a given clustering) instead of clustering functions. See more on this topic in section 3.4.

Despite the impossibility to construct a clustering function that fulfills all three axioms, it is still beneficial to use these and other properties of clustering functions to taxonomize clustering algorithms (Ackerman et al., 2010). Ackerman et al. (2010) concentrate on k-clustering functions, i.e. clustering functions that (besides X and d) take a natural number k as input and output a partition consisting of k clusters. Such a function can satisfy scale invariance, consistency and k-richness (i.e. any k-partition can be obtained) at the same time. Further properties described by Ackerman et al. (2010) include isomorphism invariance (a clustering function should be invariant to a change of labels of the data) and modifications of richness and consistency (inner, threshold and outer richness and inner and outer consistency). These properties can be used to distinguish between clustering algorithms and to make a more informed decision when choosing a clustering algorithm that suits the given task (Ackerman et al., 2010).

2.3.3 Clustering Algorithms

2.3.3.1 Overview As described in section 2.3.1, there are several desirable properties that clusters could have, some of them conflicting with each other. There exists a wide variety of clustering algorithms that incorporate these different concepts of clusters. Overviews can be found in Saxena et al. (2017) or Jain et al. (1999).

Hierarchical methods don't output a concrete clustering but a sequence of clusterings that can be represented by a *dendrogram*. The data is either step by step divided into smaller clusters until each point is a cluster on its own (*divisive* clustering) or one starts with all points as single clusters and gradually merges them (*agglomerative* clustering) (Saxena et al., 2017). Popular algorithms are *single-*, *complete-* and *average-linkage* clustering, which differ in the way they measure distances between clusters. For agglomerative clustering, the two clusters with the smallest distance are merged in every step. For divisive clustering, the cluster is split that yields two clusters with the largest distance (Saxena et al., 2017).

Partitional algorithms do not yield a dendrogram but a concrete partition of the data, usually by optimizing an objective function (Jain et al., 1999). They include a wide range of methods like probabilistic ones, which assume that the data is drawn from a mixture of distributions (e.g. Gaussian mixtures) whose parameters are estimated, as well as the famous and simple k-means algorithm (Jain et al., 1999). The objective function for k-means is the sum of squared errors, i.e. $\sum_{j=1}^{k} \sum_{i=1}^{n_j} ||x_i^{(j)} - c_j||^2$, where $x_i^{(j)}$ is the i-th point in cluster j, n_j is the number of points and c_j the centroid of cluster j (Jain et al., 1999). A heuristic to solve this in an iterative way is to (1) choose k cluster centers (centroids), (2) assign each

point to the nearest centroid, (3) recompute the new centroids and repeat (2) and (3) until convergence (Saxena et al., 2017). k-means is suitable for data with compact, spherical clusters but it needs a number k of clusters to be pre-specified and can't deal with non-convex clusters or clusters of varying size or density (Saxena et al., 2017). It is therefore not really suited for detecting clusters that correspond to connected components. In the next section, DBSCAN, a clustering algorithm that is more appropriate to find such clusters is presented.

2.3.3.2 DBSCAN DBSCAN (Density Based Spatial Clustering of Applications with Noise) (Ester et al., 1996) is a well known and widely used density-based clustering algorithm (Schubert et al., 2017). It has two main advantages: First, DBSCAN is able to find arbitrarily shaped clusters (instead of clusters of convex shape like k-means for example). Second, it doesn't need a number k of clusters to be pre-specified, so it requires less domain knowledge to determine the input parameters (Ester et al., 1996, Hahsler et al., 2019b).

DBSCAN is developed from a density-based notion of clusters: Within each cluster, there is a typical density which is considerably higher than outside the cluster (Ester et al., 1996). The concrete idea of DBSCAN is that each cluster has *core points*, whose ε-neighborhoods contain a minimum number of points $MinPts$. A point which doesn't satisfy this requirement is assigned to a cluster C, if it is in the ε-neighborhood of a core point of C. Points which don't belong to any cluster are classified as *noise* (Ester et al., 1996).

The detailed concept of DBSCAN is introduced with the following definitions from Ester et al. (1996). Let D be a given data set and $dist(x, y)$ a distance function.

Definition 2.30 (Epsilon-neighborhood of a point)**.** The ε-neighborhood of a point x is
$\mathcal{N}_\varepsilon(x) = \{y \in D : dist(x, y) \leq \varepsilon\}$

Based on the ε-neighborhood, points can be classified as *core*, *border* or *noise points* (Hahsler et al., 2019b, Schubert et al., 2017): A point $x \in D$ is a

- *core point* if $\mathcal{N}_\varepsilon(x)$ has a high density, i.e. $|\mathcal{N}_\varepsilon(x)| \geq MinPts$ ($MinPts \in \mathbb{N}$ is a threshold specified by the user)
- *border point* if x is not a core point, but it is in the neighborhood of a core point
- *noise point* otherwise

Now, notions of *reachability* are defined in order to form clusters as regions of connected points:

Definition 2.31 (Directly density-reachable)**.** $x \in D$ is *directly density-reachable* from $y \in D$ wrt. ε and $MinPts$ if
1. $x \in \mathcal{N}_\varepsilon(y)$
2. $|\mathcal{N}_\varepsilon(y)| \geq MinPts$, i.e. y is a core point

This is only symmetric for pairs of core points, but not if one core point and one border point are involved (Ester et al., 1996).

Definition 2.32 (Density-reachable)**.** $x \in D$ is *density-reachable* from $y \in D$ wrt. ε and $MinPts$ if there exists a chain of points $p_1, ..., p_n$, $p_1 = y, p_n = x$ such that p_{i+1} is directly density-reachable from p_i for all $i = 1, ..., n - 1$.

Definition 2.33 (Density-connected)**.** $x \in D$ is *density-connected* to $y \in D$ wrt. ε and $MinPts$ if there is a point $p \in D$ such that both x and y are density-reachable from p wrt. ε and $MinPts$.

Density-reachability is not symmetric, but transitive. Density-connectivity captures the relation of border points of the same cluster; it is symmetric (Ester et al., 1996). Now, one can define a notion of a density cluster:

Definition 2.34 (Cluster (DBSCAN))**.** A *cluster* C wrt. ε and $MinPts$ is a non-empty subset of D satisfying
1. $\forall x, y \in D$: if $x \in C$ and y is density-reachable from x wrt. ε and $MinPts$, then $y \in C$ (*Maximality*)
2. $\forall x, y \in C$: y is density-connected to x wrt. ε and $MinPts$ (*Connectivity*)

Definition 2.35 (Noise (DBSCAN))**.** Let $C_1, .., C_k$ be the clusters of D wrt. ε_i and $MinPts_i$, $i = 1, ..., k$. *Noise* is the set of points that do not belong to any cluster C_i, i.e. $noise = \{x \in D | \forall i = 1, ..., k : x \notin C_i\}$

The definition of a *cluster* implies that a cluster C is uniquely determined by *any* of its core points, i.e. C contains exactly the points that are density-reachable from an arbitrary core point of C (Ester et al., 1996).

The DBSCAN-algorithm to find a cluster works as follows: First, an arbitrary point x among the core points is chosen, Then, all points that are density-reachable from x are assigned to one cluster (Ester et al., 1996). If a cluster is complete, a core point among the remaining core points is chosen and a new cluster is created. If none of the remaining points is a core point, these points are considered noise (Hahsler et al., 2019b). It is possible that a border point is density-reachable from core points in several clusters. As a unique cluster assignment is desired, the original DBSCAN-algorithm assigns such points to the first cluster they are reachable from (Hahsler et al., 2019b, Schubert et al., 2017).

Figure 10 shows two clustering solutions of the same data using DBSCAN (with $\varepsilon = 0.35$ and $MinPts = 5$) and k-means ($k = 2$). DBSCAN correctly identifies two clusters and some additional noise (grey points). k-means is not able to detect the correct clusters. This shows the advantages of DBSCAN mentioned above: It can find clusters of arbitrary shape and the user doesn't have to specify the number of clusters, as it is necessary for k-means. However, DBSCAN is quite sensitive to the choice of its parameters, especially ε. For this example, several values for ε were tried until perfect performance was achieved. In the next section, a method to make DBSCAN less sensitive to the choice of ε is presented.

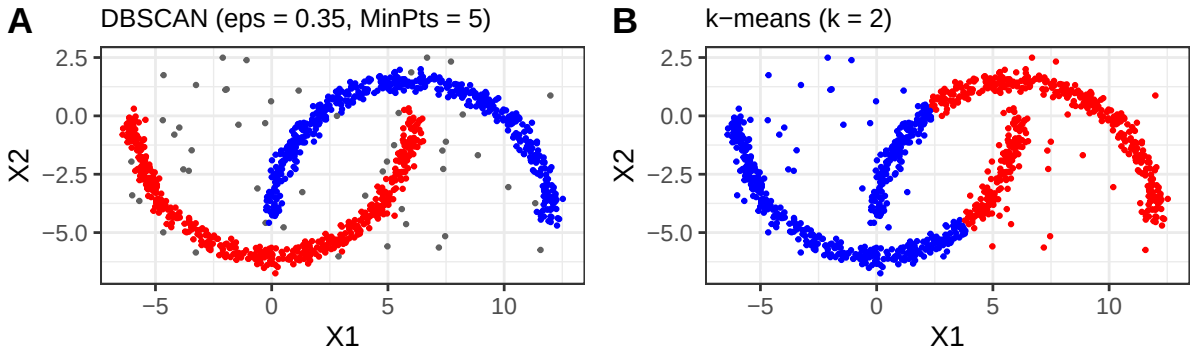

Figure 10: Example of DBSCAN and k-means Clustering

2.3.3.3 Combination of Manifold Learning Algorithms and Clustering Another approach for clustering is *spectral clustering*. Its basic idea is to construct a k-dimensional embedding of the data using spectral decomposition of a matrix related to a similarity graph. The embedding is then clustered using k-means (von Luxburg, 2007). It often outperforms traditional approaches like k-means, is easy to implement and can be solved efficiently using linear algebra (von Luxburg, 2007). Building on the idea of spectral clustering, it is natural to use manifold learning methods for pre-processing and then apply a clustering algorithm to the obtained lower-dimensional embeddings. But first, the basic algorithm for spectral clustering is described in more detail.

The input for spectral clustering is a similarity matrix $S \in \mathbb{R}^{n \times n}$ containing all pairwise similarities $s_{ij} = s(x_i, x_j)$. $s(x_i, x_j)$ is a symmetric, non-negative function, e.g. the Gaussian similarity function $s(x_i, x_j) = \exp(-|x_i - x_j|^2/(2\sigma^2))$ with $\sigma = 1$ (von Luxburg, 2007). Based on this matrix, a similarity graph G is constructed. This can be an ε-neighborhood graph, a k-NN graph or a fully connected graph. The edge weights are always s_{ij}. One can now reformulate the problem of clustering: one wants to find a partition of the graph such that the edge weights between different groups are small and the edge weights within a group are high (von Luxburg, 2007). Let now W be the weighted adjacency matrix and D be the degree matrix defined as follows:

$$W = (w_{ij})_{i,j,=1,...,n} \tag{12}$$

$$D = \text{diag}(d_1, ..., d_n) \text{ where } d_i = \sum_{j=1}^{n} w_{ij} \tag{13}$$

The *unnormalized graph Laplacian matrix* L is then given by (von Luxburg, 2007)

$$L = D - W \tag{14}$$

L is symmetric and positive semi-definite. An important result for spectral clustering is that the number of eigenvalues of L that are 0 is equal to the number of connected components in the similarity graph G (von Luxburg, 2007). The algorithm for spectral clustering[9] takes a number k of clusters as input parameter. Then the first k eigenvectors $u_1, ..., u_k$ of L are computed. The "first k eigenvectors" are the eigenvectors corresponding to the k smallest eigenvalues. This leads to a matrix $U \in \mathbb{R}^{n \times k}$ with $u_1, .., u_k$. The i-th row of U is the embedding $y_i \in \mathbb{R}^k$ of a point x_i. Finally, k-means is applied to the points $(y_i)_{i=1,..,n}$ in $\mathbb{R}^k$ (von Luxburg, 2007).

So the idea of spectral clustering is to compute a representation of the data in $\mathbb{R}^k$. k-means can then easily find the clusters in this embedding (von Luxburg, 2007). From a theoretical perspective, spectral clustering can be derived as an approximation to a graph partitioning problem. As mentioned above, one would like to find a partition of the similarity graph such that the weights of edges between the clusters are small. In addition, the clusters should be "reasonably" large. To formalize this problem, one can define an objective function to measure the quality of a partition of the graph. One can show that the above presented algorithm for unnormalized spectral clustering approximately solves this optimization problem[10] (von Luxburg, 2007).

A disadvantage of spectral clustering is that - like k-means - it requires a number k of clusters to be pre-specified. However, the idea of applying a clustering algorithm to a representation of the data that is easier to cluster than the original data is a promising approach. This leads to the combination of manifold learning algorithms and clustering: One first calculates an embedding of the original data that preservers or even emphasizes certain structures or desired characteristics and then tries to find clusters in the resulting representation. As presented in section 2.3.1.1, clustering can be seen as the task to find the connected components of the manifold underlying the data. It is therefore natural to use manifold algorithms that preserve the topology (see section 2.2) to pre-process the data. UMAP seems particularly suited for this task, as its main goal is to preserve the topological structure (McInnes et al., 2018).

Herrmann et al. (2022) show both from a practical and theoretical perspective that UMAP considerably improves the performance of DBSCAN. Their results on synthetic and real-world data indicate that applying DBSCAN to UMAP embeddings makes the clustering algorithm less sensitive to hyperparameters, especially ε. It can also lead to a perfect clustering in situations where DBSCAN alone is not able to detect the right clusters (Herrmann et al., 2022). Moreover, they explain how and why UMAP enhances clustering from a theoretical point of view: UMAP optimizes for separability by improving the distinction between dense and sparse regions (Herrmann et al., 2022). The improvement comes both from the graph construction as well as the graph layout step: The k-NN graph and the corresponding weight matrix already emphasize the cluster structure by setting all weights except those of the nearest neighbors to zero and by using a local metric to compute similarities. The graph layout step further increases the separability as the cost function (i.e. minimizing the cross entropy) with its attractive and repulsive forces leads to higher between-cluster and lower within-cluster distances (Herrmann et al., 2022). Dalmia and Sia (2021) propose a modified version of UMAP which leads to further improvements in clustering performance. They use a *mutual* k-NN graph[11] and then increase the connectivity of this graph - as it may contain isolated vertices and many disconnected components - by adding edges from a minimum spanning tree. The results of Dalmia and Sia (2021) indicate that this leads to a better separation of

[9] The algorithm described here is *unnormalized* spectral clustering. There are slightly modified versions called *normalized* spectral clustering. See von Luxburg (2007) for details.

[10] There are two common objective functions, *RatioCut* and normalized cut *Ncut*. Unnormalized spectral clustering approximates the solution of RatioCut, normalized spectral clustering approximately solves Ncut. See von Luxburg (2007) for details and derivations.

[11] A mutual k-NN graph is a sub-graph of the original k-NN graph that contains an edge between x_i and x_j if and only if both x_j is a k-nearest neighbor of x_i and vice versa (Dalmia and Sia, 2021).

classes and therefore improves clustering. Herrmann et al. (2022) also find that "dimension inflation" - embedding the data in a higher dimension than the observed one, e.g. from two to three dimensions - can improve separability.

However, enhancing the topological structure as UMAP does comes with the loss of the geometry of the data, i.e. shapes of clusters or how they are positioned relative to each other. Herrmann et al. (2022) show examples of nested spheres or circles and intertwined spirals. UMAP embeddings don't preserve their relative positions and sometimes also their shapes. Additionally, the positioning of clusters is not consistent over different embeddings. Furthermore - because of the local connectivity constraint (every x_i has at least one other point with edge weight 1, see section 2.2.1) - both outliers and noise are embedded into nearby clusters and can't be detected as noise by DBSCAN anymore (Herrmann et al., 2022).

Herrmann et al. (2022) mention that other combinations of manifold learning and/or clustering methods might be successful too. According to van der Maaten and Hinton (2008), t-SNE can reveal clusters and there is even a theoretical guarantee that under some assumptions t-SNE can correctly identify well-separated clusters as shown by Linderman and Steinerberger (2017). On the other hand, Yang et al. (2021) find that these assumptions are often violated for real-world data and they present counter-examples where t-SNE fails to find clusters even if the data is well clusterable. McInnes et al. (2018) state that UMAP preserves more of the global and topological structure than t-SNE and in their experiments, a k-NN classifier on UMAP embeddings achieves a higher accuracy than on t-SNE embeddings. However, Kobak and Linderman (2021) show that t-SNE is not worse than UMAP in preserving global structure and results indicating a poor performance of t-SNE are caused by random initialization. However, UMAP yields denser clusters with more space in between than t-SNE (Kobak and Linderman, 2021). In section 5, experiments on both synthetic and real-world data are conducted to compare UMAP and t-SNE concerning their ability to preserve or even emphasize clusters. In appendix A, concrete embeddings and their effect on the performance of DBSCAN are shown.

3 Separability

3.1 Motivation & Overview

The aim of this thesis is to investigate data separability both from a topological and practical perspective. Fernández et al. (2018) describe separability as an intrinsic characteristic that quantifies how much classes in a data set overlap, i.e. mix with each other, which leads to a more complex decision boundary in classification. This approach is based on the idea that the performance of a classifier depends on two aspects: the classifier model, i.e. the capability of the classifier on the one hand and the separability of a data set on the other hand (Guan and Loew, 2021). Separability in a classification task can not only be characterized as the degree to which different classes mix with each other, but also in terms of the number of hyperplanes needed to separate different classes or the time-cost or accuracy of a specific classifier (Guan and Loew, 2021). This "classification-based" view of separability is closely related to the *complexity* of a data set, i.e. the difficulty of a classification problem (Ho and Basu, 2002). Complexity measures aim to quantify this characteristic. An overview and categorization of these measures can be found in Lorena et al. (2019). Further works using the term *separability* also concentrate on supervised classification (Thornton, 1998, Mthembu and Greene, 2004, Mthembu and Marwala, 2008). However, the focus of this thesis lies on data separability for clustering, i.e. how well a data set is separable in given classes by a clustering algorithm. There are similarities in this "clustering-based" view and the classification-based approaches, but also some important differences. In section 3.2, these differences are described in more detail.

Ghosh et al. (2010) propose a separability index to quantify the difficulty of a clustering problem, however, their approach assumes a mixture of multivariate Gaussians and is thus not appropriate for a topological view of clustering. In order to quantify separability from a topological perspective, one could consider the persistence diagram and measure how persistent (i.e. prominent) the connected components are. This approach is presented in section 3.5. For a data set with given classes, the difficulty of the clustering problem could be measured by the degree to which these classes correspond to connected components. If this is the case and these components are well separated, the classes can easily be detected by a density-based clustering algorithm. The difficulty can be measured by quantifying both the *separation* between and the *connectedness* within classes. In section 3.6.4 such a measure is proposed, which is based both on ideas from complexity measures for classification and *clustering quality measures* (also called *internal cluster validity indices*, section 3.4). Some of these measures are introduced in section 3.6. Their validity and appropriateness as separability measures for clustering is evaluated both on synthetic and real-world data in chapter 5.

Evaluating clustering algorithms using data sets with given classes (i.e. labeled data) is a widely used approach, but Herrmann et al. (2022) raise the question if this is useful at all, as it isn't known if the given classes correspond to meaningful clusters. And, as mentioned in chapter 2.3.1, there is no unique definition of clusters, so Herrmann et al. (2022) state that it might not make sense to evaluate non-probabilistic clustering algorithms (e.g. density-based methods) on data with highly overlapping classes (see Figure 9 **B** for example). The experiments on real-world data in chapter 5 aim to investigate if some data sets that are widely used for evaluation and comparison of clustering methods are suitable for clustering from a topological and density-based perspective.

3.2 Clustering vs. Classification

As mentioned above, in the existing literature, the term separability is mostly used related to supervised classification tasks (e.g. Guan and Loew (2021), Thornton (1998)), which should to be distinguished from a clustering-based view. In both cases, the question is how easy it is to separate a data set in given classes. However, high separability with regard to a classification algorithm doesn't mean that the given classes correspond to meaningful clusters that can be detected by a clustering algorithm.

A common definition of separability is the degree to which points from different classes mix with each other (Guan and Loew, 2021). This makes sense from a classification based view, but for clustering (from a topological perspective), it is not only necessary that the classes don't overlap but also that they form connected components. Figure 11 shows two situations that can be considered as easily separable

from a classification based view but not from the perspective of clustering. In both cases, the data is linearly separable and the classes don't overlap, but they don't correspond to meaningful clusters. In **A**, the classes touch so there is only one connected component. In **B**, the points in class A don't form one connected component but two. A measure that quantifies the separability of a data set with regard to cluster analysis not only has to measure the separation (like a complexity measure for classification) but also needs to take the connectedness within classes into account (**B**). Moreover, separation from a classification view has to be distinguished from separation in a clustering context. In order to form meaningful clusters (i.e. connected components), the domains of different classes must not touch (**A**).

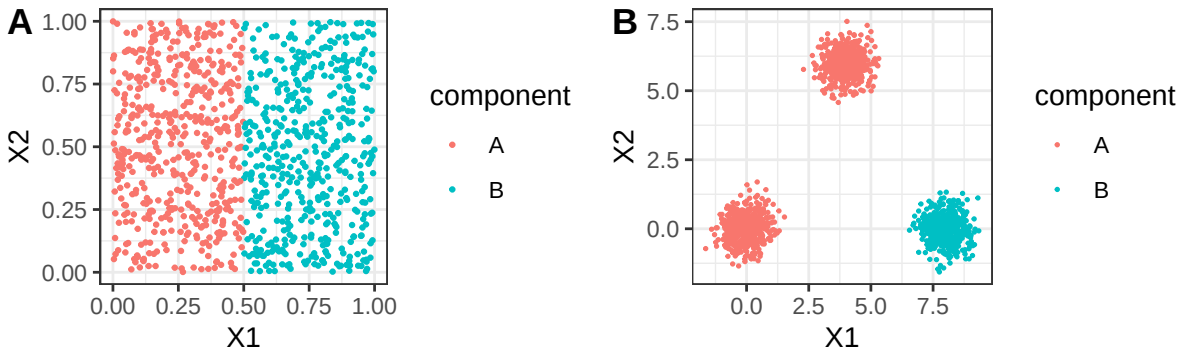

Figure 11: Separability from a classification vs clustering based view

Some definitions of separability in a classification context mention the aspect of the number of hyperplanes needed to separate the classes (Guan and Loew, 2021) and there exist several complexity measures that quantify the degree to which the classes are linearly separable (Lorena et al., 2019). From the perspective of clustering however, a data set that isn't linearly separable is not necessarily difficult to cluster (with a density-based algorithm for example), as the form and the position of the components is of minor importance as long as they are clearly separated (like the nested circles in figure 9 **A** or the moons in figure 10 for example).

In section 3.6, several measures of separability and complexity are presented. Some of them are not completely appropriate from a clustering based view as they consider the above mentioned situations as separable. However, they are included in this thesis, as they still incorporate some important aspects of separability.

3.3 Clusterability vs. Separability with Labels

Separability should not be confused with *clusterability*. This thesis focuses on the separability of a data set, i.e. to what extent it can be separated in given classes by a clustering algorithm. The term "clusterability" refers to a situation without labels; its goal is to evaluate if a data set possesses a cluster structure (Adolfsson et al., 2019). Measures of clusterability aim to quantify how strong this structure is (Ackerman and Ben-David, 2009). As there is no unique definition of true clusters (see section 2.3.1), there exist different measures of clusterability that are sometimes even pairwise incompatible, i.e. there are data sets that are well clusterable according to one measure but at the same time poorly clusterable according to another (Ackerman and Ben-David, 2009).

Ackerman and Ben-David (2009) present two categories of notions of clusterability: One category is based on clustering quality measures (CQMs, see section 3.4): a CQM takes a specific clustering as input and evaluates its "goodness" (Ben-David and Ackerman, 2008). A measure of clusterability can be defined as the optimal value of this measure over all k-clusterings (partitions consisting of k sets) of the data set (Ackerman and Ben-David, 2009). Clusterability can also be evaluated with respect to a loss function (e.g. k-means loss, section 2.3.3), e.g. one can measure the sharpness of the drop in the loss function when moving from a $(k-1)$-clustering to a k-clustering (Ackerman and Ben-David, 2009). However, the computation of some of these clusterability measures is NP-hard (Ackerman and Ben-David, 2009).

Adolfsson et al. (2019) present another category, *clusterability via multimodality*: The idea is to get a one-dimensional summary of the data via PCA or by calculating the set of pairwise dissimilarities. If the data is generated from one cluster, the distributions of the pairwise distances and the first principal component are unimodal, whereas for data from multiple clusters, these distributions are multimodal (Adolfsson et al., 2019). The next step is to apply a test of multimodality on the pairwise distances or the first principal component. If the null hypothesis (H_0: the data is generated from a unimodal distribution) can be rejected, the data is considered clusterable (Adolfsson et al., 2019). The idea of investigating pairwise distances is related to a separability measure called DSI, presented in section 3.6.2. But first, clustering quality measures are described in more detail.

3.4 Internal Cluster Validity Indices & Clustering Quality Measures

An important part of cluster analysis is to evaluate the quality of a clustering in order to chose a suitable clustering algorithm, find an appropriate number k of clusters or tune the algorithm's parameters for example (Hu and Zhong, 2019, Guan and Loew, 2020, Liu et al., 2013). Clustering validation consists of two categories, internal and external. External validation uses external information, i.e. (true) class labels to evaluate the goodness of a clustering. However, in practice, such information is often not available, so internal validation is usually the only option (Hu and Zhong, 2019). An *internal cluster validity index* (CVI) uses only the predicted labels and the data (Guan and Loew, 2020). The term *clustering quality measure* (CQM) (Ben-David and Ackerman, 2008) is used interchangeably. A CVI (or CQM) is a function that takes a clustering and the data as input and returns a real number indicating how "strong" or "conclusive" the clustering is (Ben-David and Ackerman, 2008). An overview on clustering validity indices (both internal and external) can be found in Desgraupes (2016). Ben-David and Ackerman (2008) show that, while an axiomatization of clustering functions isn't possible (see section 2.3.2), clustering quality measures can indeed be axiomatized, i.e. there exist CQMs that satisfy adapted versions of the axioms proposed in Kleinberg (2002) (scale invariance, consistency, richness) and another axiom called "isomorphism invariance".

Guan and Loew (2020) propose to use their newly developed separability measure DSI (see section 3.6.2) as an internal CVI, as the separability of clusters indicates how well the data has been separated, i.e. how good the clustering is. Guan and Loew (2020) compare their measure to some existing CVIs by running different clustering algorithms on some data sets, computing the CVIs for the resulting clusterings and comparing these values with an external validation index used as ground truth. The CVIs whose results are the closest to the ground truth are considered to perform best. However, as the "true" clusters in data sets have a great diversity and every CVI takes different aspects into account, there exists no CVI that performs well on all data sets and the big diversity of CVIs is necessary (Guan and Loew, 2020). Analogous to using separability measures as CVIs, CVIs could be used as separability measures. In order to evaluate to what extent the given classes in a data set correspond to meaningful clusters, the classes can be taken as the result of a clustering and the goodness of this result could be measured by a CVI. Several CVIs are used as separability measures in chapter 5. Their definitions, advantages and disadvantages when used as separability measures are presented in section 3.6.1.

As mentioned in section 3.1, a separability measure for clustering (from a topological perspective) could be defined based on between-cluster separation and within-cluster connectedness. Most CVIs try to measure separation between clusters and/or compactness (instead of connectedness) within clusters (Hu and Zhong, 2019, Liu et al., 2013). Separation quantifies how well separated the clusters are from each other. It is often defined as the distance between representatives of clusters, e.g. as the pairwise distances between cluster centers (e.g. DB index) or the pairwise minimum distance between objects in different clusters (Dunn index) (Hu and Zhong, 2019, Liu et al., 2013). There are also indices that don't use a single representative but include more or other information, like the CVNN-index, which measures separation based on the average percentage of nearest neighbors belonging to a different cluster[12] (Liu et al., 2013). Intra-cluster compactness quantifies how closely related the objects within a cluster are. Compactness can be measured based on variance (e.g. CH-index), the maximum or average distance to the cluster center (average: DB-index) or the maximum or average pairwise distances (maximum:

[12]The definitions and further explanations of all CVIs mentioned here can be found in section 3.6.1.

Dunn-index, average: CVNN) (Hu and Zhong, 2019, Liu et al., 2013). A CVI is then often defined as the ratio of separation and compactness (Liu et al., 2013). As some of the definitions of separation and compactness indicate, most CVIs are mainly suited for spherical clusters, so they don't perform well with arbitrarily shaped clusters (Hu and Zhong, 2019, Liu et al., 2013). Although this behavior doesn't seem suitable from a topological perspective of clustering, CVIs might - just as the complexity measures - capture certain important aspects of separability.

3.5 Proposed Methods to Quantify Topological Separability

How could one measure separability from a purely topological perspective? In section 2.1.2, persistent homology and persistence diagrams are introduced. A cluster from a topological point of view is a connected component, and as the persistence diagram indicates the presence of such connected components, it could be used to quantify the "topological separability" of clusters. In the following subsections, two methods to measure "topological separability" are proposed.

Both methods are based on the idea that if a data set consists of two clearly separable clusters (i.e. the data is sampled from two components that are well separated), the persistence diagram shows two prominent connected components, i.e. components with a long lifetime (in the persistence diagram, this is indicated by points far from the diagonal). In order to measure the separability of two clusters, one could investigate the component with the second longest lifetime, called the "second component" from now on. If a data set consists of one component only or two components that aren't well separated, there is only one long-living component in the persistence diagram, so the lifetime of the second component (if it exists at all) is relatively short. The more prominent the two clusters are, the more persistent is the second component. In order to turn the persistence of the second component into a real number (preferably between 0 and 1), one could calculate the p-value of the second component (section 3.5.1) or its lifetime relative to the lifetime of the first component (i.e. the component with the longest lifetime) (section 3.5.2). Both proposed measures can take any value in the interval $[0, 1]$, with 1 as best value ("highest separability"). The proposed methods could be generalized to situations with k clusters either by calculating the measures for the k-th component (i.e. the component with the k-th longest lifetime) or by computing the pairwise separability between each pair of two clusters and taking the average or minimum separability.

By using the persistence diagram and considering the second component, one implicitly assumes that the connected components represent the "true" classes. However, it is usually unknown if the given labels correspond to connected components, i.e. meaningful clusters from a topological perspective. If this isn't the case, it doesn't make sense to investigate the persistence of the second component. Consequently, the approaches presented here cannot be directly applied to data with classes that aren't sampled from different components (like it could be the case for real-world data) and are only applied to synthetic data that is generated from two components and thus consists of meaningful clusters. The separability of real-world data sets in section 5.3 is only evaluated based on the separability measures presented in section 3.6. These measures aim to quantify both separation and connectedness (or compactness), whereas the proposed methods for topological separability are only able to capture the separation of two existing components.

3.5.1 Topological Separability based on p-Value

As mentioned in section 2.1.3, it is possible to obtain a confidence band for a persistence diagram and decide if connected components (or higher-dimensional features) are statistically significant (see definition 2.16)[13] A feature with birth and death time (u, v) is significant at level α if for its lifetime $t = |u - v|$ it holds $t/2 > c_\alpha/\sqrt{n}$, where c_α is given by (Chazal et al., 2014)

$$P(||\hat{\delta} - \delta||_\infty > \frac{c_\alpha}{\sqrt{n}}) = \alpha \left(\iff P(||\hat{\delta} - \delta||_\infty \leq \frac{c_\alpha}{\sqrt{n}}) = 1 - \alpha \right) \tag{15}$$

So if $t/2 = c_\alpha/\sqrt{n}$, the p-value of the component is α. In general, the p-value of a component can be calculated as follows:

[13] As the validity of the bootstrap isn't shown for the k-nearest neighbor density estimator, this method is only applied to the distance function and DTM.

Definition 3.1 (p-Value of a component)**.** The p-value of a point x in a persistence diagram with lifetime t_x is given by $p(t_x) = P\left(||\hat{\delta} - \delta||_\infty > \frac{t_x}{2}\right)$ where δ is the true distance function/DTM and $\hat{\delta}$ is the estimated version used for the persistence diagram.

The "topological separability" of two components could be defined as $1 - p(t_x)$ where $p(t_x)$ is the p-value of the second component. If the second component is very prominent, the separability approaches 1 (as the p-values tends to 0). As the lifetime (i.e. the Manhattan distance to the diagonal) of the second component decreases, the p-value increases so the separability decreases and is 0 if the p-value is 1.

The p-value can be estimated using bootstrap, analogously to the bootstrap estimation of c_α mentioned in section 2.1.3. The details of the algorithm can be found in Fasy et al. (2014a): Given a sample X, an empiric distance function or DTM $\hat{\delta}$ is calculated. Then, B bootstrap samples X_i^*, $i = 1, ..., B$ are drawn. For each bootstrap sample $\theta_i^* = ||\hat{\delta}_i^* - \hat{\delta}||_\infty$ is computed, where $\hat{\delta}_i^*$ is the estimated function using X_i^*. To obtain a $(1-\alpha)$-confidence band for δ or the corresponding persistence diagram, one calculates the $1-\alpha$-quantile of $\theta_1^*, ..., \theta_B^*$ (Fasy et al., 2014a). The p-value $p(t)$ of a component with lifetime t can be computed by

$$p(t) = \frac{1}{B}\sum_{i=1}^{B} I(\theta_i^* > t/2) \tag{16}$$

The topological separability $1 - p(t)$ is given by $\frac{1}{B}\sum_{i=1}^{B} I(\theta_i^* \leq t/2)$.
A disadvantage of this measure might be that - by making use of statistical significance - it depends on the sample size n. So if data sets with different sample sizes are generated from the same distribution (e.g. a mixture of two Gaussians), the higher n, the smaller the p-value of the second component. However, one could argue that the separability of the components doesn't increase but stays the same. The measure proposed in the next section does not depend on the sample size.

3.5.2 Topological Separability based on Relative Lifetime

The persistence of the second component can also be measured by its lifetime t_2 relative to the lifetime t_1 of the first component[14] (i.e. the component with the longest lifetime), so "topological separability" could be defined as t_2/t_1. If there are two clearly separable clusters, the lifetime of the second component approaches the lifetime of the first component, so t_2/t_1 tends to 1. As t_2 decreases relatively to t_1, t_2/t_1 decreases as the separability of the components shrinks. If the lifetime of the second component approaches 0 (or if there is only one component in the persistence diagram, so $t_2 = 0$), t_2/t_1 tends to 0.

Figures 13 and 14 show examples of the two proposed measures with DTM and knnDE. Separability based on the p-value of the second component is only calculated for DTM. These examples aim to show differences between the proposed measures and the functions used for persistent homology as well as some drawbacks of these methods and functions. The data is shown in figure 12. Each of the four data sets is sampled from two Gaussians. Their standard deviation is always 1. The mean of the first Gaussian is always 0, the mean of the second is 8 for **A** and **C** and 4 for **B**. For **A** and **B**, $n_1 = n_2 = 50$, for **C**, $n_1 = 90, n_2 = 10$, so the total sample size is always 100. The data in **D** is the same as in **A**, but one outlier (-10) was added.

Figure 13 shows the results for DTM. The first row shows the estimated DTM functions (note the different scales of the y-axis) and the second row shows the corresponding persistence diagrams and the relative lifetime and p-value of the second component computed as described above (note that in **A2**, there are two points in the upper left corner, so the lifetime of the second component is almost equal to the lifetime of the first component). A comparison of the first two columns shows that both the p-value and the relative lifetime of the second component indicate that the data in **B** is less separable than the

[14] Note that in the case of persistent homology of a sub- or superlevel set filtration of a function on a closed interval, the birth and death time of the component with the longest lifetime are given by the minimum and maximum value of the function (see Chazal and Michel (2021), example 1). This ensures that in this case, the definition of the lifetime of the first component is meaningful, whereas for a Čech or a Vietoris-Rips filtration there remains one never dying component (Chazal and Michel (2021), example 3). In this thesis, the (multidimensional) interval where the function is evaluated is always given by the minimum and maximum values of the data.

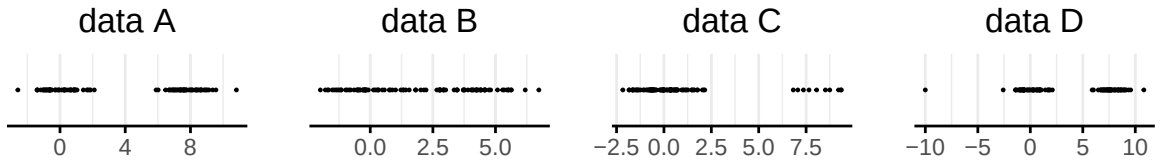

Figure 12: One-dimensional synthetic data for examples of measures of topological separability

data in **A**. The difference between the first and the third row is in the density of the Gaussians: In **A**, the density is equal in both components (as $n_1 = n_2 = 50$), whereas in **C**, the density in the left component is much higher ($n_1 = 90, n_2 = 10$). This example shows a possible disadvantage of the proposed measures when dealing with components with different densities: The measures indicate a lower separability for **C**, although this situation is not necessarily more difficult to separate than **A**, as the distance and the variance of the two components is the same. However, the DTM has higher values in regions with a lower density, so the persistence of the second component (and thereby its lifetime and p-value) decreases.

D shows a drawback when using sublevel set filtrations of DTM (and the distance function analogously) and topological separability based on the relative lifetime: In regions with very few or no observations, the values of DTM can become arbitrarily high (remember that DTM is evaluated on a grid given by the maximum and minimum values of the data). As the lifetime of the first component is defined by the difference between the maximum and the minimum value of DTM, one outlier can drastically increase the lifetime (here from 2.5 to 8) (as the grid where the function is evaluated becomes much bigger). Consequently, the second component becomes much less prominent in the persistence diagram, as its lifetime doesn't change. This leads to a lower relative lifetime, although the separability of the two components doesn't change. So in the presence of outliers, the persistence diagram is at some point misleading, as one cannot interpret the lifetime of the first component relative to the other features shown in the diagram. In such cases, the p-value is much more robust and shows which features are relevant and which not. For density estimators like knnDE, this problem doesn't exist, as the function is bounded below by 0 so it doesn't take arbitrarily extreme values in regions without observations.

The results for knnDE are shown in figure 14. The p-value is not calculated for knnDE, as the validity of the bootstrap isn't shown. Just like for DTM, the relative lifetime in **B** is much lower than in **A**, as the components are less separable. **C** shows that knnDE also cannot deal with clusters with different densities. In **D** one can see the advantage of knnDE (and density estimators in general), as one outlier doesn't lead to a drastic decrease of the relative lifetime of the second component. In chapter 5, the proposed methods to measure topological separability are investigated in more detail.

3.6 Measures of Separability

In this section, several measures of separability are presented. They are grouped in the following categories: *Internal Cluster Validity Indices* (section 3.6.1), *Distributional Approaches* (section 3.6.2) and *Graph- and Neighborhood-Based Approaches* (section 3.6.3). Section 3.6.1 includes some widely used CVIs, most of them based on separation and compactness, as mentioned in section 3.4. The following section (3.6.2) presents a different approach to quantify separability: One can measure how much points from different classes mix with each other, i.e. one quantifies the dissimilarity of the distributions. The third section (3.6.3) contains mostly complexity measures. Complexity measures can be grouped in several categories (see Lorena et al., 2019), but some of them are not suited to measure separability from a clustering based view (e.g. linearity or class imbalance measures). The complexity measures presented here all belong to the categories *neighborhood measures* and *network measures*. Neighborhood measures quantify the presence of points of same or different classes in local neighborhoods (Lorena et al., 2019). Network measures model the data as a graph and extract measures from it (Lorena et al., 2019). As these two approaches are somewhat related, they are presented in the same section.

In section 3.6.4, a new measure is proposed that quantifies separability from the perspective of density-based clustering. It builds both on ideas from CVIs (separability as a ratio of separation and compactness or connectedness) and ideas from complexity measures (modelling the data as a graph).

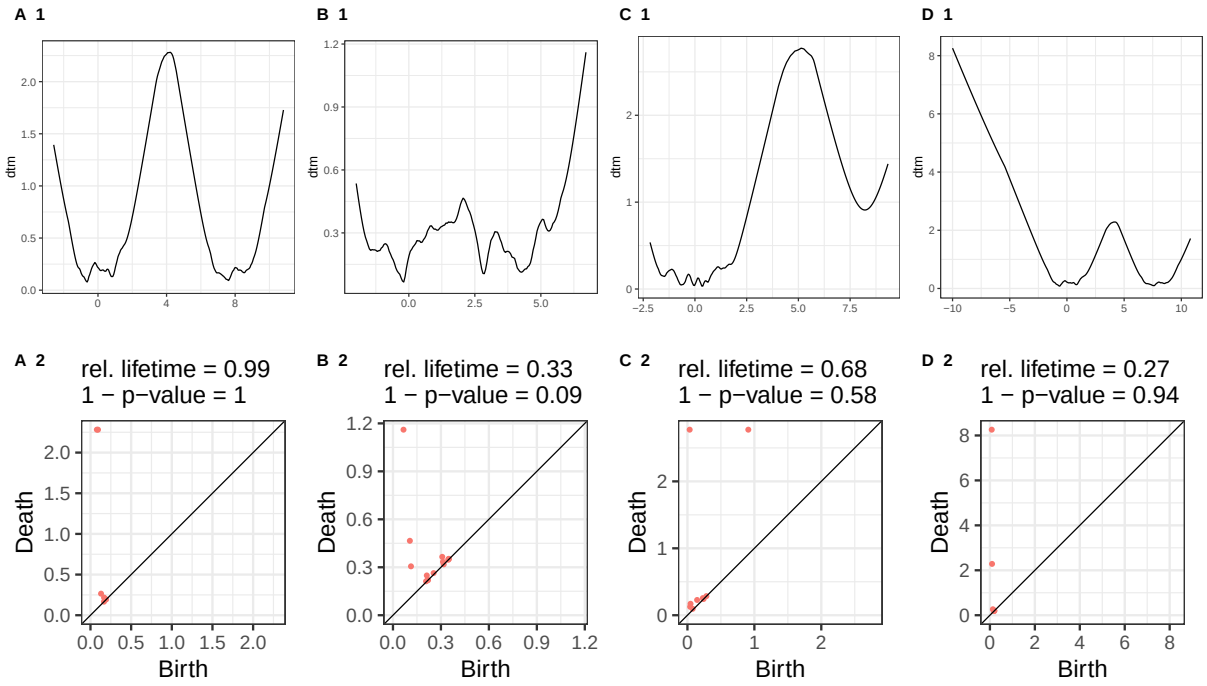

Figure 13: Measures of topological separability on one-dimensional synthetic data, DTM

For each measure, some limitations as well as advantages and disadvantages are discussed. In section 3.6.5, some properties of these measures are summarized and their behavior on exemplary data sets is evaluated. Important properties are suitability for arbitrary shaped classes (e.g. nested circles) and appropriateness for a clustering-based view of separability. The last one refers to situations like in figure 11 (**A**: linear separable with a margin of 0, **B**: three components but only two classes), i.e. a clustering-based separability measure should measure separation from a clustering-based perspective and it should take the connectedness within classes into account.

As it is desirable that all measures are in $[0, 1]$ (or $[0, 1[$ or $]0, 1]$ etc.) with 1 as best value, some measures are slightly modified which is indicated by an asterisk. The notation is as follows: $X = x_1, ..., x_n$ is a given data set with K classes $C_1, ..., C_K$ of sizes $n_1, ..., n_K$ with centers $c_1, ..., c_K$ (i.e. the mean of each class). c is the center of the whole data set. $d(x, y)$ denotes the Euclidean distance between x and y (unless otherwise stated, see section 3.6.3). For some measures, a distance $d(C_i, C_j)$ or a similarity $s(C_i, C_j)$ between two classes C_i and C_j is defined. $Sep(X)$ and $Comp(X)$ denote index specific definitions of separation and compactness. For a point x_i, y_i denotes the class label of x_i.

3.6.1 Internal Cluster Validity Indices

Dunn Index: The Dunn Index (Dunn, 1973) is the ratio of separation and compactness (see section 3.4 or Liu et al. (2013)), where separation and compactness are defined as follows: The distance between two classes C_j and C_j is the minimum distance between points of these classes. The separation $Sep_{Dunn}(X)$ of the whole data set X is given by the minimum distance between two classes (Dunn, 1973, Desgraupes, 2016):

$$d(C_i, C_j) = \min_{x \in C_i, y \in C_j} d(x, y) \tag{17}$$

$$Sep_{Dunn}(X) = \min_{i,j=1,...,K, i \neq j} d(C_i, C_j) \tag{18}$$

For each class C_k, the diameter $diam(C_k)$ is given by the maximum distance of points in this class. The compactness $Comp_{Dunn}(X)$ is defined by the maximum diameter (Dunn, 1973, Desgraupes, 2016):

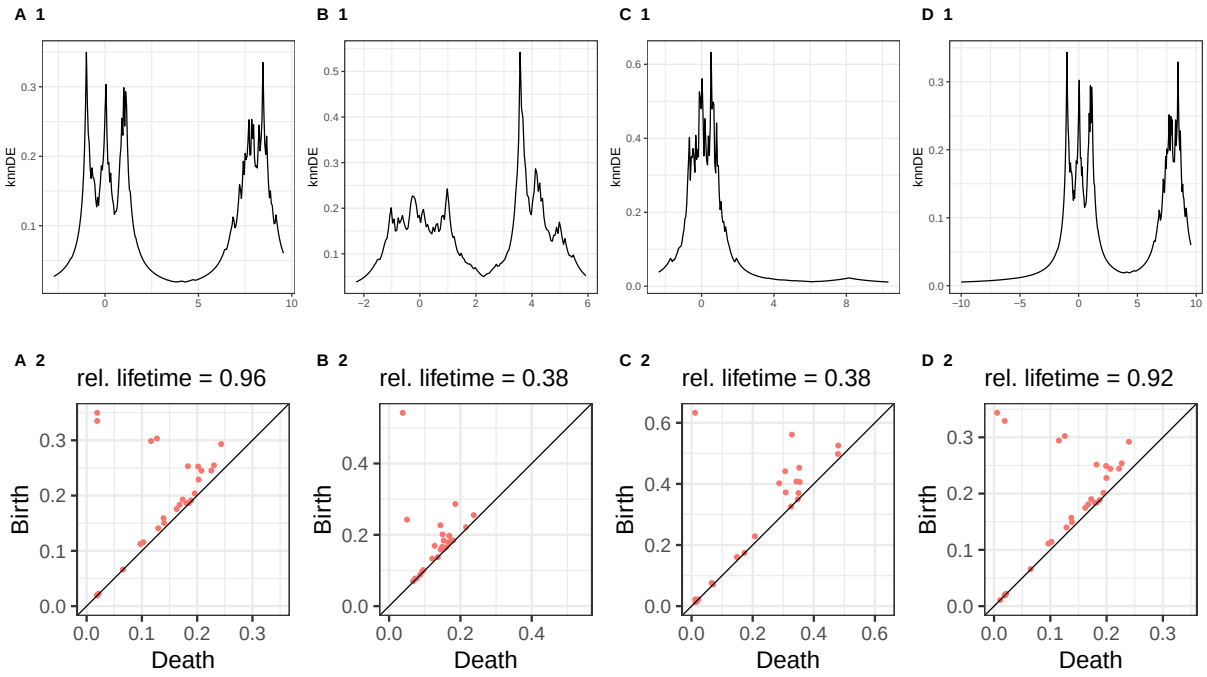

Figure 14: Measures of topological separability on one-dimensional synthetic data, knnDE

$$diam(C_k) = \max_{x,y \in C_k} d(x,y) \tag{19}$$

$$Comp_{Dunn}(X) = \max_{k=1,\dots,K} diam(C_k) \tag{20}$$

The Dunn index is then given by the ratio of Sep_{Dunn} and $Comp_{Dunn}$:

Definition 3.2 (Dunn index)**.** $Dunn(X) = \dfrac{Sep_{Dunn}(X)}{Comp_{Dunn}(X)} = \dfrac{\min_{i,j,i\neq j}(\min_{x\in C_i, y\in C_j} d(x,y))}{\max_k(\max_{x,y\in C_k} d(x,y))}$ (Dunn, 1973)

The higher the Dunn index, the better. As there is no upper limit, the Dunn index is slightly modified to be in $[0,1[$:

Definition 3.3 (Modified Dunn index)**.** $Dunn(X)^* = \dfrac{Dunn(X)}{1 + Dunn(X)}$

By taking the minimum and maximum values of between- and within-cluster distances, the Dunn index is not robust to outliers as just one observation can lead to a drastic decrease in Sep_{Dunn} or increase in $Comp_{Dunn}$. Furthermore, by measuring compactness as the diameter of a class, i.e. the maximum distance within a class, it is not suited for clusters with arbitrary shape but rather for spherical clusters. The Dunn index measures separability rather from a clustering based view than from the perspective of classification, as it doesn't classify the situation in figure 11 **A** (linear separable with a margin of zero) as well separable (because $Sep_{Dunn}(X) \to 0$). However, the evaluation of situations like in figure 11 **B** (three components but only two classes) depends on the distance and position of the components, as the Dunn index does not directly measure connectedness within classes but only the diameter.

Calinski-Harabasz Index (CH): The Calinski-Harabasz Index (CH) (Caliński and Harabasz, 1974) also takes the form $Sep_{CH}/Comp_{CH}$ (Liu et al., 2013). Separation is measured in terms of the weighted sum of squared distances of the class centers to the center c of the whole data set (Desgraupes, 2016, Liu et al., 2013):

$$Sep_{CH}(X) = \frac{1}{K-1}\sum_{i=1}^{K} n_i d(c_i, c)^2 \tag{21}$$

Compactness is based on the within-group variance (Desgraupes, 2016, Liu et al., 2013):

$$Comp_{CH}(X) = \frac{1}{n-K}\sum_{i=1}^{K}\sum_{x\in C_i} d(x, c_i)^2 \tag{22}$$

The CH index is defined as

Definition 3.4 (Calinski-Harabasz index)**.**
$CH(X) = \frac{Sep_{CH}(X)}{Comp_{CH}(X)} = \frac{\sum_{i=1}^{K} n_i d(c_i, c)^2/(K-1)}{\sum_{i=1}^{K}\sum_{x\in C_i} d(x, c_i)^2/(n-K)} = \frac{n-K}{K-1}\frac{\sum_{i=1}^{K} n_i d(c_i, c)^2)}{\sum_{i=1}^{K}\sum_{x\in C_i} d(x, c_i)^2}$ (Desgraupes, 2016, Liu et al., 2013)

The higher the CH index the better, and it can take arbitrary high values. As this index is used as a CVI and the term $\sum_{i=1}^{K}\sum_{x\in C_i} d(x, c_i)^2$ (typically) becomes smaller as the number K of clusters increases, it is corrected by multiplying with $(n-K)/(K-1)$, which decreases as K increases. However, when used as a separability measure, no correction for the number of classes is needed. The modified version of the CH index is therefore given by

Definition 3.5 (Modified Calinski-Harabasz index)**.** $CH(X)^* = \frac{CH(X)^{**}}{1+CH(X)^{**}}$
where $CH(X)^{**} = \frac{K-1}{n-K}CH(X)$

As the CH index measures (squared) distances to class centers, it is not suited for arbitrarily shaped classes. As a clustering quality measure, the CH index measures separability from a mainly clustering-based view: In figure 11 **B**, the within-class variance increases when a class consists of more than one component. In **A**, the CH index cannot directly indicate that the margin is small and its measure of separation between classes does not take into account whether classes touch or overlap. However it can at some point distinguish between big and small margins as the within-class variance increases or variance of cluster centers decreases when different classes get closer to each other and begin to touch or overlap (unlike a measure that just takes linear separability into account as some complexity measures do).

Davies-Bouldin Index (DB): The Davies-Bouldin Index (Davies and Bouldin, 1979) is also based on separation and compactness, although unlike the previous two measures, it is not given by the ratio of two values measuring these quantities. Let δ_j be the average distance of points in C_i to the center c_i of C_i (compactness) and let Δ_{ij} be the distance between the centers c_i and c_j (separation) (Desgraupes, 2016, Liu et al., 2013):

$$\delta_i = \frac{1}{n_i}\sum_{x\in C_i} d(x, c_i) \tag{23}$$

$$\Delta_{ij} = d(c_i, c_j) \tag{24}$$

The similarity between two classes is given by (Liu et al., 2013)

$$s(C_i, C_j) = \frac{\delta_i + \delta_j}{\Delta_{ij}} \tag{25}$$

For each class, the maximum similarity is computed and the Davies-Bouldin index $DB(X)$ is defined as the average of these maximum similarities:

Definition 3.6 (Davies-Bouldin index)**.**
$DB(X) = \frac{1}{K}\sum_{i=1}^{K}\max_{j,j\neq i} s(C_i, C_j) = \frac{1}{K}\sum_{i=1}^{K}\max_{j,j\neq i}\frac{\delta_i+\delta_j}{\Delta_{ij}}$ where $\delta_i = \frac{1}{n_i}\sum_{x\in C_i} d(x, c_i)$ and $\Delta_{ij} = d(c_i, c_j)$
(Liu et al., 2013, Desgraupes, 2016)

As the Davies-Bouldin index measures similarity between classes instead of distance or dissimilarity, smaller values indicate a better separation between classes. In order to transform the values to $]0,1]$ with 1 as best value, the DB index is modified as follows:

Definition 3.7 (Modified Davies-Bouldin index)**.** $DB(X)^* = \dfrac{1}{1+DB(X)}$

By measuring distances to or between cluster centers, the Davies-Bouldin index is mainly suited for center-based clustering and not for clusters of arbitrary shape. Just like the CH index, the DB index rather represents a clustering-based view on separability as it behaves similar to the CH index in situations like in figure 11: A class that consists of more than one component leads to a lower separability as the average distance to its class center increases. Analogous to CH, DB cannot directly indicate a margin of zero but it can distinguish between big and small margins.

Silhouette Index (Sil): The silhouette index (Rousseeuw, 1987) is not based on a ratio of separation and compactness but on the differences of between- and within-cluster distances (Liu et al., 2013). First, a so called *silhouette width* $s(x)$ is calculated for each point x: Let $a(x)$ be the average distance of a point x in class C_i to the other $n_i - 1$ points in C_i, let $\delta(x, C_k)$ be the average distance to the points of another cluster C_k and let $b(x)$ be the minimum of $\delta(x, C_k)$ over all other classes $k \neq i$, i.e. the minimum distance of x to another class; the "second-best choice" for x (Rousseeuw, 1987, Desgraupes, 2016):

$$a(x) = \frac{1}{n_i - 1} \sum_{y \in C_i, y \neq x} d(x, y) \text{ for } x \in C_i \tag{26}$$

$$\delta(x, C_k) = \frac{1}{n_k} \sum_{y \in C_k} d(x, y) \tag{27}$$

$$b(x) = \min_{k=1,\dots,K, k \neq i} \delta(x, C_k) \text{ for } x \in C_i \tag{28}$$

The silhouette width $s(x)$ for each observation x is given by the following quotient (Rousseeuw, 1987, Desgraupes, 2016):

$$s(x) = \frac{b(x) - a(x)}{\max\{a(x), b(x)\}} \tag{29}$$

$s(x)$ is between -1 and 1 and indicates if x is assigned to the "right" cluster: $s(x)$ becomes 1 if $a(x)$ is much smaller than $b(x)$ which means that the average distance to the second-best choice (the class for which the minimum of $\delta(x, C_k)$ is attained) is much higher than the average within-class distance $a(x)$. When $s(x)$ is close to zero, this means that $a(x)$ and $b(x)$ have approximately the same value, i.e. x lies equally far from both its actual class and the second best choice. The worst situation is a silhouette width close to -1 which indicates that $a(x)$ is much bigger than $b(x)$, so x is much closer to the second-best choice than to its actual class (Rousseeuw, 1987).

The $s(x)$ of all points can be plotted and used for graphical evaluations of clusterings (Rousseeuw, 1987). In order to obtain a single value $Sil(X)$ that indicates the goodness of a given clustering (or given classes), one computes the mean silhouette width of each cluster and takes the mean of these values:

Definition 3.8 (Silhouette index)**.** $Sil(X) = \dfrac{1}{K} \sum_{i=1}^{K} \dfrac{1}{n_i} \sum_{x \in C_i} \dfrac{b(x) - a(x)}{\max\{a(x), b(x)\}}$

where $a(x) = \dfrac{1}{n_i - 1} \sum_{y \in C_i, y \neq x} d(x, y)$ and $b(x) = \min_{k=1,\dots,K, k \neq i} \left(\dfrac{1}{n_k} \sum_{y \in C_k} d(x, y) \right)$ for $x \in C_i$ (Desgraupes, 2016, Liu et al., 2013)

As $Sil(X) \in [-1, 1]$ and higher values indicate a better separation, the silhouette index is transformed to $[0, 1]$ as follows:

Definition 3.9 (Modified Silhouette index)**.** $Sil(X)^* = \dfrac{Sil(X) + 1}{2}$

Similar to the other CVIs, the silhouette index is not really suited for arbitrarily shaped classes, as it compares distances within and between clusters, i.e. it measures compactness rather than connectedness within classes. It represents a clustering-based view on separability and behaves similar to the DB and CH index in situations like in figure 11.

CVNN: The CVNN (clustering validation index based on nearest neighbors) (Liu et al., 2013) is a CVI that aims to overcome some limitations of existing CVIs. As it was developed for clustering evaluation and it is based on notions of separation and compactness, the CVNN is presented in this section and not together with other measures that also use nearest neighbors in section 3.6.3. As mentioned above, most CVIs (including those presented in this section) cannot handle clusters of arbitrary shape (Liu et al., 2013). One reason for that is that many indices measure separation based on representatives of clusters, e.g the cluster center like the DB and CH index (Liu et al., 2013). The CVNN uses nearest neighbors to evaluate separation: Let k be a number of nearest neighbors (e.g. $k = 10$) and denote by $q(x)$ the number of k nearest neighbors of x in class C_i that are not in C_i. Separation is defined as the maximum average proportion of nearest neighbors in other clusters (Liu et al., 2013):

$$Sep_{CVNN}(X) = \max_{i=1,\dots,K} \frac{1}{n_i} \sum_{x \in C_i} \frac{q(x)}{k} \tag{30}$$

The lower the value of Sep_{CVNN}, the better the separation between classes. The compactness within classes is given by the sum of average pairwise distance between points in the same class (Liu et al., 2013):

$$Comp_{CVNN}(C_i) = \frac{2}{n_i * (n_i - 1)} \sum_{x,y \in C_i} d(x, y) \tag{31}$$

$$Comp_{CVNN}(X) = \sum_{i=1}^{K} Comp_{CVNN}(C_i) = \sum_{i=1}^{K} \frac{2}{n_i * (n_i - 1)} \sum_{x,y \in C_i} d(x, y) \tag{32}$$

(the factor $\frac{2}{n_i*(n_i-1)}$ is the inverse number of pairwise distances $d(x, y)$ for $x, y \in C_i, x \neq y$). Smaller values of $Comp_{CVNN}$ indicate a better intra-class compactness. Liu et al. (2013) normalize both Sep_{CVNN} and $Comp_{CVNN}$ to $[0, 1]$ and add them up in order to obtain a single value (i.e. $CVNN(X) = Sep_{CVNN,norm} + Comp_{CVNN,norm}$). The smaller the CVNN, the better. As normalization factor, they use the maximum value of Sep_{CVNN} and $Comp_{CVNN}$ among clustering results with different numbers K of clusters (Liu et al., 2013). While this makes sense when comparing clusterings for different numbers of clusters, this is not possible when the CVNN is used as a separability measure, as there are no partitions for different numbers of classes available.

The modified version of CVNN used in this thesis is defined as follows: With the above definition of $Comp_{CVNN}$, this value depends highly on the scale of the distances in the data sets. The modified compactness is given by the mean of $Comp_{CVNN}(C_i)$ (instead of the sum) normalized by the mean pairwise distance in the data set:

$$Comp_{CVNN}(X)^* = \frac{\frac{1}{K} \sum_{i=1}^{K} \frac{2}{n_i*(n_i-1)} \sum_{x,y \in C_i} d(x, y))}{\frac{2}{n*(n-1)} \sum_{x,y \in X} d(x, y)} \tag{33}$$

Now, the sum of $Comp_{CVNN}(X)^*$ and $Sep_{CVNN}(X)$ is transformed to $]0, 1]$ with 1 as best value:

Definition 3.10 (Modified CVNN index)**.**

$$CVNN(X)^* = \frac{1}{1 + Comp_{CVNN}(X)^* + Sep_{CVNN}(X)}$$

$$\text{where } Comp_{CVNN}(X)^* = \left(\frac{1}{K} \sum_{i=1}^{K} \frac{2}{n_i * (n_i - 1)} \sum_{x,y \in C_i} d(x, y)) \right) / \left(\frac{2}{n * (n - 1)} \sum_{x,y \in X} d(x, y) \right)$$

$$\text{and } Sep_{CVNN}(X) = \max_{i=1,\dots,K} \frac{1}{n_i} \sum_{x \in C_i} \frac{q(x)}{k}$$

($q(x)$ denotes the number of k nearest neighbors of x that are not in the same class as x.)

Unlike the other CVIs, CVNN measures separation in a way that is suited for classes of arbitrary shape. However, its notion of compactness is still based on the idea of spherical clusters and quantifies the separability of nested circles for example as low. Just like the other CVIs, CVNN represents a clustering-based perspective on separability, as for classes with more than one component like in figure 11 **B**, the increase in the average pairwise within-class distances leads to a lower separability. For situations like in **A** (linear separable with a margin of zero), CVNN indicates a lower separability because the number of nearest neighbors belonging to the other class increases.

There are further attempts to develop CVIs that are able to deal with non-spherical clusters, for example the CVDD (cluster validity index based on density-involved distance) by Hu and Zhong (2019). Their notion of compactness uses path-based distances (Fischer and Buhmann, 2003) and is somewhat related to the idea of connectedness used for the separability measure proposed in section 3.6.4. The definition of separation in Hu and Zhong (2019) aims to be robust to outliers and to be able to cope with density-separated clusters as well as distance-separated cluster, whereas existing CVIs usually favor the latter.

3.6.2 Distributional Approaches

DSI: The approach of Guan et al. (2020) to separability is different than the one of classical CVIs, as it is mainly based on the perspective of classification. However, their *distance-based separability index DSI* (DSI) can also be used for cluster validation (Guan and Loew, 2020). The DSI is based on the idea that the most difficult situation to separate is when two classes mix with each other, i.e. have the same distribution (Guan et al., 2020). Consequently, separability can be defined in terms of the similarity of the distributions in different classes. However, as the dimensions of these distributions can be very high, the idea of Guan et al. (2020) is to consider (one-dimensional) sets of pairwise distances. Let $ICD(C_i)$ be the set of intra-class distances, i.e. the set of distances between any two points of C_i, and let $BCD(C_i)$ be the set of between-class distances, i.e. the set of distances between any two points x, y where $x \in C_i, y \notin C_i$ (Guan et al., 2020):

$$ICD(C_i) = \{d(x, y) : x, y \in C_i, x \neq y\} \tag{34}$$

$$BCD(C_i) = \{d(x, y) : x \in C_i, y \notin C_i\} \tag{35}$$

Note that these "sets" are multisets, i.e. they can have duplicate elements (here distances) (Guan and Loew, 2021). Guan et al. (2020) show that when $n_i, n_j \to \infty$, if and only if two classes C_i and C_j have the same distribution, the distribution of the ICD and BCD sets is identical (in the case of two classes, so $BCD(C_i) = \{d(x, y) : x \in C_i, y \in C_j\}$). So instead of measuring the similarity of the original distributions, one examines the ICD and BCD sets. Guan et al. (2020) apply the Kolmogorov-Smirnov test (KS) to compare the distributions of the ICD and BCD sets $ICD(C_i), BCD(C_i)$ and measure their dissimilarity $d(C_i)$. The KS test is the maximum distance between two cumulative distribution functions (CDFs). Let F_{ICD_i} and F_{BCD_i} be the CDFs of $ICD(C_i)$ and $BCD(C_i)$. Then $d(C_i)$ is given by (Guan et al., 2020)

$$d(C_i) = KS(ICD(C_i), BCD(C_i)) = \sup_x |F_{ICD_i}(x) - F_{BCD_i}(x)| \tag{36}$$

An alternative would be to use the Wasserstein distance $W(ICD(C_i), BCD(C_i)) = \int |F_{ICD_i}(x) - F_{BCD_i}(x)|dx$ instead of the KS test, but Guan et al. (2020) find that the Wasserstein distance is less sensitive in measuring separability. Higher values of $d(C_i)$ (i.e. close to 1) indicate that class C_i is well separated from the others, as the distribution of the ICD and BCD set are very different. The distance-based separability index (DSI) is defined as the mean of the $d(C_i)$:

Definition 3.11 (DSI). $DSI(X) = \dfrac{\sum_{i=1}^{K} d(C_i)}{K}$
where $d(C_i) = KS(ICD(C_i), BCD(C_i)) = \sup_x |F_{ICD_i}(x) - F_{BCD_i}(x)|$
and $ICD(C_i) = \{d(x, y) : x, y \in C_i, x \neq y\}$, $BCD(C_i) = \{d(x, y) : x \in C_i, y \notin C_i\}$ (Guan et al., 2020)

The DSI is between 0 and 1 and higher values indicate a higher separability. By comparing the within- and between-class distances regarding their distributions, the DSI can cope with classes of arbitrary shape. However, it represents a classification based view on separability, as it doesn't take the connectedness within classes into account but only measures separation.

There are many other ways to measure the similarity of distributions, e.g. divergence measures like the Jensen-Shannon divergence (Lin, 1991), however all approaches based on similarity of distributions only quantify separation but not connectedness.

3.6.3 Graph- & Neighborhood-Based Approaches

This section presents measures from the categories *neighborhood measures* and *network measures* in Lorena et al. (2019). Network measures model the data as a graph and extract information from it. Many neighborhood-based approaches can also be interpreted as graph-based, as some of these measures can also be extracted from (weighted) k-NN graphs or involve the construction of a particular graph or tree (like N1), so these two categories are combined in one section. The first four measures (N1, N2, N3, LSC) are neighborhood measures. The last two measures (Density and ClsCoef) are network measures. They are both extracted from an ε-NN graph, i.e. a graph where two points x_i, x_j are connected if and only if $d(x_i, x_j) < \varepsilon$. Lorena et al. (2019) use the Gower distance (Gower, 1971) for both the neighborhood and the network measures, so in this section, $d(x_i, x_j)$ denotes the Gower distance (however, all these measures can also be used with the Euclidean or any other distance instead). The Gower distance is some kind of normalized Manhattan distances and takes values between 0 and 1 (Gower, 1971, Lorena et al., 2019). To build the ε-NN graph, ε is set to 0.15 in Lorena et al. (2019). Then, the resulting graph is pruned: each edge between observations of different classes is removed (Lorena et al., 2019). The pruned graph is used to extract measures of complexity or separability: The more edges are removed, the lower is the separability. The final graph is denoted by $G = (V, E)$, where $|V| = n$ and $0 \leq |E| \leq \frac{n*(n-1)}{2}$. v_i is the i-th vertex and an edge between v_i and v_j is denoted by e_{ij}.

The complexity measures from Lorena et al. (2019) are all in $[0, 1]$ with 1 indicating the highest possible complexity, i.e. lowest separability. Here, each complexity measure $C(X)$ is presented as $1 - C(X)$. All definitions are taken from Lorena et al. (2019). Some of them can also be found in Ho and Basu (2002).

Fraction of Borderline Points (N1): To obtain this measure, one first builds a minimum spanning tree (MST) (see definition in 3.6.4) from the data. One then computes the percentage of observations that are connected to points from other classes (borderline points, here denoted by $Bord(X)$). Such points are either on the border or in regions with overlapping classes or noise that is surrounded by points from a different class. So the higher the percentage of such points, the lower the separability. Let $(x, y) \in MST(X)$ denote that the points x, y are connected by an edge in the MST build from the data X and let $|Bord(X)|$ be the cardinality of $Bord(X)$. The separability measure $N1(X)$ is given by the proportion of non-borderline points:

Definition 3.12 (Fraction of borderline points (N1))**.** $N1(X) = 1 - \frac{1}{n}|Bord(X)|$
where $x_i \in Bord(X) \iff \exists x_j \in X : (x_i, x_j) \in MST(X) \wedge y_i \neq y_j$ (Lorena et al., 2019)

As a complexity measure, N1 is suited for all shapes of classes. However, it only measures separation and doesn't take connectedness within classes into account, so for situation **B** in figure 11, N1 would indicate a high separability. For linear separable data with a small margin (**A**), a disadvantage of N1 is that when more points far away from the border between the classes are added, N1 indicates a higher separability because the proportion of borderline points decreases although this situation is not easier to separate (from a clustering-based view) than before.

Ratio of Intra/Extra Class Nearest Neighbor Distance (N2): For N2, one compares the sum of distances between each point x_i and its closest neighbor from the same class ($\min_j\{d(x_i, x_j)|y_i = y_j\}$) and the sum of distances between each point and its closest neighbor from a different class ($\min_j\{d(x_i, x_j)|y_i \neq y_j\}$):

Definition 3.13 (Ratio of intra/extra class nearest neighbor distance (N2))**.**

$$N2(X) = \frac{1}{1 + intra\_extra(X)}$$
$$\text{where } intra\_extra(X) = \frac{\sum_{x_i \in X} \min_j \{d(x_i, x_j) | y_i = y_j\}}{\sum_{x_i \in X} \min_j \{d(x_i, x_j) | y_i \neq y_j\}} \text{ (Lorena et al., 2019)}$$

Small values of $intra\_extra(X)$ mean that the distances to the nearest neighbor from the same class are small compared to the distances to the nearest neighbor from a different class, which indicates a high separability.

Just like N1, this complexity measures can deal with classes of arbitrary shape, but it doesn't measure connectedness but only separation. Concerning the linear separable data with a small margin in figure 11 **A**, N2 behaves similar to N1: as the number of observation far away from the border increases, N2 would indicate a higher separability.

Error Rate of the Nearest Neighbor Classifier (N3): N3 is computed from the error rate of a 1-nearest neighbor classifier using a leave-one-out estimate:

Definition 3.14 (Error rate of the nearest neighbor classifier (N3))**.** $N3(X) = 1 - \frac{1}{n}|Err_{NN}(X)|$

where $x_i \in Err_{NN}(X) \iff NN(x_i) \neq y_i$ and $NN(x_i)$ is the predicted label from a 1-NN classifier (Lorena et al., 2019).

$|Err_{NN}(X)|$ denotes the cardinality of $Err_{NN}(X)$, the set of points in X that are misclassified using a 1-NN classifier, so the separability measure N3 is the proportion of points that are correctly classified by this classifier. The more the classes mix with each other, the higher the percentage of misclassified points and the lower the separability. N3 is the same as the proportion of points whose nearest neighbor is in the same class, so it is somewhat related to the notion of separation for CVNN with $k = 1$: Sep_{CVNN} with $k = 1$ is the maximum 1-NN error rate among all classes.

Similar to N1 and N2, N3 is suited for arbitrarily shaped classes but represents a classification-based perspective on separability, which leads to undesired behavior when it comes to classes that consist of more than one component or situations with small margins.

Local Set Average Cardinality (LSC): For LCS, one considers the cardinality of so-called *Local Sets* LS: The LS of an observation x_i is defined as the set of points x_j that are closer to x_i than x_i's closest neighbor from a different class. The local set average cardinality is then given by

Definition 3.15 (Local set average cardinality (LSC))**.** $LSC(X) = \frac{1}{n^2} \sum_{x \in X} |LS(x)|$

where $LS(x_i) = \{x_j | d(x_i, x_j) < \min_l \{d(x_i, x_l) | y_i \neq y_l\}\}$ (Lorena et al., 2019)

In the "least separable" case, each observation x_i is closest to a point from a different class, so each local set has a cardinality of 1 (as it contains only x_i), resulting in a LSC of $1/n$. High values of LSC indicate that the classes are well separated from each other. Note that the maximum possible value of LSC depends on the sizes of the classes.

Unlike N1 and N3, for a well separable data set (e.g. two Gaussians that don't overlap), LSC can reveal how far the classes are from each other (Lorena et al., 2019). Contrary to the other neighborhood-based measures, LSC favors classes of spherical shapes, as it takes within-class distances into account. It can therefore - at least to some extent - measure the connectedness of classes, so the perspective LSC is based on is neither clearly classification- nor clustering-based.

Average density of the network (Density): This network measure is the number of edges in the final (i.e. pruned) graph divided by the maximum number of edges that can exist between n points $(n * (n-1)/2)$:

Definition 3.16 (Average density of the network (Density))**.** $Density(X) = \frac{2|E|}{n * (n-1)}$ (Lorena et al., 2019)

A dense graph (i.e. high values of $|E|$) indicates that there are dense regions within classes, so the separability is high (Lorena et al., 2019).

Density slightly favors spherical shaped classes, as the connectivity in a component with a Gaussian distribution is higher than in a circle for example. Furthermore, density doesn't take the connectedness within classes into account, so it represents a rather classification-based view on separability.

Clustering coefficient (ClsCoef): This network measure quantifies how much vertices of the same class form cliques: For each vertex (i.e. observation) v_i, one calculates the ratio of the number of edges between its neighbors and the maximum number of edges that could exist between them (Lorena et al., 2019). $N_i = \{v_j : e_{ij} \in E\}$ denotes the neighborhood set of v_i and k_i is the size of N_i, so there are $k_i * (k_i - 1)/2$ possible edges between the neighbors of v_i. $|\{e_{jk} | v_j, v_k \in N_i\}|$ is the number existing edges between neighbors of v_i. The clustering coefficient (ClsCoef) is the average proportion of existing edges:

Definition 3.17 (Clustering coefficient (ClsCoef))**.** $ClsCoef(X) = \frac{1}{n} \sum_{i=1}^{n} \frac{2|\{e_{jk} | v_j, v_k \in N_i\}|}{k_i * (k_i - 1)}$

where $N_i = \{v_j : e_{ij} \in E\}$ and $k_i = |N_i|$ (Lorena et al., 2019)

Similar to Density, this measure favors spherical shaped classes and doesn't take the within-class connectedness into account, thereby representing a rather classification-based view.

There are some other complexity or separability measures that can be found in literature. The separability index (SI) by Thornton (1998) is the same as N3 (both can also be extended to more neighbors than just one) (Lorena et al., 2019). A measure called *Hypothesis margin* (HM) (Mthembu and Marwala, 2008) is similar to N2, as it compares distances to the nearest neighbor of the same class with distances to the nearest neighbor of a different class (Lorena et al., 2019). Mthembu and Marwala (2008) combine HM and Thornton's SI to a new hybrid measure that is able to differentiate between situations with a SI of 100% (i.e. situations where no observation has a nearest neighbor from a different class).

The idea by Zighed et al. (2005) is somewhat similar to the network measures: One first builds a graph that connects nearby observations, however they don't use an ε-NN or k-NN graph but a so-called "Relative Neighborhood Graph" (RNG) that contains a vertex between x_i and x_j if and only if the intersection of two hyperspheres centered on x_i and x_j with radius $d(x_i, x_j)$ is empty[15] (Zighed et al., 2005). The next step is similar to the pruning-step in Lorena et al. (2019): all edges that connect observations from different classes are removed. Then, the relative weight of the removed edges (the "cut edge weight statistic") is computed. Zighed et al. (2005) derive the distribution of this statistic under the null hypothesis H_0 that the labels are assigned randomly and then calculate the p-value to evaluate the separability. Similar to most other neighborhood- and graph-based measures, this approach doesn't quantify connectedness but only separation from a classification based view.

3.6.4 Proposed Separability Measure for Density-Based Clustering

As seen in the previous sections, many existing separability measures face disadvantages when dealing with classes of arbitrary shape (most CVIs) or measure separability from a classification-based view like most complexity measures. In this section, a new index is proposed: the *Density Cluster Separability Index* (DCSI) aims to evaluate if given classes in a data set correspond to clusters (i.e. connected components) that can be detected by density-based algorithms like DBSCAN. As mentioned in the introduction of this chapter, such a separability measure should quantify both *separation* (how well are the classes separated from each other) and *connectedness* (how well are the observations within one class connected). Similar to many CVIs, the proposed measure is the ratio of both values. The development of both components is explained below. It uses a notion of core points that is similar to DBSCAN (section 2.3.3.2):

Definition 3.18 (Core points DCSI)**.** The set of core points $\mathcal{C}_i$ of a class C_i wrt. ε_i and $MinPts$ is defined as $\mathcal{C}_i = \{x \in C_i : |\mathcal{N}_{\varepsilon_i}(x)| \geq MinPts\}$ where $\mathcal{N}_{\varepsilon_i}(x) = \{y \in C_i : d(x, y) \leq \varepsilon_i\}$ for $x \in C_i$

[15] The mathematical definition that an edge e_{ij} between two points x_i and x_j is included in the graph (i.e. $e_{ij} \in E$) is: $e_{ij} \in E \iff \forall k, k \neq i, j : d(x_i, x_j) \leq \max\{d(x_i, x_k), d(x_j, x_k)\}$ (Zighed et al., 2005).

Compared to DBSCAN, the core points are calculated separately for each class: each C_i has its own ε_i and the ε_i-neighborhood $\mathcal{N}_{\varepsilon_i}(x)$ of a point $x \in C_i$ contains only observations from C_i.
A possible choice of ε_i is described later. $MinPts$ is a global parameter, however it could also be chosen specific to each class.

Separation: As seen above, separation should not be measured based on a few representatives like class centers, as two nested circles might have the same center for example. Measures based on the mean distance between classes or on nearest neighbors like N1, N2 and N3 might suffer from the following problem: Consider linearly separable one-dimensional data with a margin of zero (e.g. class 1: $x > 0$, class 2: $x \leq 0$) sampled uniformly from an interval $[-a, a]$. As both classes touch, such data is not separable from a clustering-based view. Measures like those mentioned above might increase (i.e. indicate a better separability) if the interval becomes wider, as the mean distance to the nearest neighbor from a different class increases or the fraction of points whose nearest neighbor is in a different class decreases. However, from the perspective of clustering, the separability remains unchanged. This could be indicated by computing the minimum distance between classes, but such a measure is too sensitive to outliers (like the Dunn index). One therefore needs a different way to quantify the "minimum distance" between two classes (or - in the case of more than two classes - between a class and the other classes). Sorting the pairwise distances between points of different classes and taking the 5%-quantile would be robust to outliers but suffers from the same problem as the measures mentioned above: a wider interval would lead to an unwanted increase in separability. The proposed solution to get a robust minimum is to take only the set of core points $\mathcal{C}_i$ into account, so the separation between a class C_i and the other classes could be defined as the minimum distance between core points $x \in \mathcal{C}_i, y \in \mathcal{C}_j, j \neq i$:

Definition 3.19 (Separation DCSI). $Sep_{DCSI}(C_i) = \min_{x \in \mathcal{C}_i, y \in \mathcal{C}_j, j \neq i} d(x, y)$

This measure of separation is relatively robust to outliers but at the same time remains unchanged when further observations that don't affect the separability are added.

Connectedness: Most of the presented measures are not able to adequately measure the connectedness within classes. The CVIs presented here aim to measure compactness, so they favor spherical shaped classes while complexity measures usually don't take anything like compactness or connectedness into account, as this aspect is less relevant for classification. Unlike compactness, connectedness should not be measured based on maximum or average distances within clusters, as they don't indicate if the data is connected and can take relatively high values if the data forms a circle for example. The proposed solution is to connect all data points within one class and compute the biggest distance. A natural way to connect all observations is to build a minimum spanning tree (MST): A MST of a graph is an acyclic subset of edges such that all vertices are connected and such that the sum of all edge weights is the smallest possible (Zhong et al., 2010). Here, the MST of a specific class could be built on a fully connected graph of this class where the edge weights are given by the corresponding distances. The maximum edge weight of the resulting MST indicates if the data within one class is connected. However, if the MST is built on all observations of a class, this measure is very sensitive to outliers. Taking a certain quantile of the edge weights (e.g. the 95%-quantile) can not adequately capture connectedness, as a class that is sampled from two components (figure 11 **B**) has just one extremely high edge weight that connects both components. The solution is again to take only core points into account: The MST is built on the core points and the maximum edge weight of the resulting tree[16] is taken as measure of connectedness within one class:

Definition 3.20 (Connectedness DCSI). $Conn_{DCSI}(C_i) = \max_{i,j:e_{ij} \in V} d(x_i, x_j)$, where V is the set of vertices of $MST(\mathcal{C}_i)$, a minimum spanning tree built from the core points $\mathcal{C}_i$ of a class C_i.

Note that this corresponds to the maximum path-based distance as defined in Hu and Zhong (2019) and Fischer and Buhmann (2003), however Hu and Zhong (2019) take the average path-based distance for their CVI.

[16] There can exist multiple valid MSTs for the same set of points (Lorena et al., 2019). However, as these trees only differ in edges of the same length, the maximum edge weight of the valid MSTs is always the same.

DCSI: So for each class C_i, one has a value of separation $Sep_{DCSI}(C_i)$ between class C_i and the other classes and a value of connectedness $Conn_{DCSI}(C_i)$ within class C_i. Higher values of Sep_{DCSI} and smaller values of $Conn_{DCSI}$ indicate a better separability. So analogous to many CVIs, one could consider the ratio of separation and connectedness as a measure of separability. In order to combine the values of several classes, one can take the minimum (i.e. worst) value of separation and the maximum (i.e. worst) value of connectedness. The ratio is then transformed to $[0, 1[$:

Definition 3.21 (DCSI). $DCSI(X) = \frac{Sep_{DCSI}/Conn_{DCSI}}{1 + Sep_{DCSI}/Conn_{DCSI}}$, where $Sep_{DCSI} = \min_{i=1,\dots,K} Sep_{DCSI}(C_i)$ and $Conn_{DCSI} = \max_{i=1,\dots,K} Conn_{DCSI}(C_i)$ with the above definitions of $Sep_{DCSI}(C_i)$ and $Conn_{DCSI}(C_i)$.

An alternative would be to compute the ratio $Sep_{DCSI}(C_i)/Conn_{DCSI}(C_i)$ for each class and take the average for example. However, $\min Sep_{DCSI}(C_i)$ and $\max Conn_{DCSI}(C_i)$ are stricter and have a "nicer" interpretation, as $\min Sep_{DCSI}(C_i)$ is the minimum distance between core points of different classes for the whole data set and $\max Conn_{DCSI}(C_i)$ is the maximum edge weight among all MSTs. The DSCI takes a value of 0 when the separation between classes is 0. The highest possible separability ($DCSI(X) \to 1$) is achieved when $Sep_{DCSI} \gg Conn_{DCSI}$, i.e. when the minimum distance between core points of different classes is much higher than the maximum path-based distance (i.e. the maximum weight in the MST) between core points of the same class. A DCSI of 0.5 indicates that $Sep_{DCSI} = Conn_{DCSI}$.

Choice of parameters: The DCSI has several parameters needed for the definition of core points: $MinPts \in \mathbb{N}$ and a $\varepsilon_i > 0$ for each class C_i. A point $x \in C_i$ is a core point, if it has at least $MinPts$ observations from C_i in its ε_i-neighborhood. $MinPts$ could be set to similar values as for DBSCAN (in this thesis, $MinPts = 5$ is always used). The choice of ε_i is more complicated, as the range of meaningful values highly depends on the distances within classes. In this thesis, a simple way that might yield meaningful values for ε_i is presented, however the sensitivity of DCSI to its parameters (both ε_i and $MinPts$) needs further research. ε_i could be set to the median distance between points $x \in C_i$ and their $(MinPts * 2)$-th nearest neighbor in C_i:

Definition 3.22 (Proposed choice of epsilon). $\varepsilon_i = median_{x_j \in C_i} d(x_j, x_{(j,MinPts*2)})$, where $x_{(j,k)}$ is the k-th nearest neighbor of x_j in C_i.

Unlike for DBSCAN, there is no global ε but a specific ε_i for each class. As the densities in different classes can highly vary, a single global ε can lead to the effect that some classes with lower density (i.e. higher distances) don't have any core points at all. In order to calculate the connectedness within a class, at least two core points are needed. The proposed choice of ε_i ensures that each class has a reasonable amount of core points. The median (i.e. the 50%-quantile) is chosen (instead of the mean) as it is robust to outliers. However, one could also chose another quantile. The effect of the choice of ε_i and the sensitivity of the DCSI with regard to its parameters need further investigation.

3.6.5 Summary of Properties & Examples

In this section, the behavior of the presented measures on 9 exemplary data sets is investigated. These data sets are shown in figure 15. They reflect different interesting situations: **A**, **B** and **C** are drawn from mixtures of Gaussian distributions. The covariance within the components is always the same, the distance between the means is 2, 4, and 8. These data sets are used to investigate the sensitivity of the presented measures with regard to the distance of components. **D** shows the same data as **C** but one blue outlier is added. **E** and **F** depict classes of non-spherical shape. The data in **G** is drawn from one Gaussian distribution, so this situation could be considered the least separable. **H** and **I** reflect the idea that a separability measure for clustering should behave different from a measure for classification. The data sets resemble those in figure 11, however the data in **I** is not linearly separable (unlike the data in figure 11 **B**). This situation could still be considered as easy in terms of classification whereas for clustering, the given classes don't correspond to meaningful clusters.

Table 1 shows the separability measures evaluated on the 9 data sets. The order of the columns is the same as in figure 15. Almost all measures are able to differentiate between **A**, **B** and **C** (first three

columns), however most neighborhood-based complexity measures have very high values for all three data sets. A comparison of the columns *dist = 8* and *outlier* shows that the Dunn index and LSC are not robust to outliers. All CVIs have rather low values for *moon* and *circle* (**E** and **F**) as they favor classes of spherical shape, but also some other measures have a relatively low separability for these two data sets, like Density, LSC and DSI. All measures assign the random data in **G** the lowest value (or one of the lowest values) of separability. The last two columns show that some measures take relatively high values for data sets that are easy for classification but don't consist of meaningful clusters. Especially the last column (**I**, three components) shows a big difference between CVIs and DCSI on the one hand and complexity measures on the other. These examples summarize the already mentioned properties and disadvantages of the presented measures: CVIs (first five rows) are not suited for classes of arbitrary shape and represent a clustering-based view. Complexity measures (rows 6 to 11) evaluate separability from the perspective of classification, however they don't favor classes of a certain (e.g. spherical) shape. The DCSI aims to overcome the disadvantages of both categories: It is suited for arbitrarily shaped classes but at the same time represents a clustering-based view as it assign a low separability to data that might be easy for classification but don't consist of meaningful clusters. This is achieved by measuring connectedness instead of compactness (**I**) and by quantifying separation from the perspective of clustering (**H**). In chapter 5, these measures are applied to several synthetic and real-world data sets. The conducted experiments are described in the next chapter.

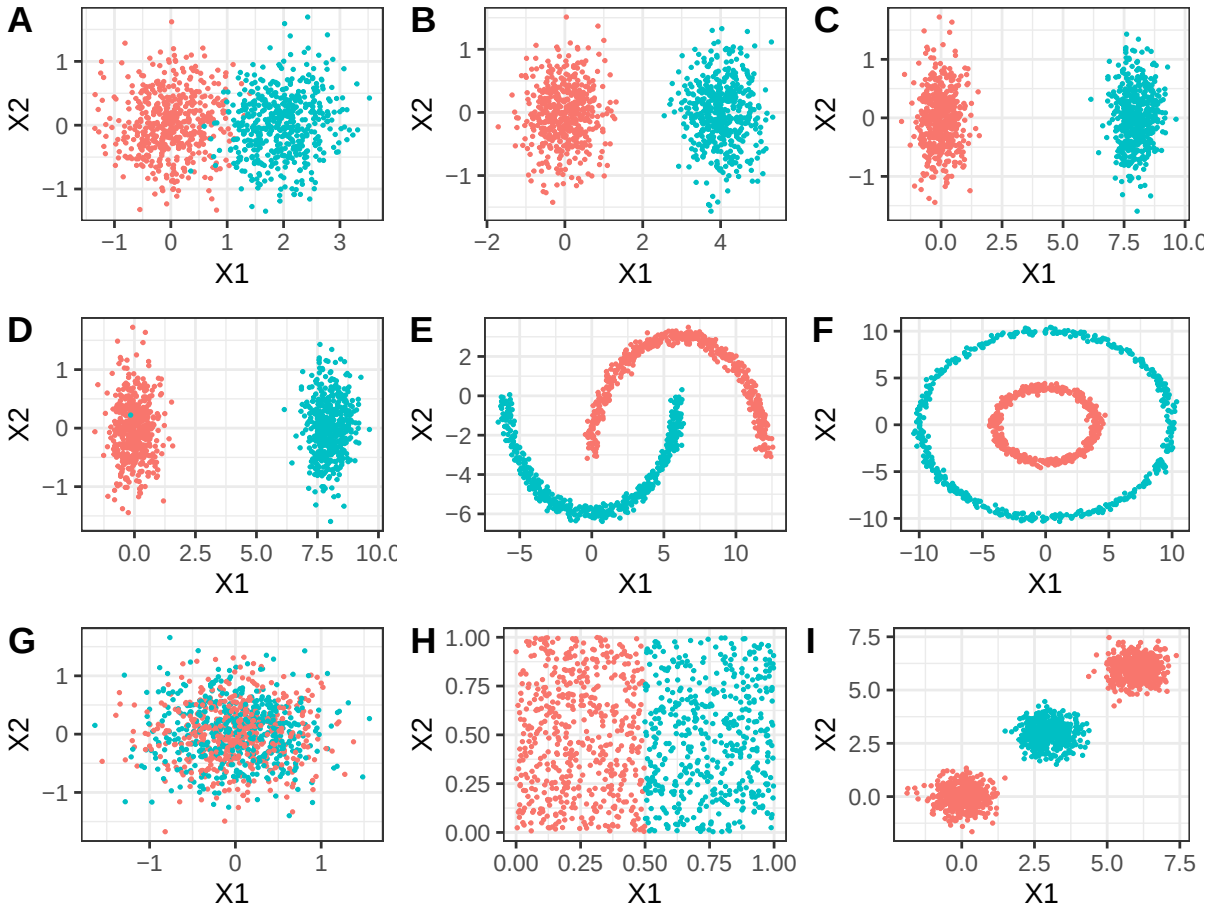

Figure 15: Exemplary data sets to evaluate presented separability measures

Table 1: Separability measures on 9 exemplary data sets

	dist = 2	dist = 4	dist = 8	outlier	moon	circle	random	lin. sep.	3 comp.
CH*	0.66	0.89	0.97	0.97	0.39	0.00	0.00	0.38	0.00
DB*	0.61	0.77	0.87	0.86	0.46	0.05	0.02	0.46	0.00
Dunn*	0.01	0.29	0.57	0.00	0.15	0.18	0.00	0.01	0.09
Sil*	0.78	0.89	0.94	0.94	0.67	0.58	0.50	0.68	0.68
CVNN*	0.61	0.74	0.83	0.83	0.57	0.52	0.40	0.56	0.59
N1	0.96	1.00	1.00	0.99	1.00	1.00	0.31	0.98	1.00
N2	0.88	0.97	0.98	0.95	0.97	0.97	0.50	0.90	0.98
N3	0.97	1.00	1.00	1.00	1.00	1.00	0.52	0.99	1.00
LSC	0.15	0.43	0.50	0.34	0.17	0.15	0.00	0.13	0.33
Density	0.17	0.19	0.19	0.18	0.15	0.13	0.09	0.15	0.19
ClsCoef	0.67	0.70	0.73	0.73	0.78	0.75	0.62	0.68	0.72
DSI	0.70	0.99	1.00	1.00	0.36	0.58	0.01	0.44	0.75
DCSI	0.39	0.91	0.93	0.93	0.85	0.84	0.01	0.23	0.27

4 Experiments - Outline & Data Sets

4.1 Synthetic Data Sets

The goal of the experiments on synthetic data is to find out about (1) the behavior of the presented separability measures and their ability to quantify separability as well as (2) the changes of the performance of DBSCAN and of the separability measures on UMAP and t-SNE embeddings. 9 experiments were conducted in order to answer these questions. Each experiment consists of several data sets and aims to investigate the behavior of the separability measures and the manifold learning methods in a certain situation. Every data set consists of two classes with $n_1 = n_2 = 500$ (except for experiment 3) that are sampled from two (more or less separated) components. For each combination of parameters, there is one data set (e.g. for experiment 1, there are 49 values for d and 31 for σ, so $49 * 31 = 1519$ data sets in total).

- **Experiment 1:**
  - Two two-dimensional Gaussians of varying distance and covariance, 1519 data sets
  - Mean first component: $(0, 0)$, mean second component: $(d, 0)$ with $d = 2, 2.125, 2.25, ..., 7.875, 8$
  - Covariance: the same in both components, $\sigma^2 I$ with $\sigma = 0.5, 0.55, ..., 1.95, 2$
  - Aim E1: investigate the behavior on a simple manifold consisting of two points and two-dimensional noise, same density in both components
- **Experiment 2:**
  - Two two-dimensional Gaussians with different densities, 1525 data sets
  - Mean first component: $(0, 0)$, mean second component: $(d, 0)$ with $d = 2, 2.125, 2.25, ..., 4.875, 5$
  - Covariance first component: $0.5^2 I$, covariance second component: $\sigma^2 I$ with $\sigma = 0.5, 0.55, ..., 3.45, 3$
  - Aim E2: investigate the behavior on a simple manifold consisting of two points and two-dimensional noise with different densities in the components
- **Experiment 3:**
  - Two two-dimensional Gaussians connected by a bridge, 775 data sets
  - Mean first component: $(0, 0)$, mean second component: $(d, 0)$ with $d = 4, 4.25, ..., 9.75, 10$
  - Covariance: the same in both components, $0.5^2 I$
  - A bridge of points (X_1, X_2) is built between the classes by sampling $X1$ from a uniform distribution on $[0, d]$ and $X2$ from $\mathcal{N}(0, \sigma^2)$ with σ being 0.2 of the observed standard deviation

of $X2$. To obtain labels for the points on the bridge, each point is added to the closest component.
  - Density of the bridge: The amount of points sampled for the bridge is $c * n$ ($n = 1000$) with $c = 0, 0.05, ..., 1.45, 1.5$
  - Aim E3: investigate the behavior if two components are connected by a bridge of varying density

- **Experiment 4:**
  - Two two-dimensional Gaussians and additional irrelevant features, 324 data sets
  - Mean first component: $(0, 0)$, mean second component: $(d, 0)$ with $d = 1.5$, $1.75, \ldots$, 4.75, 5, 10, 20, 50
  - Covariance: the same in both components, $0.5^2 I$
  - Additionally, n_{irrev} further features are sampled uniformly from $[0, 1]$ with $n_{irrev} = 0$, $1, \ldots$, 9, 10, 15, 20, 50, 100, 500, 1000, 2000 (i.e. the total number of features is $2 + n_{irrev}$)
  - Aim E4: investigate the behavior if the dimension is artificially increases by adding irrelevant features

- **Experiment 5:**
  - Two multidimensional Gaussians, 228 data sets
  - The data is sampled from two p-dimensional Gaussian with $p = 2, 3, \ldots, 9, 10, 15, 20, 50, 100, 500, 1000, 2000$
  - Mean first component: $(0, 0, ..., 0)$, mean second component: $(d, 0, ..., 0)$ with $d = 1.5, 1.75, \ldots, 4.75, 5, 10, 20, 50$
  - Covariance: the same in both components, $0.5^2 I$
  - Aim E5: investigate the behavior on a simple manifold consisting of two points and multidimensional noise

- **Experiment 6:**
  - Two two-dimensional moons, 820 data sets
  - The data is sampled uniformly from a (2-D) circle with radius 6 and center $(0, 0)$. The upper moon is shifted horizontally by 6 units. Then, the upper moon is shifted vertically by $shift * 6$ with $shift = 0, 0.05, ..., 0.9, 0.95$ (i.e. for $shift = 1$, the moons would touch)
  - Two-dimensional Gaussian noise is added with covariance $\sigma^2 I$ with $\sigma = 0, 0.05, ..., 1.95, 2$
  - Aim E6: investigate the behavior on a manifold consisting of curves

- **Experiment 7:**
  - Two two-dimensional nested circles, 861 data sets
  - One component is sampled uniformly from a circle with radius 4, the other uniformly from a circle with radius r with $r = 5, 5.125, ..., 9.875, 10$. The center of both circles is $(0, 0)$
  - Two-dimensional Gaussian noise is added with covariance $\sigma^2 I$ with $\sigma = 0, 0.05, ..., 0.95, 1$
  - Aim E7: investigate the behavior on a more complex manifold consisting of nested circles

- **Experiment 8:**
  - Two two-dimensional spirals, 51 data sets
  - The data is sampled uniformly from two intertwined (2-D) spirals
  - Two-dimensional Gaussian noise is added with covariance $\sigma^2 I$ with $\sigma = 0, 0.05, ..., 2.45, 2.5$
  - Aim E8: investigate the behavior on a complex manifold consisting of intertwined structures

- **Experiment 9:**
  - Two nested n-spheres, 135 data sets
  - One component is sampled uniformly from a n-sphere with radius 4, the other uniformly from a n-sphere with radius r with $r = 10, 20, 50$. The center of both spheres is $(0, 0)$, $n = 2, 3, \ldots$, 9, 10, 15, 20, 50, 100, 500, 1000 (note that a 2-sphere is 3-dimensional etc., so the highest dimensionality is 1001)
  - Two-dimensional Gaussian noise is added with covariance $\sigma^2 I$ with $\sigma = 0, 0.25, 0.5$

– Aim E9: investigate the behavior on a multidimensional manifold consisting of nested spheres

So E1, E2 and E3 represent different aspects of variation for two-dimensional Gaussians. The interesting aspect of E2 is the different density in both components. E3 aims to answer the questions at which point two component that are (slightly) connected with each other cannot be seen as two clusters in a topological sense anymore. With E4 and E5, the effect of an artificially high dimension is investigated. E6, E7, E9 and E9 represent different non-spherical shapes of varying complexity. E9 is the only setting where the dimension of the manifold is high.

The most "extreme" data sets for each experiment are shown in figure 49, e.g. for experiment 1, these are the data sets with $(d, \sigma) \in \{(8, 0.5), (2, 0.5), (8, 2), (2, 2)\}$ (in this order, the easiest data set in the first column, the most difficult data set in the last one). For E4 and E5, only the two-dimensional data sets are shown, i.e. the data set without irrelevant features (E4) and with two-dimensional noise (E5). As these two-dimensional data sets have the same parameters (same d, same covariance) for E4 and E5, only the data sets from E4 are shown. For E8, only one parameter (the covariance) is varied, so only two data sets are plotted. For E9, the data set with the lowest dimension is three-dimensional and therefore not shown. The corresponding two-dimensional data sets (i.e. 1-spheres) with $\sigma = 0$, $r = 10$ and 50 are shown instead.

The experiments were conducted as follows:

1. For each data set, the topological separability (relative lifetime of second component) and the separability measures are calculated on the raw data. The topological separability is not calculated on the raw data for the high-dimensional experiments (E4, E5, E9) for computational reasons.
2. DBSCAN is applied to the raw data with $\varepsilon \in [0.01, 10]$, step size: 0.01. For the high-dimensional experiments, $\varepsilon \in [0.01, 50]$ (same step size) is used because of the bigger distances in the data set. The clustering for each ε is evaluated using the *Adjusted Rand Index* (ARI) (Hubert and Arabie, 1985) and the *Normalized Mutual Information* (NMI) (Vinh et al., 2009). Note that for E2, E7 and E9, an adjustment was made that is explained below.
3. A UMAP and a t-SNE embedding are computed and the topological separability and the separability measures are calculated on the embeddings. DBSCAN is applied as described above (same ε-range, same step size and evaluation). 2-D embeddings are calculated for all experiments. For E9, 3-D embeddings are computed additionally and their results are compared to the 2-D embeddings.

In the next chapter, the results of these experiments are evaluated. The correlations of the separability measures with maximum ARI are investigated as well as the change in performance and separability on the embeddings.

As mentioned above, for E2, E7 and E9, a slightly different version of ARI and NMI is calculated. The labels that DBSCAN assigns are 1, 2, ... for the clusters and 0 for the noise points. If these labels are compared to the true labels via ARI or NMI, all noise points are taken as one class although DBSCAN didn't assign them to the same class but to *no* class. So it might be critical to simply take the labels from the DBSCAN output and compare them to the true labels via ARI or NMI, especially in cases where the densities of the components strongly differ and DBSCAN perfectly detects the component of higher density but classifies all points from the other component as noise. Such a situation is shown in figure 48. The data set is taken from experiment 2 and is shown on the left (together with the true labels). On the right, the clustering that achieves the highest ARI is shown; the noise points are grey. This clustering has an ARI close to 1, as almost all points of the red component form one cluster and almost all points of the other component of higher density are noise points, so the label assigned by DBSCAN is 0. However, one might argue that this clustering solution is far from perfect, as DBSCAN doesn't state anything about the similarity of noise points (their only commonality is that they don't belong to any cluster). As these situations might be common for E2, E7 and E9 (as for these experiments, the density is not the same in both components), adjusted versions of ARI and NMI (ARI_2, NMI_2) are computed for these settings: Each noise point is assigned to its own cluster and these alternative labels are compared to the true ones via ARI and NMI. If DBSCAN perfectly detects one component and classifies the other component as noise points, ARI_2 is 0.5 while ARI is 1.

4.2 Real-World Data Sets

Five real-world data sets are also evaluated with the aim to investigate if some frequently used data sets are suitable for clustering from a topological perspective and if UMAP and t-SNE can improve the separability. The procedure for the real-world data was similar to the synthetic experiments except for the following changes:

1. No topological separability is computed, as this doesn't make sense on classes that aren't sampled from components of a manifold.
2. A 2-D and a 3-D embedding is calculated for all data sets, however the 2-D embedding is only used for visualization and the 3-D embedding is used for the calculation of separability and for DBSCAN.
3. The separability measures are not only calculated for the whole data set but also for each pair of classes in order to investigate which classes are well separated and which not.
4. The ε-ranges for DBSCAN differ, as meaningful values depend on the dimensionality and the scale of the data. The step size is always 0.01.

The data sets used for the experiments are described below. They aim to reflect a variety of difficulties (e.g. varying number of classes, observations, dimensions). For very large data sets, one subsample was drawn for computational reasons.

- **Iris (Fisher, 1936, Anderson, 1935):**
  - $n_{obs} = 150, n_c = 3, p = 4$ (n_{obs} = number of observations, n_c = number of classes, p = number of features)
  - Measures of sepal and petal length and width for three types of iris plants
  - The features were scaled
  - $\varepsilon_{raw} \in [0.01, 5], \varepsilon_{umap}, \varepsilon_{tsne} \in [0.01, 30]$
- **Wine (Forina et al., 1998, Dua and Graff, 2017):**
  - $n_{obs} = 178, n_c = 3, p = 13$
  - Results of a chemical analysis of three types of wines
  - The features were standardized
  - $\varepsilon_{raw} \in [0.01, 10], \varepsilon_{umap} \in [0.01, 15], \varepsilon_{tsne} \in [0.01, 30]$
- **MNIST (Lecun et al., 1998):**
  - $n_{obs} = 10000$ (subsample of the original data with $n = 70000$), $n_c = 10, p = 784$
  - Handwritten digits, 28x28 grayscale images
  - The whole data set was standardized, i.e. not column-wise but the data was treated as a matrix
  - $\varepsilon_{raw} \in [1, 40], \varepsilon_{umap} \in [0.01, 10], \varepsilon_{tsne} \in [0.01, 30]$
- **FMNIST-10 (Xiao et al., 2017):**
  - $n_{obs} = 10000$ (subsample of the original data with $n = 70000$), $n_c = 10, p = 784$
  - Fashion products of 10 classes, 28x28 grayscale images
  - Classes: 0 = T-Shirt/Top, 1 = Trouser, 2 = Pullover, 3 = Dress, 4 = Coat, 5 = Sandal, 6 = Shirt, 7 = Sneaker, 8 = Bag, 9 = Ankle boot
  - The whole data set was standardized, i.e. not column-wise but the data was treated as a matrix
  - $\varepsilon_{raw} \in [1, 40], \varepsilon_{umap} \in [0.01, 15], \varepsilon_{tsne} \in [0.01, 30]$
- **FMNIST-5 (Mukherjee et al., 2019):**
  - $n_{obs} = 10000$ (subsample of the original data with $n = 70000$), $n_c = 5, p = 784$
  - 5-class version of FMNIST-10
  - Classes: 1 = T-Shirt/Top, Dress, 2 = Trouser, 3 = Pullover, Coat, Shirt, 4 = Bag, 5 = Sandal, Sneaker, Ankle Boot
  - Same ε-range as for FMNIST-10, same standardization
- **CIFAR-10 (Krizhevsky, 2009):**
  - $n_{obs} = 10000$ (subsample of the original data with $n = 60000$), $n_c = 10, p = 1024$

  - Images of animals and vehicles, 32x32 color images. A grayscale, normalized version was used (https://github.com/mlampros/DataSets)
  - Classes: 1 = airplane, 2 = automobile, 3 = bird, 4 = cat, 5 = deer, 6 = dog, 7 = frog, 8 = horse, 9 = ship, 10 = truck
  - $\varepsilon_{raw} \in [0.01, 20], \varepsilon_{umap} \in [0.01, 30], \varepsilon_{tsne} \in [0.01, 30]$

4.3 Parameters & Software

The parameters for the experiments were chosen as follows:

- **Topological separability:** $m_0 = 0.01$ (DTM), $k = 0.1n$ (nearest neighbor parameter for knnDE, n is the sample size), the three functions (DTM, knnDE, distance functions) are evaluated on a grid given by the maximum and minimum values of the data, the space between the points of the grid is 0.1 in each dimension.
- **Separability measures:** The ε-value for the network measures is 0.15 (as in Lorena et al. (2019)), the nearest neighbor parameter for CVNN is $k = 10$ (as in Liu et al. (2013)), the $MinPts$ parameter for DCSI is 5 and the ε_i are chosen as proposed in 3.6.4.
- **DBSCAN:** $MinPts = 5$, the ε-ranges are described above.
- **UMAP:** UMAP was always used with spectral initialization and *min-dist* $= 0.1$, the dimension of the output (2- or 3-D) is described above, the nearest neighbor parameter is $k = 15$ for all synthetic experiments (this value was chosen based on results of a pilot study) and $k = 10$, as this value yields the best results for most data sets in Herrmann et al. (2022)).
- **t-SNE:** the perplexity parameter is $k = 30$ for the real-world data and all synthetic experiments except E4 and E5, $k = 40$ for E4 and E5 (these values were chosen based on results of a pilot study).

The experiments can be reproduced with the provided code. All analyses were conducted in *R* (R Core Team, 2021). The synthetic data is sampled using the *TDA* (Fasy et al., 2022) and the *tdaunif* package (Brunson et al., 2020) for the circles, n-spheres and spirals. The persistence diagrams are calculated using the *TDA* package. The complexity measures are computed with the package by Lorena et al. (2019), *ECoL* (Garcia and Lorena, 2019) and all CVIs except CVNN with the *clusterCrit* package (Desgraupes, 2018). CVNN, DSI and DCSI were calculated using own implementations. The packages used for DBSCAN, UMAP and t-SNE are *dbscan* (Hahsler et al., 2019a), *umap* (Konopka, 2022) and *Rtsne* (Krijthe, 2015).

5 Experiments - Results

5.1 Evaluation of Methods to Quantify Topological Separability

5.1.1 p-Value

In order to evaluate the methods to quantify the topological information on separability proposed in section 3.5, the two measures are computed for synthetic data sets. Figure 16 shows the separability measure based on the p-value of the second component (i.e. $1 - \text{p-value}$) for synthetic data sampled from two Gaussians. The mean of the first Gaussian is always $(0, 0)$ and the mean of the second Gaussian is $(x, 0)$ with $x = 15, 14.5, ..., 2.5, 2$. The distance of the means (i.e. x) is plotted on the x-axis, the topological separability of the corresponding data set is shown on the y-axis. The covariance matrix for both Gaussians is always the identity matrix I_2. From both Gaussians, $n_i = 1000$ observations are sampled (so $n = 2000$). Note that the data is only sampled once and then the desired mean of the second Gaussian for each data set is added. The p-value is calculated based on $B = 100$ Bootstrap samples. The smoothing parameter m_0 for DTM is 0.1.

Figure 16 shows that for both the distance function (denoted by *dist*) and DTM, the topological separability quite rapidly goes from 0 to 1. For the distance function, this is the case between $x = 8$ and $x = 12.5$, while for DTM, there is only one data set whose separability is not 0 or 1 ($x = 3.5$). The data sets and persistence diagrams corresponding to these rapid increases in separability are shown in the following figures.

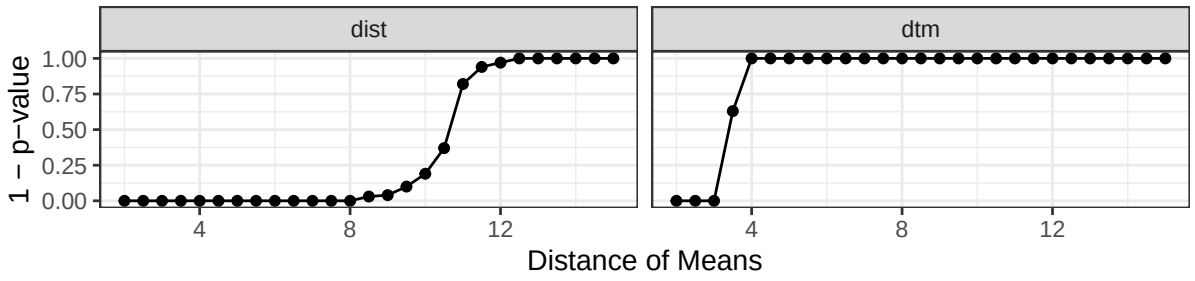

Figure 16: Evaluation of topological separability: p-value, overview

Figure 17 shows the data sets for $x = 12, x = 10, x = 8$. The corresponding persistence diagrams obtained from the distance function are plotted in figure 18. The topological separability decreases from 0.97 for $x = 12$ to 0 for $x = 8$, although the data in figure 17 still clearly depicts two components for $x = 8$. So the p-value of the second component when using the distance function doesn't seem suitable to quantify the topological information on separability.

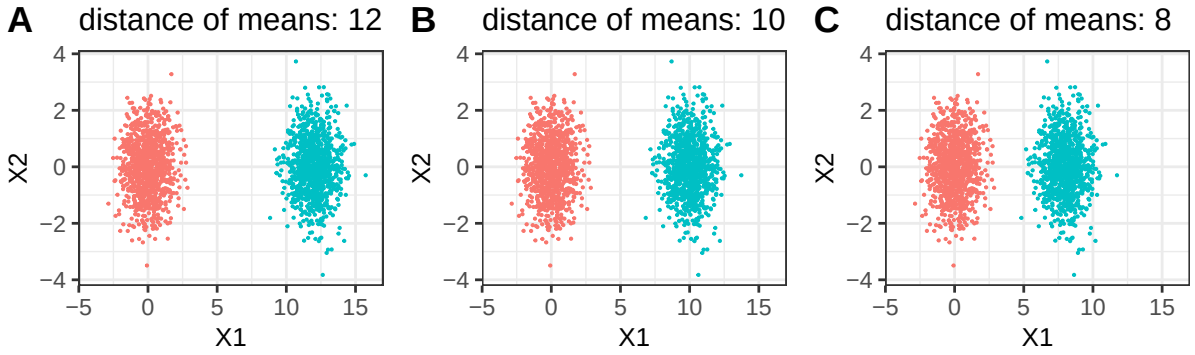

Figure 17: Evaluation of topological separability: selected data sets (distance function)

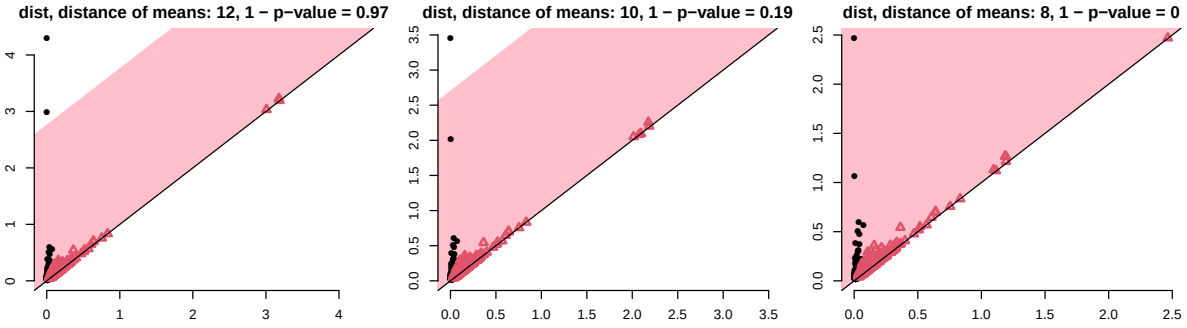

Figure 18: Evaluation of topological separability: p-value and persistence diagrams distance function

Figure 19 depicts the data sets for $x = 4, x = 3.5, x = 3$. The corresponding persistence diagrams obtained from DTM are shown in figure 20 (the confidence bands are much smaller than for the distance function, as DTM is much more robust to small changes in the data). The topological separability decreases from 1 for $x = 4$ to 0 for $x = 3$, which means that the p-value for DTM is not really sensitive for changes in the distance of the two components, as for all values of $x \geq 4$, it outputs a separability of 1. For $x = 4$, the components overlap so this situation should be considered as less separable than data without overlap like in figure 17. Furthermore, the different values obtained for the distance function and DTM show that the p-value highly depends on the variability of the evaluated function. The p-value seems therefore not suited to capture the topological information on separability, so this approach is not investigated anymore in the following.

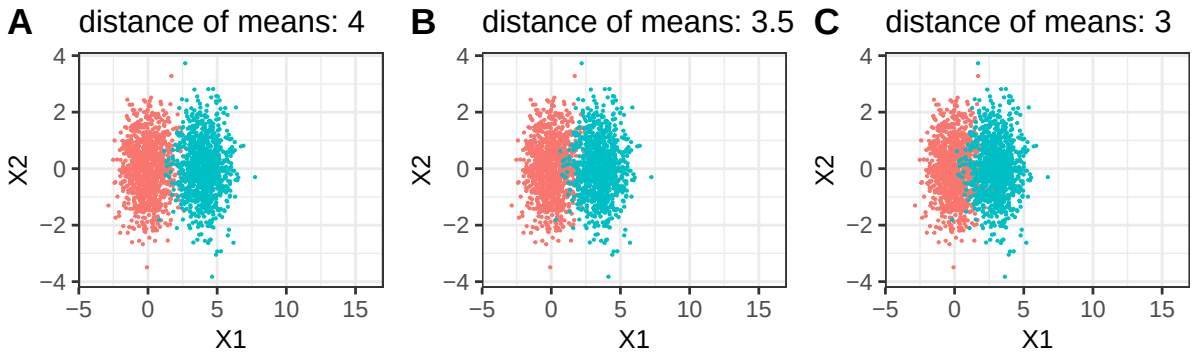

Figure 19: Evaluation of topological separability: selected data sets (DTM)

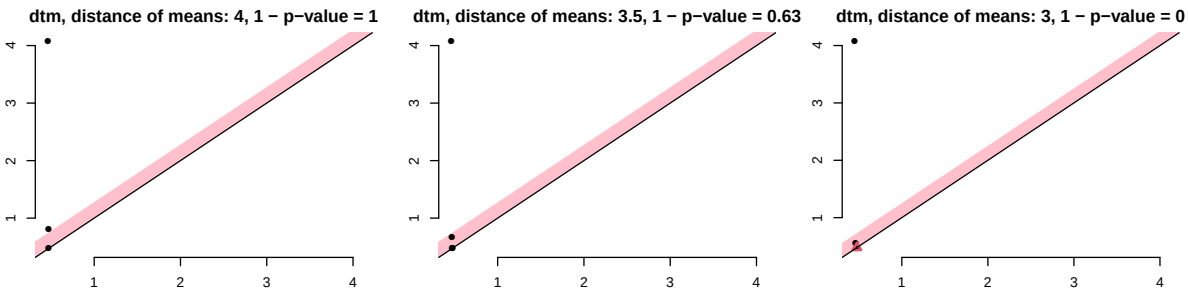

Figure 20: Evaluation of topological separability: p-value and persistence diagrams DTM

5.1.2 Relative Lifetime

Figure 21 shows the evaluation of topological separability based on the relative lifetime of the second component. The data sets are the same as in the previous section. The relative lifetime for DTM and the k-NN density estimator (knnDE) is a strictly increasing function of the distance between the two Gaussians, however only for knnDE, the topological separability takes values close to 1 for well separable data like those in figure 17. For distances $x \geq 6.5$, the relative lifetime using the distance function strictly increases, but for small values, the separability is 0.25 and doesn't take values close to 0 like DTM and knnDE. A possible reason might be that the distance function is less robust to noise than DTM and knnDE, so even for data that is hardly separable, there is a relatively long-living second component when using the distance function.

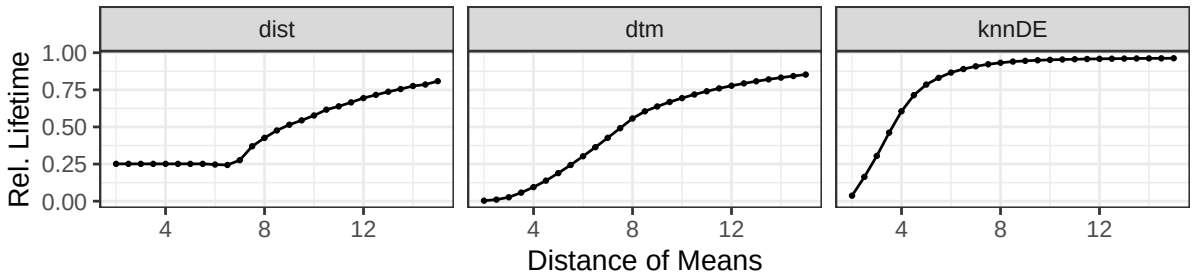

Figure 21: Evaluation of topological separability: relative lifetime, overview

Figure 22 shows the variability of the three measures: For each distance $x = 3, 4, ..., 10$, 100 data sets consisting of two Gaussians with means $(0, 0)$ and $(x, 0)$ are sampled ($n_i = 500$ so $n = 1000$, covariance is I_2 in both components). The figure shows that DTM is the most robust of the three functions.

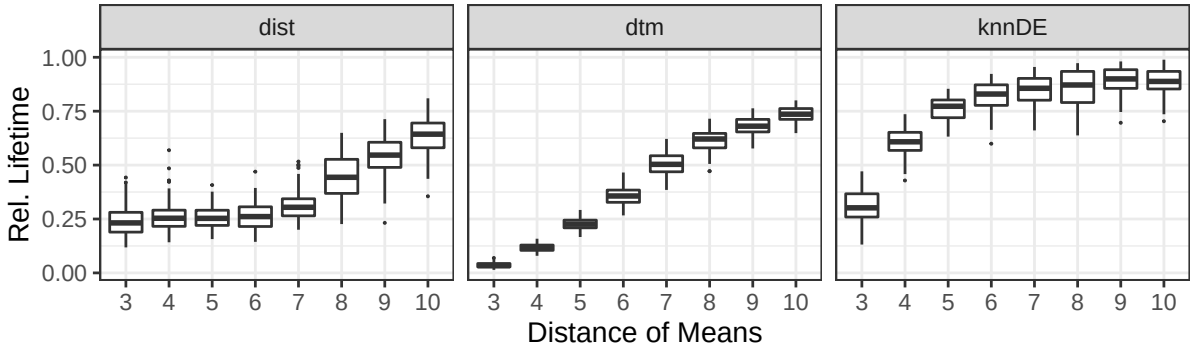

Figure 22: Evaluation of topological separability: relative lifetime, variability

Figure 23 shows the sensitivity to the smoothing parameter m_0 when using DTM. The first row depicts three exemplary data sets. The second row shows the relative lifetime computed for different values of m_0 between 0.01 and 0.15. The moon data is the least sensitive to changes in m_0. For the nested circles, the separability increases for smaller values of m_0, whereas for the two Gaussians in the last column (*point*, as a Gaussian can be described as data lying on a manifold that is a point plus Gaussian noise), the separability slightly decreases for smaller values of m_0. As the nested circles in in the first column are clearly separable and this data set is the most sensitive to changes in m_0, $m_0 = 0.01$ (which yields the "best" value for the nested circles) is used for the experiments in the next sections[17].

[17]The relatively small values of separability for these well-separated circles might be explained as follows: The density on the outer circle is lower than on the inner circle (as the same amount of points is sampled from both circles). As mentioned before, this leads to a lower estimated DTM-function on the component of lower density and therefore a lower topological separability. Another reason might be in the shape of the circles, which leads to higher distances of the nearest neighbors. This effect is explained and investigated in more detail in the evaluations of experiments 7 and 8 in sections 5.2.7 and 5.2.8

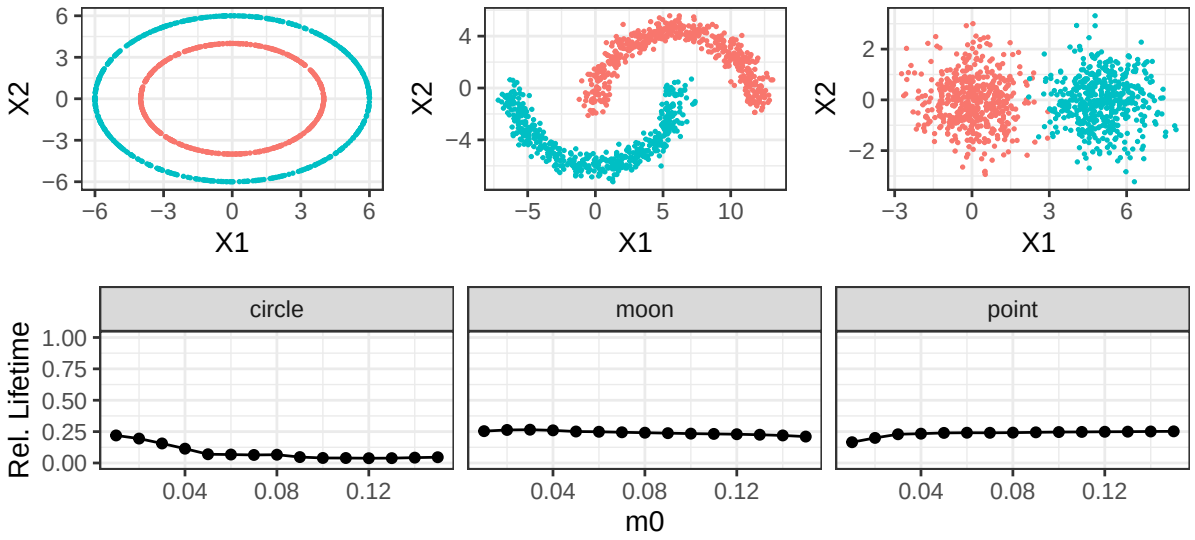

Figure 23: Evaluation of topological separability: relative lifetime, parameter sensitivity DTM

The results shown in figure 21 and 22 suggest that the topological information on separability for these synthetic settings (i.e. no outliers, see section 3.5.2) could be adequately measured by the relative lifetime of the second component. This measure is therefore used in the next sections for large scale experiments on synthetic data. Further results and plots can be found there.

5.2 Separability Measures and Manifold Learning Methods on Synthetic Data

5.2.1 Experiment 1

The results of experiment 1 (two two-dimensional Gaussians, varying distance and covariance, 1519 data sets) are summarized in figures 24 and 25. The structure of these plots is the same for all experiments. Figure 24 **A** shows boxplots of the maximum values of ARI and NMI[18], the topological separability (relative lifetime of second component) for DTM, the distance function and k-nn density estimator as well as the values of thirteen separability measures. The boxplots depict the raw data and the UMAP and t-SNE embeddings. This plot gives an overview on the range of each measure as well as the change between the original data set and the embeddings. Note that all separability measures and the topological separability have a theoretical range of (or close to) $[0, 1]$. A good separability measure should take values on the whole range and thereby differentiate between different data sets in the best possible way.

B shows the Spearman correlation[19] between the separability measures and ARI. The first row shows the correlation on raw data, whereas the correlation on the embeddings is shown in the second and third row. The aim of this plot is to show how strong the separability measures correlate with the performance of DBSCAN measured by maximum ARI. A good separability measure should be highly correlated with the clustering performance. The correlation between NMI and the separability measures is not shown here, as NMI is highly correlated with ARI.

The scatterplots in **C** show the relation between ARI and some separability measures in more detail. The first row shows the topological separability, the second one shows DCSI, DSI, the best complexity measure and the best CVI. The complexity measure and the CVI are chosen based on the correlation with ARI and the range of observed values: The higher the correlation with ARI and the wider the range of observed values, the better.

[18] Here and in the following, the maximum value of ARI and NMI among all values for different ε is always considered.

[19] Here and in the following, Spearman correlation is always used instead of Pearson, as the aim is not to quantify the linear correlation (Pearson), but to measure any kind of monotonic relationship as Spearman correlation does.

D shows the relation of maximum ARI, topological separability using knnDE and DCSI and parameters of the data sets. For experiment 1 for example, the distance of means of the two Gaussians and the covariance ($\sigma^2 I$, the same in both components) was varied. For each combination of distance (x-axis) and standard deviation σ (y-axis), there is one data set. The corresponding values of this data set for the raw data and the two embeddings are shown in this figure. The most difficult data sets (here: smallest distance of means and highest standard deviation) are always shown in the upper left corner, the easiest data sets are in the lower right corner. The aim of this plot is to show how strong the values of ARI, $Topol_{knnDE}$ and DCSI correlate with the parameters of the data set for the original data and the embeddings.

The focus of figure 25 lies on the change in the performance and the separability measures on the embeddings. For each data set, the change of all measures is computed, i.e. $ARI_{UMAP} - ARI_{raw}$, $ARI_{t-SNE} - ARI_{raw}$ etc. Boxplots of these values are shown in **A**. Note that the scale of the plots varies.

B shows the correlation of the change, i.e. the correlation of $ARI_{UMAP} - ARI_{raw}$ and $DCSI_{UMAP} - DCSI_{raw}$ etc. This plots aims to show if changes in ARI correspond to changes in the separability measures and vice versa.

The structure of **C** is similar to the one of **D** in figure 24. Here, the change of ARI, $Topol_{knnDE}$ and DCSI depending on the parameters of the data sets is plotted, so this plot shows for which data sets the embeddings improve the performance or separability.

The plots for the other experiments have the same structure as figures 24 and 25. The complexity measure and the CVI shown in 24 **C** are chosen separately for each experiment and and the axes of 24 **D** and 25 **C** show the parameters specific to each experiment.

Figure 24 **A** shows that the median of maximum ARI and NMI is higher for the embeddings than for the raw data. UMAP leads to a slightly better performance than t-SNE. However, not all separability measures show this improvement as well. $Topol_{knnDE}$ for example has lower values on the t-SNE embeddings than on the original data. This finding is discussed in more detail later. The median of DCSI drops for both UMAP and t-SNE embeddings, while the results of DSI are relatively similar to the performance measured by ARI.

The range of observed values differs among the measures: the network-based measures *Density* and *ClsCoef* (section 3.6.3) only take values in a very small range. Recall that *Density* is the number of edges in the pruned graph divided by the maximum number of edges that can exist between n points. It seems plausible that - for this specific situation of two Gaussians - this measure takes only values in a small interval, as no data set yields a pruned graph without edges ($Density = 0$) or a nearly fully connected graph ($Density \to 1$). Despite having a theoretical range of $[0, 1]$, for a given data set with two classes and $n_1 = n_2 = n/2$, approximately half of the edges that could exist between n vertices ($\approx n^2/2$) cannot be contained in the pruned graph, as these $n^2/4$ edges connect points of different classes, so *Density* doesn't take values higher than 0.5. The clustering coefficient *ClsCoef* quantifies how much vertices of the same class form cliques (i.e. subsets of vertices such that every two vertices are connected by an edge). This measure is not likely to vary much among the data sets evaluated in this experiment, as it is not much affected by the distance of means for example.

Besides some high values, Dunn* mostly takes values close to zero, as it just takes the minimum distance between and the maximum distance within classes into account. DCSI, DSI and CH* take values across the whole range, while the values of the neighborhood-measures N1, N2 and N3 are relatively close to one, as the amount of points having a nearest neighbor from a different class (N3) or on the "border" between the classes (N1) is relatively small for all data sets. LSC (local set average cardinality) doesn't take values higher than 0.5. Note that this is due to its definition (section 3.6.3): Despite having a theoretical range of $[1/n, 1]$, for a setting with two classes and $n_1 = n_2 = n/2$, each local set $LS(x_i)$ can at most have a cardinality of $n/2$, yielding a LSC of at most 0.5.

The Spearman correlation in **B** shows that on the raw data, all measures except $Topol_{dist}$ are highly correlated with ARI. The low correlation of $Topol_{dist}$ can be explained by the lack of robustness of the distance function. The correlation on the embeddings is slightly smaller for DCSI, $Topol_{dtm}$ and $Topol_{knnDE}$ than on the original data. This phenomenon is discussed later. The correlation of *ClsCoef* is much smaller on the embeddings than on the original data. Plots of the embeddings (figure 50) show that

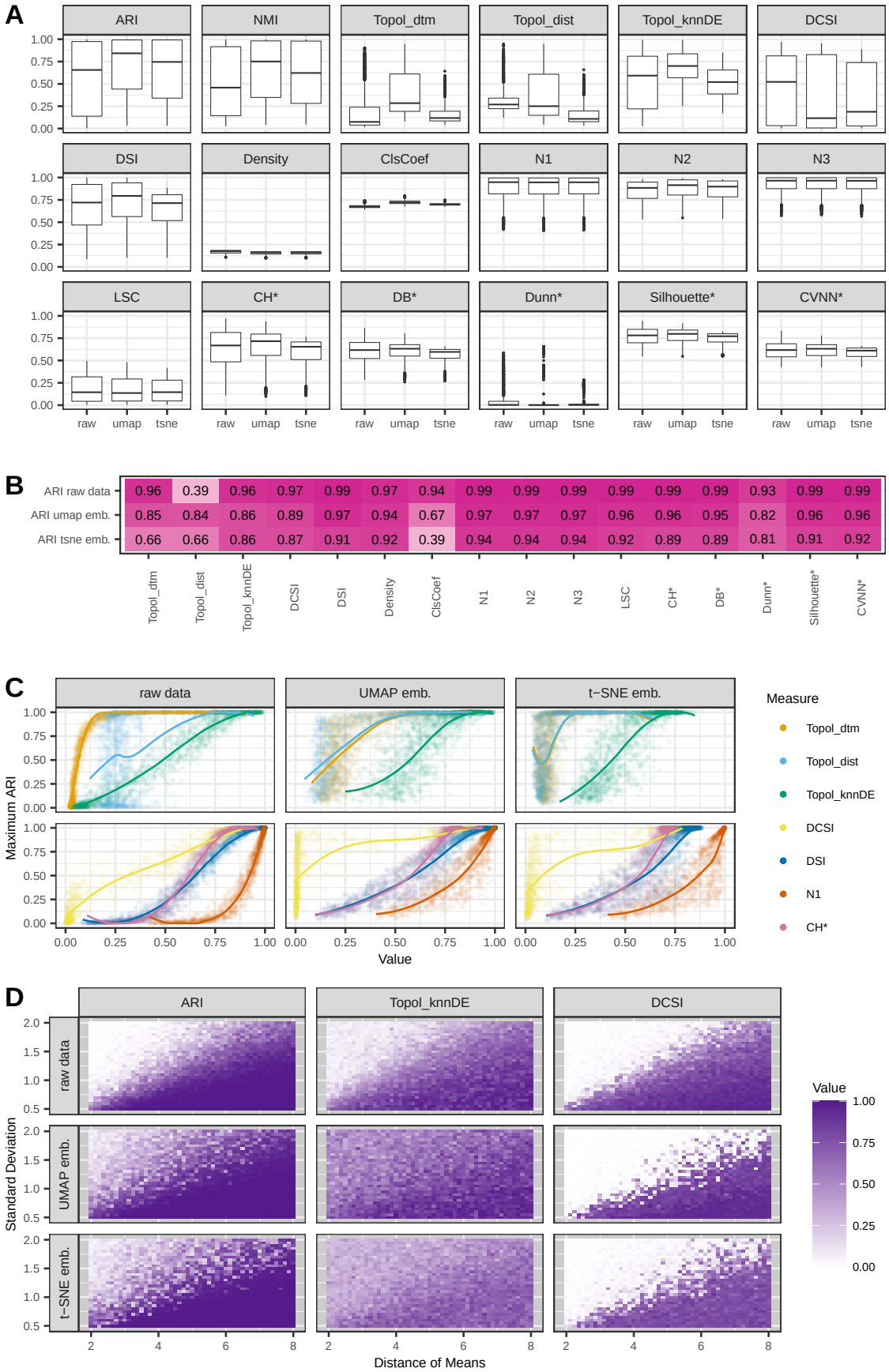

Figure 24: Results Experiment 1: performance and separability measures on raw data and embeddings

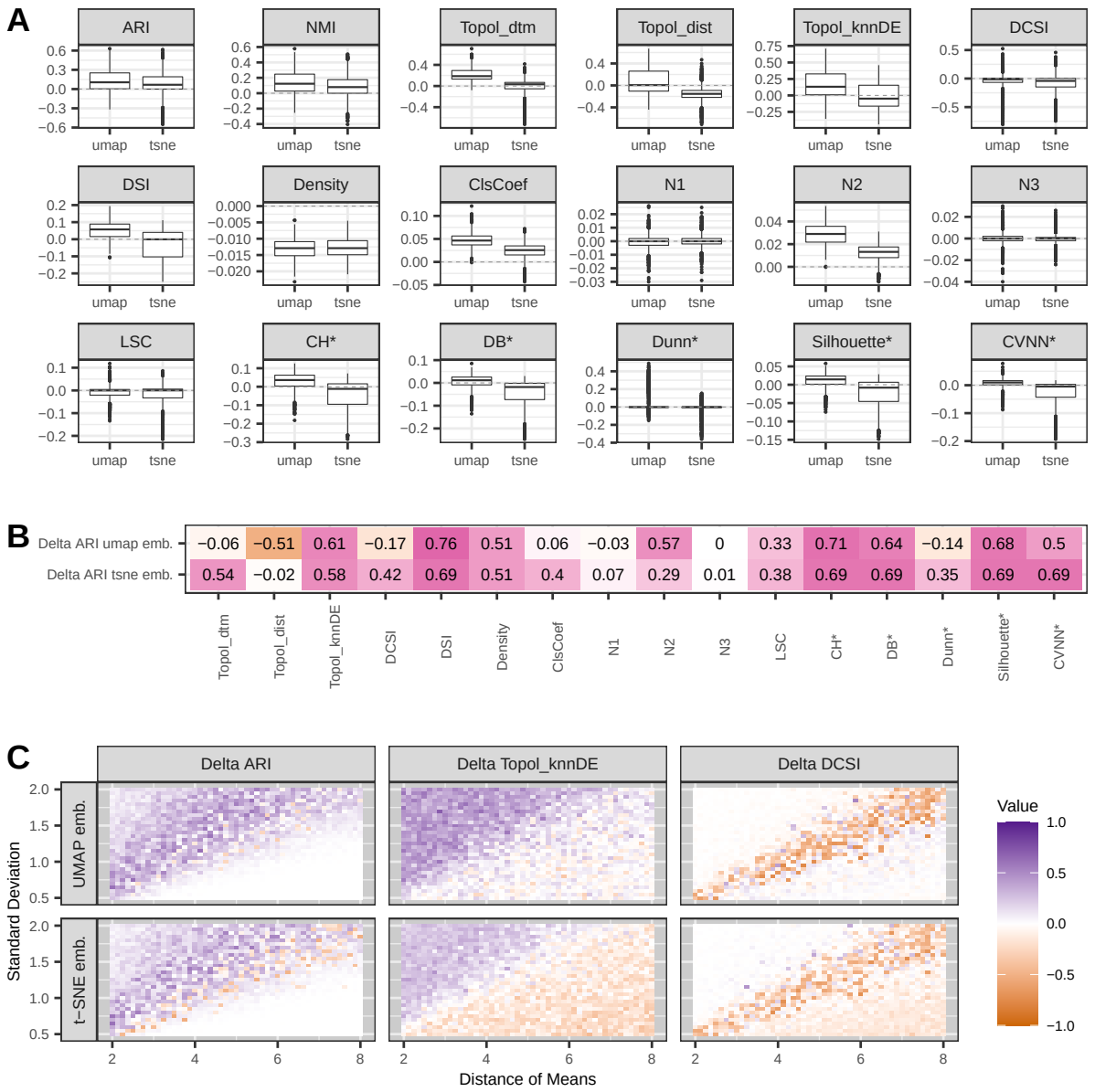

Figure 25: Results Experiment 1: change in performance and separability measures on embeddings

often, despite forming two well-separated components (e.g. the UMAP embedding in figure 50 **B**), the data within the components tends to "cluster less", as there are several regions of low density within the classes. This might explain why the correlation of *ClsCoef* with ARI is relatively low on the embeddings, as ARI mostly depends on the overall separation between classes while *ClsCoef* quantifies how much the data forms clusters within the classes.

The scatterplots in **C** show the relation of some measures with ARI in more detail. On the raw data, $Topol_{dist}$ mostly takes values around 0.25 and rarely correlates with ARI. On the embeddings, there is a stronger relation (and higher correlation, see **B**), which might be explained by the fact that the embeddings merge outliers to components, which makes $Topol_{dist}$ less sensitive. The second row shows that on the embeddings, there are a lot of data sets with a small DCSI, relatively independent of the performance measured by ARI. As mentioned above, this is discussed later.

D shows that for ARI and DCSI, the relation with the parameters of the data sets (distance of means and standard deviation) is quite strong both on the embeddings and the original data. For DCSI,

there is some sort of "cut" on the embeddings: the transition of difficult data sets (upper left corner) to easy data sets (lower right corner) is less smooth for the embeddings than for the original data, whereas for ARI, the transition is relatively smooth in all three cases. $Topol_{knnDE}$ correlates with the parameters on the raw data but less on the embeddings. This behavior and the change of these values is investigated in more detail in the next figure and with some concrete embeddings shown in figure 50.

Figure 25 **A** shows boxplots of changes between the raw data and the embeddings. Values above zero indicate a higher value for the embedding[20]. The plot shows that both UMAP and t-SNE embeddings lead to an improved or equal performance for approximately 0.75% of the data sets, with a slightly better performance of UMAP. However, this improvement isn't reflected in most of the separability measures, as the median is in many cases close to or below zero. Nevertheless, most measures have higher values for UMAP than for t-SNE, indicating that for this experiment, UMAP yields more successful embeddings.

B shows the Spearman correlation of change. For some measures, the change between the original data and the embeddings is relatively high correlated with the change in ARI (e.g. $Topol_{knnDE}$, DSI, CH*, DB*, Silhouette*). Again, the correlation of $Topol_{dist}$ and ARI is low (here even negative), as the change of $Topol_{dist}$ is probably more influenced by outliers being added to components than a general improvement of the separability. DCSI and Dunn* have a slightly negative correlation for UMAP embeddings. A possible reason could be that these two measures are less robust than most of the others as they use maxima and minima, so an embedding that is in general successful and has a higher ARI can have relatively low values for DCSI and Dunn*, if one (core) point is merged to the wrong cluster (see figure 50 **B** for example).

C shows for which data sets the embeddings lead to an increase or decrease of ARI, $Topol_{knnDE}$ and DCSI. Most of the easy data sets (lower right corner) already have an ARI of (or close to) one (figure 24 **D**) and the performance remains unchanged for the embeddings. For the difficult data sets, there is an improvement in most cases, however there are also data sets where the performance decreases for the embeddings. Figure 50 shows the data sets with the highest decrease in ARI for UMAP (**A**) and t-SNE (**C**). The original data is shown in the first column, the embeddings are shown in the third one. The performance measured by ARI for different values of ε is plotted in the second and fourth column. In both cases, the performance decreases due to a bridge of relatively high density connecting the two classes in the embedding, while a relatively high performance could be achieved with the right choice of ε for the raw data (maximum ARI > 0.75). So there are some cases where the embeddings are harder to cluster than the original data.

The second column of figure 25 **C** shows the change of $Topol_{knnDE}$. One notices that while the most difficult data sets (upper left corner) have a higher topological separability on the embeddings, $Topol_{knnDE}$ decreases for many well separated data sets. The data sets wit the highest decrease in $Topol_{knnDE}$ are shown in figure 51. Although the classes are well separated in both cases (UMAP embedding for **A**, t-SNE for **B**), the separability is lower than for the original data according to $Topol_{knnDE}$. On the original data, knnDE propably has two clear maxima, the means of the two components. On the embeddings on the other hand, the estimated density does not look like the mixture of two Gaussians anymore but probably has multiple local maxima as well as regions with low density within components. It's therefore much harder to describe these data as "manifold + noise", as the manifold doesn't consist of a single point anymore (like for the Gaussians in the raw data). This phenomenon can lead to a decrease in $Topol_{knnDE}$ (see values in the fourth column in figure 51) for well separable data and might also explain the lower correlation of $Topol_{knnDE}$ and ARI on the embeddings as well as the relatively low correlation of $Topol_{knnDE}$ with the parameters of the data sets (figure 24, **B** and **D**).

The plots in the third column of **C** show that DCSI decreases mainly for data sets around the diagonal, i.e. of medium difficulty. This decrease leads to the sharp drop in DCSI around the diagonal observed on the embeddings in figure 24 **D**. The data sets with the highest decrease in DCSI are shown in figure 50 **B** and **D**. For UMAP (**B**), six points from the blue class were merged to the red one. Although the embedding is very successful and shows some of the advantages of combining manifold learning with DBSCAN (see fourth column, the maximum ARI of the embedding is close to one and the range of good ε-values is much wider), DCSI decreases to almost zero, as the minimum distance between core points goes to zero. This example shows that DCSI might lack robustness when there is one core point that is

[20] Note the different scales on the y-axes in order to also be able to see changes for those measures with a small range of observed values. Zero is always marked with a dashed grey line.

close to another class. For t-SNE, DCSI decreases because a bridge was built between the two classes (**D**).

Figure 25 **C** furthermore shows that the DCSI slightly decreases for relatively easy data sets on t-SNE embeddings, e.g. for cases like the data set shown in figure 51 **B** ($DCSI_{raw} = 0.93$, $DCSI_{t-SNE} = 0.79$, $DCSI_{UMAP} = 0.95$): the UMAP embedding places the two classes far away from each other whereas for t-SNE, there is much less space between the components which leads to a lower DCSI. This corresponds to the finding of Kobak and Linderman (2021), that UMAP yields more compact clusters with more white space in between than t-SNE. Note that this change of separation and the fact that UMAP produces a better embedding than t-SNE in this case is not reflected in maximum ARI, which is 1 for the raw data as well as both embeddings. The different separability is only indicated by the range of ε-values that lead to a perfect performance (not shown here). This range is wider for UMAP than for t-SNE. So the maximum ARI cannot capture all aspects of separability, as it is not sensitive in cases where at least one ε exists that leads to a perfect performance.

Summary: Most of the separability measures - despite their different observed ranges - have a high correlation with ARI both on the original data and the UMAP and t-SNE embeddings. UMAP leads to slightly better results than t-SNE (according to ARI), which is also indicated by most separability measures, however some of them have lower values on the embeddings than on the original data. This can at some point be explained by the structure of the embeddings (no clear two maxima of the density lead to a decrease of $Topol_{knnDE}$, (groups of) outliers are merged to one class, which yields a lower DCSI if a core point merged to the wrong class). Furthermore, in some cases bridges are built between components. It seems that - similar to findings in literature - UMAP yields better separated clusters, which cannot always be captured by maximum ARI if the performance on the original data is already high. In such cases, separability measures can indicate that the separability of the UMAP embedding is higher than for t-SNE and/or the original data.

5.2.2 Experiment 2

The results of experiment 2 (two two-dimensional Gaussians, varying distance, fixed covariance in one component, varying covariance in the other, 1525 data sets) are summarized in figures 26 and 27. The boxplots in figure 26 **A** show that - as expected - ARI takes higher values on the raw data than ARI_2, while on the embeddings, their correlation is 1 (**B**). The boxes of ARI_2 show that the embeddings clearly improve the performance of DBSCAN, with UMAP being slightly better than t-SNE. However, the difference between UMAP and t-SNE is smaller than in experiment 1. $Topol_{dtm}$ and $Topol_{knnDE}$ have lower values than for experiment 1, which can be explained by the different densities of the two components (recall section 3.5.2, figures 13 and 14). Again, most separability measures indicate that UMAP leads to a slightly better performance than t-SNE. Similar to experiment 1, the observed ranges differ from $[0, 1]$ for some measures. DCSI has relatively low values as for the density separated clusters in this experiment, the distance between core points of different classes is often very small (recall that ε_i is chosen separately for each class), see the data in figure 52 for example. It is furthermore relatively likely that for slightly overlapping clusters, a core point is merged to the wrong component, which could explain the low values of DCSI on the embeddings.

The Spearman correlations in **B** show that ARI is much less correlated with the other measures than ARI_2, as ARI can have very high values for difficult data sets (i.e. high covariance in the second component) if most points of the second component are labelled as noise points. The correlations of the separability measures with ARI_2 are similar to experiment 1 but slightly smaller. For the raw data, this could be due to the fact that some measures aren't designed to cope with density separated clusters (like DCSI, as mentioned above). For the embeddings, this could be explained by points being merged to the wrong cluster (so DCSI and Dunn* have low values for well separable data). Furthermore, UMAP and t-SNE seem to be likely to form bridges between overlapping clusters (see figure 53 for example). This can lead to a drastic decrease in ARI_2, while separability measures that are based on cluster centers (some CVIs) or take the distribution of of the classes into account (DSI) are not affected by the bridge. The findings in **C** are similar to experiment 1.

D shows that ARI takes values close to 1 for the most difficult data sets, as there exist ε-values for which one cluster is correctly detected and the other class is almost completely classified as noise points.

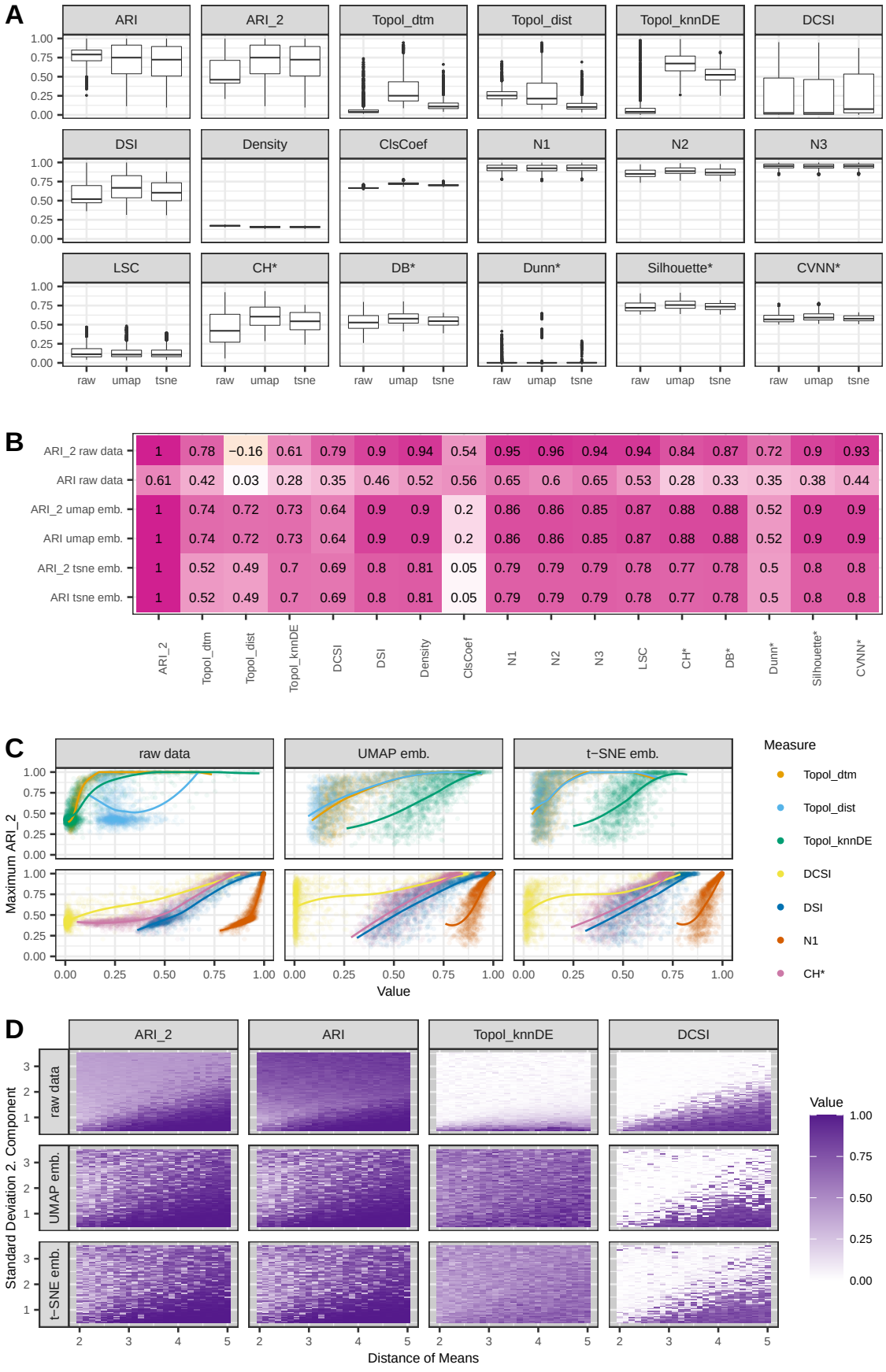

Figure 26: Results Experiment 2: performance and separability measures on raw data and embeddings

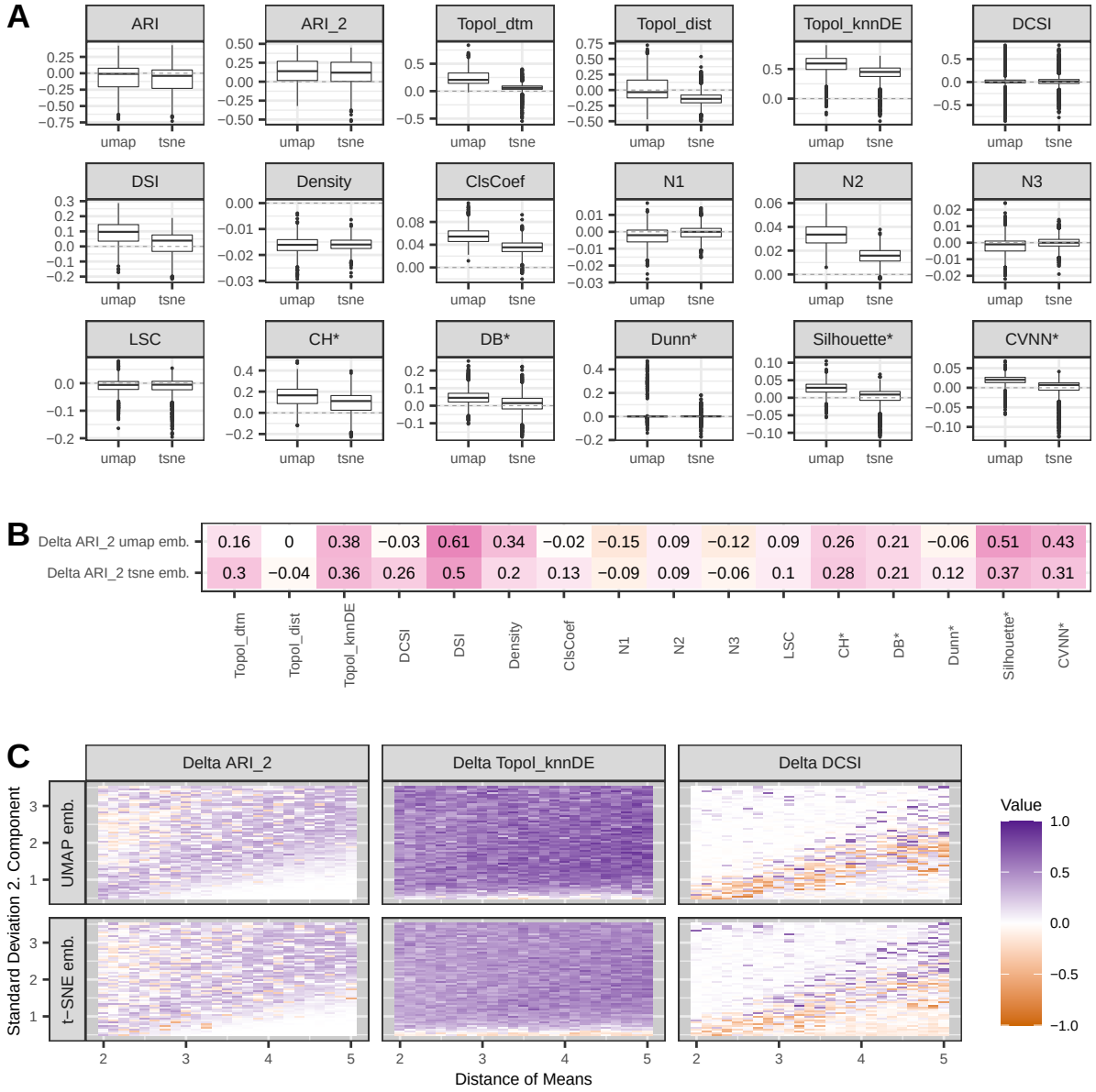

Figure 27: Results Experiment 2: change in performance and separability measures on embeddings

ARI_2 on the other hand takes values of 0.5 in such cases. The relation between the parameters of the data set and ARI_2 on the embeddings seems less strong than for experiment 1, which could be explained by bridges that are sometimes built between the classes and sometimes not, depending on very few points (compare the data in figures 52 and 53 for example). The low values of $Topol_{knnDE}$ on the raw data were explained by the different densities in the components above. The third column of **D** supports this explanation, as the only high values are observed for data sets with similar densities in both components (smallest standard deviation of second component). The correlation of $Topol_{knnDE}$with the parameters of the data sets is again very low on the embeddings, probably for similar reasons as in experiment 1. The findings for DCSI are also similar to experiment 1: the transition between difficult and easy data sets is less smooth for the embeddings than for the original data, as DCSI can be highly influenced by core points being merged to the wrong component, which leads to a low DCSI for an embedding that would otherwise be very successful.

The change in performance and separability for experiment 2 is shown in figure 27. The boxplots in

A show that the performance measured by ARI_2 increases for 75% of the data sets, with again UMAP performing slightly better than t-SNE. The improved performance is also indicated by some separability measures, however the correlations in **B** are smaller than for experiment 1. The reasons could be similar to the explanation of the lower correlations in figure 26 **B**.

The plots in **C** show that ARI_2 decreases for some data sets (especially difficult ones), as some data sets have an ARI_2 close to 0.5 on the raw data (as one cluster is correctly detected). ARI_2 drops for the embeddings if a bridge is built, like in figure 53 ($ARI_{2,raw} = 0.50$, $ARI_{2,umap} = 0.18$, $ARI_{2,tsne} = 0.25$). However there are also a lot of data sets where DBSCAN performs better on the embeddings. The data set with the highest increase in ARI_2 is shown in figure 52. The maximum ARI_2 for the UMAP embedding is 0.94 compared to 0.45 for the original data. This example shows that density based clustering can benefit from manifold learning methods, as these methods are able to separate density separated clusters that cannot be detected by DBSCAN. This is due to the local metric UMAP uses for each X_i to define similarity and the way t-SNE defines the probabilities $p_{j|i}$ (which could also be interpreted as some kind of local metric of similarity for each X_i). $Topol_{knnDE}$ is much higher on most of the embeddings, as it cannot cope with clusters of different densities and just like DBSCAN benefits from the separation of density separated clusters. DCSI again decreases for data sets of medium difficulty and slightly decreases for easy data sets on t-SNE embeddings, probably for the same reasons as in experiment 1.

Summary: Experiment 2 shows that the use of ARI might not be meaningful in some cases, which is also indicated by the low correlation between ARI and the separability measures. So here, ARI_2 is used to evaluate the performance of DBSCAN. Most findings are similar to experiment 1, however the topological separability has low values for clusters of different densities and the correlation of most separability measures with ARI_2 is less high than for experiment 1, which could be due to different reasons: Sometimes, ARI_2 drops if a bridge is built, which isn't indicated by some measures that aren't influenced by the bridge. In other cases, there are (core) points merged to the wrong cluster, which leads to small values of DCSI although ARI_2 is relatively high. DBSCAN can benefit from the ability of UMAP and t-SNE to separate density-separated clusters and this improvement of separability is also indicated by several separability measures.

5.2.3 Experiment 3

The results of experiment 3 (two two-dimensional Gaussians with a bridge of varying density in between and varying distance of means, fixed covariance, 775 data sets) are summarized in figures 28 and 29. The boxplots in **A** show that the performance of DBSCAN improves on the embeddings, with again UMAP being better than t-SNE. The range of most separability measures is much smaller than for the other experiments, as most of these measures aren't affected by the bridge between the classes. N1, N2 and N3 for example always have values close to one, as there are only very few points on the border between the classes. The proportion of these points remains relatively unchanged, even if the density of the bridge increases, as points that don't lie on the border are also added. DCSI has fewer high values than for the first two experiments (compare the 75%-quantiles for example), as there are a lot of data sets with close core points of different classes.

The relatively high values of $Topol_{knnDE}$ seem surprising, as from a topological perspective, two Gaussians connected by a bridge cannot be considered two unconnected components anymore. The reason might be that - even if the density of the bridge is high - there are still two points with a relatively high density, the means of the Gaussians, so in most cases, the lifetime of the second component is relatively high. In extreme cases, the density of the bridge is even higher than in the components (see figure 54 **C** for example), so the two highest values of the density don't even correspond to the two components. One might argue that in this case with bridges of high density, it's not really appropriate to use the topolgical separability (and compare it to ARI), as there are no two clusters from a topological point of view. DSI and several CVIs have lower values for the embeddings than on the original data. Especially those methods that use cluster centers have high values for all raw data sets, as the centers are always well separated and the classes don't overlap. For the embeddings, the centers might be less clearly separated, as UMAP and t-SNE sometimes produce two components of a curved shape (see figure 54 **C** for example), which leads to lower values for some separability measures.

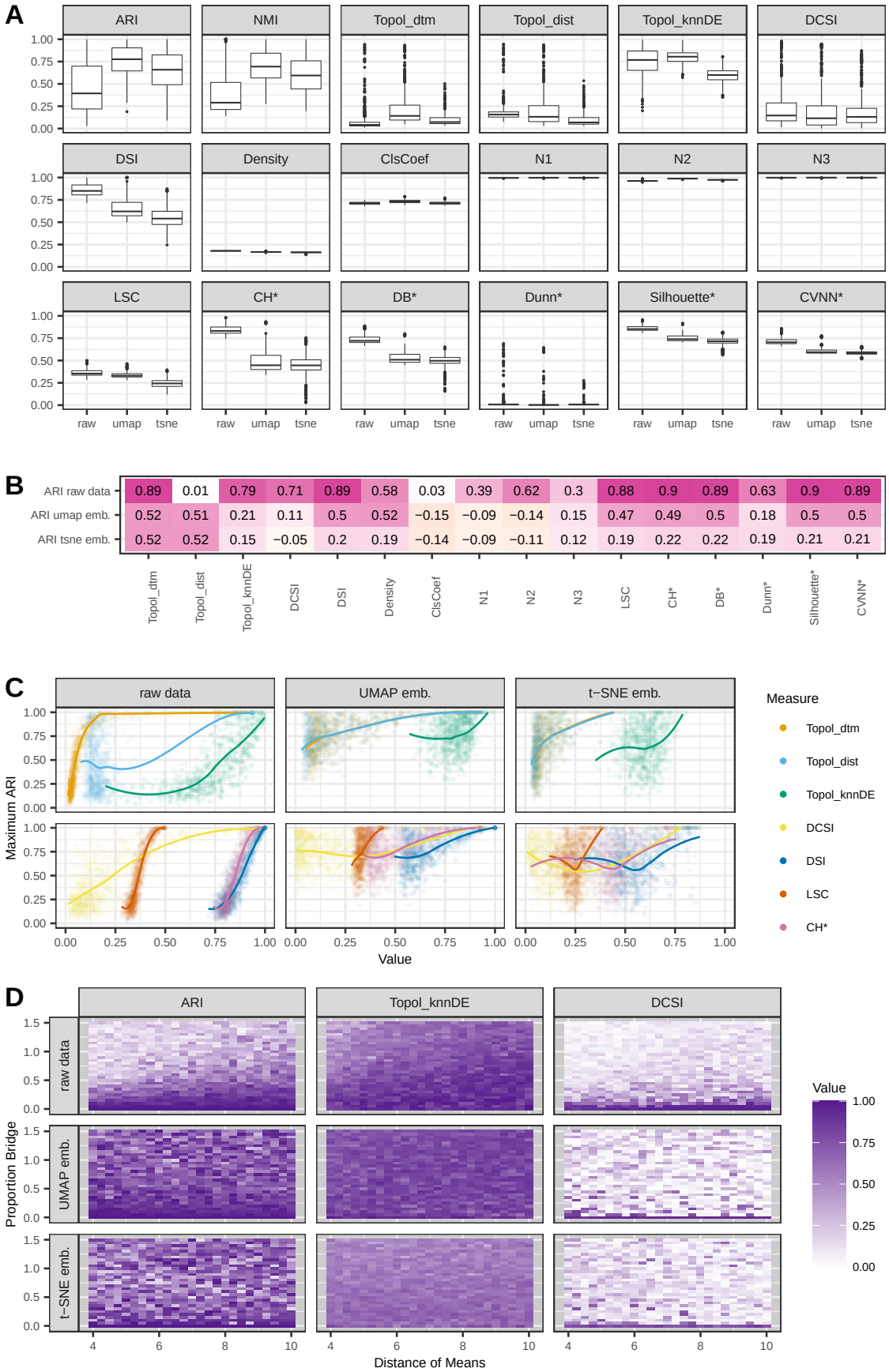

Figure 28: Results Experiment 3: performance and separability measures on raw data and embeddings

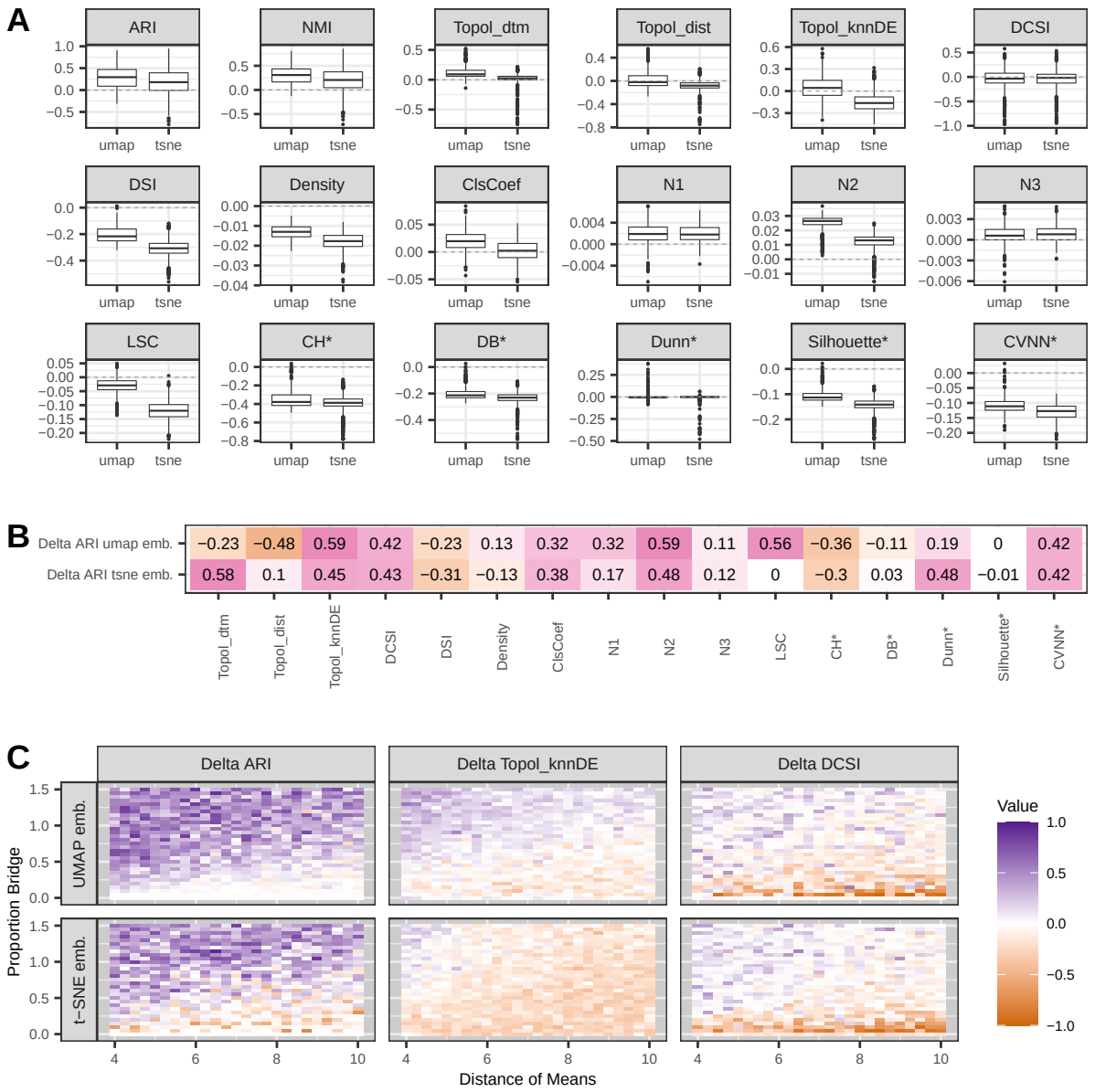

Figure 29: Results Experiment 3: change in performance and separability measures on embeddings

The correlations in **B** are relatively low for some complexity measures (both on the raw data and the embeddings). As these measures aren't able to differentiate between the data sets in this experiment (e.g. the values of N1 are between 0.99 and 1 for all raw data sets and embeddings), this finding should probably not be overinterpreted. The low correlations for the embeddings, especially t-SNE, might also be explained by the curved shape (which leads to low values for some CVIs despite high values of ARI) and by the bridges that sometimes remain in the embeddings (see figure 54 **D** for example), leading to a small ARI while several measures aren't affected by the bridge. The low correlations of $Topol_{knnDE}$ and DCSI could be due to similar reasons like in the other experiments: DCSI is highly affected by single core points being merged to the wrong cluster, while $Topol_{knnDE}$ might decrease because of the specific structure of the embeddings without two clear maxima of the density or closer components (for t-SNE). The low correlation can also be seen in **C**.

D shows that the distance (x-axis) affects the values of ARI, $Topol_{knnDE}$ and DCSI much less than the density of the bridge (here given by the proportion of points that are added to the bridge, i.e. if the

proportion is 0.5, $0.5n$ points are added to the bridge). In most cases, the manifold learning methods are able to separate the bridge, leading to a higher performance indicated by ARI. As mentioned above, DCSI has relatively low values for all data sets, as most of them contain close core points.

Figure 29 **A** shows some of the findings discussed above in more detail. ARI performs slightly better than t-SNE, which is indicated by some separability measures. Most CVIs take lower values on the embeddings than on the original data, as mentioned above.

B shows that the change of some measures (especially CVIs and DSI) is not or even negatively correlated with the change in ARI. The reasons for the low correlation of some measures with ARI on the embeddings mentioned above probably also apply to the change of these measures: the separability according to most CVIs and DSI was relatively high on the original data, as the clusters didn't overlap and were well-separated. The spherical shape of the clusters isn't preserved in the embeddings, leading to a decrease that isn't correlated with changes in ARI for several CVIs. DSI might decrease as the distributions of the two components are more similar in the embeddings. Note that from the perspective of classification, on might also argue that the separability doesn't decrease as the density of the bridge increases.

D shows that ARI increases on the embeddings most data sets with a bridge of high density. The data sets with the highest improvement in performance are shown in figure 54 **A** and **C**. ARI decreases for some relatively easy data sets for t-SNE, as there are cases where DBSCAN can separate the raw data but t-SNE yields a bridge of higher density than in the original data (figure 54 **D**). Both $Topol_{knnDE}$ and DCSI decrease for data sets with bridges of low density and/or a big distance of means. Decreases of DCSI can mostly be explained by core points being merged to the wrong cluster or core points of different classes being placed close to each other (see figure 54, **B**[21]). The decreases in $Topol_{knDE}$ (especially for t-SNE) might be explained by the relatively high values for the original data and less separated clusters especially for t-SNE (see figure 54 **D** for example).

Summary: For experiment 3, the manifold learning methods again lead to an increase in performance of DBSCAN. However, this is not reflected by most separability measures, which might be explained by relatively high separability values for the raw data (as the clusters don't overlap and the centers are well separated) compared to smaller values for the embeddings, induced by components of non-spherical shape or points from the bridge that are added to the wrong cluster. The observed correlations, especially on the embeddings, are therefore lower than for the previous experiments. From a topological perspective, one could argue that at some point, there don't exist two clusters anymore (e.g. figure 54 **A**, **C**), which makes the assignment of true labels consisting of two classes and the evaluation with regard to these labels somewhat critical.

5.2.4 Experiment 4

The results of experiment 4 (two two-dimensional Gaussians, varying distance, fixed covariance, between 0 and 2000 additional irrelevant features sampled uniformly from $[0, 1]$, 324 data sets) are summarized in figures 30 and 31. Note that the topological separability was not calculated on the original data as for d-dimensional data, it involves the evaluation of a d-dimensional function over a grid, which is computationally unfeasible for large d. The boxplots in figure 30 **A** show that ARI is much higher on the embeddings than on the original data. Most separability measures indicate a better separability for UMAP than for t-SNE, which isn't indicated by ARI. All measures (except for Dunn* on t-SNE) have higher values on the embeddings than on the original data.

The correlations shown in **B** are in some cases higher on the embeddings than on the raw data, contrary to the findings of the previous experiments. The lower correlations of some complexity measures on the raw data could be due to the high dimensionality and the amount of irrelevant features in the data: With a lot of irrelevant features, the pairwise distances within the data set increase and the intra- and inter-cluster distances become more similar. It might therefore be more likely that the nearest neighbor of a point belongs to the other class, leading to lower values for N1, N2 and N3 compared to the previous experiments. However this property of the data set doesn't directly affect the performance, as a perfect

[21] In the t-SNE embedding (not shown) of the same data set, some blue points were added to the red cluster, leading to a DCSI of almost zero.

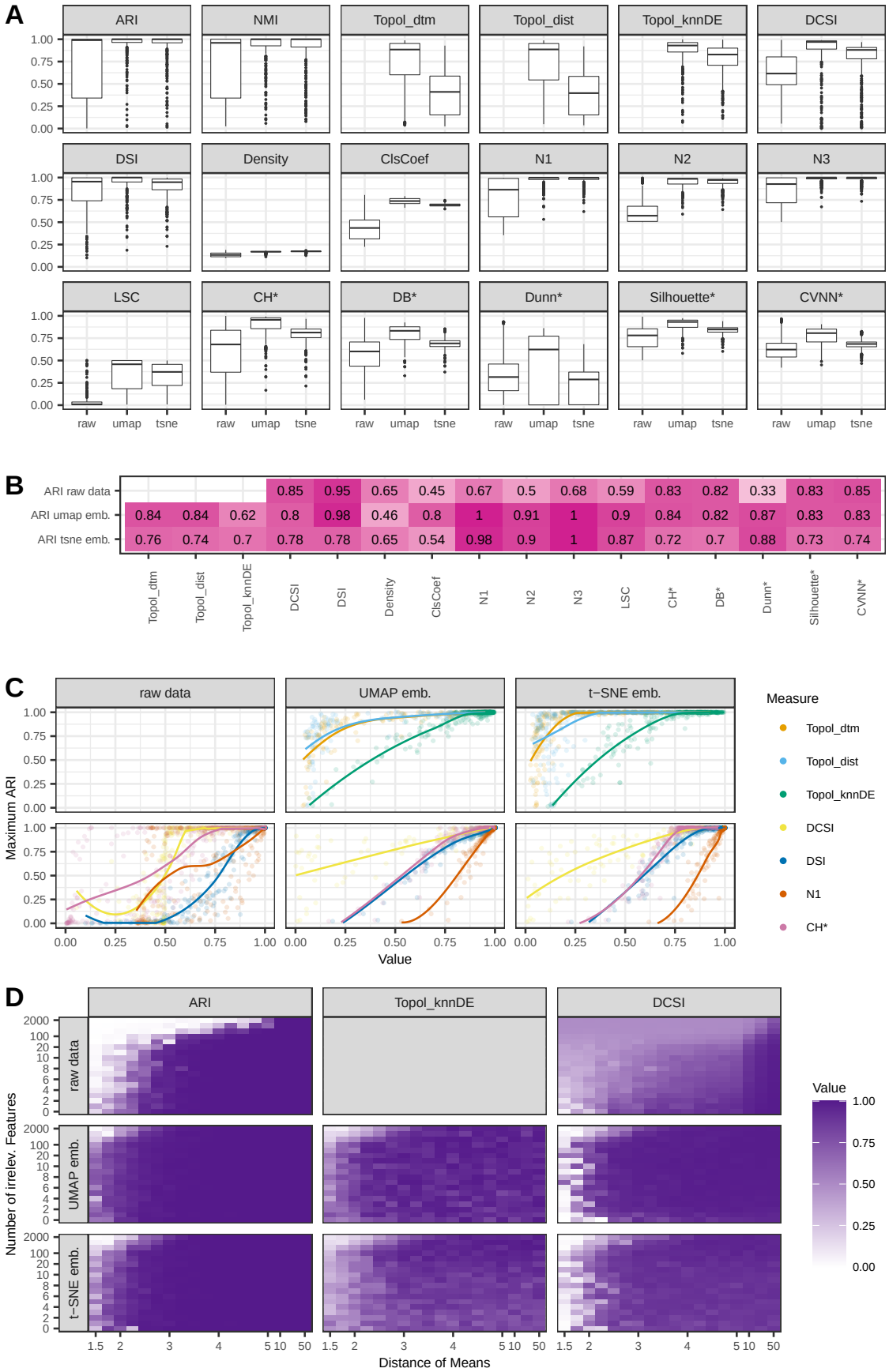

Figure 30: Results Experiment 4: performance and separability measures on raw data and embeddings

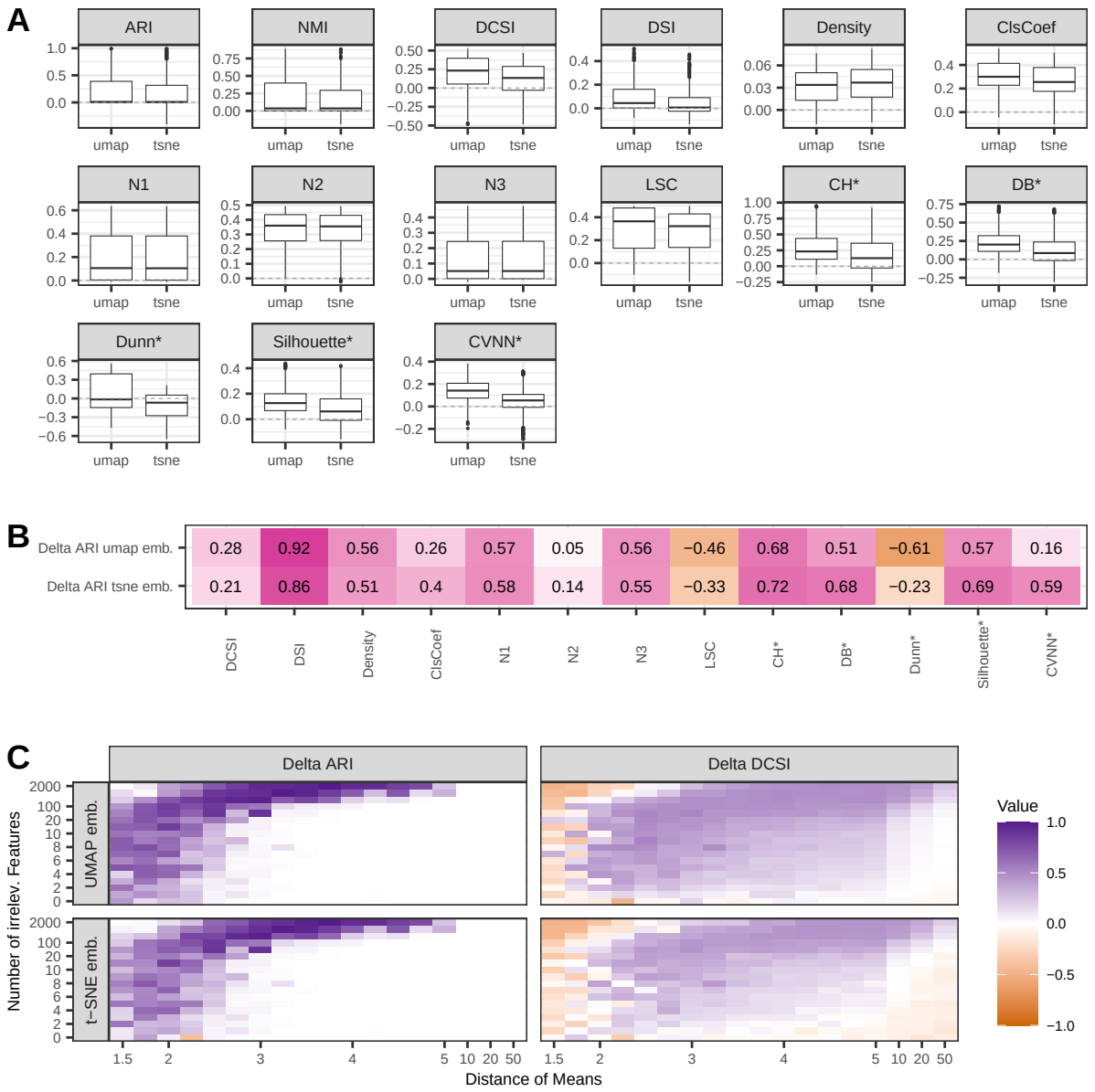

Figure 31: Results Experiment 4: change in performance and separability measures on embeddings

clustering can still be achieved if such two close points from different classes are border points of their clusters.

C shows that there are data sets with an ARI of 1, but N1 is close to 0.5. The relatively low correlation of Dunn* and ARI_2 might be for similar reasons: As the dimension increases, the pairwise distances increase and are more similar (relatively), so the ratio of the minimum inter-cluster and the maximum intra-cluster distance approaches 1 (and Dunn* goes to 0.5), even though the data isn't separable and has low values on the low-dimensional data sets.

The relation of ARI, $Topol_{knnDE}$ (not calculated on raw data) and DCSI with the parameters of the data sets are shown in **D**. Note that both axes aren't linear. The plot shows that a perfect performance is achieved in many cases already on the raw data, but the embeddings considerably improve the performance for some rather difficult data sets, especially in very high dimensions (2000 irrelevant features). The behavior of $Topol_{knnDE}$ and DCSI is relatively similar to ARI on the embeddings.

However, DCSI has values close to 0.5 for the most difficult data sets (upper left corner). The separa-

bility for the lowest distance of means (1.5) increases with the dimensionality, which seems contradictory. The most extreme data set (distance 1.5, 2000 irrelevant features) is shown in figure 55. For the original data, only the first two dimensions (i.e. the two-dimensional Gaussians) are plotted. The embeddings in the second and third column show that UMAP and t-SNE are not able to separate the classes, which leads to a DCSI close to zero (ARI is close to zero for both embeddings and the raw data)[22]. The DCSI on the original data was 0.49, which is much higher than the DCSI for the data set with the same distance of means but no irrelevant features (0.06). These values are shown in table 5. The table furthermore shows the values of separation (smallest distance between core points of different classes) and connectedness (biggest distance in a MST of the core points of the same class). The values close to 0.5 of DCSI can be explained by the high dimensionality: As the dimension increases, the pairwise distances become larger and differ less between the classes. So for the data set in figure 55 (bottom row of the table), both separation and connectedness are between 17 and 18. If separation is approximately equal to connectedness, the DCSI is 0.5. This example shows a weakness of DCSI when it comes to high-dimensional data as it is possible that connectedness and separation are approximately equal although the data is not separable whereas for low-dimensional data, separation tends to zero for classes that are not separable (table 5, second row). A related effect is observed for the data set with a distance of means of 50 and 2000 irrelevant features (third row of the table). DBSCAN is able to correctly detect the classes and the absolute values of separation and connectedness indicate that the data set is well-separated, however DCSI is not close to one (like for the low-dimensional data set, first row of the table) because the ratio of connectedness and separation doesn't tend to zero as the connectedness is relatively high. So for high-dimensional data, DCSI is not able to appropriately distinguish between well-separated data sets and data sets with a low separability.

Figure 31 **A** shows boxplots of the changes. As mentioned above, almost all measures as well as the performance increase for the embeddings. The correlations (**B**) differ, with the change in DSI being highly correlated with the change in performance, whereas the change in Dunn* and LSC has a negative correlation with the change in ARI. The negative correlation of Dunn* might be due to points being merged to the wrong class. LSC had very small values (median = 0.01) and a relatively low correlation with ARI on the original data. Similar to the other neighborhood measures N1, N2 and N3, LSC was probably affected by the high dimensionality, leading to local sets of a low cardinality. While ARI mainly improved on difficult data sets (and couldn't improve on most data sets with big distances, as ARI_{raw} was already 1), LSC had the biggest increases for high-dimensional data sets with big distances (plots not shown here), as UMAP and t-SNE yield well-separated clusters in these cases

The results in **C** were already discussed above: UMAP and t-SNE lead to better performances especially in high dimensions while DCSI decreases for high-dimensional data sets that aren't separable. The decreases of DCSI for lower dimensional data with small distances are mainly induced by core points being merged to the wrong cluster, while the slight decrease for easy data sets on t-SNE embeddings was already observed in the previous experiments.

Summary: The results for experiment 4 differ at some points from the previous results, mostly due to the high dimensionality of the data. UMAP and t-SNE can considerably improve the performance of DBSCAN, especially for high-dimensional data, which is also indicated by the separability measures. Some surprising effects can be explained by the high-dimensionality: The correlation on the raw data of some measures (especially complexity measures) with ARI is lower than in the previous experiments, as the curse of dimensionality leads to less distinct pairwise distances in the data set. DCSI is not able to distinguish between data sets of low and high separability in high dimensions and yields values close to 0.5 instead, as the values of connectedness and separation are closer in high dimensions.

5.2.5 Experiment 5

The results of experiment 5 (two Gaussians, varying distance, fixed covariance, dimension between 2 and 2000, 288 data sets) are summarized in figures 32 and 33. Note that the topological separability was not calculated on the original data, as for d-dimensional data, it involves the evaluation of a d-dimensional

[22] Note that here, PCA works much better as the projection based on the first two principal components basically looks like the first two dimensions of the original data.

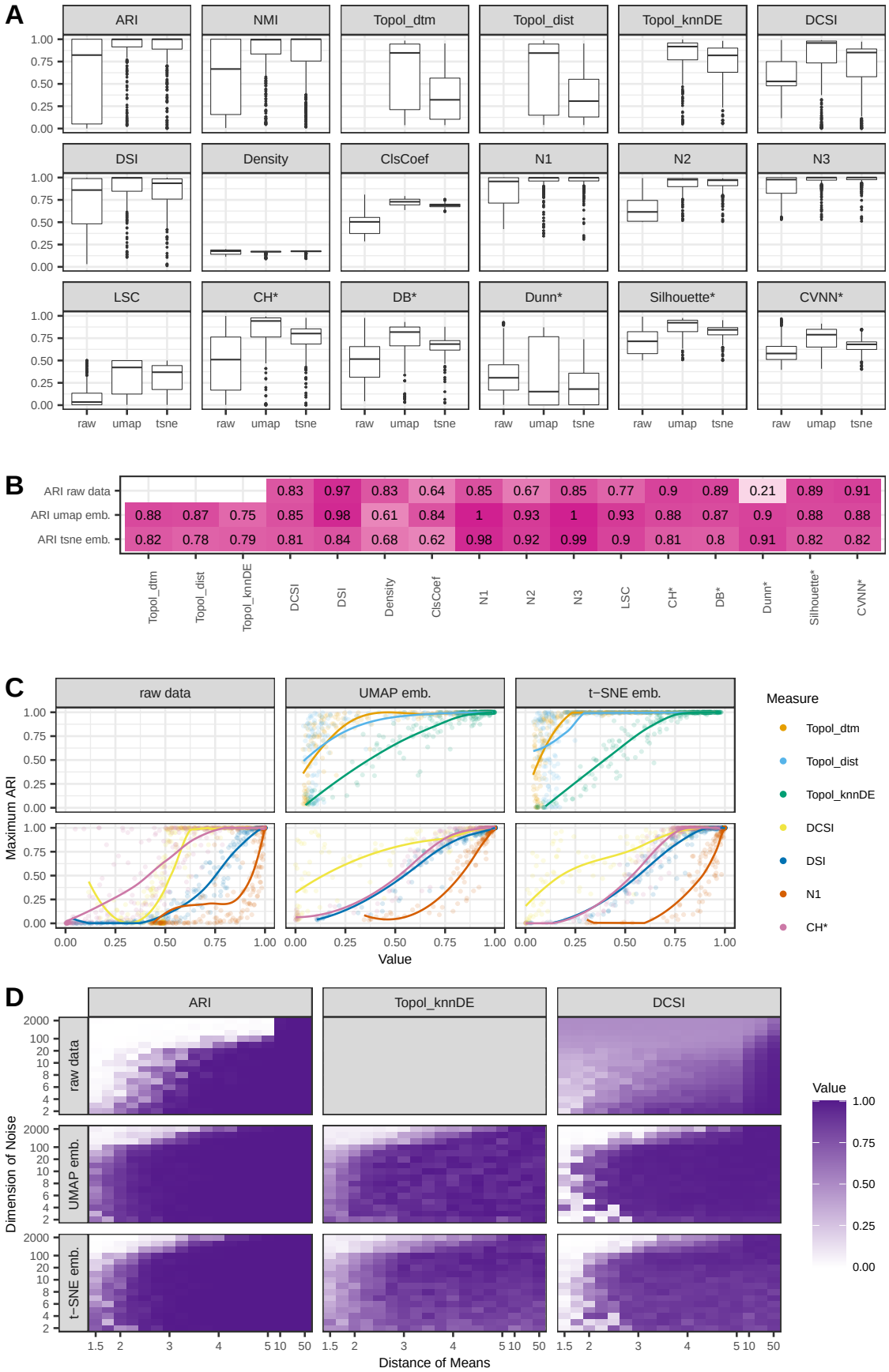

Figure 32: Results Experiment 5: performance and separability measures on raw data and embeddings

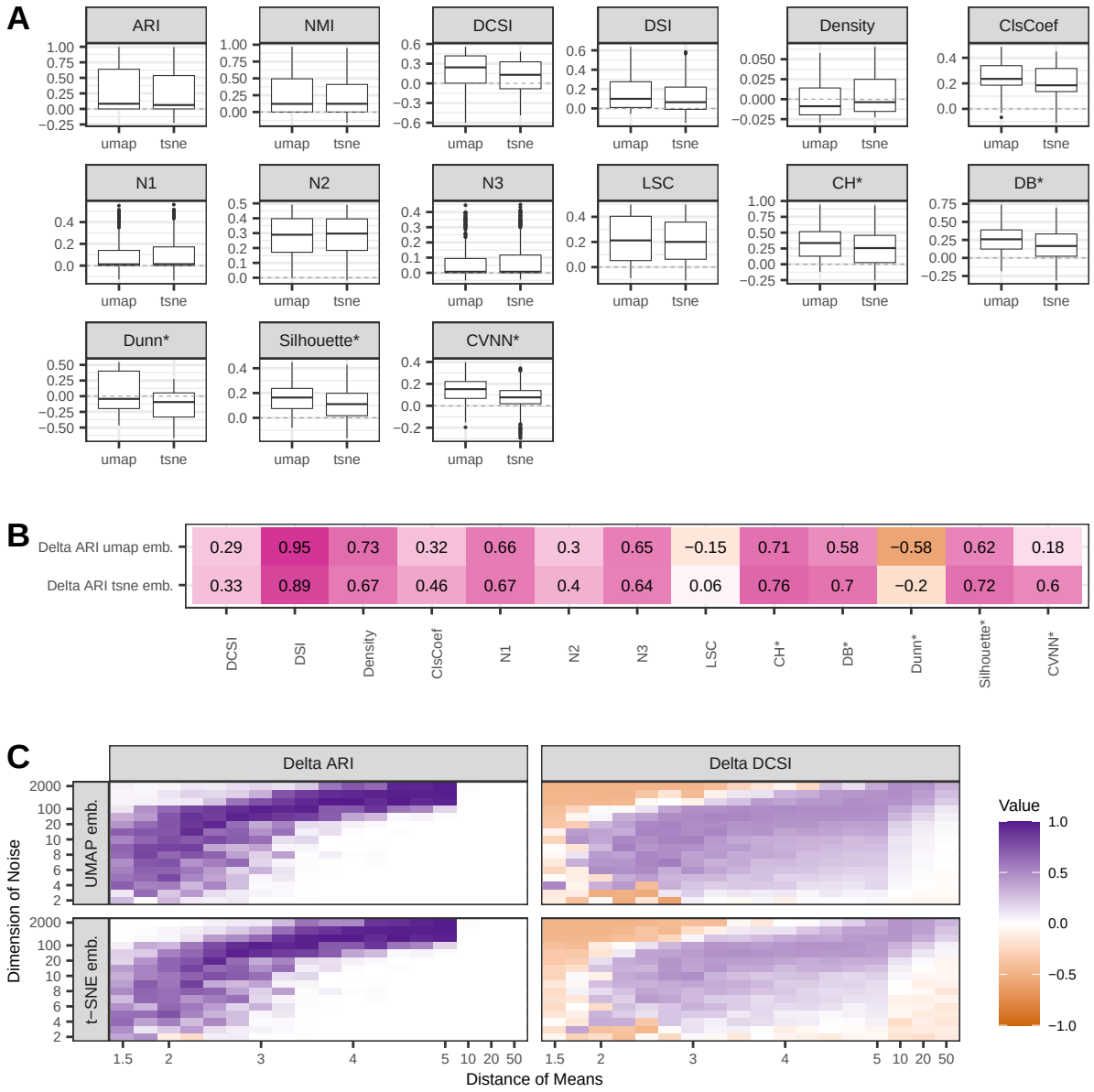

Figure 33: Results Experiment 5: change in performance and separability measures on embeddings

function over a grid, which is computationally unfeasible for large d.

Summary: All results are relatively similar to experiment 4[23]. UMAP and t-SNE lead to a higher performance, which is also indicated by the separability measures (except Dunn*) with UMAP having higher values of separability (figure 32 **A**). The correlation is again lower on the original data than on the embeddings for some complexity measures, probably for similar reasons as in experiment 4. The plot in **D**[24] is also extremely similar to the same plot for experiment 4 with ARI increasing especially for high-dimensional data sets and DCSI yielding values close to 0.5 in high dimensions. The findings of figure 33 are also very similar to experiment 4: DCSI decreases for high-dimensional data sets that aren't separable as the embeddings cannot separate the classes, which leads to a DCSI close to zero.

[23] No concrete embeddings are shown for experiment 5. The embeddings of the most difficult data set and the values of DCSI, separation and connectedness resemble those of experiment 4 (figure 55 and table 4).

[24] Note again the non-linearity of the axes.

5.2.6 Experiment 6

The results of experiment 6 (data sampled from two moons, varying standard deviation, varying shift of the moons, 820 data sets) are summarized in figures 34 and 35. The boxplots in figure 34 **A** show that contrary to the other experiments, the performance increases only slightly on the embeddings (compare the 75%-quantiles). Here, t-SNE leads to slightly better results than UMAP, which is also different to the other experiments. However, several separability measures indicate a higher separability for UMAP than for t-SNE embeddings. Except for DCSI, most measures have higher or equal values on the embeddings compared to the raw data. The maximum values of many separability measures are lower for this experiment than for the others: DSI had values close to one for each of the previous experiments and most CVIs also output a lower separability, while the general performance of DBSCAN is not worse. This can be explained by the inability of some measures (especially CVIs) to cope with clusters of non-spherical shape, which was already shown in table 1.

In accordance with that, some correlations in **B** are slightly lower than for some of the previous experiments. The correlation of CH^* and DB^* with ARI on t-SNE embeddings is relatively small. A possible explanation could be that the values of these two center-based CVIs can be low for clusters of curved shape, while ARI doesn't depend on the shape of the classes and their center but on the question if they are well-separated or not (e.g. because of a bridge). t-SNE seems to be more likely to produce curved embeddings (see figure 56 **B** and **D**), where the performance often depends on bridges existing between the two components or not (whereas bridges between the components are irrelevant for CH^* and DB^*). The correlation of $Topol_{knnDE}$ with ARI on the embeddings (especially UMAP) is relatively low. This will be discussed below.

D shows that ARI correlates with the parameters of the data sets and increases for the majority of difficult data sets (upper left corner). The data sets with the biggest increase in ARI are shown in figure 56 **A** and **B**. However, there are some relatively easy data sets where the performance of DBSCAN decreases on the embeddings, especially for t-SNE. Such data sets are shown in 56 **C** and **D**. For the data set with the biggest decrease in ARI for t-SNE (**D**), the UMAP embedding is also shown. Here, UMAP is able to separate the two components relatively well, although the performance still decreases compared to the original data, as UMAP "tears apart" some parts of the moon. t-SNE on the other hand just unfolds the original data but the components still almost touch at two points, leading to a decrease in ARI. The transition from difficult to easy data sets on the raw data is relatively sharp for ARI. This can also be seen in 56 **B** and **C**, raw data (first column). The maximum performance (second column) can drastically change depending on very few points, jumping from values close to zero (**B**) to relatively high values of ARI for the right choice of ε (**C**). DCSI has relatively low values for some of the easy data sets, especially for t-SNE, which is mainly due to bridges with close core points of different classes.

There is a correlation of $Topol_{knnDE}$ with the parameters on the raw data. On the UMAP embeddings however, there are several difficult data sets (upper left corner) with relatively high values of topological separability, which also explains the low correlation of $Topol_{knnDE}$ and ARI on UMAP embeddings. The data set with the highest increase in $Topol_{knnDE}$ for UMAP embeddings is shown in figure 56 **E**. The topological separability on the raw data is low, as there is only one component. The embeddings on the other hand consist of several components, which leads to a relatively high topological separability (0.77 for UMAP). However, $Topol_{knnDE}$ is not able to measure if the embedding was successful in the sense that DBSCAN can separate the data, as $Topol_{knnDE}$ doesn't take into account if the components correspond to the true classes. So in the case of **E**, the maximum ARI only slightly increases on the embeddings as the two components in the embeddings do not exactly correspond to the true classes (0.02 on the raw data versus 0.22 on the UMAP embedding). These situations seem common for data sets with high standard deviation and high moon shift (upper left corner), which leads to a relatively low correlation of $Topol_{knnDE}$ and ARI on UMAP embeddings.

The changes in performance and separability measures are shown in figure 35. The boxplots in **A** show what was already mentioned above: t-SNE leads to slightly higher increases in performance, while most separability measures rather improve on UMAP embeddings. **B** shows that - despite the low correlation of $Topol_{knnDE}$ and ARI on the embeddings (figure 34) - their change is relatively high correlated. The reason is probably that both $Topol_{knnDE}$ and ARI mainly improve on difficult data sets, for different reasons though (ARI: successful embeddings/components that can be separated with the right choice of ε (figure 56), $Topol_{knnDE}$: embedding consists of two (or more) components,

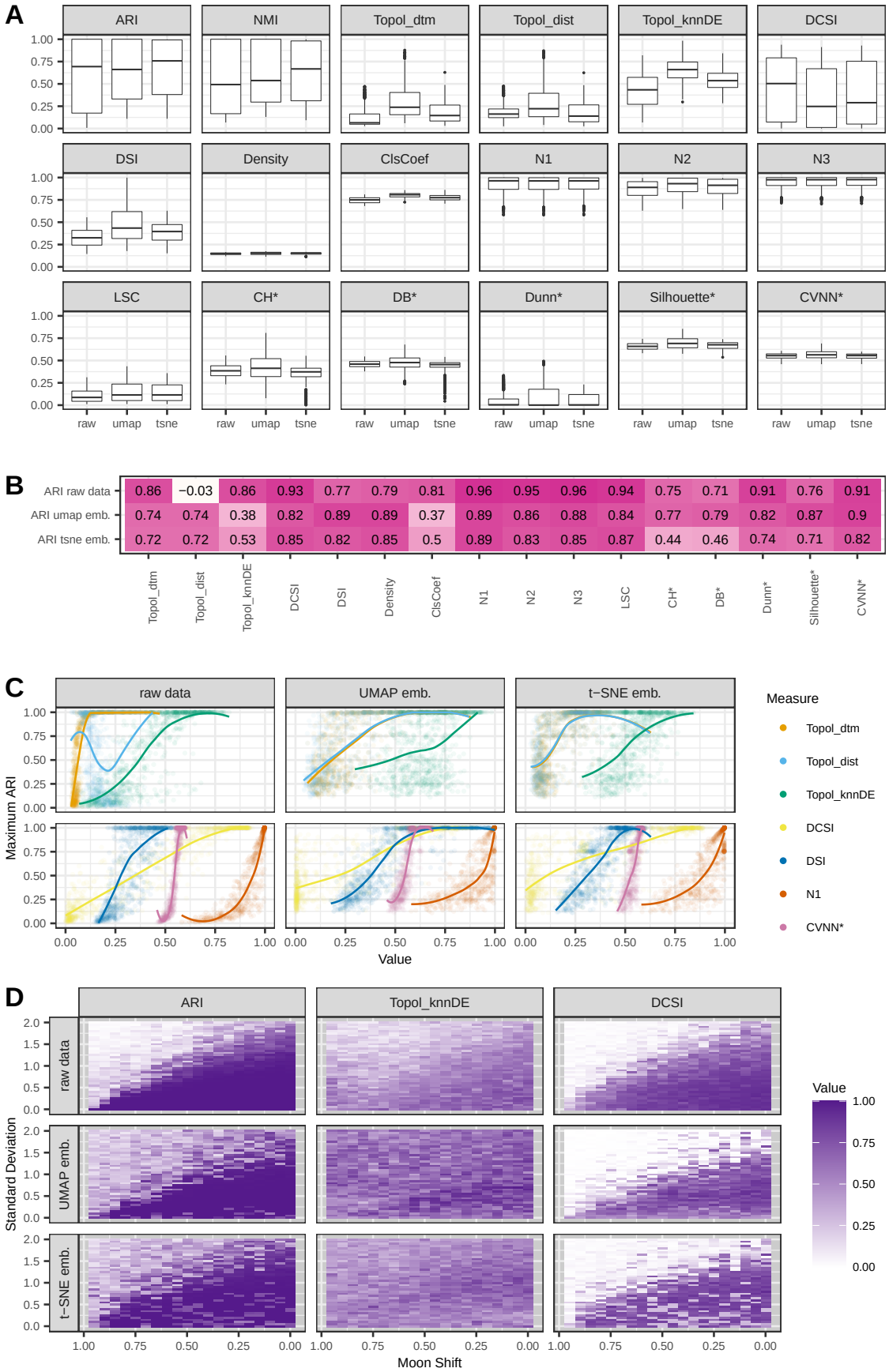

Figure 34: Results Experiment 6: performance and separability measures on raw data and embeddings

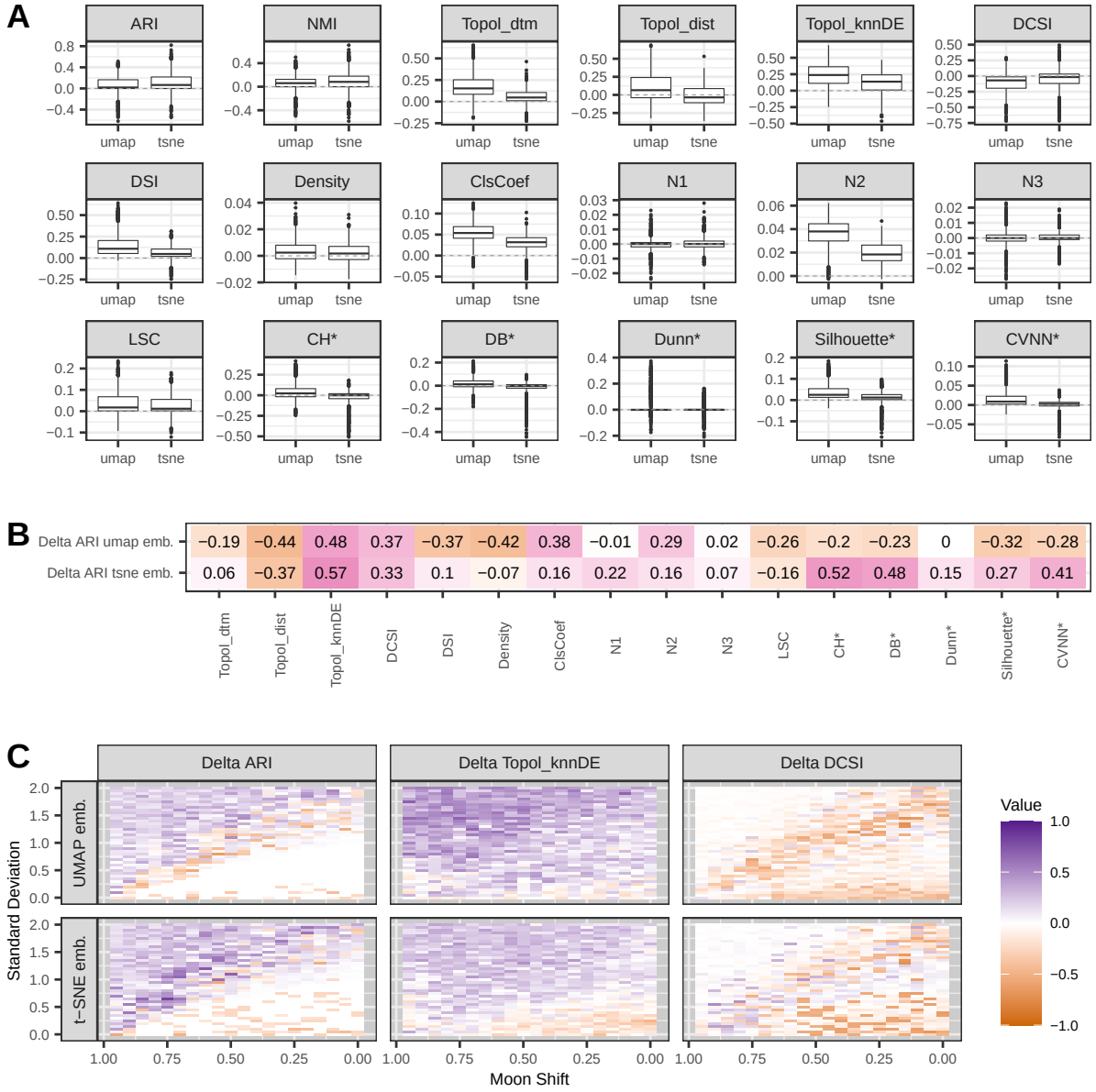

Figure 35: Results Experiment 6: change in performance and separability measures on embeddings

independent of the correspondence of these components to the true classes). The change of several separability measures has a relatively low correlation with the change of ARI, especially for UMAP. The reason is that many of these measures (DSI, CH*, DB* etc.) mainly increase on UMAP embeddings for easy data sets (lower right corner) as UMAP further separates the components, while ARI cannot increase on these easy data sets as a perfect performance is already achieved on the original data, which explains the low correlation of changes for UMAP. Heatmaps of the changes of some separability measures are shown in figure 57. This shows again that some improvements cannot be measured by maximum ARI, as it doesn't differentiate between well separable data sets. However, these improvements are indicated by the separability measures. The findings of **C** were already discussed above (figure 34 **D**).

Summary: Contrary to the previous experiments, clusters of non-spherical shape are considered in this experiment. This leads to lower values of some separability measures are well as slightly lower correlations with ARI. The benefits of manifold learning for clustering are less clear for this experiment.

Bridges built between the classes sometimes lead to lower performances on the embeddings than on the original data. However some separability measures are able to indicate that (especially UMAP) can further increase the separability of well-separated data sets.

5.2.7 Experiment 7

The results of experiment 7 (data sampled from two nested circles, varying standard deviation, varying radius outer circle, 861 data sets) are summarized in figures 36 and 37. The boxplots in figure 36 **A** show that the embeddings don't lead to an improvement in the performance of DBSCAN. The values of $Topol_{knnDE}$ and its correlation with ARI (**B**) are relatively low. There are probably two reasons for this: First, the density on the outer circle is lower than on the inner circle as the same amount of points is sampled from both circles[25]. The higher the radius of the outer circle (i.e. the more separable the components are), the lower the density on the outer circle. As seen before, $Topol_{knnDE}$ has some difficulties when dealing with components of different densities. The second reason could be the shape of the components: The knn density estimator only takes the distance to the k-th nearest neighbor into account (here $k = 100$). For components of non-spherical shape like circles, the distance to the k-th neighbor can become relatively high. For two close circles, the estimated density might even be higher between the two circles and might therefore not consist of two components. Smaller values of k might mitigate this effect, however for small k, the estimated density is less smooth and lacks robustness. This phenomenon also applies to the intertwined spirals in experiment 8, where the effect of k is investigated in more detail. $Topol_{dtm}$ works much better here (e.g. has a correlation of 0.9 with ARI on the raw data), as it takes the distances to all k nearest neighbors into account[26].

CH* and DB* have relatively low values and no correlation with ARI on the raw data, as these two measures compare the cluster centers, which is in this case the same point (the origin) for both classes. Most measures indicate a higher separability for UMAP embeddings than for t-SNE although the maximum ARI is lower for UMAP embeddings. Similar to previous experiments (e.g. experiment 6), this could be due to higher increases on UMAP embeddings for data sets that already have a maximum ARI of 1 on the original data. Dunn* is the only CVI that has a very high correlation with ARI on the raw data, which is plausible as both ARI and Dunn* highly depend on the radius of the outer circle and the covariance of the noise. Note that for this reason and despite its small range of observed values, Dunn* is shown in **C**.

D shows that ARI highly correlates with the parameters of the data sets both on the raw data and the embeddings, while $Topol_{knnDE}$ on the other hand has no correlation with parameters for the above mentioned reasons. On the embeddings, $Topol_{knnDE}$ indicates a relatively high separability of UMAP embeddings for difficult data sets (upper left corner), which explains its low correlation with ARI on UMAP (**B**). These increased values of $Topol_{knnDE}$ on UMAP embeddings are discussed later. $Topol_{dtm}$ yields much better results and is therefore shown additionally. The high correlation of DCSI with ARI (**B**) can also be seen here. **D** shows similar "cuts" in the transition from difficult to easy data sets for DCSI as for ARI.

The changes in performance and separability measures are shown in figure 37. As mentioned above, the performance is relatively unchanged or decreases, whereas some separability measures indicate a higher separability on the embeddings, mostly due to their definition and inability to cope with nested circles on the original data sets. **C** shows that ARI mostly decreases for data sets of medium difficulty (UMAP) and data sets with a low standard deviation and radius (t-SNE). The data set with the biggest decrease in ARI for UMAP embeddings is shown in figure 58 **B**. A perfect performance was achieved on the original data (with the right choice of ε), but UMAP tears the circles apart. The biggest decrease for t-SNE embeddings is for a similar reason (not shown here). However there are also situations where the performance increases on the embeddings (**A**, **D**). Furthermore, even for data sets whose maximum ARI is 1 on the original data, an improvement is possible as the data sets with the highest increase in DCSI (**C**, **E**) show. The range of ε-values that leads to a perfect clustering is much wider on the embeddings than on the raw data (second versus fourth column).

[25] Note that for this reason, ARI_2 was also computed for experiment 7. As both ARI_2 and ARI had a similar observed range and a correlation of > 0.99, ARI_2 isn't shown here.

[26] k is given by $k = m_0 * n$.

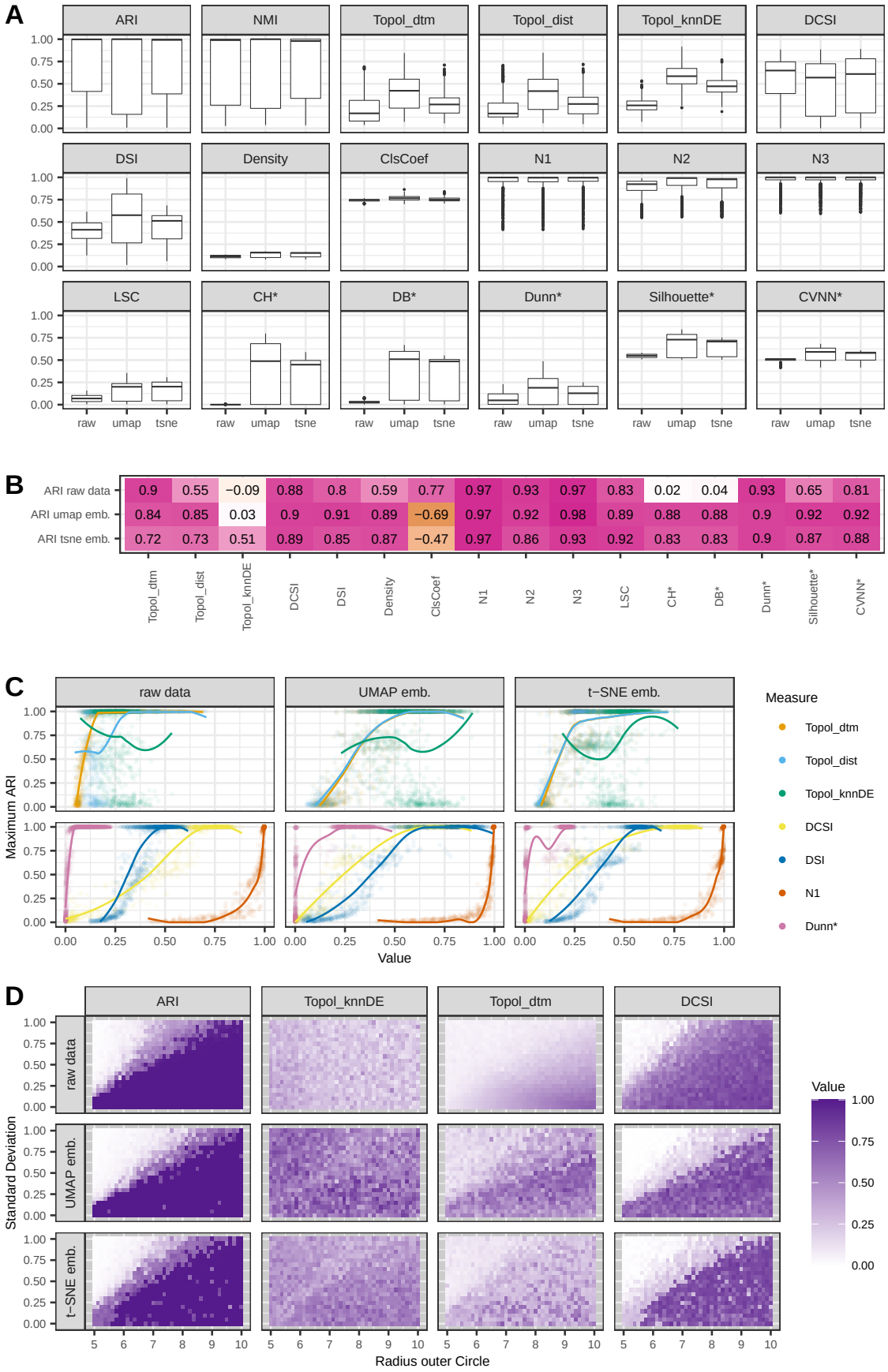

Figure 36: Results Experiment 7: performance and separability measures on raw data and embeddings

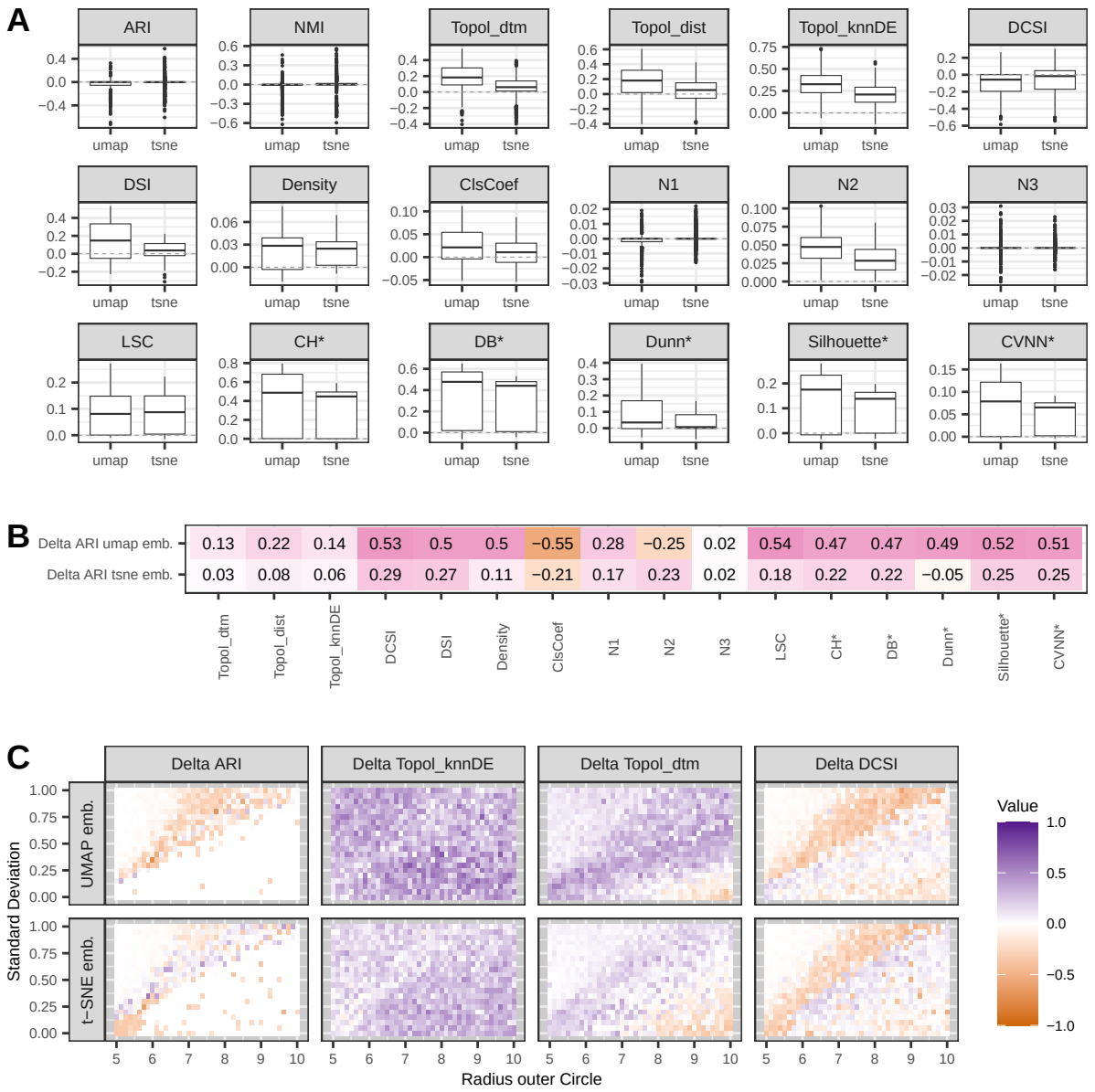

Figure 37: Results Experiment 7: change in performance and separability measures on embeddings

The second column of figure 37 **D** shows that $Topol_{knnDE}$ increases for almost all data sets. As mentioned above, $Topol_{knnDE}$ has relatively high values on UMAP embeddings of difficult data sets. Such a data set with a high increase in $Topol_{knnDE}$ is shown in figure 59. In the original data, one cannot distinguish the two classes and $Topol_{knnDE}$ consequently outputs a relatively low value (0.19). On the embeddings, it's still impossible to separate the two classes. But as especially the UMAP embedding consists of several relatively unconnected parts of high density, $Topol_{knnDE}$ is high on the embedding (0.81). This example shows again a drawback of topological separability on the embeddings, as it isn't able to detect if components consist of points of the same class or not.

For both UMAP and t-SNE, there is a certain group of data sets above the diagonal whose DCSI is smaller on the embeddings than on the original data. These are data sets where the manifold learning methods are not able to completely separate the data and tear the circles apart, leading to a low connectedness (see figure 58 **B** for example).

Summary: Similar to experiment 6, DBSCAN doesn't benefit as much from the manifold learning methods as in the previous experiments, as UMAP and t-SNE tend to tear the circles apart if they are close and/or the data is noisy. Some separability measures using the cluster centers are not able to measure separability for nested circles and therefore have a low correlation with ARI (on the raw data). Additionally, $Topol_{knnDE}$ also has difficulties in dealing with classes of this shape: As the distance to the k-th nearest neighbor can be relatively high, the knn density estimator does not depict two components in many (well separable) cases. Another drawback of $Topol_{knnDE}$ is that it isn't able to measure if components in the embeddings correspond to the true classes and therefore sometimes outputs high values for data sets that aren't separable.

5.2.8 Experiment 8

The results of experiment 8 (data sampled from two intertwined spirals, varying standard deviation, 51 data sets) are summarized in figures 38 and 39. The boxplots in figure 38 show that for this experiment, the performance decreases on the embeddings in many cases. The median ARI is higher for t-SNE, however the 75%-quantile is higher for UMAP. Most separability measures output slightly better values for UMAP than for t-SNE. Similar to experiment 7, some measures have very low values on the raw data and the correlation of CH* and DB* with ARI is again close to zero, as the cluster center is approximately the same for both classes. The correlation of $Topol_{knnDE}$ with ARI is again negative on the raw data (**B**). Similar to the nested circles, the knn density estimator isn't really suited for this type of data, as it only takes the distance to the k-th neighbor into account. This can lead to the effect that the estimated density is higher on points that don't lie on the spirals but between them.

The behavior of $Topol_{knnDE}$ is investigated in more detail in figure 61. The first row depicts three data sets with different standard deviations of the Gaussian noise. $Topol_{knnDE}$ was calculated for the three data sets with $k = 2, 5, 10, 15, 50, 100$. The results are shown in the second row. One can see that the absolute values of $Topol_{knnDE}$ extremely depend on k, as for $k = 2$, $Topol_{knnDE}$ is close to 1 for $sd = 0$, where for $k = 100$, it is approximately 0.25. For most values of k, there is no clear decrease in separability as the standard deviation increases, so $Topol_{knnDE}$ is not really able to differentiate between the three data sets. The estimated two-dimensional density for the data set without noise ($sd = 0$) is shown in the bottom row for different values of k. For $k = 5$, one cannot detect the two spirals and there are just a few points with high values, as the function lacks robustness and smoothness. $k = 15$ yields better results: it's now possible to detect the two spirals. As k increases, the smoothness of the function increases (see $k = 50$). For $k = 100$ (used for the experiments), the density is higher in the space between the two spirals, as in this case, the 100-th neighbor is closer than on the spirals. So $k = 100$ seems not suitable for this setting. However, $k = 15$ and $k = 50$ are also not able to properly distinguish between the three data sets (middle row). Again, $Topol_{dtm}$ works much better than $Topol_{knnDE}$ (correlation of 0.93 with ARI on the raw data), as it takes the distances to all k nearest neighbors into account.

Figure 38 **D** shows that ARI, $Topol_{dtm}$ and DCSI correlate with the standard deviation on both the raw data and the embeddings. As the standard deviation is the only parameter that was varied for this experiment, scatterplots are shown instead of the heatmaps of the previous experiments. Note that while the easiest data sets were located in the lower right corner for the heatmaps, the easiest data sets of experiment 8 are on the left (smaller standard deviation). $Topol_{knnDE}$ shows no correlation with the standard deviation for the reasons mentioned above.

Figure 39 summarizes the changes in performance and separability measures. As mentioned above, most separability measures indicate a higher separability on the embeddings, especially for UMAP, while the performance decreases in most cases. UMAP seems to be more likely to separate the two spirals (see figure 60 **B** for example, biggest increase in DCSI for UMAP), which can be measured by some separability measures like the CVIs and DSI. The correlation of changes are close to zero or negative for almost all measures. Most separability measures probably increase rather for well-separated data sets whose ARI is already 1 on the raw data, whereas ARI remains the same in most cases for UMAP (and decreases on some "easy" data sets) and decreases most for easy data sets for t-SNE. The data set with the biggest decrease in ARI and DCSI for UMAP is shown in figure 60. While a perfect clustering performance was possible on the original data, UMAP unrolls the spirals but "cuts" one class into pieces. The biggest decrease in ARI or DCSI for t-SNE is similar: the spirals are cut into several pieces. There are no data sets where the maximum ARI considerably increases on the embeddings: The data set with

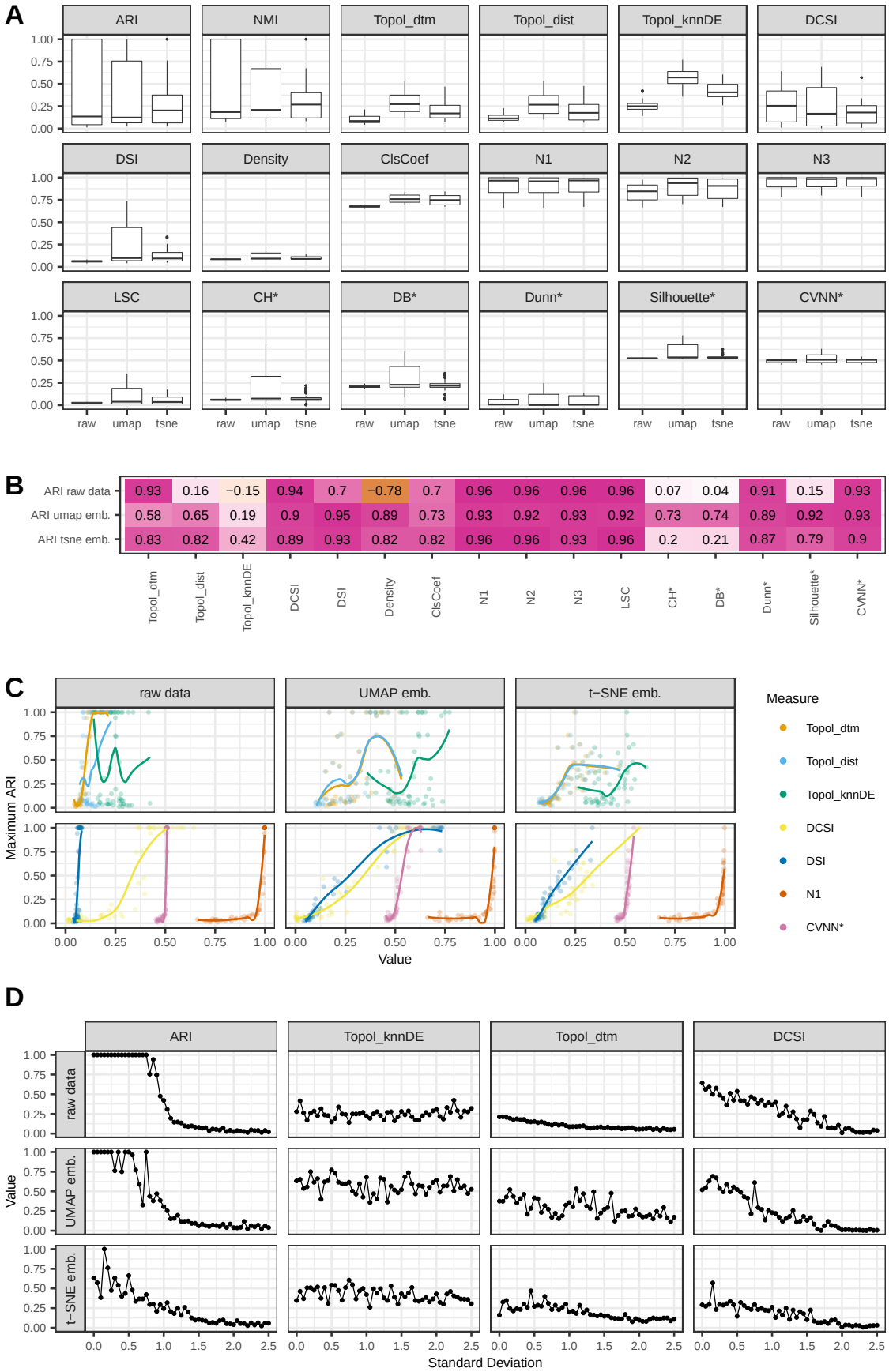

Figure 38: Results Experiment 8: performance and separability measures on raw data and embeddings

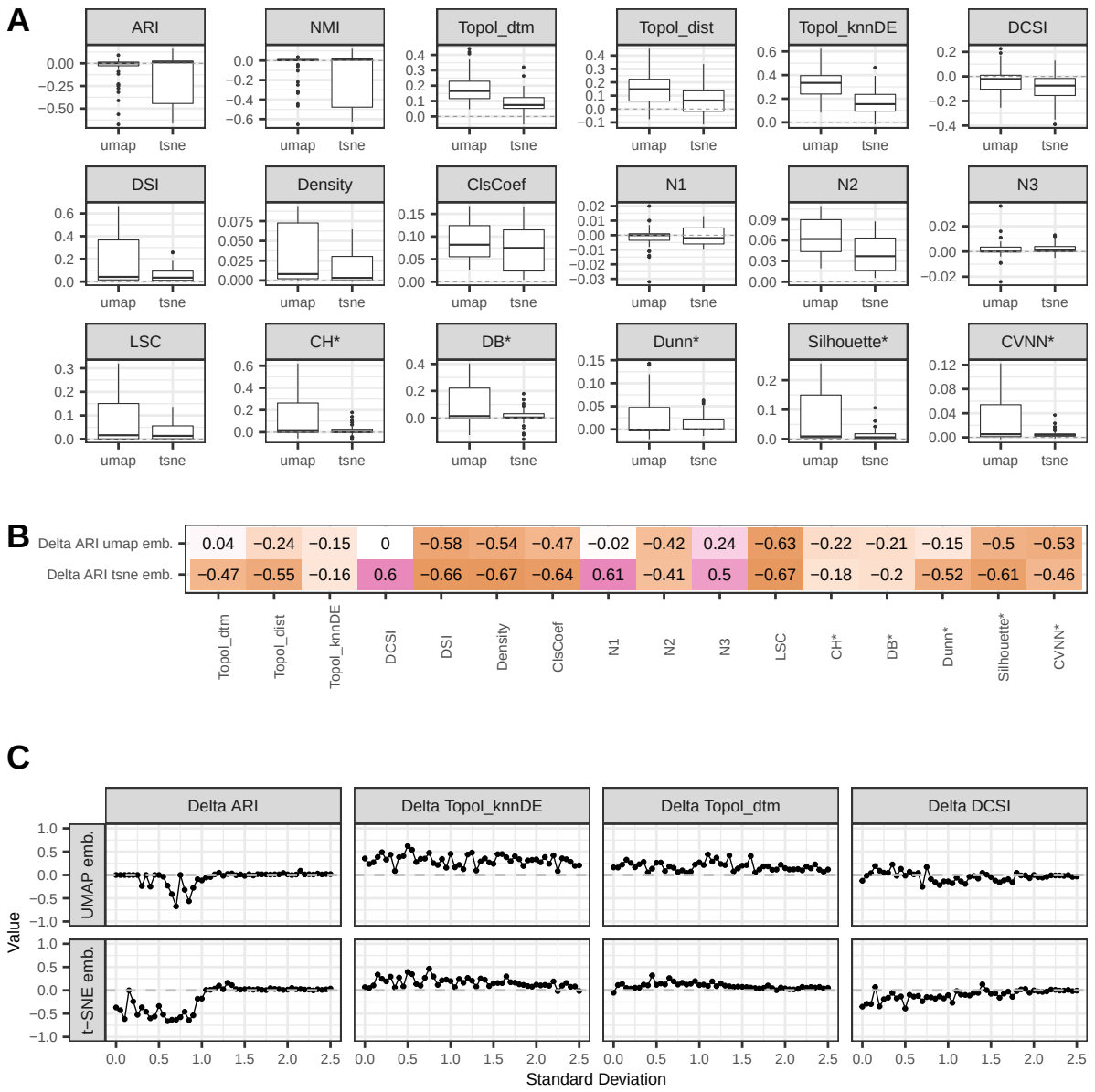

Figure 39: Results Experiment 8: change in performance and separability measures on embeddings

the highest increase in ARI for t-SNE is shown in figure 60 **C**. The spirals are more separated from each other on the embedding than on the raw data, but as they are cut into pieces, the performance increases only slightly.

$Topol_{knnDE}$ increases on the embeddings for almost all data sets. However, similar to experiment 7, $Topol_{knnDE}$ detects unconnected components of high density (which leads to a higher separability for very noisy data) independent of the changes in performance (see slightly negative correlations in **B**).

Summary: The results of experiment 8 are relatively similar to experiment 7 (nested circles). An investigation of the knn density estimator and its dependence on k shows that the absolute values of $Topol_{knnDE}$ highly depend on k and for no k, $Topol_{knnDE}$ can properly distinguish between data sets with different standard deviations. Furthermore, it's at some point questionable if it makes sense to apply the topological separability in settings were it's possible that data from different classes mix, as it is the case for the embeddings. For similar reasons, the topological separability is not applied to real-world

data (see explanation in section 3.5). One might argue that it is only valid to apply these measures to the raw synthetic data and not to the embeddings.

5.2.9 Experiment 9

The results of experiment 9 (data sampled from two nested n-spheres, varying dimension of the spheres, varying standard deviation, varying radius second sphere, 135 data sets) are summarized in figures 40 and 41. Note that the topological separability was not calculated on the original data as for d-dimensional data, it involves the evaluation of a d-dimensional function over a grid, which is computationally unfeasible for large d. Both ARI and ARI_2 were computed. As the maximum ARI is 1 for all raw data sets (this means it's always possible to detect the inner sphere as a cluster and classify the outer sphere as noise points), only ARI_2 is shown. On the embeddings, both the correlation and the observed values are very similar for ARI and ARI_2 (see figure 62).

This is the only experiment where the dimension of the manifold is high. For the other high-dimensional settings (E4, E5), the dimension is only artificially high as the manifold itself is low-dimensional and high-dimensional noise is added. To evaluate if in the case of data with a high intrinsic dimensionality embeddings of a higher dimensionality lead to better results, 2- and 3-D embeddings were computed for all data sets. A comparison of ARI and the separability measures on the 2- and 3-D embeddings is shown in figure 62. The boxplots show that the differences between the embeddings are very small, so only the results of the two-dimensional embeddings are presented in figures 40 and 41.

Figure 40 **A** shows that the maximum ARI_2 is always equal to or greater than 0.5 as for every data set, it is possible to correctly detect the inner sphere and classify the outer sphere as noise points (which leads to an ARI of 1 and an ARI_2 of 0.5). The performance decreases on the embeddings with UMAP having more extreme values (higher 75%- and lower 25%-quantile) than t-SNE. The topological separability was only calculated on the (2-D) embeddings, with higher values for UMAP. Some measures indicate the decrease in performance (e.g. DCSI and DSI), while others like CH^* and DB^* have lower values on the raw data (as they use the cluster center, which is the origin for both classes) than on the embeddings. For similar reasons as in the other high-dimensional experiments 4 and 5, the smallest values of DCSI are close to 0.5, as the pairwise distances become more similar as the dimension increases. Again, most separability measures have higher values for UMAP embeddings, as UMAP is more likely to produce compact, well-separated clusters (see figure 65 for plots of concrete embeddings).

With a few exceptions, the Spearman correlations in **B** are all relatively high. The low correlation of CH^* and DB^* with ARI_2 on the raw data (similar to experiment 7, the two-dimensional analogue of this experiment) can be explained by their use of cluster centers, which is inappropriate for circles and spheres. In order to investigate the low correlation of DSI and $Dunn^*$ with ARI_2 on the raw data, their values depending on the parameters of the data sets are shown in figure 63.

The negative correlation of $Dunn^*$ and ARI_2 might be due to the lower density of the spheres induced by the high dimensionality: Figure 63 shows that $Dunn^*$ slightly increases as the dimension of the n-sphere increases. This can be explained as follows: The maximum (Euclidean) distance between two points on a n-sphere of radius r is $2r$, regardless of n. If a fixed amount of points are sampled from the sphere and the observed maximum distance is computed, the distance decreases as the dimension of the sphere increases, because the density on the sphere decreases. For $Dunn^*$, the maximum within-cluster distance is computed, which is - in theory - $2r_2$ with r_2 being the radius of the outer circle but in practice, this value decreases as the dimension of the sphere increases, which leads to higher values of $Dunn^*$. The minimum between-cluster distance probably suffers from the same effect: in theory, the minimum distance is given by $r_2 - r_1$, but as the high dimensional data gets sparser, the minimum distance increases, again leading to higher values of $Dunn^*$. As ARI_2 decreases as the dimension of the sphere increases, the correlation of ARI_2 and $Dunn^*$ is negative.

DSI is also negatively correlated with ARI_2 on the raw data. Figure 63 shows that DSI tends to increase as the dimensionality increases. This can be explained using figure 64: The plot shows the distributions of the between-class distances (BCD) and intra-class distances (ICD) for 2- and 1000-spheres and $r = 10, sd = 0$ (the radius of the inner sphere is always 4). While the mean of the intra-class distances and between-class distances is approximately the same for both data sets, the variance is much lower for the 1000-dimensional spheres, so the distributions of ICD and BCD are less similar. This explains the high values of DSI on the raw data for high dimensional data sets as well as the negative correlation with

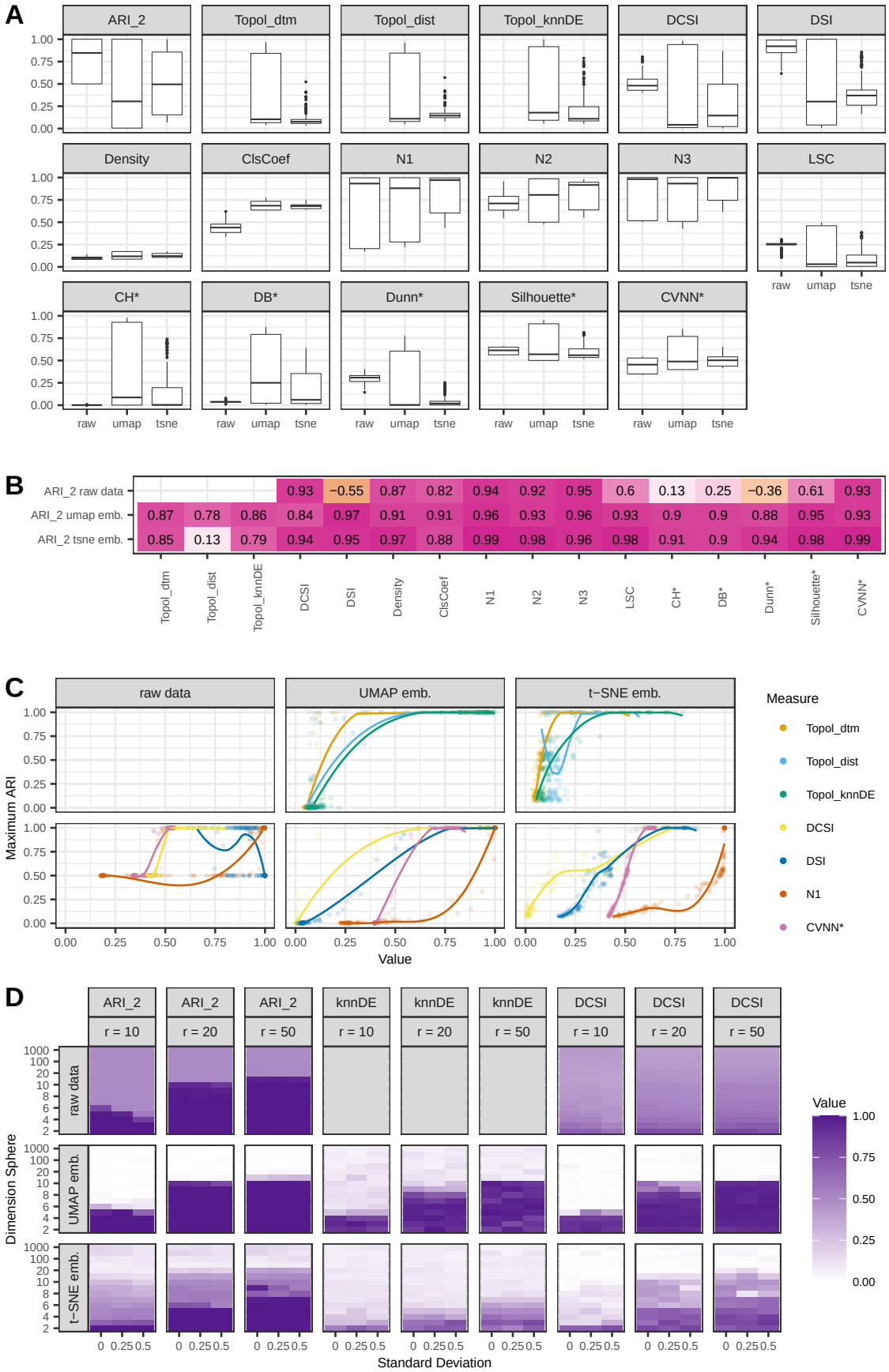

Figure 40: Results Experiment 9: performance and separability measures on raw data and embeddings

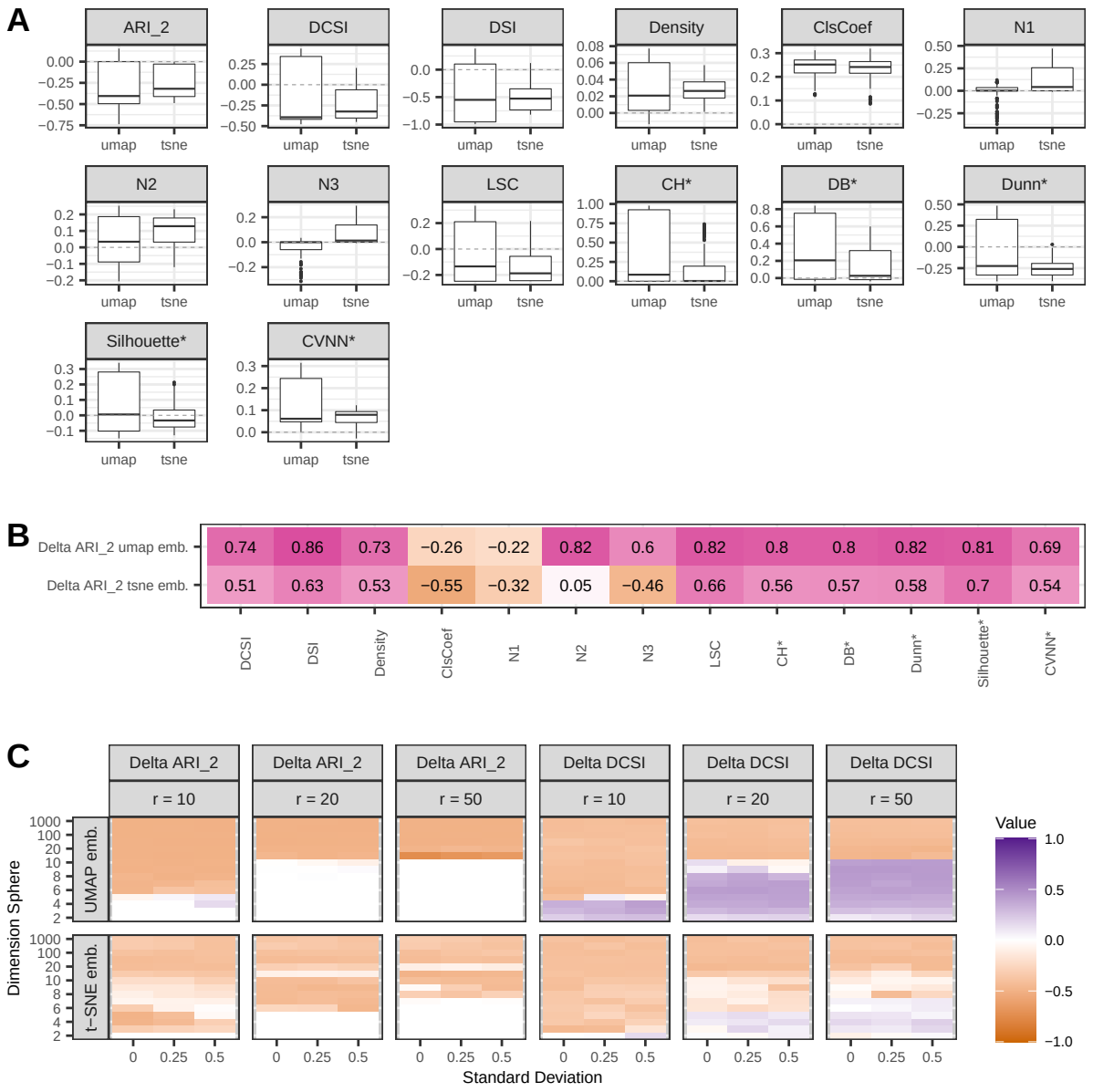

Figure 41: Results Experiment 9: change in performance and separability measures on embeddings

ARI_2.

Figure 40 **D** shows the values of ARI_2, $Topol_{knnDE}$ (not calculated for raw data) and DCSI. Note that for this experiment, 3 parameters were varied (standard deviation, radius outer sphere, dimension of the spheres). There is almost no effect of the standard deviation and a relatively low effect of the radius of the outer circle. The dimensionality however highly influences both the performance and the separability measures. Those raw data sets that weren't separable ($ARI_2 = 0.5$) are also not separable on the embeddings, with a decrease in performance. t-SNE furthermore decreases on some well-separable data sets. While UMAP has an ARI_2 of 0 on data sets that aren't separable, t-SNE has values above zero in the most cases. The changes and concrete embeddings are discussed later in more detail. Both $Topol_{knnDE}$ and DCSI correlate high with the performance.

Figure 41 **A** shows that the performance decreases for almost all embeddings, while (similar to previous experiments) some measures indicate a higher separability for the embeddings and better values for UMAP. The changes of ARI_2 and DCSI are shown in **C**. Concrete embeddings for spheres of different

dimensionalities are shown in figure 65. The radius of the outer circle is 50, the standard deviation is 0.25. The 7-dimensional sphere (**A**) is separated by both UMAP and t-SNE, while UMAP produces more white space between the components. The 8-dimensional sphere (**B**) is separated by UMAP, while t-SNE yields a component consisting of the inner sphere (except for some points) with the outer sphere around it, so t-SNE preserves the nested structure of the components. The t-SNE embedding of the 15-dimensional data (**C**) looks similar, while UMAP isn't able to separate the components anymore and also yields some sort of nested structure, but without the accuracy of t-SNE. This example also explains why the clustering performance on difficult data sets is slightly better on t-SNE than on UMAP embeddings. Note that on the original data, a perfect performance was achieved for a small ε-range. The 2000-dimensional data set (**D**) could not be clustered by DBSCAN and both UMAP and t-SNE don't separate the data. Again, there is more structure in the embedding of t-SNE. Recall that while the goal of UMAP is to learn the underlying manifold and preserve its connected components, t-SNE was mainly designed for visualization of high-dimensional data. A practical reason why t-SNE preserves more of the outer geometry could be that for UMAP, only the k nearest neighbors have a positive weight in the graph construction step whereas for t-SNE, the conditional probability $p_{j|i}$ is positive for each j.

Summary: Contrary to the previous experiments, the intrinsic dimensionality is high for the data sets in experiment 9. There are cases where a clustering of the raw data is possible, but UMAP and t-SNE aren't able to separate the components. 3-D embeddings don't seem to perform better for this experiment. While UMAP tends to yield compact, well-separated clusters, t-SNE preserves more of the global structure, e.g. that the spheres are nested. While most correlations with the separability measures are high, some measures suffer from different aspects of the curse of dimensionality.

5.2.10 Summary & PCA

In order to draw general conclusions on the separability measures, the results of all 6298 data sets from the 9 experiments are taken together and evaluated jointly. As the topological separability wasn't calculated for all raw data sets, it isn't included in this analysis. For experiment 9, only the values of the 2-D embeddings are evaluated, as they are very similar to the 3-D embeddings and no data set should be included twice. For experiments 2 and 9, ARI_2 is taken instead of ARI.

Figure 42 **A** shows the correlations of the 13 separability measures with ARI on the raw data and the embeddings. The correlations on the raw data are lower than most observed values for the separated experiments (e.g. for DSI and N2). This might be due to their different ranges for different experiments: DSI for example correlates highly with ARI for both experiment 1 and 7 but has much smaller values for the nested circles in experiment 7 than for the two-dimensional Gaussians in experiment 1, while ARI takes values across the whole range for both experiments. DCSI has the highest correlation of all separability measures on the raw data. This indicates that DCSI is able to quantify separability in different settings, e.g. independent of the shape of the classes. Similar to some other measures with high correlations (N1, N3), DCSI doesn't favor classes of a certain shape. CH^* and DB^* on the other hand cannot adequately measure separability on classes of arbitrary shape (e.g. nested circles), which is indicated by the lowest correlations with ARI of all measures (on the raw data).

The correlations of almost all measures are higher on the embeddings than on the raw data. UMAP and t-SNE often yield clusters of a certain structure (compact, spherical clusters that aren't nested), so the embeddings are less diverse than the original data, which probably explains the higher correlations. Note that CH^* and DB^* as well as some other measures have a lower correlation on t-SNE embeddings, as t-SNE is more likely to keep nested or non-spherical structures. From now on, only the results on the raw data are analyzed, as they are more diverse and therefore incorporate more aspects of separability (e.g. high dimensionality (both intrinsic and artificial), different shapes, nested clusters) than the embeddings.

Figure 42 **B** shows the correlations of the 13 separability measures on the raw data. There are several blocks of highly correlated measures, as many of these measures quantify similar aspects of separability. The complexity measures N1, N2 and N3 are highly correlated, as they all quantify the presence of nearest neighbors of different classes. There are also very high correlations among the CVIs (except Dunn^*), as they all aim to measure the compactness of classes. Dunn^* also measures compactness, but in a different way (i.e. by the maximum distance within a class). As there are some conceptual similarities between Dunn^* and DCSI (e.g. in measuring separation), they have a high correlation. ClsCoef is relatively low

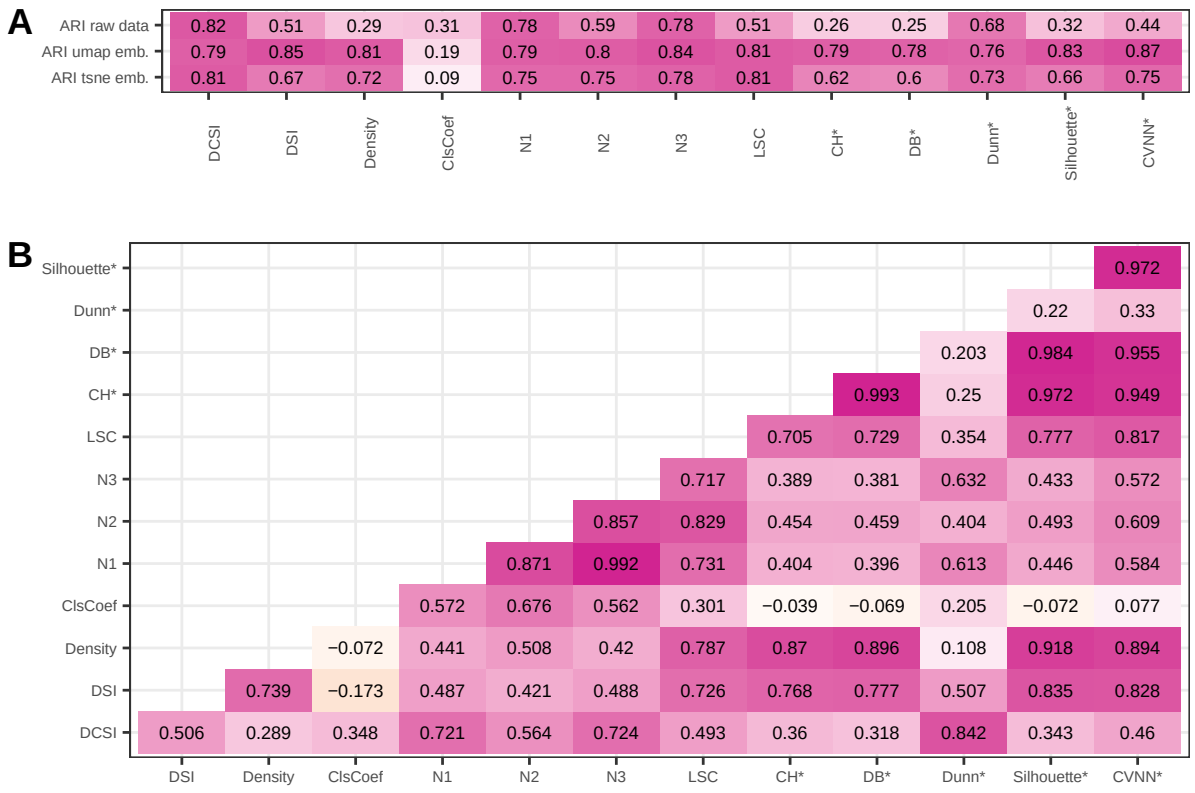

Figure 42: Summary experiments: correlation with ARI and correlation among separability measures on raw data

Table 2: Summary experiments: PCA, loadings first three principle components

	DCSI	DSI	Density	ClsCoef	N1	N2	N3	LSC	CH*	DB*	Dunn*	Sil*	CVNN*
PC1	-0.17	-0.30	-0.32	-0.10	-0.27	-0.26	-0.27	-0.32	-0.33	-0.31	-0.05	-0.35	-0.35
PC2	-0.05	-0.25	-0.04	0.53	0.35	0.39	0.34	-0.03	-0.18	-0.18	-0.39	-0.17	-0.11
PC3	0.62	0.12	-0.28	-0.07	0.17	0.12	0.18	0.07	-0.20	-0.33	0.52	-0.16	0.01

correlated with most measures (and with ARI, see **A**), as it measures an aspect that is not directly related to the difficulty of clustering (measured by ARI) and to the aspects of separability measured by most separability measures.

Each of the 6298 data sets can be seen as a point in the 13-dimensional space given by the 13 separability measures. However, as many separability measures are highly correlated, the intrinsic dimensionality of the distributions of these data points might be lower. Ho and Basu (2002) perform a principal component analysis in order to uncover the intrinsic dimensionality of their high-dimensional space given by 12 complexity measures and to identify the independent factors that determine a data set's difficulty. The results of a PCA on the data sets are shown in figure 43 and table 2. As the observed scales of the measures differ, the data was scaled first. Figure 43 **A** shows the percentage of variance explained by each component. Similar to Ho and Basu (2002), the first component explains about half of the variance (55.4%), the second component explains 20.4% and the third 12.4%. All other components explain less than 5% of the variance.

B shows the projections of the data on the first two components. This plot shows that most experiments form clusters, so they incorporate different aspect of separability. The two-dimensional data sets of non-spherical shape (E6, E7, E8) are close to each other with the moons (E6) being the closest among

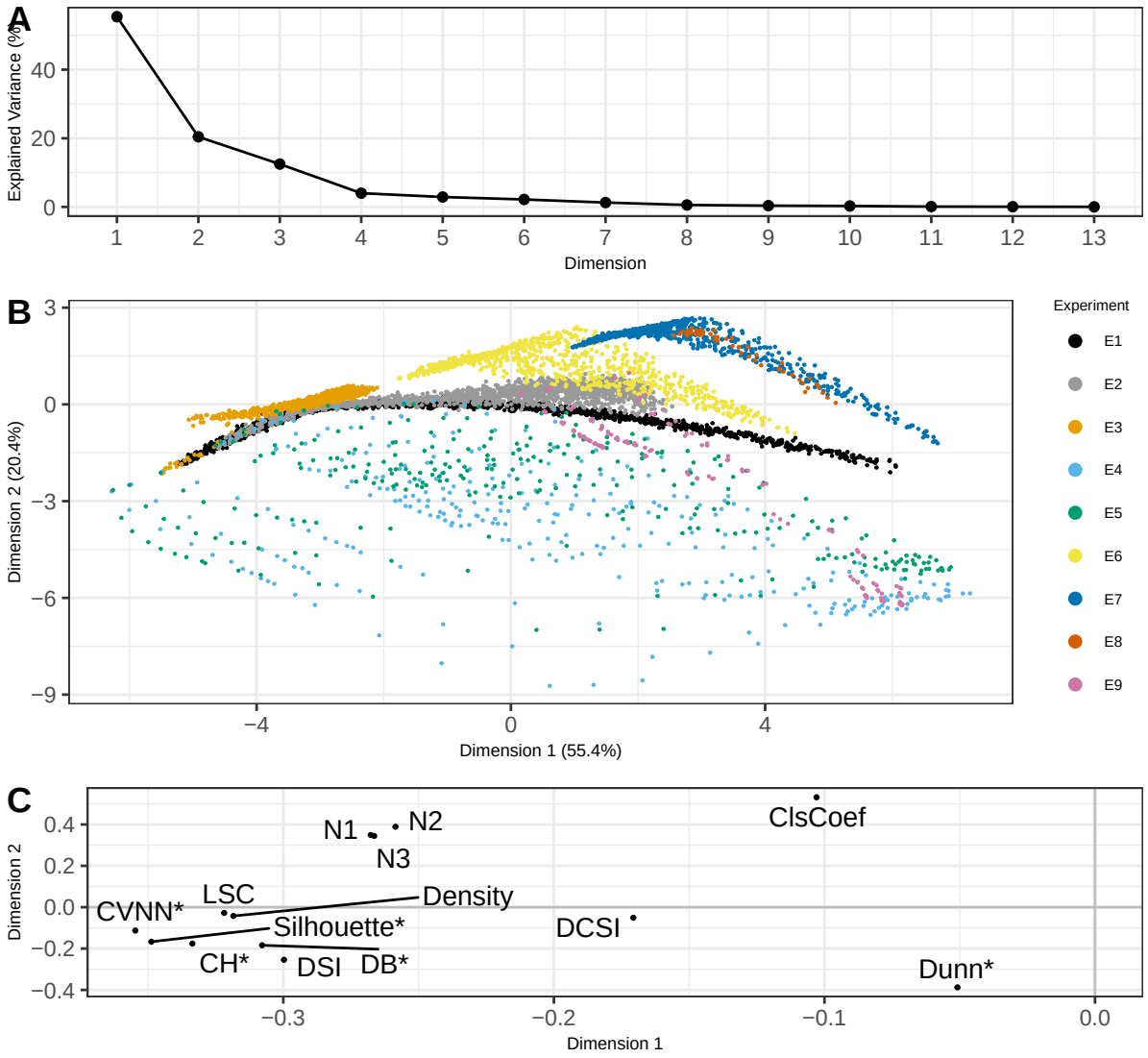

Figure 43: Summary experiments: results PCA

these three to the Gaussian experiments (E1, E2, E3). The three high-dimensional experiments (E4, E5, E9) don't form compact clusters but spread widely across the whole range.

C shows the loadings of the first two principal components. These values can also be found in table 2. Note that an overview of the values of the separability measures on the raw data for all experiments can be found in figure 66.

The neighborhood measures and all CVIs except Dunn* contribute strongly negative to the first component. The CVIs (apart from Dunn*) mainly measure the compactness of the classes and the distance of their centers while the neighborhood measures quantify how close neighbors from the opposite class are. So the first component could be interpreted as the joined effect of these two aspects. These results are at some point similar to the results of Ho and Basu (2002), who identify their first component (explaining over 50% of the variance) as the "joint effect of linearity of class boundaries" (quantified by measures of overlap and linearity not included here) and "proximity of opposite class neighbors" (Ho and Basu, 2002). With the linearity measures replaced by CVIs, the results found here strongly correspond to those of Ho and Basu (2002). These findings also match with the projections of the data sets, with E3 having the lowest values of all experiments for the first component (here, the data set is always linearly separable, doesn't overlap and the cluster centers are far from each other) and the non-spherical

experiments E6, E7, E8 (as well as E9) having the highest values for the first dimension. For E1, the proximity of the classes highly varies, explaining the high variance of the data sets from E1 in the first dimension.

The second component seems to be related to the dimensionality of the data sets. Note that in **B**, there is no clear difference in the second dimension between E9 (high intrinsic dimensionality) and E4 and E5 (dimensionality only artificially high). Looking at figure 66, the measures that have a positive loading for the second component (ClsCoef and N1-N3) have relatively low values for high-dimensional data sets while Dunn* has the highest negative loading and has relatively high values for high-dimensional data sets.

The third component (table 2) has high positive contributions from DCSI and Dunn*, so this component seems to be related to the minimal distance between the classes. Note that this is again similar to Ho and Basu (2002), whose third component quantifies the "intrusion of samples into the wrong class".

5.3 Separability Measures and Manifold Learning Methods on Real-World Data

Table 3: Experiments real-world data: maximum ARI and some separability measures

Data	Embedding	max ARI	DCSI	DSI	N2	CH*
iris	Raw	0.75	0.51	0.81	0.80	0.90
iris	UMAP	0.89	0.72	0.89	0.87	0.97
iris	t-SNE	0.89	0.76	0.88	0.83	0.97
wine	Raw	0.44	0.42	0.64	0.64	0.44
wine	UMAP	0.81	0.82	0.82	0.80	0.85
wine	t-SNE	0.82	0.65	0.83	0.78	0.85
mnist	Raw	0.10	0.35	0.35	0.60	0.21
mnist	UMAP	0.77	0.02	0.82	0.76	0.89
mnist	t-SNE	0.78	0.01	0.74	0.84	0.79
fmnist5	Raw	0.10	0.20	0.43	0.62	0.31
fmnist5	UMAP	0.76	0.00	0.79	0.80	0.82
fmnist5	t-SNE	0.70	0.00	0.67	0.83	0.63
fmnist10	Raw	0.07	0.20	0.47	0.56	0.40
fmnist10	UMAP	0.41	0.00	0.72	0.66	0.87
fmnist10	t-SNE	0.41	0.00	0.67	0.71	0.75
cifar	Raw	0.03	0.18	0.08	0.49	0.06
cifar	UMAP	0.03	0.00	0.07	0.37	0.11
cifar	t-SNE	0.03	0.00	0.07	0.37	0.10

The results on the real-world data sets are summarized in table 3. The table shows the maximum ARI and the values of DCSI, DSI, N2 and CH* for the raw data and the 3-D embeddings. N2 and CH* were chosen among the complexity measures/CVIs because the values of some other measures with higher correlations with ARI (figure 42) had almost no variability on the real-world data[27]. The Iris data set is the easiest data set according to ARI, which is also indicated by the separability measures. The clustering performance highly benefits from manifold learning for Wine, MNIST and FMNIST (both with 5 and 10 classes) and slightly for Iris. The improvement is also indicated by all separability measures except DCSI, which has very low values on the embeddings for all data sets except Iris and Wine. So on real-world data sets (and especially their embeddings), the lack of robustness of DCSI seems to be an even more serious issue than on synthetic data sets: DCSI doesn't differentiate between the embeddings of MNIST, FMNIST and CIFAR although there are clear differences in the clustering performance on these data sets. DSI and CH* have difficulties in differentiating between FMNIST-5 and 10. Both measures have higher values on the raw data of FMNIST-10 than on the raw data of FMNIST-5 and similar values for both

[27] N1 and N3 for example had values close to one for most data sets and Dunn* was close to zero for almost all embeddings.

label sets on the embeddings. Only N2 indicates that FMNIST-5 is easier to cluster than FMNIST-10. For CIFAR, the true classes weren't detected on the raw data as well as on the embeddings. Consequently, this is the data set with the lowest separability according to the selected measures. Note that for each of MNIST, FMNIST and CIFAR, the analysis was performed on a subsample of 10 000 observations (due to computational reasons)[28]. However, the results for MNIST and FMNIST don't differ much from the results in Herrmann et al. (2022) (and CIFAR wasn't used there).

In order to investigate which classes might be critical for clustering from a topological view, the separability measures were calculated for all pairs of classes. The results are shown in the following figures, together with the 2-D embeddings of the data sets. Note that the separability and the clustering was calculated/performed on 3-D embeddings; the 2-D embeddings were calculated for visualization only. The results for Iris and Wine are shown in appendix A, figures 67 and 68. For Iris, the visualization of both embeddings shows that *versicolor* and *virginica* are less separable than *setosa* is from the others. This is also indicated by all of the selected separability measures. The separability increases on the embeddings according to all measures, which can also be seen in a higher maximum ARI and a wider ε-range (also note the different x-axes in the performance plots).

The results for Wine are similar (figure 68): The embeddings lead to an increase in performance (higher maximum ARI and wider ε-range) which is also indicated by the separability measures. The plots of the embeddings show that classes 1 and 3 are the most separable and these two classes have the highest pairwise separability for all measures. So for Iris and Wine, the groups defined by the class labels seem to correspond to connected components relatively well.

The results for MNIST are shown in figure 44. The embeddings indicate that there are some digits that are more difficult to separate, especially 3 and 5 as well as 4 and 9. Note that the results of the t-SNE embedding look very similar to the results of van der Maaten and Hinton (2008) (figure 2), who performed t-SNE on a subsample of 6000 images from MNIST. Similar to the experiments on synthetic data, UMAP yields more compact, separated clusters (which leads to higher values in DSI and CH^*), however the performance of DBSCAN is similar on both embeddings. The second row of figure 44 even shows that t-SNE performs sightly better (wider ε-range), which is only indicated by N2 (table 3). Looking at the embeddings, t-SNE seems to be slightly more successful in separating 8 from 3 and 7 from 9. The plots of the pairwise separability show that the separability increases for almost all measures and pairs of classes, however DCSI has lower values on the embeddings than on the raw data for some hardly separable classes, which was already observed in some synthetic experiments. While the total DCSI is low (table 3), there are a lot of classes that have a high pairwise separability. The groups of digits that are very close ($\{4, (7), 9\}$ and $\{3, 5, (8)\}$) are also indicated by the pairwise separabilities of DCSI, DSI and CH^*, however DSI is the only one that reveals these hardly separable groups of classes already on the raw data. These two groups might be critical from a topological perspective: While the results of McInnes et al. (2018) (figure 3) suggest that one might separate all 10 digits with the "right" choice of parameters for UMAP, $\{4, 9\}$ and $\{3, 5, 8\}$ don't form separated components in any of the embeddings shown in Dalmia and Sia (2021) (UMAP and extensions/improvements of the algorithm, figure 3).

The results for FMNIST-10 and -5 are shown in figures 45 and 46. Similar to the UMAP embedding in Dalmia and Sia (2021), the plots of the embeddings of FMNIST-10 indicate that there are some hardy-separable groups of classes: the different types of shoes (5, 7, 9) and the different types of clothes except trousers (0, 2, 3, 4, 6), while bags and trousers (8 and 1) are well separated from the other classes. T-Shirts/Tops and dresses (0 and 3) are slightly more separated from the other clothes (pullovers, coats, shirts, 2, 4, 6), which seem to form one component. Again, the groups of difficult classes are indicated by the pairwise separabilities, with DSI (and CH^* to some extent) slightly revealing these classes already on the raw data.

So it might be critical to use this label set as true classes. The results of Dalmia and Sia (2021) (figure 3) suggest that one can separate some of the groups with modified versions of UMAP, e.g. the three types of shoes can be separated from each other, however there remains a subset of sneakers (7) that is added to ankle boots (9). To overcome the problem of hardly separable groups, one can also merge similar categories as Mukherjee et al. (2019) does. The results of this 5-class data set

[28] The subsample for FMNIST-5 and FMNIST-10 was the same, so that the clustering was only calculated once and evaluated for both label sets.

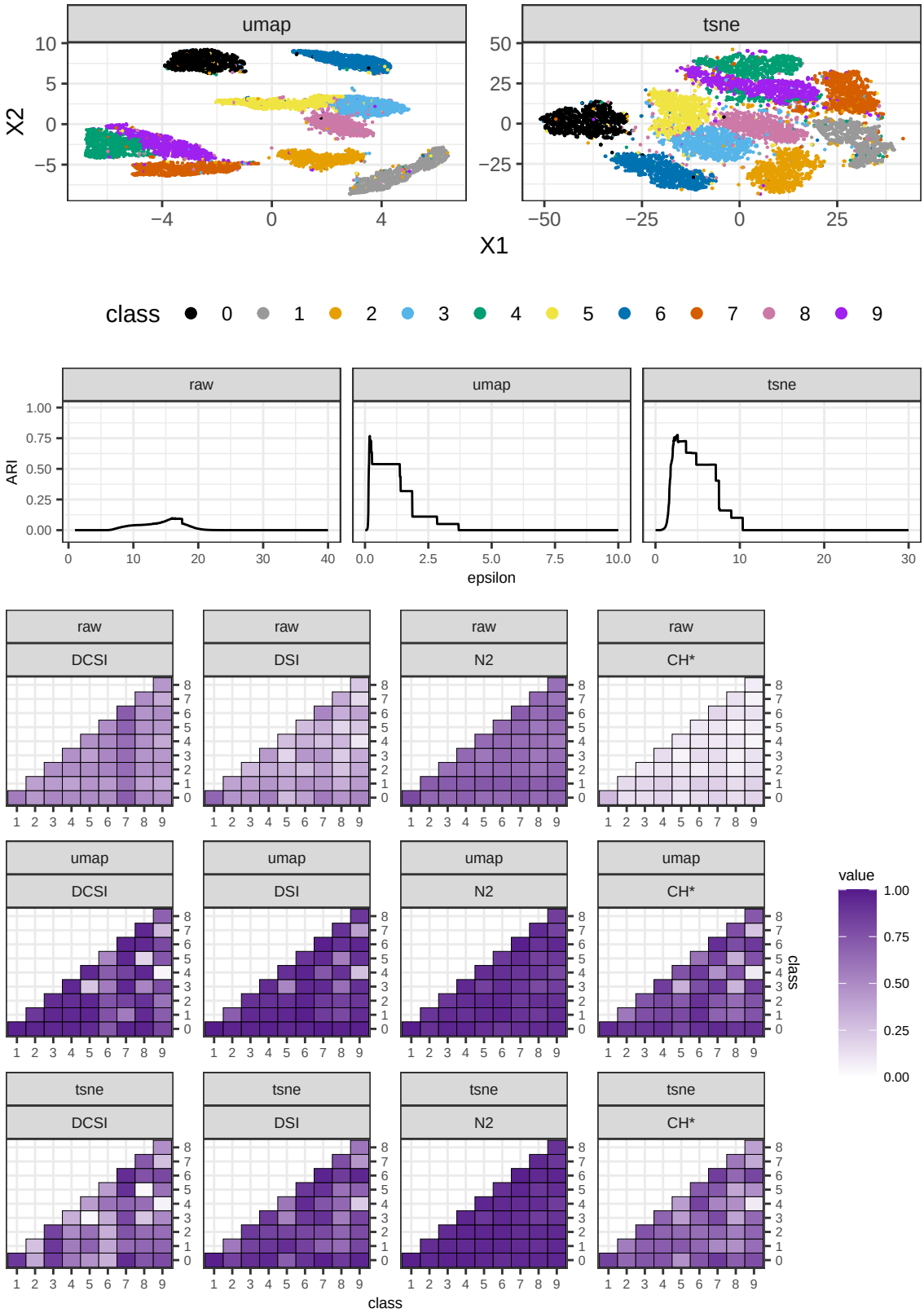

Figure 44: Experiments real-world data: results MNIST

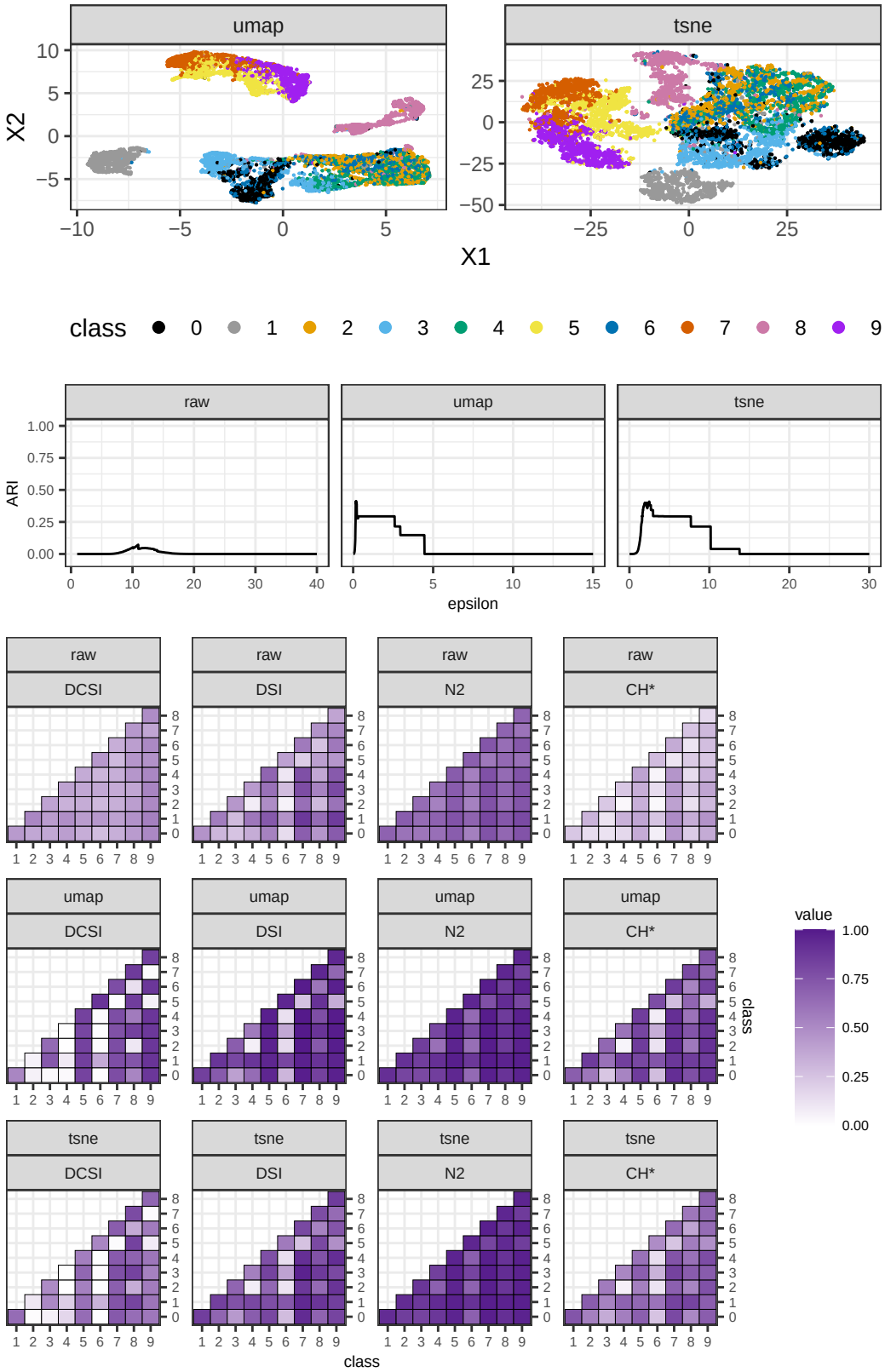

Figure 45: Experiments real-world data: results FMNIST-10

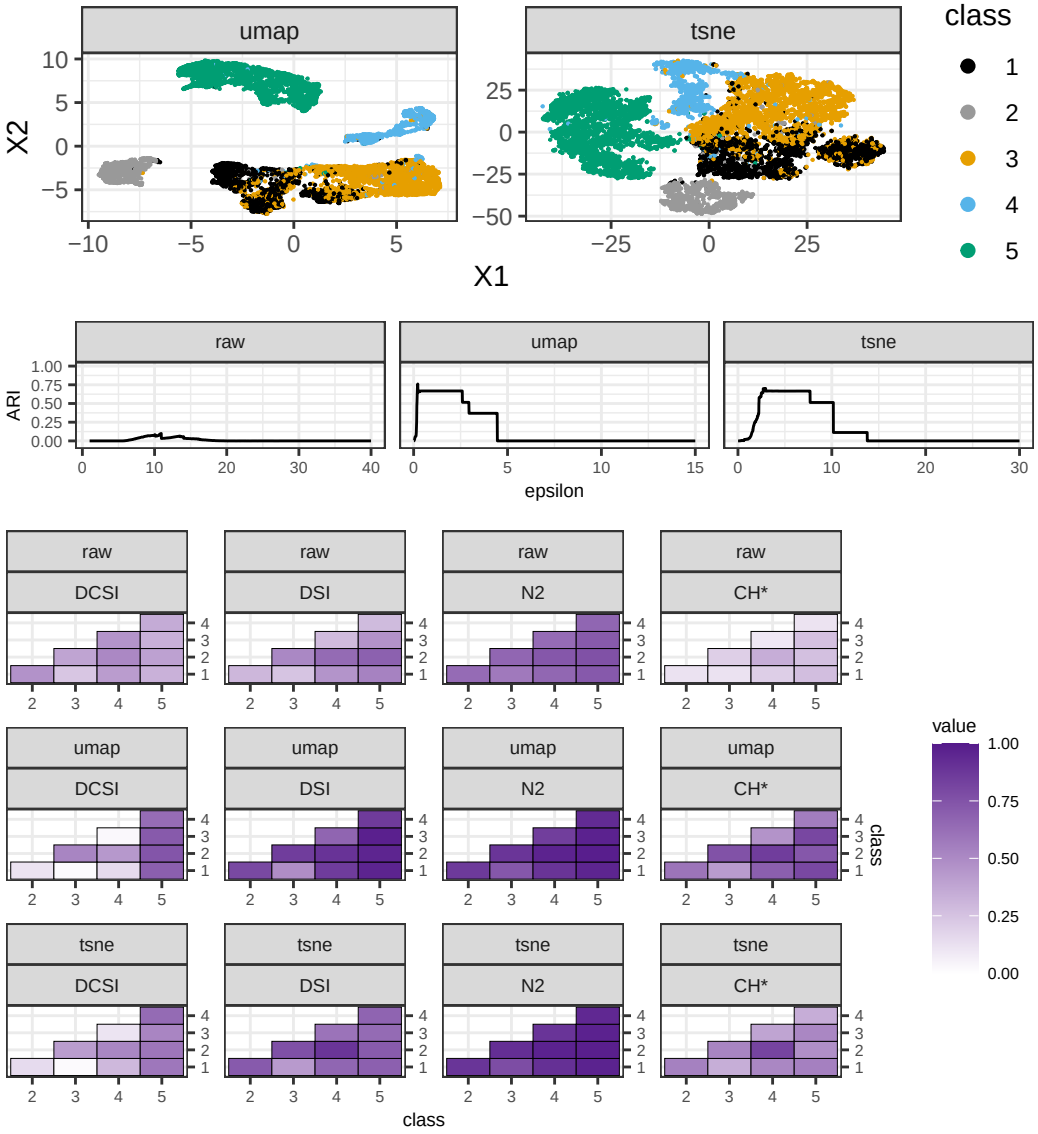

Figure 46: Experiments real-world data: results FMNIST-5

are shown in figure 46. The class of shoes now forms one well separated component, which is also indicated by the separability measures. However the classes of clothes (1 = {T-Shirt/Top, Dress}, 3 = {Pullover, Coat, Shirt}) still seem somewhat problematic in the visualizations of the embeddings. DCSI has very low values for this pair of classes (1, 3), while the lowest values of pairwise separability for DSI and CH* are higher for FMNIST-5 than for FMNIST-10. The increased separability is also indicated by a higher performance of DBSCAN on FMNIST-5. This data set might need further investigation about which classes to use or maybe which observations to exclude as they are too similar to observations of other classes and their true label might be ambiguous (e.g. consider the subset of sneakers that is merged to the class of ankle boots for all embeddings in Dalmia and Sia (2021)).

The results for CIFAR-10 are shown in the appendix in figure 69. The maximum ARI is close to zero for the raw data as well as the embeddings. Both UMAP and t-SNE yield one point cloud with relatively few structure, which is also indicated by very low values of separability for all measures. However, one can see that the groups of animals (3, 4, 5, 6, 7, 8) are closer to each other than to the group of vehicles and vice versa. Another 2-D UMAP embedding for visualization was calculated for 30 000 observations (instead

of 10 000), but the results were similar. There are clustering algorithms based on deep neural networks that achieve a high accuracy ($> 90\%$) and ARI (> 0.8) (Niu et al., 2021) on this data set. Most of these high-performing methods were only recently developed (see https://paperswithcode.com/sota/image-clustering-on-cifar-10 for example), so the CIFAR-10 data set seems to be more difficult than the other real-world data sets investigated here. Furthermore, the grayscale versions used here are probably much harder cluster than the original images, as the color of the images might be an important aspects of the clusters. Another reason for the difficulty of this data set might be its dimensionality, especially the intrinsic one. A subsample size of 10 000 (and even 30 000) may not be enough to learn the topological structure, if the dimension of the underlying manifold is too high[29]. Furthermore, the classes in CIFAR-10 are probably based on more different aspects than the classes in MNIST and FMNIST, where it's mainly the shape that varies between the classes. For CIFAR-10, these different aspects are - besides the shape - probably also the color (for the original data), scale, location or texture. These aspects are different subspaces of the data set, and there can be different clustering solutions in different subspaces (i.e. it can be meaningful to cluster objects with similar shape but different color and vice versa) (Zimek and Vreeken, 2013). Furthermore, the relevant subspaces are not necessarily the same for different clsuters (Zimek and Vreeken, 2013): while for some classes of animals (e.g. deer), the color is similar for all objects of the class, color is of less importance for cars or ships. Using Euclidean distances in the combined space of shape, texture etc. might not be meaningful (Zimek and Vreeken, 2013). Finally, there is no guarantee that the classes in CIFAR-10 correspond to structures that density-based algorithms like DBSCAN are able to discover. The evaluation of clustering on data with class labels might not be valid, as Zimek and Vreeken (2013) emphasize that the difference between clusters and classes has to be taken into account.

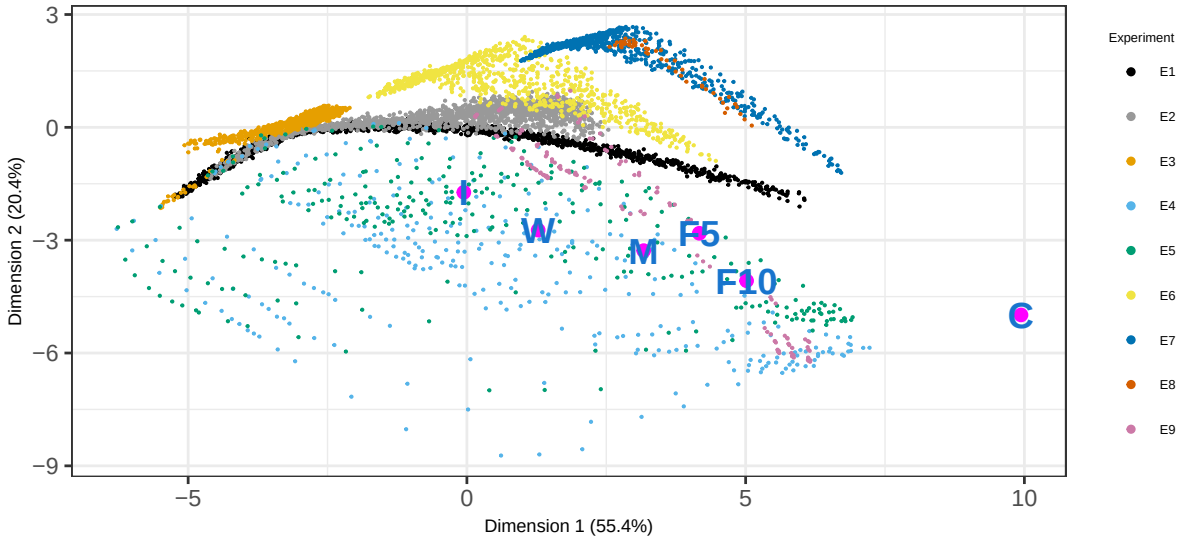

Figure 47: Experiments real-world data: principal component projections

Figure 47 shows the PCA projections of the real-world data sets investigated in this section. As seen before, the first dimension captures the most important aspects of separability - the proximity of class centers and the presence of neighbors of the wrong class. The positions of the 6 data sets along the first dimension corresponds to their difficulty measured by maximum ARI. The order of the data sets along the second dimension is relatively consistent with their dimensionality. CIFAR-10 is far away from any of the other data sets (both the synthetic and real ones), which indicates that the classes in CIFAR-10 are more difficult to learn than those in the other data sets investigated in this chapter.

[29] See Niyogi et al. (2011) for more information on how the sample size needed to learn the homology of an underlying manifold from a noisy sample depends on the dimension of the manifold and the data.

6 Discussion & Conclusion

The aim of this thesis was to investigate the topic of separability from the (topological) perspective of clustering. A literature review has shown that the existing measures of separability cover certain different aspects of separability, while no measure is able to incorporate all principles necessary to quantify the separability of clusters from a topological point of view. Complexity measures and the DSI focus on classification, so they don't measure connectedness but only the between-class separation. There are further differences between separability from the perspective of classification compared to clustering: Measures like N1 and N3 (the fraction of border points and the error rate of the NN-classifier) indicate a higher separability when points far from the border between the classes are added, although the separability from the perspective of clustering might not change. Most cluster validity indices (CVIs) on the other hand favor classes of spherical shape as they take the compactness of classes into account.

In order to overcome some disadvantages of the existing measures, a new measure for density based clustering (DCSI) was developed. It quantifies both within-class connectedness and between-class separation in a way that seems suitable for clustering from a topological perspective. Furthermore, two ways to quantify separability from a purely topological perspective were proposed. Both measures aim to quantify the persistence of the component with the second longest lifetime in a persistence diagram. The limitations of this approach are discussed later.

The experiments on synthetic data sets show that most separability measures have a high correlation with the clustering performance (measured by maximum ARI). There are some exceptions for certain measures on certain data sets, e.g. some CVIs use the cluster centers, which isn't appropriate for circles and some measures suffer from effects of the curse of dimensionality when it comes to high-dimensional data sets (e.g. DSI on nested n-spheres). The newly developed DCSI correlates highly with ARI in most cases, but lacks robustness especially when it comes to the embeddings, where a core point being merged to the "wrong" class can lead to a drastic decrease in separability according to DCSI. Furthermore, its sensitivity to the ε_i and $MinPts$ parameter needs to be investigated in more detail.

The summary of all 9 different settings of the synthetic experiments shows that for some measures, the correlation with ARI on all data sets is lower than for the specific experiments, as these measures favor clusters of a certain shape. For these measures (most CVIs and DSI), the correlation with ARI is higher on the UMAP and t-SNE embeddings than on the raw data, as the structure of the embeddings is less diverse than the original data sets. This indicates that some measures are less suited to quantify separability in general (and not in a certain setting, e.g. for spherical clusters) than those measures with a high correlation on both the original data and the embeddings (DCSI, N1, N3). The PCA shows that the 9 settings incorporate different aspects of difficulty and separability and leads to similar interpretations of some components as in Ho and Basu (2002).

An important finding of the experiments is that the evaluation of DBSCAN clustering with ARI might be critical when the noise points are treated as one cluster. A possible solution might be to assign each noise point to its own cluster for the evaluation with ARI.

The benefits of manifold learning for clustering presented in Herrmann et al. (2022) were confirmed in many cases, which is often also indicated by the separability measures. However especially for data sets with a low separability, UMAP and t-SNE can lead to a decrease in performance by building bridges between the classes or by merging (parts of) the components. t-SNE embeddings sometimes have a lower separability according to the measures than UMAP embeddings, while the performance of DBSCAN is relatively similar. This finding corresponds to the literature which states that UMAP produces compact clusters with more white space in between than t-SNE (Kobak and Linderman, 2021).

The nested n-spheres in experiment 9 reveal interesting differences between UMAP and t-SNE: While UMAP seems to be slightly better in separating the spheres, t-SNE preserves more of the outer geometry (i.e. the nestedness of the spheres). This indicates that UMAP - as it is mainly designed for preserving the topological structure - might be more suited for clustering whereas t-SNE has advantages when it comes to visualization.

The goal of the experiments on real-world data sets was to find out if some frequently used data sets are suited for the evaluation of clustering and to what extent manifold learning can increase the separability of these data sets. The results show that most measures can both quantify the overall separability of these data sets as well as the separability of pairs of classes, so they can indicate which classes have a high overlap and might not be suited for the evaluation of clustering (e.g. some classes of clothes in

FMNIST). Similar to the synthetic experiments, manifold learning increases the overall separability and the performance of DBSCAN, while the separability of classes that aren't well separated might decrease.

DCSI again lacks robustness, in the case of multi-class data sets also because of its definition using the worst values of separation and connectedness of all classes. Taking the mean of separation and connectedness instead might mitigate this weakness. However, DCSI is well suited to indicate the separability of pairs of classes, similar to other measures like DSI and N2. The "topological separability" on the other hand is not suited for real-world data. Its further limitations and disadvantages are now discussed.

In order to measure separability from a purely topological perspective, one could quantify the persistence of the component with the second longest lifetime in a persistence diagram either by its p-value or by its lifetime relative to the lifetime of the longest living component. However, both methods have certain limitations: The p-value of the second component does not only depend on the sample size (which might not be wanted when it comes to quantifying separability) but also on the robustness of the used function. Furthermore, some evaluations have shown that the p-value is not very sensitive in measuring changes of the separability, so it wasn't used for the synthetic experiments.

The relative lifetime of the second component yielded relatively good results (wide observed range, high correlation with ARI) for some settings, especially those with Gaussian mixtures. However, this measure has several drawbacks: It has difficulties in dealing with clusters of different densities, even if they are well separated. For sublevel set filtrations (used for DTM and the distance function), the lifetime of the longest living component is somewhat arbitrary, as it depends on the grid where the function is evaluated. So the persistence diagram for DTM and the distance function should always be interpreted with caution, as the visual impression is influenced by the evaluated grid. This finding also shows the importance of p-values for persistence diagrams. The knn density estimator is robust to changes in the evaluated grid but it fails to correctly estimate the density when it comes to complicated shapes like intertwined spirals. Note that the parameter sensitivity of the used functions (DTM and knnDE) wasn't investigated in detail and the experiments were restricted to DTM, knnDE and the distance function but could easily be extended to further (kernel) density estimators.

While for the original, synthetic data, the data generation ensures that the two most persistent component correspond to the true classes (in most cases), this is not the case on the embeddings anymore. This is also the reason why it doesn't make sense to use the topological separability on real-world data sets as in this case, one does not only have to measure the separation of the components but also if they correspond to the true classes. So it remains an open question whether persistent homology can provide useful information for the quantification of separability in practice.

When it comes to synthetic data, another possibility might be to compare the second and third component (i.e. the component with the third longest lifetime). If the lifetime of the second component is big relative to the third one, this might indicate that the data set consists of two clusters. This measure does not depend on the evaluated grid, which would be an advantage compared to the method used in this thesis. However, if the lifetime of the third component is small (or even zero, if there is no third component), this measure wouldn't be sensitive to changes in separability anymore. The behavior of this measure could be investigated in future work.

Another limitation of this work is that the intrinsic dimensionality for the synthetic experiments was (except for the n-spheres) relatively low. While low-dimensional manifolds correspond to the model presented in Niyogi et al. (2011), the results of experiment 9 (nested n-spheres) suggest that there are several interesting effects to investigate for high-dimensional manifolds. For nested n-spheres for example, one could conduct further experiments or calculations in order to investigate their separability depending on the sample size and the dimensionality:

What makes the separation of these two spheres difficult is the fact that the mean distance between two points on the outer sphere is bigger than the mean distance between a point on the outer and a point on the inner sphere. As the dimensionality increases, the variance of the intra- and between-class distances decreases, so it's less likely that the nearest neighbors of a point on the outer sphere lie on the outer sphere as well, which explains why UMAP isn't able to separate the two spheres. However, with an arbitrary high sample size (on the outer sphere), it should be possible to separate the components as the between-class distances are bounded below by the difference of the two radii ($r_2 - r_1$), while the

within-class distances can become arbitrarily small. It would be interesting to investigate how fast the sample size has to increase as the dimensionality of the spheres increases.

The sensitivity of UMAP and t-SNE to the nearest neighbor/perplexity parameter could also be investigated in more detail. For the experiments, these values were chosen based on the literature and results of a pilot study. It would be interesting to better understand the behavior of UMAP and t-SNE depending on these parameters, especially for high-dimensional data, as the pilot study for example showed that for the high-dimensional experiments 4 and 5, a higher perplexity value (t-SNE) yields better results while this wasn't the case for the nested spheres in experiment 9.

The results of this thesis suggest that separability is a crucial issue when manifold learning is used to enhance cluster analysis: Only if a data set consists of several well-separated components, manifold learning methods can preserve and even emphasize this structure and thereby increase the performance of clustering algorithms like DBSCAN. None of the presented measures seems to be able to adequately measure separability in its entirety, however a combination of different measures can compensate weaknesses of single measures. Similar to clustering, there is no general definition of separability and each measure incorporates its own idea of "good" clusters. DCSI aims to quantify separability from a topological, density-based perspective of clustering. In this context, it could also be used as a CVI.

Not only when manifold learning is combined with clustering but also for cluster analysis in general, it's important to clearly specify what kind of perspective on clustering is taken, as this determines if a data set consists of meaningful, well-separated clusters that can be detected with a suitable algorithm.

A Additional Figures

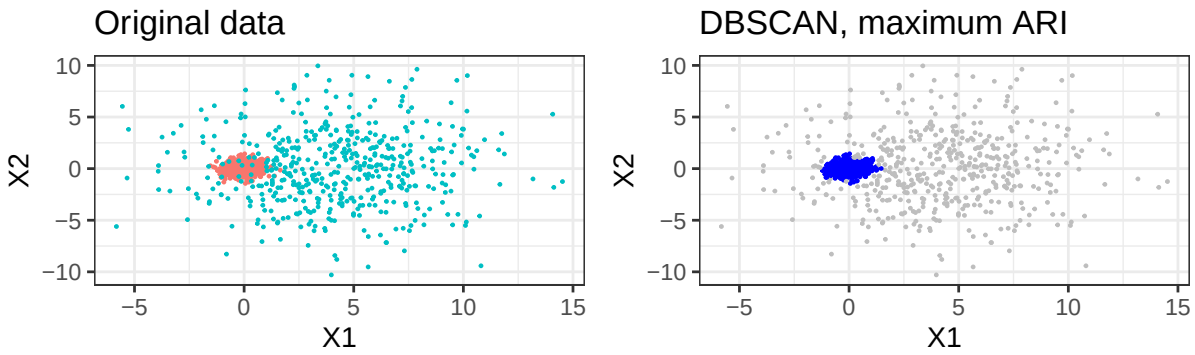

Figure 48: Motivation of ARI_2

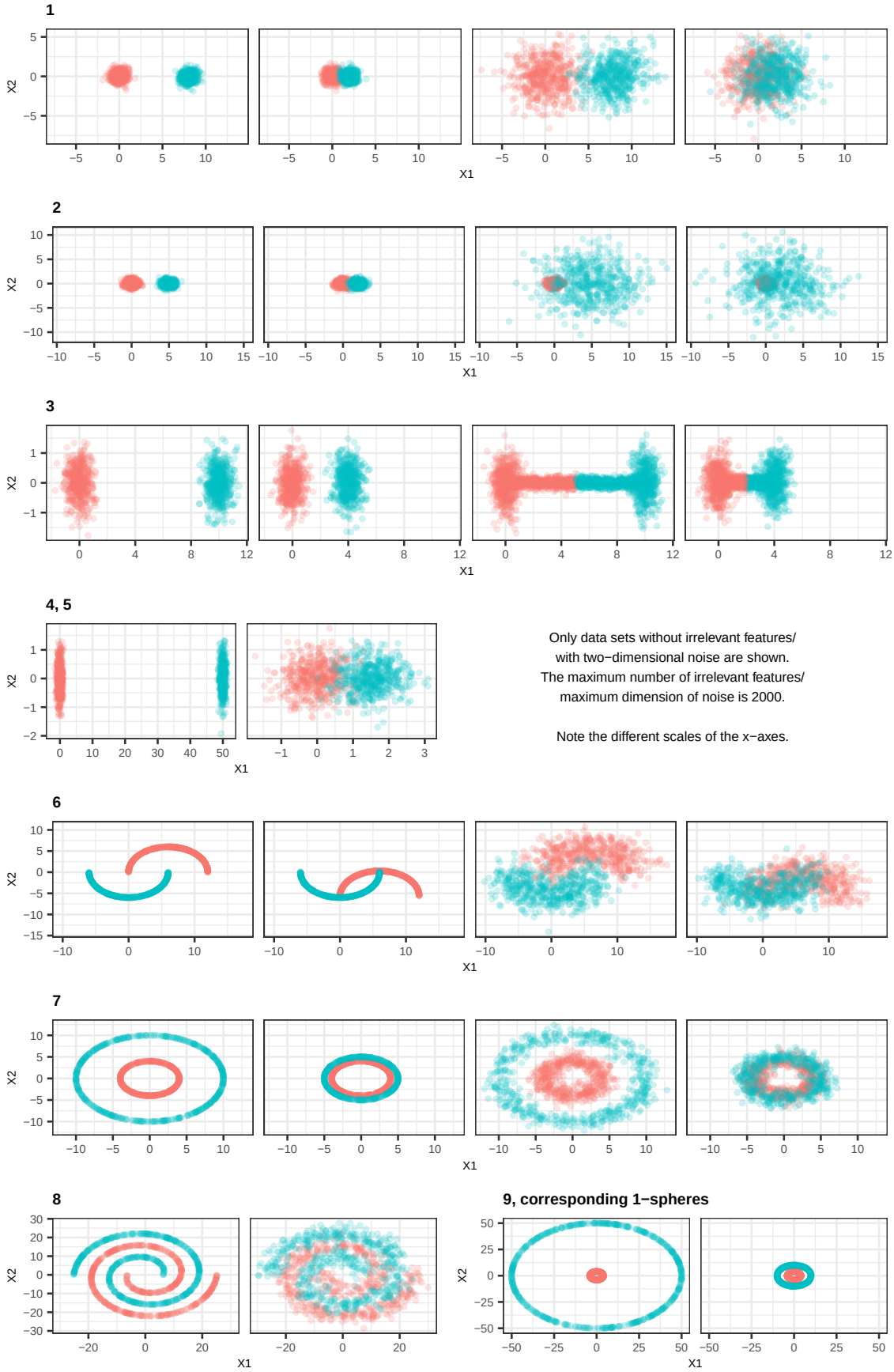

Figure 49: Data sets synthetic experiments

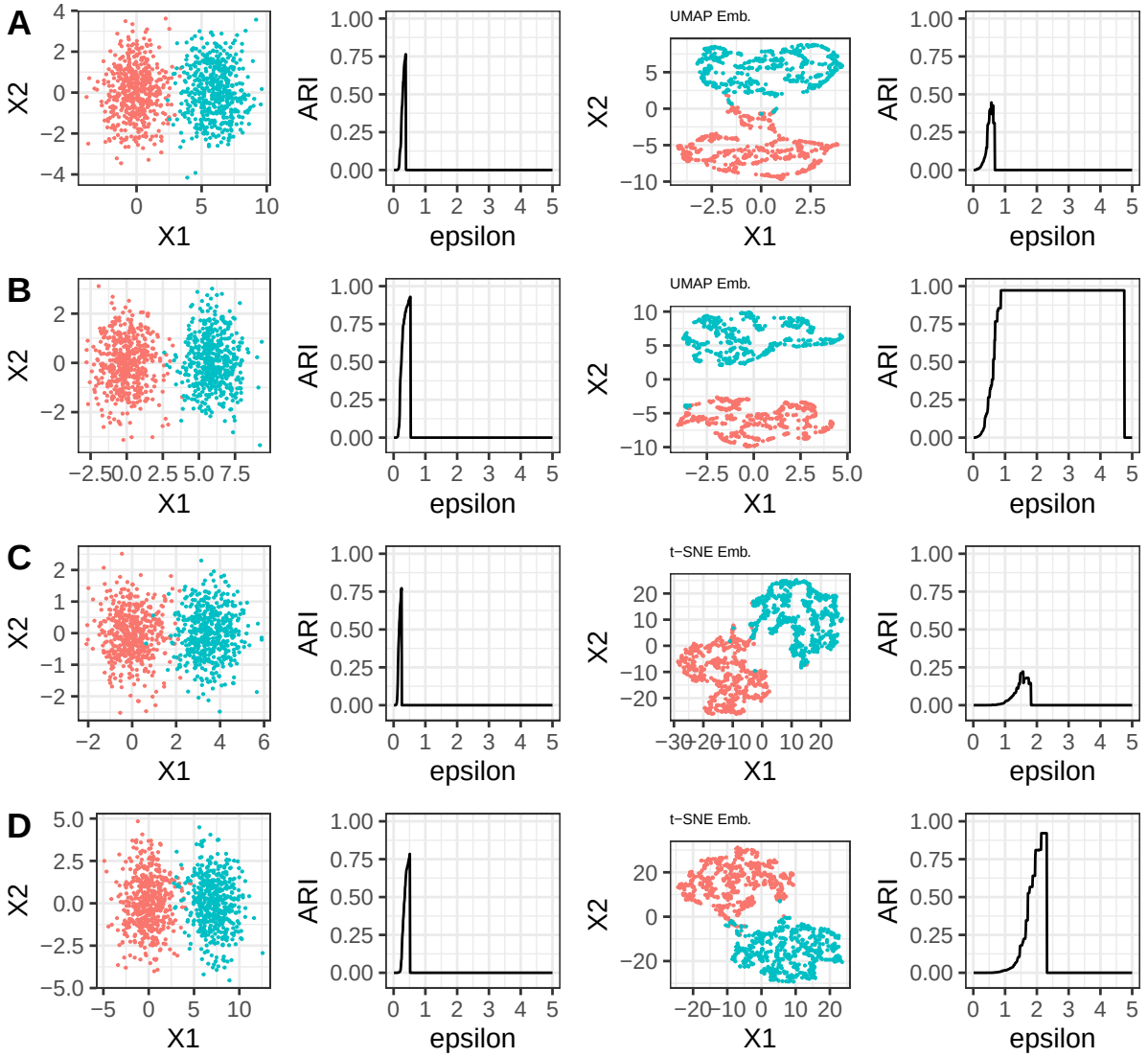

Figure 50: Results Experiment 1: data sets with biggest decrease in ARI (A, C) and DCSI (B, D) for UMAP and t-SNE

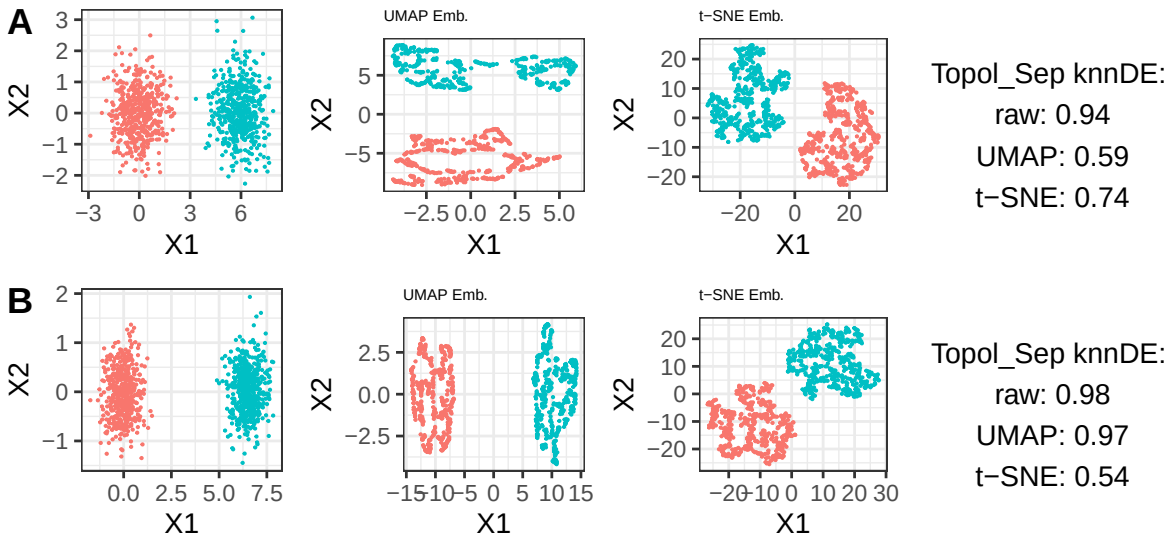

Figure 51: Results Experiment 1: data sets with biggest decrease in Topol. Sep. knnDE for UMAP and t-SNE

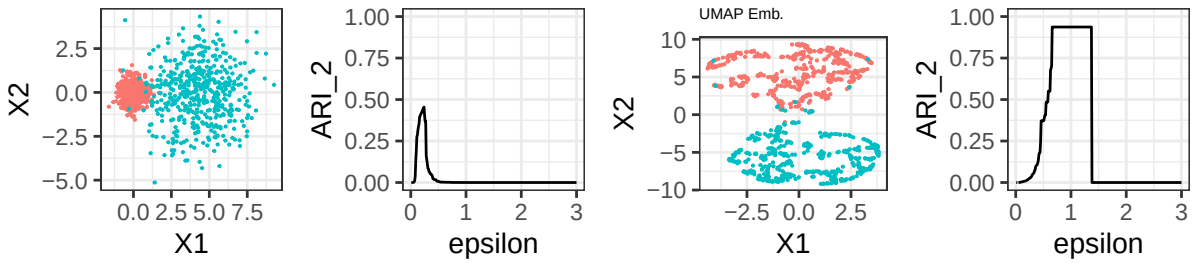

Figure 52: Results Experiment 2: manifold learning methods can separate density separated clusters

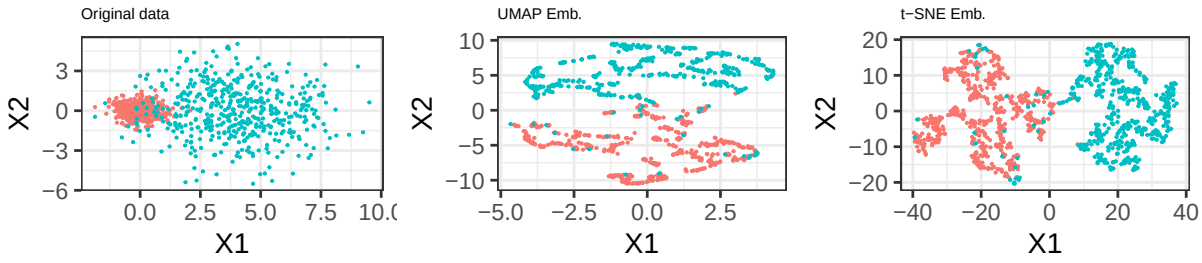

Figure 53: Results Experiment 2: data set with the biggest decrease in ARI_2 for UMAP

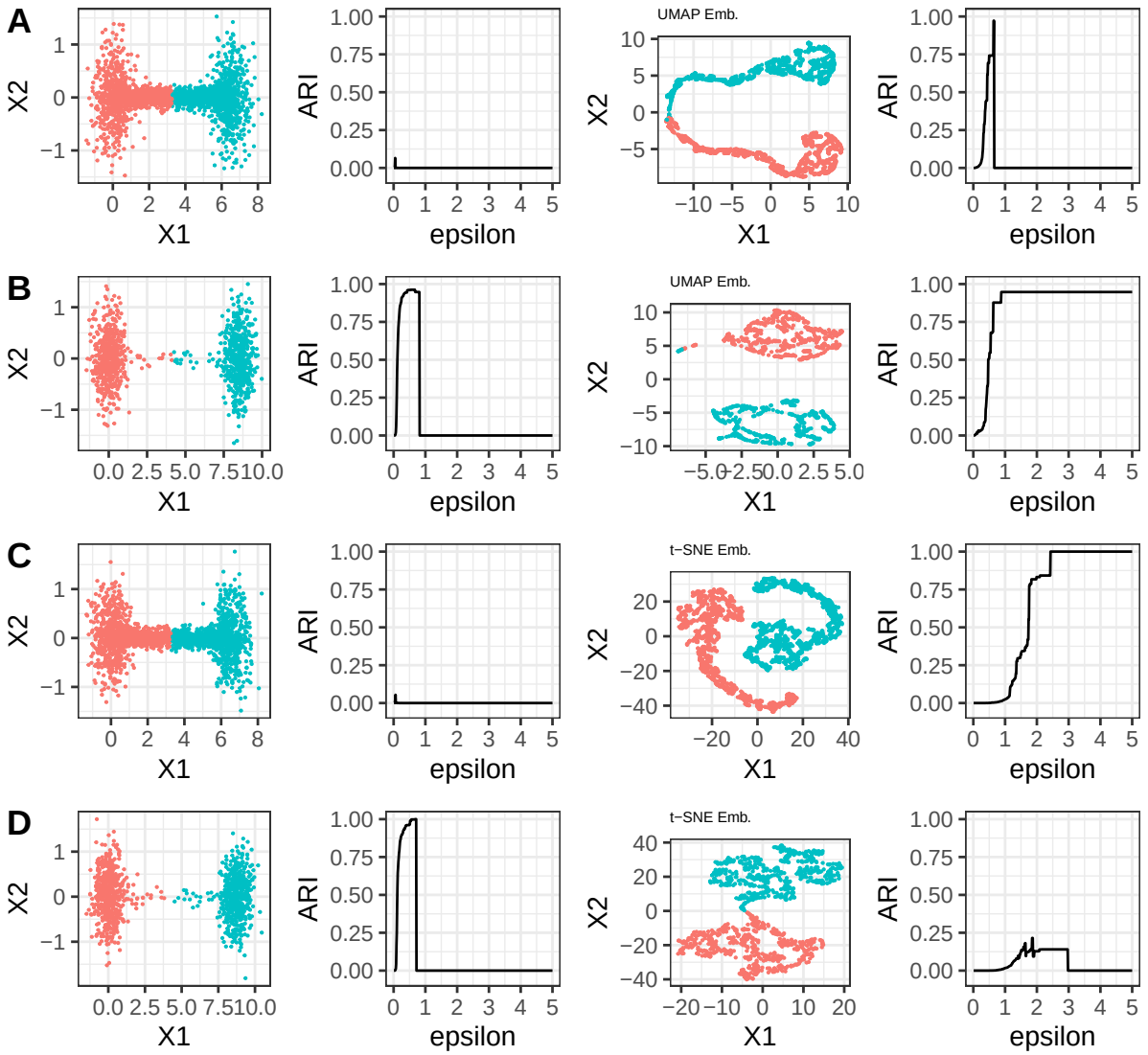

Figure 54: Results Experiment 3: data sets with highest increase in ARI for UMAP and t-SNE (A, C), biggest decrease in DCSI for UMAP and t-SNE (B) and biggest decrease in ARI for t-SNE (D)

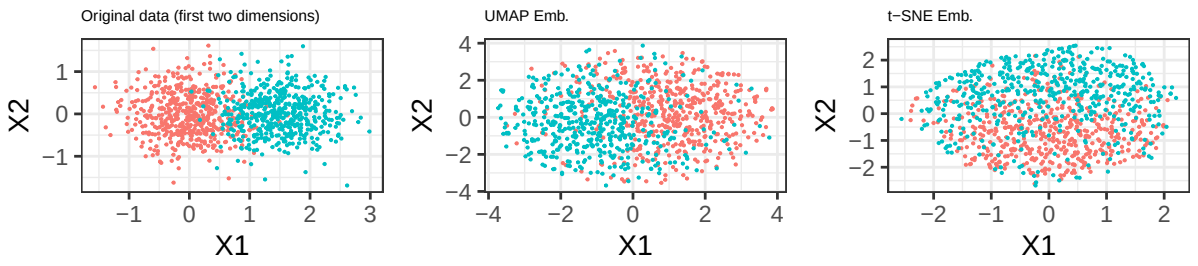

Figure 55: Results Experiment 4: data set with highest decrease in DCSI for t-SNE and second highest decrease in DCSI for UMAP

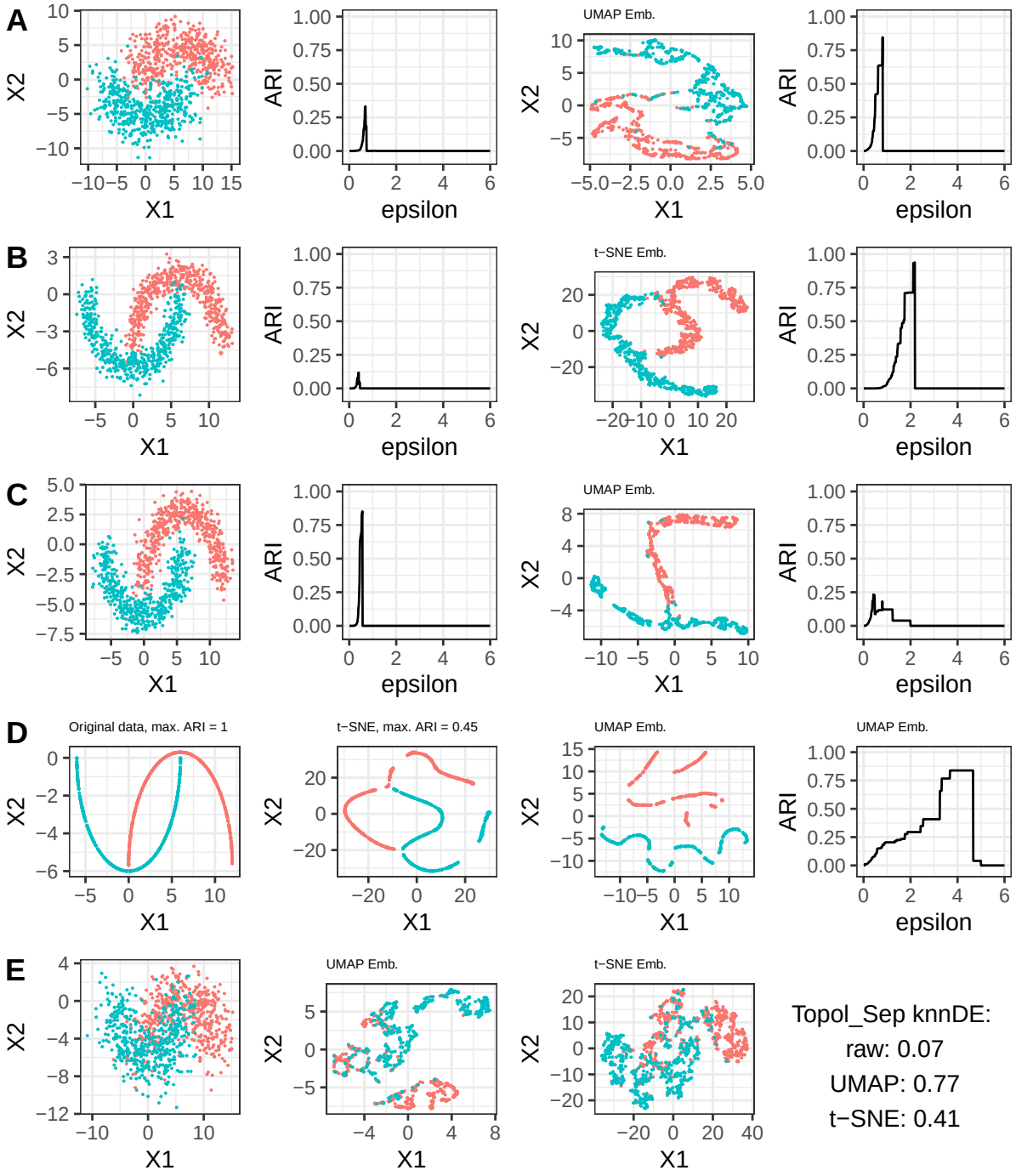

Figure 56: Results Experiment 6: data sets with biggest increase in ARI for UMAP and t-SNE (A, B), biggest decrease in ARI for UMAP and t-SNE (C, D) and biggest increase in Topol. Sep. knnDE for UMAP (E)

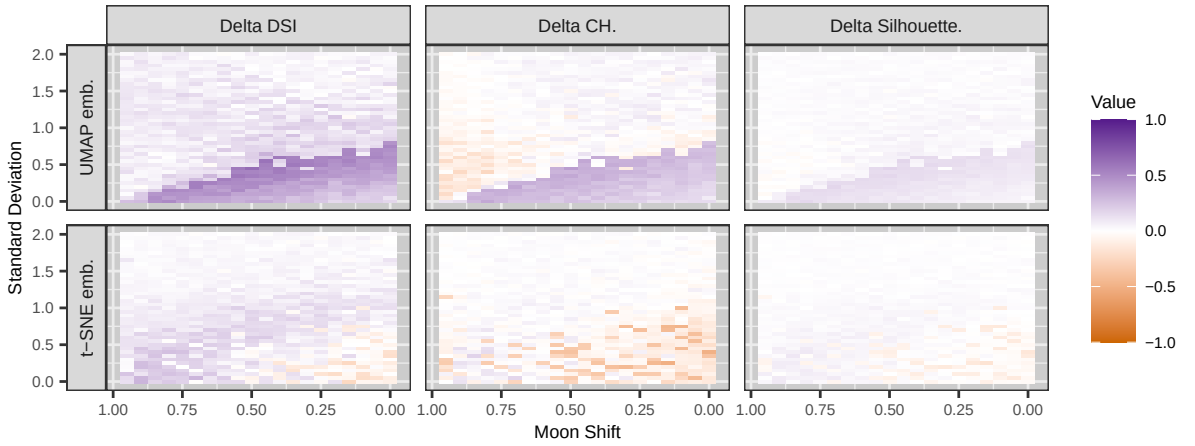

Figure 57: Results Experiment 6: heatmaps of change of some separability measures

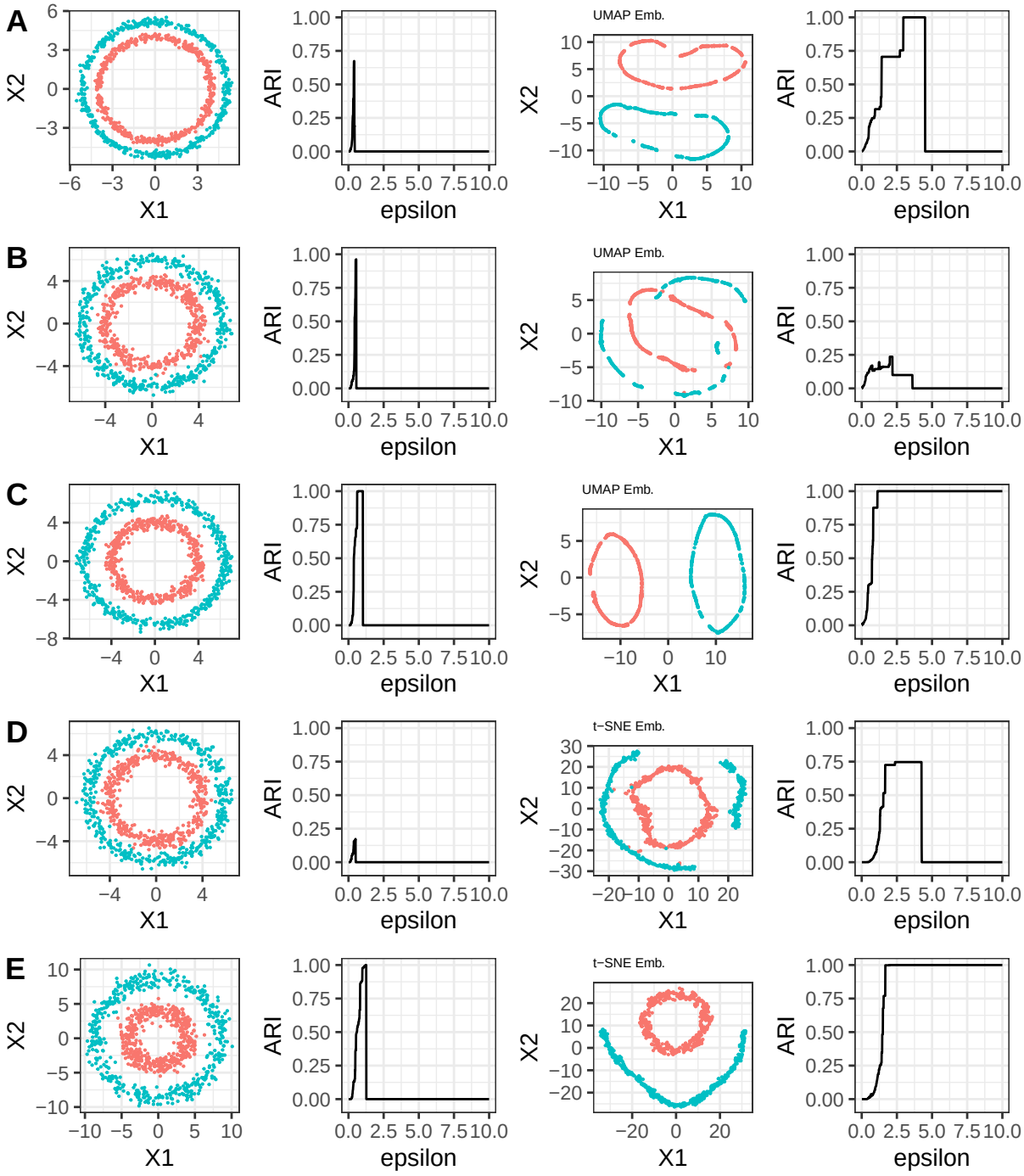

Figure 58: Results Experiment 7: data sets with biggest increase in ARI for UMAP and t-SNE (A, D), biggest decrease in ARI for UMAP (B), biggest increase in DCSI for UMAP and t-SNE (C, E)

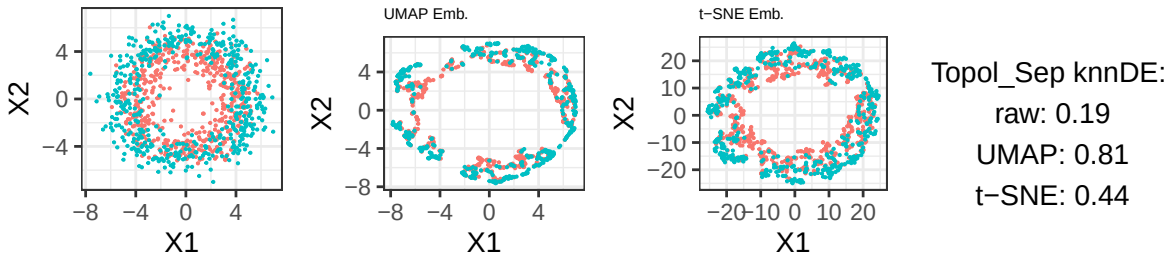

Figure 59: Results Experiment 7: data set with a high increase in Topol. Sep knnDE for UMAP

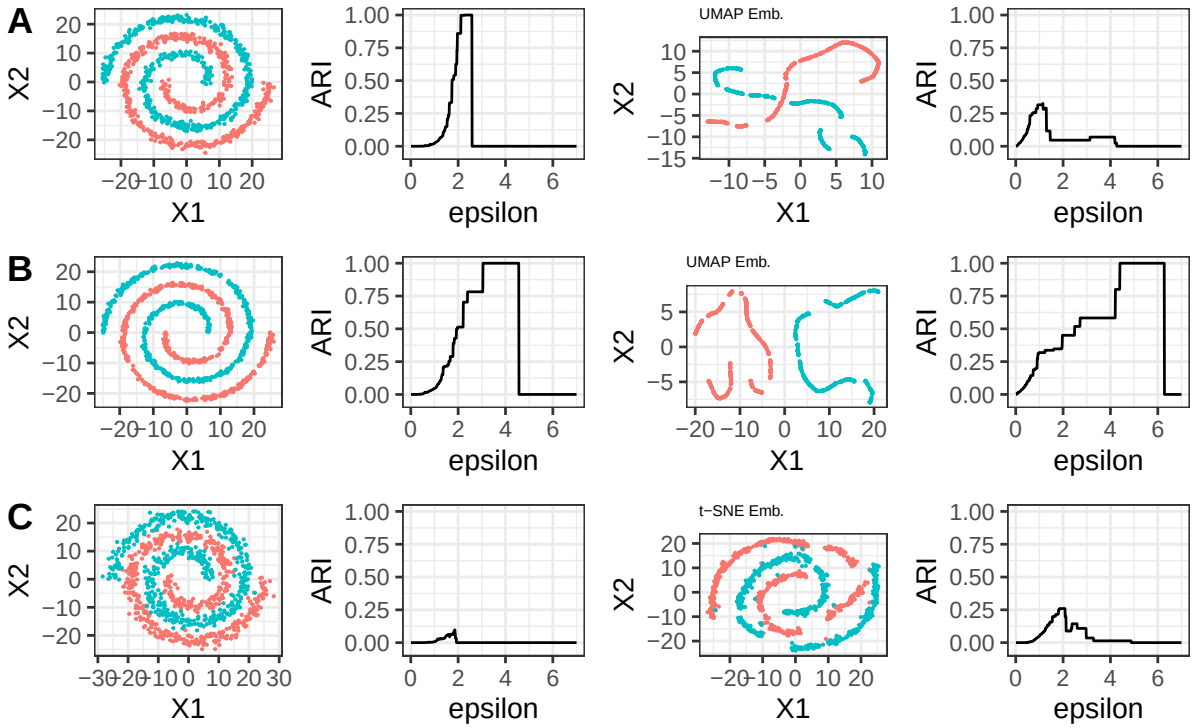

Figure 60: Results Experiment 8: data sets with biggest decrease in ARI and DCSI for UMAP (A), biggest increase in DCSI for UMAP (B), biggest increase in ARI for t-SNE (C)

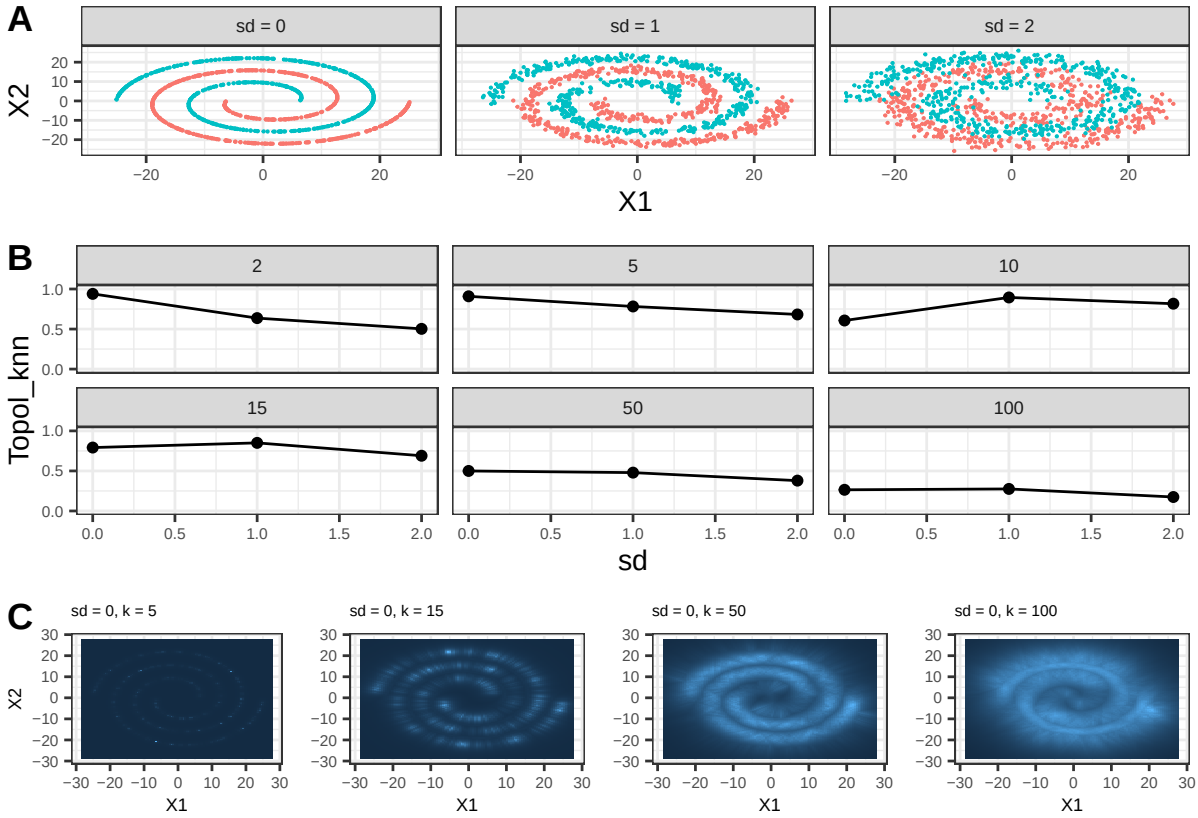

Figure 61: Experiment 8: topological separability (knnDE) for different values of k

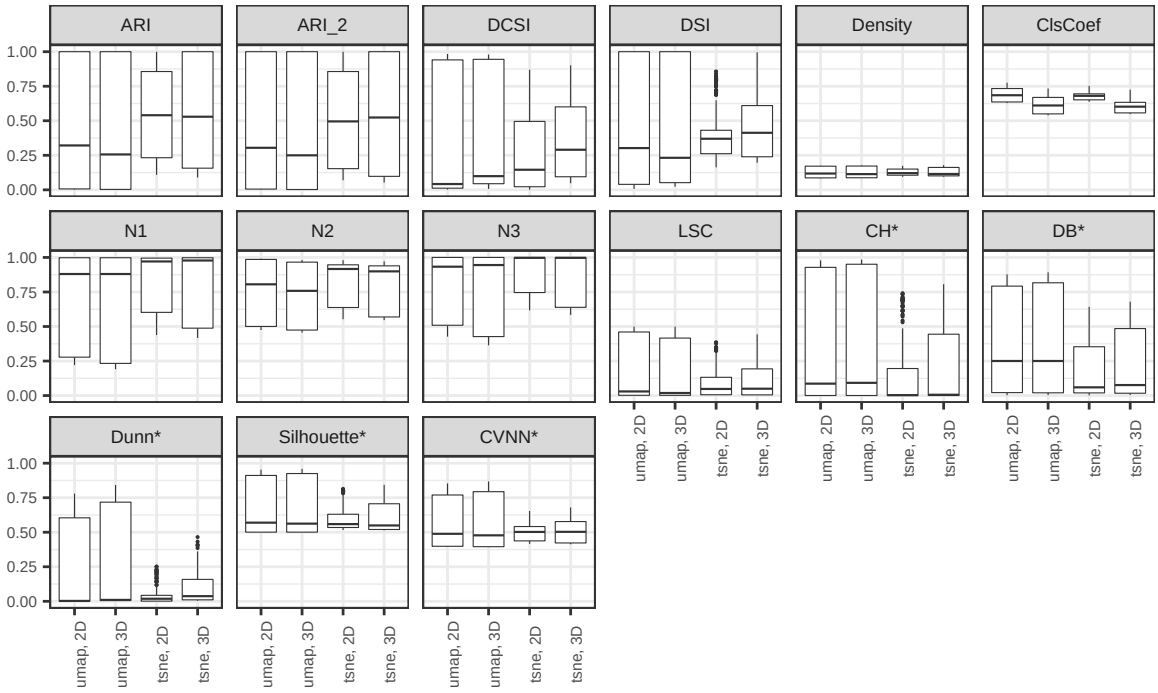

Figure 62: Results Experiment 9: performance and separability measures on 2-D and 3-D embeddings

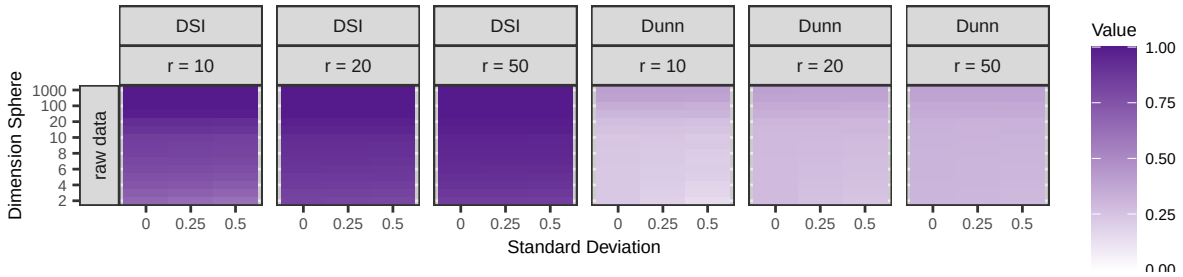

Figure 63: Results Experiment 9: heatmaps of some separability measures

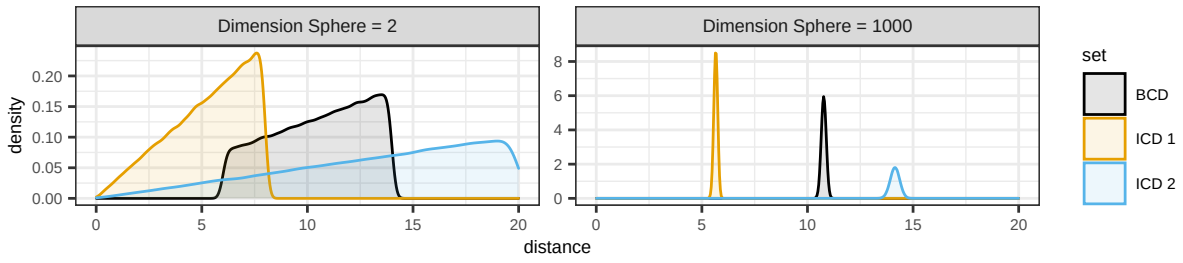

Figure 64: Results Experiment 9: examples of ICD and BCD sets (DSI) for r = 10, sd = 0

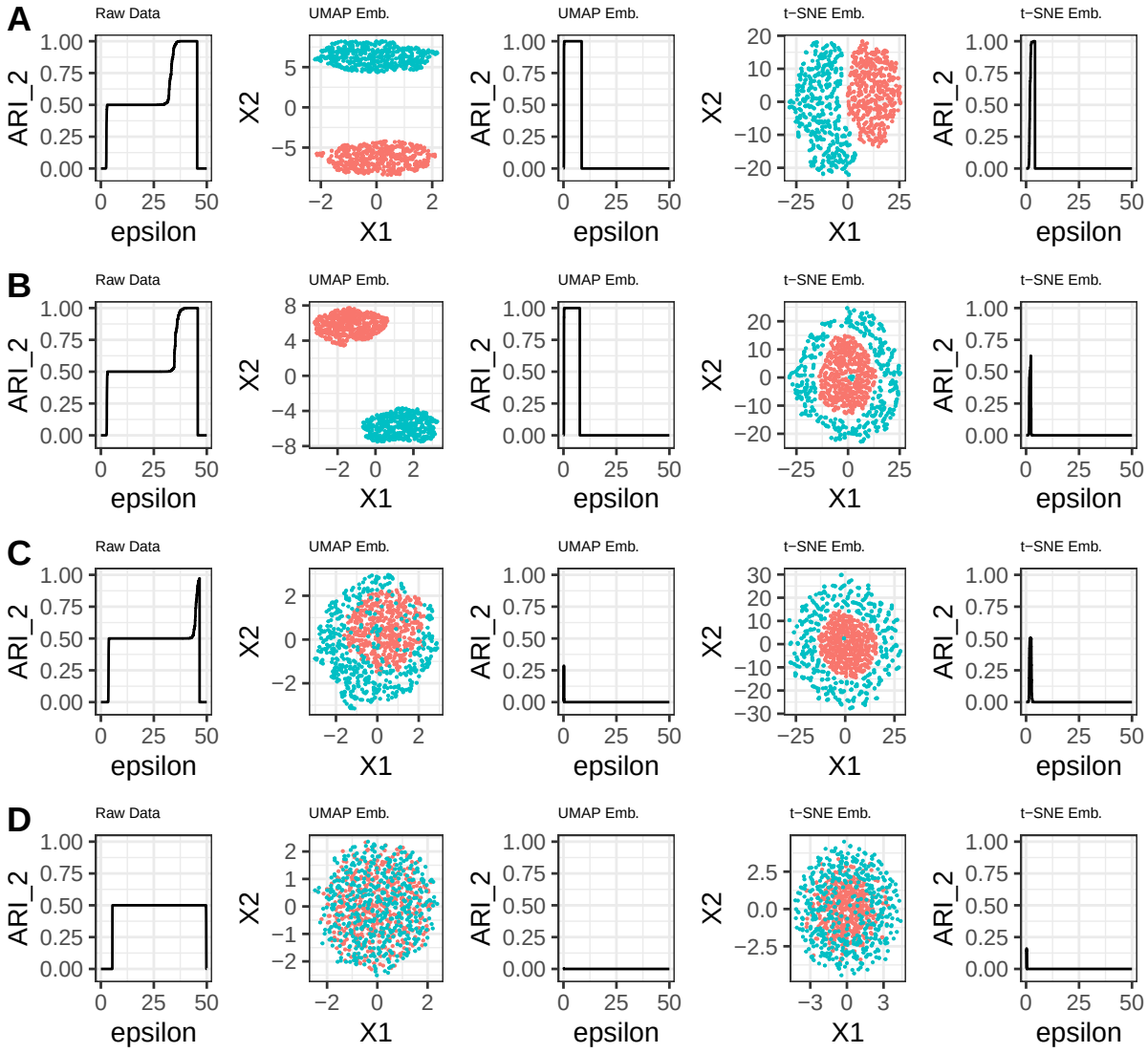

Figure 65: Results Experiment 9: embeddings and ARI on 7-, 8-, 15- and 1000-dimensional spheres

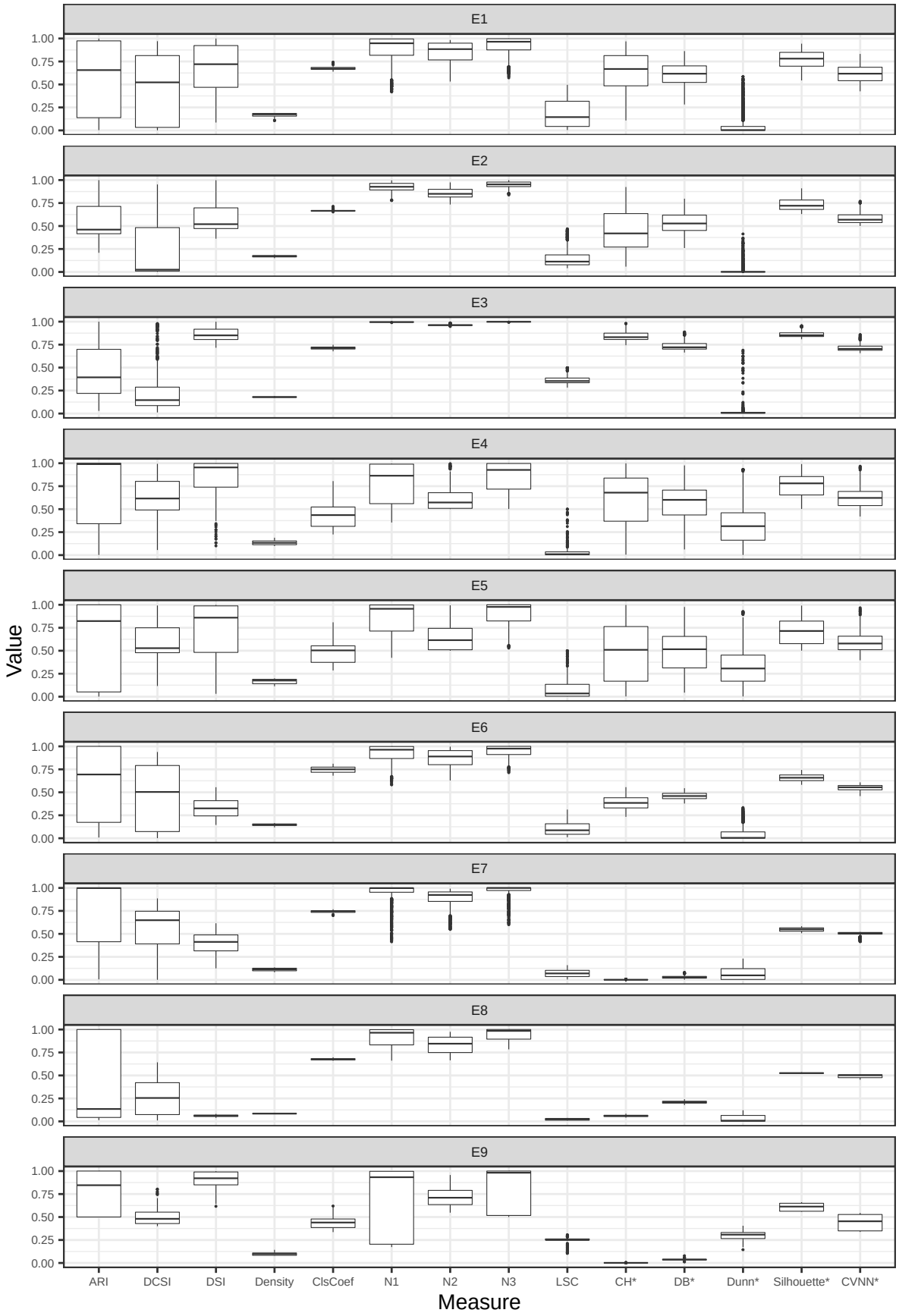

Figure 66: Summary experiments: boxplots of separability measures and ARI on raw data

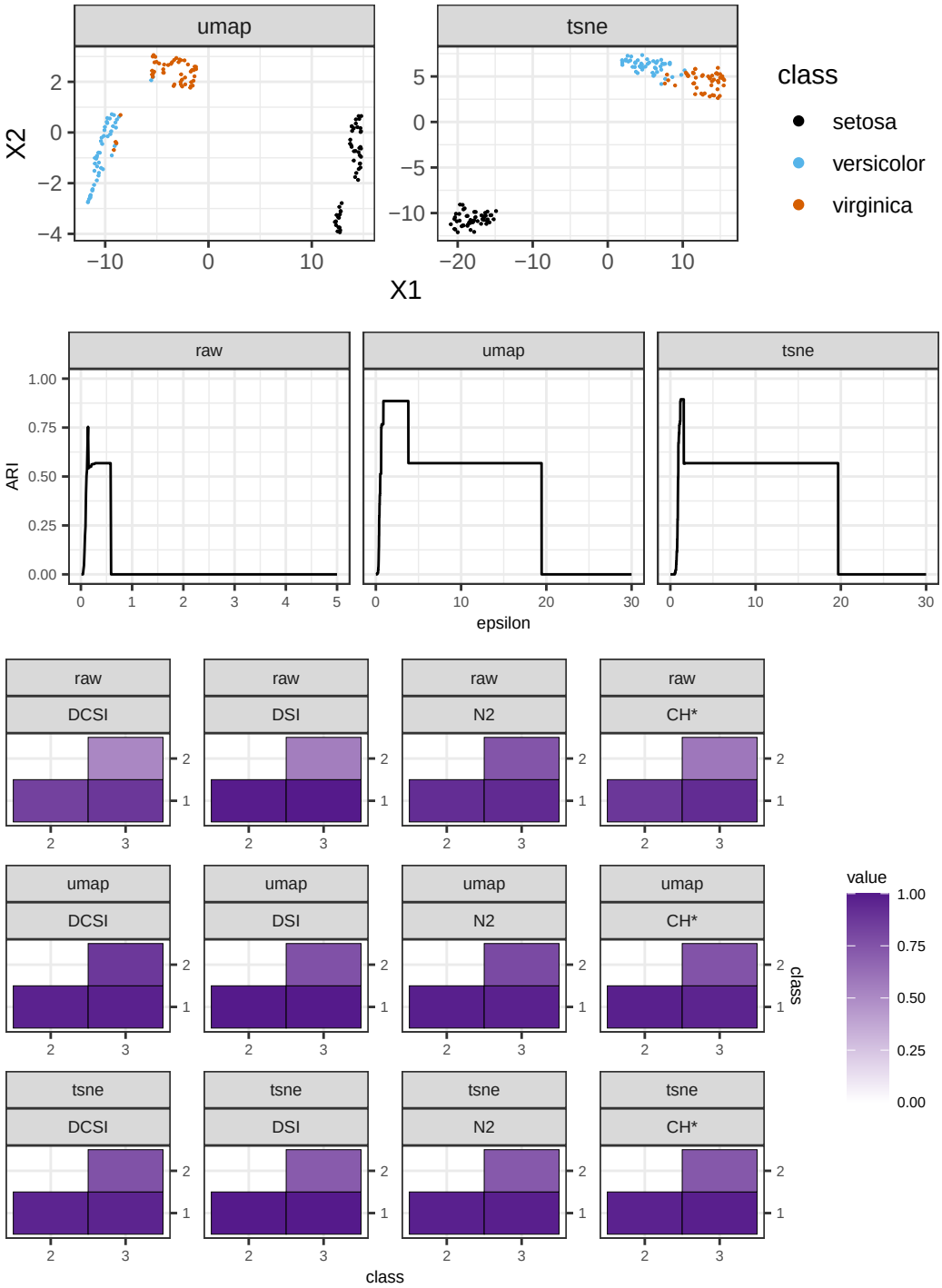

Figure 67: Experiments real-world data: results Iris

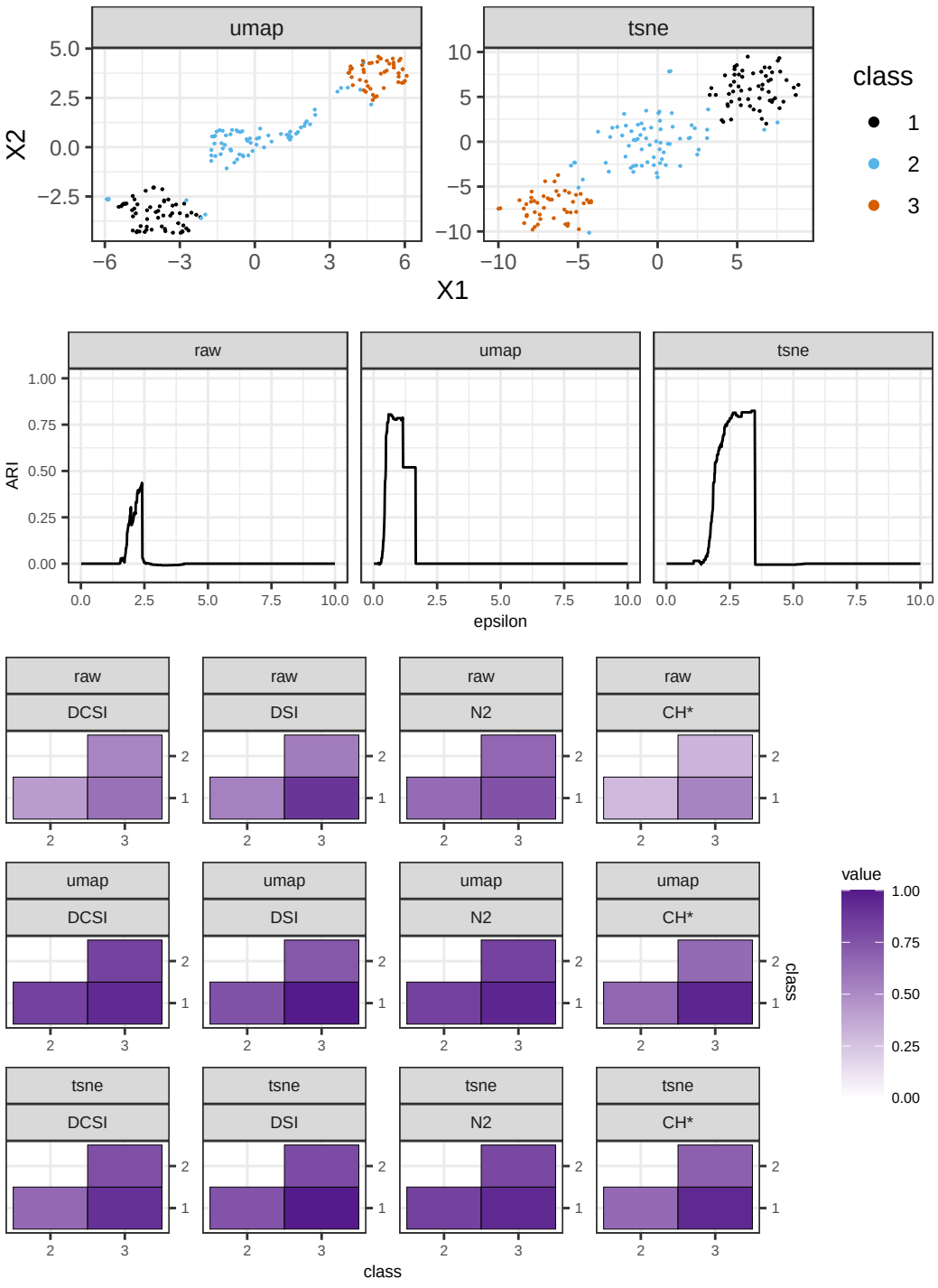

Figure 68: Experiments real-world data: results Wine

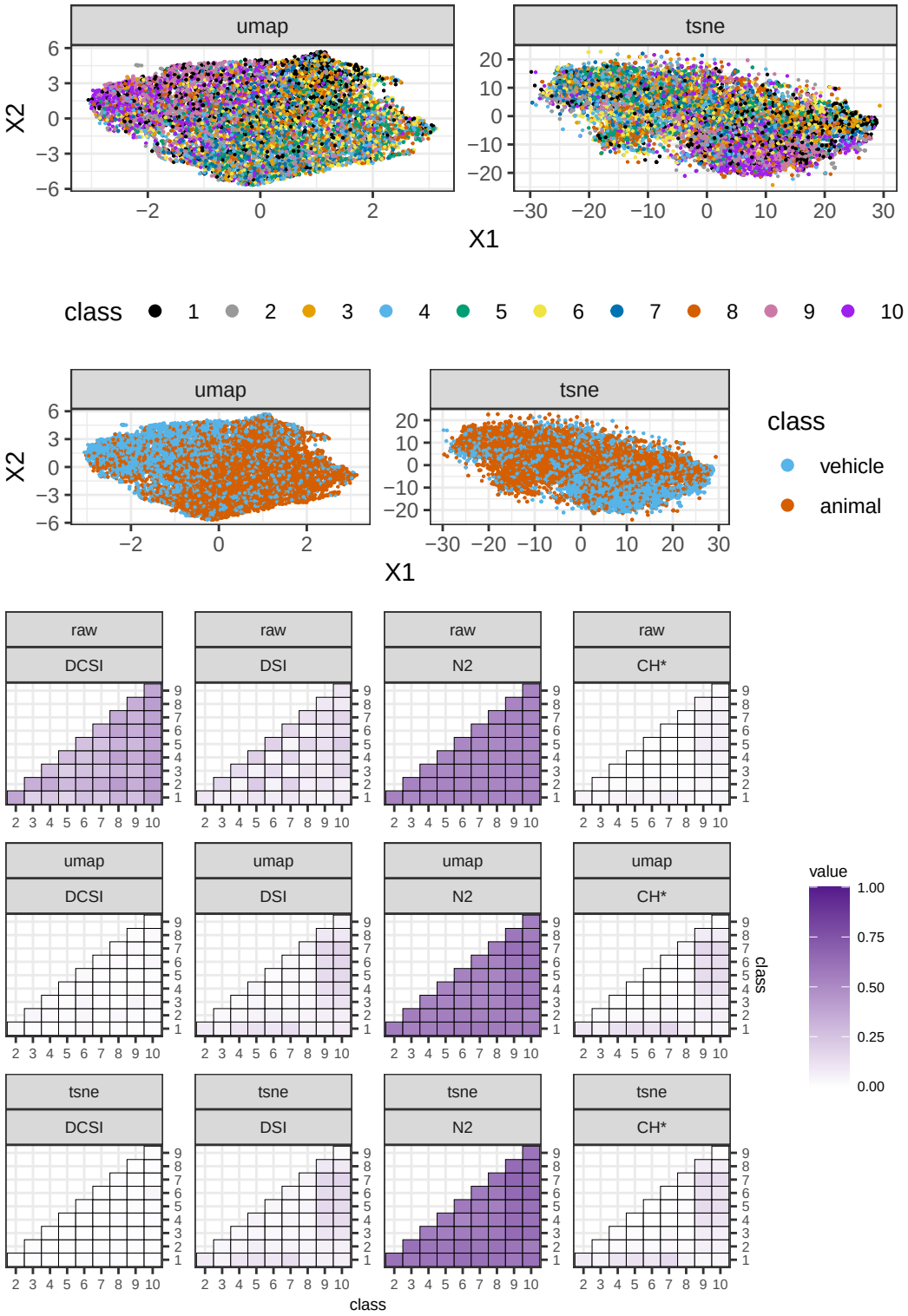

Figure 69: Experiments real-world data: results CIFAR

B Additional Tables

Table 4: Experiment 4: Separation, Connectedness and DCSI on exemplary data sets (r) and their embeddings (u, t)

dist	n_irrev	Sep r	Conn r	DCSI r	Sep u	Conn u	DCSI u	Sep t	Conn t	DCSI t
50	0	48.03	0.35	0.99	14.28	0.81	0.95	11.80	1.57	0.88
1.5	0	0.02	0.30	0.06	0.01	5.97	0.00	0.24	13.43	0.02
50	2000	50.55	17.78	0.74	18.32	0.27	0.99	14.50	0.82	0.95
1.5	2000	17.19	17.75	0.49	0.01	0.61	0.01	0.01	0.75	0.01

Table 5: Experiment 9: ARI_2 and DCSI on exemplary data sets with radius = 50, sd = 0.25

Dimension n-Sphere	ARI raw	ARI UMAP	ARI t-SNE	DCSI raw	DCSI UMAP	DCSI t-SNE
7	1.00	1.00	1.00	0.56	0.94	0.62
8	1.00	1.00	0.62	0.54	0.95	0.11
15	0.97	0.29	0.50	0.50	0.02	0.40
1000	0.50	0.00	0.16	0.42	0.01	0.02

References

Margareta Ackerman and Shai Ben-David. Clusterability: A theoretical study. 5:1–8, 16–18 Apr 2009. URL https://proceedings.mlr.press/v5/ackerman09a.html.

Margareta Ackerman, Shai Ben-David, and David Loker. Towards property-based classification of clustering paradigms. 23, 2010. URL https://proceedings.neurips.cc/paper/2010/file/f93882cbd8fc7fb794c1011d63be6fb6-Paper.pdf.

Andreas Adolfsson, Margareta Ackerman, and Naomi C. Brownstein. To cluster, or not to cluster: An analysis of clusterability methods. *Pattern Recognition*, 88:13–26, apr 2019. doi: 10.1016/j.patcog.2018.10.026. URL https://arxiv.org/pdf/1808.08317.pdf.

E. Anderson. The irises of the gaspe peninsula. *Bull. Am. Iris Soc.*, 59:2–5, 1935.

Shai Ben-David and Margareta Ackerman. Measures of clustering quality: A working set of axioms for clustering. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc., 2008. URL https://proceedings.neurips.cc/paper/2008/file/beed13602b9b0e6ecb5b568ff5058f07-Paper.pdf.

Gérard Biau, Frédéric Chazal, David Cohen-Steiner, Luc Devroye, and Carlos Rodríguez. A weighted k-nearest neighbor density estimate for geometric inference. *Electronic Journal of Statistics*, 5(none): 204 – 237, 2011. doi: 10.1214/11-EJS606. URL https://doi.org/10.1214/11-EJS606.

Jason Cory Brunson, Brandon Demkowicz, and Sanmati Choudhary. *tdaunif: Uniform Manifold Samplers for Topological Data Analysis*, 2020. URL https://CRAN.R-project.org/package=tdaunif. R package version 0.1.0.

T. Caliński and J Harabasz. A dendrite method for cluster analysis. *Communications in Statistics*, 3(1): 1–27, 1974. doi: 10.1080/03610927408827101. URL https://www.tandfonline.com/doi/abs/10.1080/03610927408827101.

Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 01 2009. doi: 10.1090/S0273-0979-09-01249-X. URL http://www.ams.org/journal-getitem?pii=S0273-0979-09-01249-X.

Lawrence Cayton. Algorithms for manifold learning. 2005. URL https://axon.cs.byu.edu/~martinez/classes/778/Papers/Manifold_Learning.pdf.

Frédéric Chazal and Bertrand Michel. An introduction to topological data analysis: fundamental and practical aspects for data scientists. *arXiv:1710.04019 [cs, math, stat]*, 02 2021. URL http://arxiv.org/abs/1710.04019. arXiv: 1710.04019.

Frédéric Chazal, David Cohen-Steiner, and Quentin Mérigot. Geometric inference for probability measures. *Foundations of Computational Mathematics*, 11(6):733–751, 12 2011. doi: 10.1007/s10208-011-9098-0. URL http://link.springer.com/10.1007/s10208-011-9098-0.

Frédéric Chazal, Brittany T. Fasy, Fabrizio Lecci, Bertrand Michel, Alessandro Rinaldo, and Larry Wasserman. Robust topological inference: Distance to a measure and kernel distance. 2014. doi: 10.48550/ARXIV.1412.7197. URL https://arxiv.org/abs/1412.7197.

Ayush Dalmia and Suzanna Sia. Clustering with umap: Why and how connectivity matters. 2021. doi: 10.48550/ARXIV.2108.05525. URL https://arxiv.org/abs/2108.05525.

David L. Davies and Donald W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227, 1979. doi: 10.1109/TPAMI.1979.4766909. URL https://ieeexplore.ieee.org/document/4766909.

Bernard Desgraupes. Clustering indices. 2016. URL https://cran.r-project.org/web/packages/clusterCrit/vignettes/clusterCrit.pdf.

Bernard Desgraupes. *clusterCrit: Clustering Indices*, 2018. URL https://CRAN.R-project.org/package=clusterCrit. R package version 1.2.8.

Dheeru Dua and Casey Graff. Uci machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

J. C. Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3(3):32–57, 1973. doi: 10.1080/01969727308546046. URL https://doi.org/10.1080/01969727308546046.

Herbert Edelsbrunner and John Harer. *Computational Topology: An Introduction*. Applied Mathematics. American Mathematical Society, 2010. URL https://www.maths.ed.ac.uk/~v1ranick/papers/edelcomp.pdf.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. page 226–231, 1996. URL https://dl.acm.org/doi/10.5555/3001460.3001507.

Brittany Terese Fasy, Jisu Kim, Fabrizio Lecci, and Clément Maria. Introduction to the R package TDA. 2014a. doi: 10.48550/ARXIV.1411.1830. URL https://arxiv.org/abs/1411.1830.

Brittany Terese Fasy, Fabrizio Lecci, Alessandro Rinaldo, Larry Wasserman, Sivaraman Balakrishnan, and Aarti Singh. Confidence sets for persistence diagrams. *The Annals of Statistics*, 42(6), 12 2014b. doi: 10.1214/14-AOS1252. URL http://arxiv.org/abs/1303.7117. arXiv:1303.7117 [cs, math, stat].

Brittany Terese Fasy, Jisu Kim, Fabrizio Lecci, Clement Maria, David L. Millman, and Vincent Rouvreau. *TDA: Statistical Tools for Topological Data Analysis*, 2022. URL https://CRAN.R-project.org/package=TDA. R package version 1.8.7.

Alberto Fernández, Salvador García, Mikel Galar, Ronaldo C. Prati, Bartosz Krawczyk, and Francisco Herrera. *Data Intrinsic Characteristics*, pages 253–277. Springer International Publishing, Cham, 2018. ISBN 978-3-319-98074-4. doi: 10.1007/978-3-319-98074-4_10. URL https://doi.org/10.1007/978-3-319-98074-4_10.

B. Fischer and J.M. Buhmann. Path-based clustering for grouping of smooth curves and texture segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(4):513–518, 2003. doi: 10.1109/TPAMI.2003.1190577. URL https://ieeexplore.ieee.org/document/1190577.

R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936. doi: https://doi.org/10.1111/j.1469-1809.1936.tb02137.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-1809.1936.tb02137.x.

Michele Forina, Riccardo Leardi, Armanino C, and Sergio Lanteri. *PARVUS: An Extendable Package of Programs for Data Exploration*. 01 1998. ISBN 0-444-43012-1.

Luis Garcia and Ana Lorena. *ECoL: Complexity Measures for Supervised Problems*, 2019. URL https://CRAN.R-project.org/package=ECoL. R package version 0.3.0.

Arka P. Ghosh, Ranjan Maitra, and Anna D. Peterson. A separability index for distance-based clustering and classification algorithms. 2010.

J. C. Gower. A general coefficient of similarity and some of its properties. *Biometrics*, 27(4):857–871, 1971. ISSN 0006341X, 15410420. URL http://www.jstor.org/stable/2528823.

Shuyue Guan and Murray Loew. An internal cluster validity index using a distance-based separability measure. In *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 827–834, 2020. doi: 10.1109/ICTAI50040.2020.00131. URL https://ieeexplore.ieee.org/document/9288314.

Shuyue Guan and Murray Loew. A novel intrinsic measure of data separability, 2021. URL https://arxiv.org/abs/2109.05180.

Shuyue Guan, Murray Loew, and Hanseok Ko. Data separability for neural network classifiers and the development of a separability index, 2020. URL https://arxiv.org/abs/2005.13120.

Michael Hahsler, Matthew Piekenbrock, and Derek Doran. dbscan: Fast density-based clustering with R. *Journal of Statistical Software*, 91(1):1–30, 2019a. doi: 10.18637/jss.v091.i01. URL https://cran.r-project.org/web/packages/dbscan/index.html.

Michael Hahsler, Matthew Piekenbrock, and Derek Doran. dbscan: Fast density-based clustering with *r*. *Journal of Statistical Software*, 91(1), 2019b. doi: 10.18637/jss.v091.i01. URL http://www.jstatsoft.org/v91/i01/.

Christian Hennig. What are the true clusters? 2015. doi: 10.48550/ARXIV.1502.02555. URL https://arxiv.org/abs/1502.02555.

Moritz Herrmann. *Towards more reliable machine learning: conceptual insights and practical approaches for unsupervised manifold learning and supervised benchmark studies*. PhD thesis, 2022.

Moritz Herrmann, Daniyal Kazempour, Fabian Scheipl, and Peer Kröger. Enhancing cluster analysis via topological manifold learning. 2022. doi: 10.48550/ARXIV.2207.00510. URL https://arxiv.org/abs/2207.00510.

Geoffrey E Hinton and Sam Roweis. Stochastic neighbor embedding. 15, 2002. URL https://proceedings.neurips.cc/paper/2002/file/6150ccc6069bea6b5716254057a194ef-Paper.pdf.

Tin Kam Ho and M. Basu. Complexity measures of supervised classification problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):289–300, 2002. URL https://ieeexplore.ieee.org/document/990132.

Lianyu Hu and Caiming Zhong. An internal validity index based on density-involved distance. *IEEE Access*, 7:40038–40051, 2019. doi: 10.1109/ACCESS.2019.2906949. URL https://ieeexplore.ieee.org/document/8672850.

Lawrence J. Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2:193–218, 1985. URL https://link.springer.com/article/10.1007/BF01908075.

A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review. *ACM Comput. Surv.*, 31(3):264–323, sep 1999. ISSN 0360-0300. doi: 10.1145/331499.331504. URL https://doi.org/10.1145/331499.331504.

Jon Kleinberg. An impossibility theorem for clustering. 15, 2002. URL https://proceedings.neurips.cc/paper/2002/file/43e4e6a6f341e00671e123714de019a8-Paper.pdf.

Dmitry Kobak and George Linderman. Initialization is critical for preserving global data structure in both t-SNE and UMAP. *Nature Biotechnology*, 39, 2021. doi: 10.1038/s41587-020-00809-z. URL https://www.nature.com/articles/s41587-020-00809-z.

Tomasz Konopka. *umap: Uniform Manifold Approximation and Projection*, 2022. URL https://CRAN.R-project.org/package=umap. R package version 0.2.9.0.

Jesse H. Krijthe. *Rtsne: T-Distributed Stochastic Neighbor Embedding using Barnes-Hut Implementation*, 2015. URL https://github.com/jkrijthe/Rtsne. R package version 0.16.

Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. URL https://www.cs.toronto.edu/~kriz/cifar.html.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791. URL http://yann.lecun.com/exdb/mnist/.

Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991. doi: 10.1109/18.61115. URL https://ieeexplore.ieee.org/document/61115.

George C. Linderman and Stefan Steinerberger. Clustering with t-SNE, provably. 2017. doi: 10.48550/ARXIV.1706.02582. URL https://arxiv.org/abs/1706.02582.

Yanchi Liu, Zhongmou Li, Hui Xiong, Xuedong Gao, Junjie Wu, and Sen Wu. Understanding and enhancement of internal clustering validation measures. *IEEE Transactions on Cybernetics*, 43(3):982–994, 2013. doi: 10.1109/TSMCB.2012.2220543. URL https://ieeexplore.ieee.org/document/6341117.

Ana C. Lorena, Luís P. F. Garcia, Jens Lehmann, Marcilio C. P. Souto, and Tin Kam Ho. How complex is your classification problem? A survey on measuring classification complexity. *ACM Comput. Surv.*, 52(5), sep 2019. ISSN 0360-0300. doi: 10.1145/3347711. URL https://doi.org/10.1145/3347711.

Leland McInnes. How UMAP Works — umap 0.5 documentation. https://umap-learn.readthedocs.io/en/latest/how_umap_works.html, 2018. (Accessed on 2022-08-19).

Leland McInnes, John Healy, and James Melville. UMAP: Uniform manifold approximation and projection for dimension reduction. 2018. doi: 10.48550/ARXIV.1802.03426. URL https://arxiv.org/abs/1802.03426.

Linda Mthembu and John Greene. A comparison of three class separability measures. 2004. URL https://open.uct.ac.za/handle/11427/24145.

Linda Mthembu and Tshilidzi Marwala. A note on the separability index, 2008. URL https://arxiv.org/abs/0812.1107.

Sudipto Mukherjee, Himanshu Asnani, Eugene Lin, and Sreeram Kannan. ClusterGAN: Latent space clustering in generative adversarial networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):4610–4617, Jul. 2019. doi: 10.1609/aaai.v33i01.33014610. URL https://ojs.aaai.org/index.php/AAAI/article/view/4385.

Chuang Niu, Hongming Shan, and Ge Wang. SPICE: Semantic pseudo-labeling for image clustering, 2021. URL https://arxiv.org/abs/2103.09382.

P. Niyogi, Stephen Smale, and Shmuel Weinberger. Finding the homology of submanifolds with high confidence from random samples. *Discrete & Computational Geometry*, 39(1-3):419–441, 03 2008. doi: 10.1007/s00454-008-9053-2. URL http://link.springer.com/10.1007/s00454-008-9053-2.

P. Niyogi, S. Smale, and S. Weinberger. A topological view of unsupervised learning from noisy data. *SIAM Journal on Computing*, 40(3):646–663, 01 2011. doi: 10.1137/090762932. URL http://epubs.siam.org/doi/10.1137/090762932.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2021. URL https://www.R-project.org/.

Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987. ISSN 0377-0427. doi: https://doi.org/10.1016/0377-0427(87)90125-7. URL https://www.sciencedirect.com/science/article/pii/0377042787901257.

Amit Saxena, Mukesh Prasad, Akshansh Gupta, Neha Bharill, Om Prakash Patel, Aruna Tiwari, Meng Joo Er, Weiping Ding, and Chin-Teng Lin. A review of clustering techniques and developments. *Neurocomputing*, 267:664–681, 2017. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2017.06.053. URL https://www.sciencedirect.com/science/article/pii/S0925231217311815.

Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN. *ACM Transactions on Database Systems*, 42(3):1–21, 08 2017. doi: 10.1145/3068335. URL https://dl.acm.org/doi/10.1145/3068335.

Chris Thornton. Separability is a learner's best friend. In *4th Neural Computation and Psychology Workshop, London, 9–11 April 1997*, pages 40–46, London, 1998. Springer London. ISBN 978-1-4471-1546-5. URL https://link.springer.com/chapter/10.1007/978-1-4471-1546-5_4.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL http://jmlr.org/papers/v9/vandermaaten08a.html.

Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Is a correction for chance necessary? ICML '09, page 1073–1080, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605585161. doi: 10.1145/1553374.1553511. URL https://doi.org/10.1145/1553374.1553511.

Ulrike von Luxburg. A tutorial on spectral clustering. 2007. doi: 10.48550/ARXIV.0711.0189. URL https://arxiv.org/abs/0711.0189.

Larry Wasserman. Topological data analysis. *Annual Review of Statistics and Its Application*, 5(1):501–532, 2018. doi: 10.1146/annurev-statistics-031017-100045. URL https://doi.org/10.1146/annurev-statistics-031017-100045. _eprint: https://doi.org/10.1146/annurev-statistics-031017-100045.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017. URL https://arxiv.org/abs/1708.07747.

Zhirong Yang, Yuwei Chen, and Jukka Corander. t-SNE is not optimized to reveal clusters in data. *ArXiv*, abs/2110.02573, 2021. URL https://arxiv.org/pdf/2110.02573v1.pdf.

Caiming Zhong, Duoqian Miao, and Ruizhi Wang. A graph-theoretical clustering method based on two rounds of minimum spanning trees. *Pattern Recognition*, 43(3):752–766, 2010. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2009.07.010. URL https://www.sciencedirect.com/science/article/pii/S0031320309002945.

Djamel A. Zighed, Stéphane Lallich, and Fabrice Muhlenbach. A statistical approach to class separability: Research articles. *Appl. Stoch. Model. Bus. Ind.*, 21(2):187–197, mar 2005. ISSN 1524-1904. URL https://hal.archives-ouvertes.fr/hal-00383773/document.

Arthur Zimek and Jilles Vreeken. The blind men and the elephant: on meeting the problem of multiple truths in data from clustering and pattern mining perspectives. *Machine Learning*, 98, 01 2013. doi: 10.1007/s10994-013-5334-y. URL https://link.springer.com/article/10.1007/s10994-013-5334-y.

Declaration of authorship

I hereby declare that the thesis submitted is my own unaided work. All direct or indirect sources used are acknowledged as references. I am aware that the thesis in digital form can be examined for the use of unauthorized aid and in order to determine whether the report as a whole or parts incorporated in it may be deemed as plagiarism. For the comparison of my work with existing sources I agree that it shall be entered in a database where it shall also remain after examination, to enable comparison with future theses submitted. Further rights of reproduction and usage, however, are not granted here. This thesis was not previously presented to another examination board and has not been published.

Munich, 2 November, 2022 Jana Gauß