USING nD-LAPLACE TO TRAIN PRIVACY-PRESERVING CLUSTER ALGORITHMS ON DISTRIBUTED N-DIMENSIONAL DATA

by

Tjibbe van der Ende

in partial fulfillment of the requirements for the degree of

Master of Science
in Software Engineering

at the Open University, faculty of Management, Science and Technology
Master Software Engineering
to be defended publicly on Day Month DD, YYYY at HH:00 PM.

Student number:	852372917	
Course code:	IMA0002	
Thesis committee:	Dr. Ir. Clara Maathuis (chairman),	Open University
	Dr. Ir. Mina Sheikhalishahi (supervisor),	Open University

Contents

1

INTRODUCTION

1.1. RESEARCH QUESTIONS

Main question:
How can the nD-Laplace algorithm be applied in training privacy-preserving clustering algorithms on distributed n-dimensional data?

1. RQ1: How can 2D-Laplace be used to protect the data privacy of 2-dimensional data which is employed for training clustering algorithms?
2. RQ2: How can 3D-Laplace be extended to protect the data privacy of n-dimensional data which is employed for training clustering algorithms?
3. RQ3: What is the impact of different privacy budgets, dataset properties, and other clustering algorithms on the research conducted for research question 2?

2

LITERATURE REVIEW

This chapter lays out the theoretical foundation of this work. To review the past literature, it is first necessary to gather the required knowledge for it.

2.1. DIFFERENTIAL PRIVACY

Explain general notion of privacy

MATH SYMBOLS

$K(x)(Z)$ Randomization method for $x \in X$ and output $z \in Z$..

$Pr(K(x_i) \in (Z))$ Probability of reporting $x \in X$ for $z \in Z$.

X Set of locations for a user. (R^2).

Z For every $x \in X$ a perturbed location $z \in Z$ is reported..

ϵ Privacy budget.

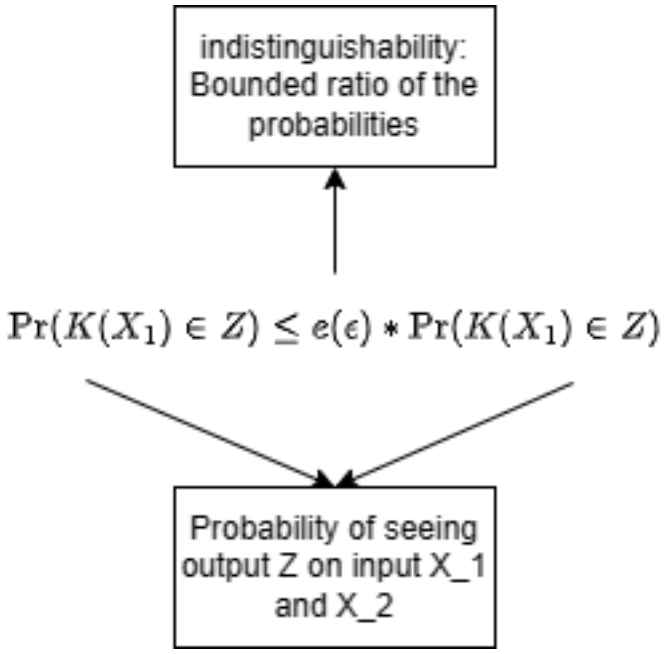

Figure 2.1: Randomization function K gives ϵ-differential privacy for all elements in D_1 and D_2 if they differ at most one element. [Dwork, 2006]

The privacy budget ϵ determines the amount of noise that is added.

2.1.1. LAPLACE ALGORITHM

One way to achieve ϵ-DP is using sampling noise from the Laplace or Gaussian distributions.

Explain gaussian / laplace distributions

The noise is then based on the sensitivity of a function f. This is the maximal possible change when adding or removing a single record [Dwork, 2006; Friedman and Schuster, 2010].

$$\Delta f = max_{D_1,D_2} ||f(D_1) - f(D_2)|| \tag{2.1}$$

Explain differential privacy implementation with La place distribution

2.1.2. Local differential privacy
2.1.3. Geo-indistinguishability
2.1.4. Evaluation methods

It is possible to evaluate and measure the impact of the noise between two distributions by calculating the error between the non-private and private data [Xiong et al., 2020]. Two metrics that are proposed by the same study are Mean Squared Error (MSE) and Mean Average Error (MAE). These metrics can be used to calculate the error between X and the perturbed dataset Z.

Just as it is possible to measure the utility, this can also be done with privacy. When performing a privacy algorithm, it can be proven whether a method meets the privacy requirements. These are metrics such as ϵ-differential-privacy (2.1) and ϵ-geo-indistinguishability (see next chapter 3.1). Although these methods give an idea of privacy, it can only be "yes" or "no". Furthermore, it can give a distorted image, since a chance of 70% also gives a "yes" according to the definition of geo-indistinguishability [Oya et al., 2017]. In other words, to gain more insight into the amount of privacy (such as with MSE or MAE), other metrics are needed.

For this reason, Oya et al. introduced a metric for geo-indistinguishability that makes it possible to give percentages in their study [Oya et al., 2017]: As an example, an adversary is given that guesses between two locations: $x \in X$ and $x' \in X$.

$$p_e(x, x', z) \leq p_e^* = \frac{1}{1 + e^{e * d(x,x')}} \tag{2.2}$$

Where privacy level p_e^* is the lower bound of the probability of an adversary guessing correctly. The method is called ϵ-geo-indistinguishability as error. Based on this metric, it can be calculated that an adversary has an average of 90% chance to guess a location correctly. In that case, the algorithm would be ϵ-geo-indistinguishability, but in practice not.

2.2. CLUSTERING

2.2.1. METHODS

In this chapter, a brief description is given of each clustering algorithm on how it works. The different parameters for the algorithm are also highlighted.

K-MEANS

The K-Means algorithm is based on the algorithm of Loyd et al. . The starting point is determined randomly by choosing k number of centroids. Each point is then assigned to a centroid based on the Euclidean distance. A new centroid is then determined based on the cluster average. This is repeated until the given number of iterations is reached or until the results are stable.

Source

Choosing K: The most important parameter of the K-Means algorithm is the value of k. This value determines the number of clusters to consider and has a big influence on the results [Ahmed et al., 2020].

The first method is called an "elbow" plot [Kodinariya and Makwana, 2013]. This method can be used to determine the best k by applying the algorithm multiple times and estimating the best k.

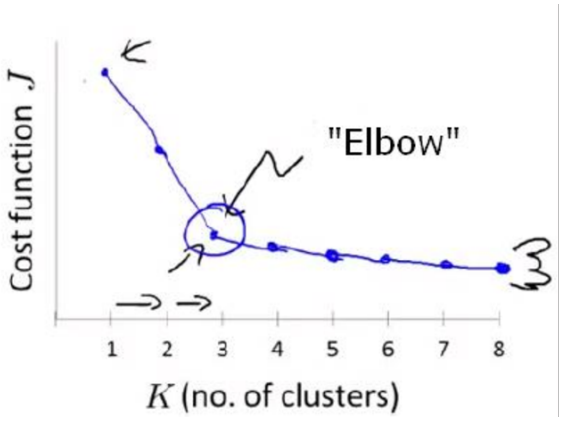

Figure 2.2: Illustration of determining k using the "elbow" method [Kodinariya and Makwana, 2013]

In this situation, a Silhouette plot can be used to determine the k. This method uses calculates the Silhouette coefficient for each cluster [SAPUTRA et al., 2020]. It takes into consideration the separation and cohesion of the cluster. The plot can then be made by plotting the silhouette coefficient on the y-axis and k on the x-axis [SAPUTRA et al., 2020]. Then the K with the highest coefficient can be selected based on that.
Another popular method is the Gap statistic method [Yuan and Yang, 2019]. It compares the total within-cluster variation for different values of k with their expected values under a null reference distribution of the data [Tibshirani et al., 2001]. A practical appliance of this method uses a line plot for comparing the k-value and gap value [Yuan and Yang, 2019]. Based on the visual change of the line, someone is be-able to select the best k.

All in all, there is no fixed method to choose a good k for K-Means. The elbow method is common in the existing literature and is very popular due to its simplicity. However, one disadvantage is that it can be difficult to determine the "elbow" point, as it is not always present [Kodinariya and Makwana, 2013]. In that case, the silhouette method or gap statistic method can be chosen, with the algorithm for silhouette being the most obvious choice due to its simplicity.

Affinity Propagation

Affinity Propogation (AP) is an algorithm that clusters data points by iteratively passing messages between them. Each point sends and receives messages about the attractiveness of other points as cluster centers (exemplars) and the suitability of itself as a center [Keller et al., 2021]. The method was introduced by Frey et al. and does not require any hyperparameters [Frey and Dueck, 2007]. Still, there are important properties that could potentially impact the clustering [Wang et al., 2007].

Choosing preference(p): Indicates the preference that a data point is selected as cluster center [Wang et al., 2007]. It highly influences the number of clusters, a high one would lead to more clusters and a small one to less [Moiane and Machado, 2018]. A good choice depending on the data is to set the p to the median of all data similarities [Wang et al., 2007]. But, the effectiveness of this could highly be influenced based on the dataset. To validate if the preference is correctly set, it is possible to analyze the silhouette coefficient [Moiane and Machado, 2018].

Choosing damping factor($lam \in [0,1]$): The damping factor is used to improve the stability (convergence) of the algorithm [Wang et al., 2007]. By default, this value is 0.5 and can be increased to 1 to reduce the impact of numerical oscillations. This can be applied manually, by re-running the algorithm and finding the optimal or just being increased. However, both approaches take a lot of time, especially for bigger datasets [Wang et al., 2007].

To conclude on this, damping is important if big datasets are considered. However, this research does not use large datasets or consider time complexity as a metric. The preference on the other hand could influence the results a lot. For this, the silhouette coefficient can be evaluated to choose the best option. In general, it should be sufficient to take the median.

DBSCAN

Density-based spatial clustering of applications with noise (DBSCAN) was introduced by [Ester et al.] and works by drawing a radius (neighborhood) around data points. It then groups all points within this radius as clusters. The main advantage is its ability to find arbitrarily shaped clusters and detect outliers [Liu et al., 2012]. To do this, the DBSCAN algorithm uses the inputs $minPts$, $radius(\epsilon)$ and a distance function [Schubert et al., 2017]. The ϵ is used to draw a neighborhood and the $minPts$ is used as a weight to evaluate which points should be inside the neighborhood. For the distance function, the Euclidean distance is used, to be consistent with the other algorithms.

Choosing radius(ϵ): The desired ϵ can be calculated using the K-NearestNeighbours algorithm [Ester et al.; Schubert et al., 2017]. The general approach for this is to choose a $K = 2 * N - 1$ (where N is the number of features) and plot the distance for each point. This can then be plotted using a k-dist plot and the best "elbow" can be chosen for deciding the ϵ (similar to choosing the k for K-means) [Elbatta and Ashour, 2013].

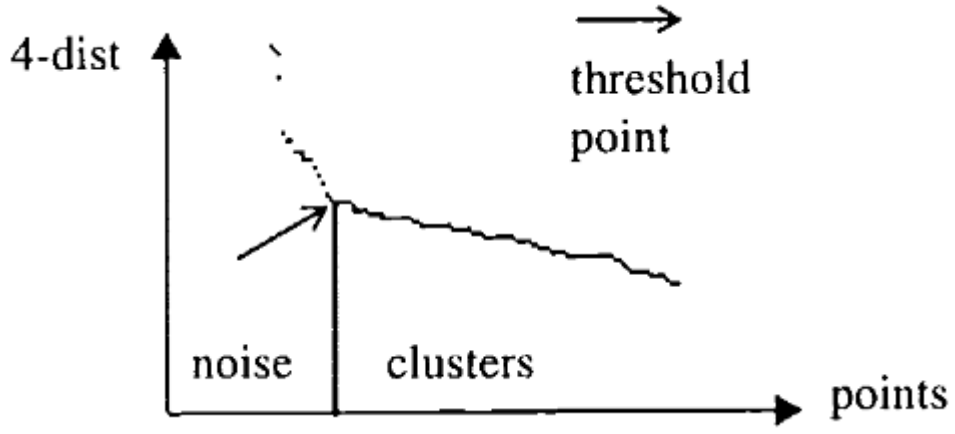

Figure 2.3: K-dist plot example for $minPts = 4$ based on a 2-dimensional dataset [Ester et al.]

Choosing minimum points ($minPts$): This hyperparameter is considered by a paper written by Sander et al. The work describes a way of calculating this parameter by doing two times the feature amount [Sander et al., 1998]. So, using this approach a dataset with two features will have an $minPts$ of four. This is confirmed by Schubert et al. to use the default $minPts = 4$ for a 2-dimensional dataset [Schubert et al., 2017].

In conclusion, the parameters of DBSCAN are a little harder to estimate than for AP. But, as with K-means for k there exists a selection method the ϵ. Although DBSCAN requires the most hyperparameters out of the three clustering algorithms, determining them is clearer than for K-means. There are clear methods for determining both the $minPts$ and ϵ.

Overfitting

Overfitting is dangerous for machine learning models and one of the biggest mistakes with training machine learning models [Demšar and Zupan, 2021]. It occurs when a machine learning model is trained on samples that are not representative of future test data [Bashir et al., 2020]. A common mistake is to evaluate a model, using the same data as it is trained on [Demšar and Zupan, 2021]. The model appears to have high accuracy but just memorized the properties of a training dataset. Another cause of overfitting can be the size of the training data [Valdenegro-Toro and Sabatelli, 2022].

To measure if a model overfits, it is necessary to compute the generalization gap [Valdenegro-Toro and Sabatelli, 2022].

$$L_{gap} = L_{val} - L_{train} \tag{2.3}$$

Where $Lval$ and L_{train} are the validation and training splits of the dataset. For splitting a dataset, a common approach is to have 50% training data, 30% for validation and 20% test [Chicco, 2017]. In addition to this, if a dataset is too small cross-validation can be considered.

Needs more work for unsupervised / cluster problems

In summary:

1. It is necessary to evaluate a machine learning model on a dataset that was not used to train the model.
2. Use a representative training dataset, to work well on unseen data.
3. To reduce the chance of overfitting for supervised learning, it is a good practice to validate using a 30% subset.

2.2.2. Evaluation methods

Clustering comparison measures are important in cluster analysis for external validation by comparing clustering solutions to a "ground truth" clustering [Vinh et al.]. These external validity indices are a common way to assess the quality of unsupervised machine learning methods like clustering [Warrens and van der Hoef, 2022]. A method that could be used for this is the Rand Index [Rand, 1971]. It is a commonly applied method for comparing two different cluster algorithms [Wagner and Wagner]. An improvement of this method is adjusted for chance by considering the similarity of pairwise cluster comparisons [Vinh et al.]. Both the Rand Index (RI) and Adjusted Rand Index (ARI) [Hubert and Arabie, 1985] report a value between 0 and 1. Where 0 is for no-similarity and 1 for identical clusters. Alternatives for RI are the Fowles-Mallows Index and Mirkin Metric. However, these two methods have their disadvantages. Respectively, being sensitive to a few clusters and cluster sizes [Wagner and Wagner]. The ARI metric suffers from cluster size imbalance as well, so it only provides not a lot of information on smaller clusters [Warrens and van der Hoef, 2022]. Instead, they recommend using the cluster index metric that was proposed by Fränti et al. [Fränti et al., 2014].

Another popular group of methods is the information theoric-based measures [Vinh et al.]. This metric measures the information between centroids; the higher the value, the better [Vinh et al.]. Mutual Information (MI) is such metric, which calculates the probability of an element belonging to cluster C or C'. But, is not easy to interpret as it does not

have a maximum value [Wagner and Wagner]. To this end, Normalized Mutual Information (NMI) can be used to report a value between 0 and 1 using the geometric mean [Strehl and Ghosh, 2002]. The metric exists also in an adjusted version as Adjusted Mutual Information (AMI). This works in the same way as for the Adjusted Rank Index (ARI) and is mostly needed if the number of data items is small in comparison to the number of clusters [Vinh et al.].

Besides the external validity measurements for clustering, it is also possible to use internal validation methods. These metrics focus entirely on the intrinsic dataset properties, instead of relying on an external baseline cluster algorithm Craenendonck and Blockeel. Assessing two important concepts of clustering: compactness and separation Hassani and Seidl [2017]. Both studies, consider three different metrics and measure both concepts at the same time Hassani and Seidl [2017]:

1. Calinski-Harabasz Index (CHI) [Caliński and Harabasz, 1974] is used to measure the cluster variance (well-separated clusters) and low variance within the clusters (tightly coupled data). A high score indicates better clustering.

2. Silhouette Index [Rousseeuw, 1987] this metric is similar, by also measuring cohesion within clusters and separation of clusters. However, this metric uses the pairwise distance Hassani and Seidl [2017]. A score of -1 indicates incorrect clustering and +1 for dense clusters Rousseeuw [1987].

3. Davies-Bouldin [Davies and Bouldin, 1979] uses the average distance between centroids. A lower score indicates good clustering.

K-Means scores relatively high for CHI [Craenendonck and Blockeel; Hassani and Seidl, 2017] and SI [Craenendonck and Blockeel]. The same applies to DBSCAN, which scores relatively high on SI and DB due to the sensitivity of noise Craenendonck and Blockeel.

Existing literature

Comparable studies with differential privacy use external validation [Sun et al., 2022; Xia et al., 2020]. Their experiment setup uses a so-called non-private cluster algorithm as external validation. This cluster algorithm is trained without the perturbed data and compared with the same clustering algorithm that is trained with perturbed data. Thus, the non-private variant functions as an external validation by providing the ground truth.

They compare the mutual information between a baseline cluster algorithm using AMI [Huang et al., 2021] or NMI [Sun et al., 2022; Xia et al., 2020]. Another study for evaluating Differential Privacy (DP) with AP uses both ARI and AMI. In addition to mutual information and rand index scores, it is also not uncommon to calculate the error between the two cluster algorithm's centroids [Huang et al., 2021; Xia et al., 2020]. These two studies used Relative Error (RE) for this.

2.3. LITERATURE REVIEW

Mostly based on the preparation, and summarized here later

3

nD-Laplace

3.1. 2D-Laplace

The theory for this subject is heavily inspired by the paper that was written by Andrés et al. [Andrés et al., 2012]. This notion of Geo-indistinguishability (GI) was introduced to solve the issue of privacy and location data. It offers an alternative approach for differential privacy by adding noise to the location locally before sending it to a location-based system (LBS) like Google maps. This section starts with an introduction to mathematics for the planar and polar Laplace algorithm. For each of the different subsections, we visualize and explain open challenges and theoretic for applying them for clustering.

Math symbols

θ Angle.

l Privacy level.

r Radius.

The other symbols can be found in section 2.1.

Geo-indistinguishability

As mentioned in the previous section, the GI method can be applied to preserve the privacy using a differential privacy method specific to spatial data. The formula to measure if an algorithm preserves ϵ-geo-indistinguishability can be expressed as [Andrés et al., 2012]:

$$K(x)(y) \leq e^{\epsilon * d(x,x')} K(x')(y) \tag{3.1}$$

Where K is a probability method reporting $x, x' \in X$ as $z \in Z$. The idea of this algorithm looks a lot like that of differential privacy using the La Place method; but includes distance. The intuition for this is that it displays the distinguishability level between two secret locations/points x and x' [Chatzikokolakis et al., 2015]. An extension of this is called d_x-privacy and is a more general notation of distance-aware differential privacy. Their definition for GI is, therefore, d_2-privacy, but is essentially the same as the proof provided for GI.

3.1.1. Planar and polar Laplace

The idea of planar Laplace is to generate an area around $x0 \in X$ according to the multivariate Laplace distribution. The mechanism of planar Laplace is a modification of the Laplace algorithm to support distance [Andrés et al., 2012]. This distance method $dist(x, x')$ is defined as the Euclidean distance between two points or sets. Recalling the definition of Laplace, this method $|x - x'|$ is replaced by the distance metric. Hence, the definition of the Probability Density Function (pdf) by Andrés et al. is:

$$\frac{\epsilon^2}{2 * \pi} e(-\epsilon d(x_0, x)) \tag{3.2}$$

Which is the likelihood a generated point $z \in Z$ is close to $x0$. The method works for Cartesian coordinates but was modified to support polar coordinates by including θ. So each point is reflected as (r,θ) and can be modified by using a slight modification to work for polar Laplace. A point $z \in Z$ where $z = (r,\theta)$ is randomly generated using two separate methods for calculating r and θ.

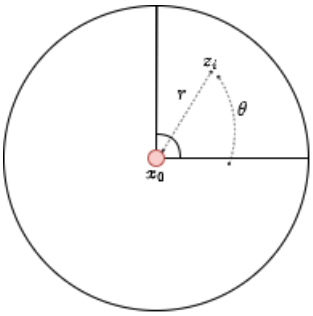

Figure 3.1: Representation of the generated $z = r\theta$ and original point $x0$.

Calculating r: This variable is described as $dist(x_0, z)$ and can be randomly drawn by inverting the CDF (**Link**) for the Laplace distribution:

$$C_\epsilon{}^{-1}(p) = -\frac{1}{\epsilon}(W\_1(\frac{p-1}{e}) + 1) \tag{3.3}$$

For this equation, W_1 is a Lambert W function with -1 branch. The Lambert w function, also called the product logarithm is defined as $W(x)e^{W(x)} = x$ [Lehtonen, 2016]. The purpose of the Lambert w function is to invert the CDF of the Laplace distribution to generate random noise for one of the coordinates (r) using the random value of p.

Calculating θ: The other coordinate (θ) is defined as a random number $[0, 2\pi]$.

To visualize these methods it is necessary to convert the polar coordinates for $z = (r,\theta)$ back to a plane (x, y). This is described as step 4 of the planar Laplace algorithm [Andrés et al., 2012] and visualized using figure **??**.

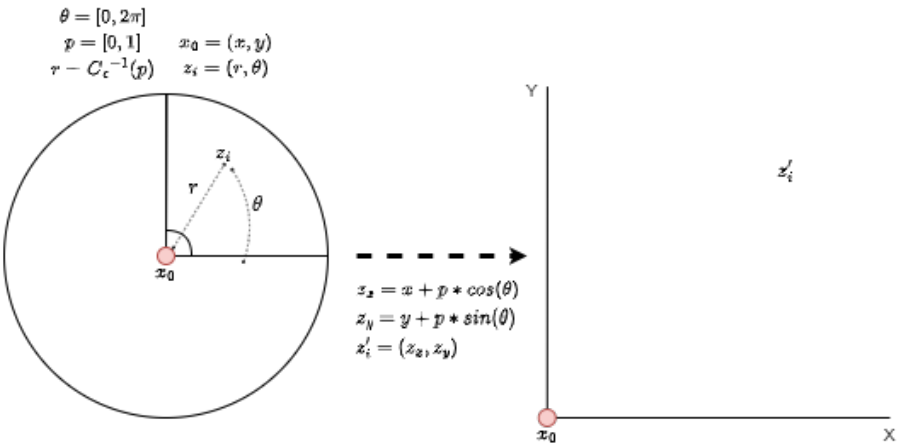

Figure 3.2: Representation of converting the perturbed point $z = (r,\theta)$ to a point z_x, z_y

3.1.2. TRUNCATION

Because we have a finite space, it can be possible the perturbed points are off-graph (outside the given domain). The solution was described in step 5 of the Laplacian mechanism for 2D space. This explains the idea of remapping to the closest admissible location in set A. For which $A \subset \mathbb{R}$, where A is the set of admissible locations [Andrés et al., 2012]. This is also described by chatzikokolakis et al, who also describes a method to do it. When a perturbed point z is located at the sea or in water, it is easily distinguishable as a fake location. They introduce a method to check this and efficiently remap to a nearby location.

Describe the method

Analyze other methods

3.1.3. OPTIMIZING FOR CLUSTERING

The decision of the parameters for the algorithm is straightforward as it depends on the ϵ. This constant is calculated by defining the radius r and the desired level of privacy l and ϵ is calculated using l/r. The l is a predefined constant $l \in R^+$ but usually will be below 10. For geographical data, the r can be configured by using meters as a unit of measure. Therefore, $r = 200$ corresponds to a radius of 200m around point x_0. So, regarding clustering, it is a challenge to define a reasonable radius.
The ϵ can be considered the inverse unit of r [Andrés et al., 2012]. A radius can be defined per-use case based on how crowded a place is [Chatzikokolakis et al., 2015].

Give the algorithm

A drawn area as shown in **??** can be expressed as a perturbation area P_{area} [Yan et al., 2022]. This metric was formulated as:

$$P_{area} = \left\{ center = x_0, radius = \frac{1}{N} \times \sum_{i=1}^{N} r_i \right\} \tag{3.4}$$

The method loops through each perturbed point r on center x_0 (recall **??**) and calculates the Euclidean distance for an n amount of perturbation points. Although the method does not contribute to the Laplace algorithm, it is useful for visualization purposes.

3.2. 3D-LAPLACE

Is considered for research question 2

3.3. ND-LAPLACE

Is considered for research question 2

4

ATTACKS ON PRIVACY

This chapter focuses on the prevalent attacks that machine learning algorithms face. Towards the end, we will assess their impact on differential privacy and discuss evaluation methods.

4.1. MEMBERSHIP INFERENCE ATTACKS

An attack model that plays a big role in machine learning is a membership inference attack (MIA). With this attack, an adversary attempts to infer the training data $x \in X$ (member) from a given data point $z \in Z$ (non-member). The attack happens exclusively on supervised learning models, which either predict labels or probabilities. Most attacks on models trained on a centralized dataset occur during the inference phase, where the trained model is used to make predictions. [Rigaki and Garcia, 2021]. This is also why we are primarily interested in this phase, as we are not using a distributed learning model.

MIA depends on the knowledge of the attacker (adversarial knowledge), which can be divided into white-box and black-box approaches [Hu et al., 2022].

1. **White-box**: The attacker has all the data that is needed. Including target model parameters, the training dataset and even the architecture [Hu et al., 2022].

2. **Black-box**: The attacker has a limited amount of information, like training data distribution and the trained model [Hu et al., 2022].

The most well-known member inference attack is training shadow models [Rigaki and Garcia, 2021]. In this attack, an attacker trains multiple models. These models do not necessarily have to be the same as the original model, and the focus is mainly on the data input/output. It is a black-box attack, but often the attacker also needs knowledge of the data distribution to create a good shadow dataset [Rigaki and Garcia, 2021].

One of the earlier works that used this attack was Shokri et al. [Shokri et al., 2017]. An attacker trains multiple models (shadow model), with as goal to overfit the original modal. This idea is based on the fact that the model gives higher scores to the data on which it was trained (overfitting). Using this approach, attackers can retrieve the training data (member data) from the model by injecting a large amount of fake data (non-membership data).

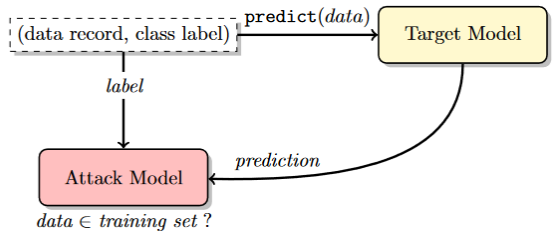

Figure 4.1: Black-box MIA attack on a machine learning model [Shokri et al., 2017]

Another approach to a black-box attack was introduced by Peng et al. and only considers that the attacker has access to the already trained model. They rescale the probabilities first using temperature scaling, to compensate for models that are overconfident

[Peng et al.]. So instead of having a probability between two classes with for example 99% against 1% it will be more evenly distributed based on the training data. They then proceed in clustering the probabilities into two clusters using K-Means and label the higher confidence scores as members.

The above attacks do rely on the model to also provide the confidence or probabilities of the predictions. This is often not the case for the practical appliance of a model, and therefore Choquette-Choo et al. introduced a label-only attack. While the existing models exploit the probability output for MIA, they solely rely on labels [Choquette-Choo et al., 2021]. For this, they make use of the "HopSkipJump" attack; a so-called decision-based attack [Chen et al., 2020]. Choquette-Choo et al. consider a more semi-black-box approach, for which the attacker still requires access to a subset of the original training data and the trained model. Another paper that also uses "HopSkipJump" requires only the trained model and achieves higher accuracy by using an approach with random data [Li and Zhang, 2021].

Access to only the output of the model is a typical characteristic of black-box attacks. If the attacker also knows architecture, for example, it is referred to as a white-box attack. Another take on this is prediction and confidence-based MIA which are both proposed by [Yeom et al., 2018]. They assume that an attacker knows the standard error and has access to the perturbation dataset. The algorithm is then able to extract the truth label by minimizing the loss.

4.2. Reconstruction attack

This attack is a threat, especially to differential privacy. The attack is primarily focused on reconstructing the data rather than focusing on machine learning.

Do research for this attack

4.3. Model inversion attack

Do research for this attack

4.4. Attack evaluation

In this section, we compile a list of attacks and evaluate which attacks are best suited for adoption in this research. We also assess whether differential privacy provides protection for each attack and discuss how this can be measured.

4.4.1. Member inference attacks

Most current research for MIA is evaluated for neural networks [Rigaki and Garcia, 2021]. Just a small percentage evaluates this attack for supervised learning, with the majority using classification with decision trees. For these attacks, most studies have used a black-box approach [Rigaki and Garcia, 2021]. This is not surprising, as these attacks have a high success rate and pose a greater risk of being exploited.

The introduction of differential privacy reduces the impact of a member inference attack [Hu et al., 2022; Rigaki and Garcia, 2021]. This is because the input to the model is perturbed. While it is still possible to retrieve the training data, the leaked privacy is significantly reduced. The leaked privacy can be measured using a metric called "adversarial advantage." This metric describes the percentage of privacy that is compromised in the event of a member inference attack [Yeom et al., 2018]. This is calculated by subtracting the False Positive Rate from the True Positive Rate. The TPR represents the number of correctly predicted member data (training data) and the FPR represents the number of correctly predicted non-member data.

In conclusion, the attacks that use member inference attacks are all based on supervised machine learning. However, in this study, we use cluster algorithms. Therefore, a semi-supervised approach can be used, as illustrated in this figure:

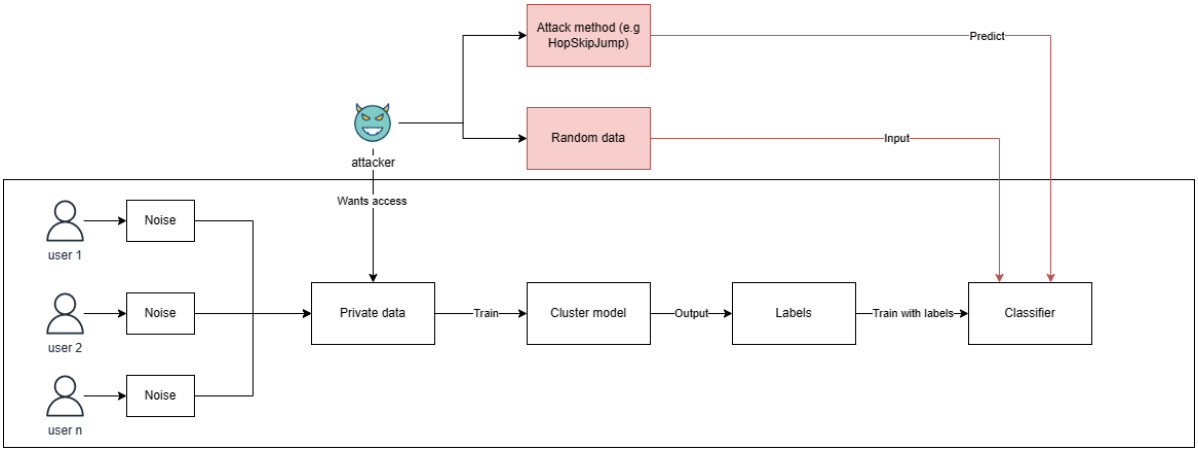

Figure 4.2: Semi-supervised black-box approach to execute a member inference attack.

Reconstruction attacks:

Evaluate after we did research for reconstruction attacks

Model inversion attacks:

Evaluate after we did research for model inversion attacks

5
METHODOLOGY

To gain insights into the proposed methods for researching the appliance of (ND)-Laplace for cluster algorithms we conducted experiments. The experiment results are used to evaluate our method against other literature. In this chapter, we explain:

1. Datasets
2. Environmental setup.
3. For each research question: Description of the different experiments.
4. For each research question: Results.

5.1. DATASETS

For this research, we will use a synthetic dataset for all three research questions. Research question 3 uses a "real-world" dataset to properly assess the different dataset properties that are the subject of this research question.

Dataset	Records	Centers	Dimensions	Standard deviation	Research
1	50	4	2	0.60	RQ 1
2	50	4	3	0.60	RQ 2
3	50	4	5	0.60	RQ 2

5.2. ENVIRONMENTAL SETUP

For running the experiments we make use of 16GB ram memory and i7-10750H 2.6Ghz processor. The experiments are run using a Docker container which runs a pre-configured distribution of Linux Alpine. It includes a pre-installed Anaconda environment for python [1,2]. We run the container using the dev-container feature for visual-studio code [3]. This allows us to create a reproducible experiment environment.

[1] https://github.com/devcontainers/images/tree/main/src/anaconda
[2] tag: mcr.microsoft.com/devcontainers/anaconda:0-3
[3] https://code.visualstudio.com/docs/devcontainers/containers

5.2.1. Libraries & code versions

We use python version 3.9.13 with Jupyter notebook for creating a reproducible experimental environment. The packages for python are:

1. Scikit-learn: 1.0.*
2. Yellow-brick: 1.5
3. Numpy: 1.24.*
4. Pandas: 1.4.*
5. Seaborn: 0.11.*
6. Mathplotlib: 3.5.*

5.3. Methods

This section explains what methods/ algorithms we used and how we evaluate them.

5.3.1. Clustering methods

For the three different algorithms: K-Means, AP and DBSCAN we analyzed the most important decisions regarding parameter selection. In this section, we give a short list and explanation of the different parameters we used throughout the experiments. For all three Scikit-learn was used, and for each of them we also provide the underlying formula.

K-Means

$$\sum_{i=0}^{n} \min_{\mu_j \in C} (||x_i - \mu_j||^2) \tag{5.1}$$

Parameter	Description	Value	Dataset
K-value	Calculated based on an "elbow" plot.	4 (see figure 7)	Dataset 1
K-value	TODO	??	Dataset 2
K-value	TODO	??	Dataset 3

Table 5.1: K-Means hyperparameters for dataset 1 - 3

Affinity Propagation

As specified in section 2.2.1, the clustering algorithm has two types of similarity. The responsibility is calculated by the following formula:

$$r(i,k) \leftarrow s(i,k) - max[a(i,k') + s(i,k') \forall k' \neq k] \tag{5.2}$$

Then the availability is given using this formula:

$$a(i,k) \leftarrow min[0, r(k,k) + \sum_{i' \ s.t. \ i' \notin \{i,k\}} r(i',k)] \tag{5.3}$$

And iteratively calculated while considering the damping factor.

Parameter	Description	Value	Dataset
Preference	We decided to use the median similarity as described in section 2.2.1	TODO	dataset 1
Preference	""	TODO	dataset 2
Preference	""	TODO	dataset 3
Damping factor	Default value as specified in section 2.2.1	0.5	dataset 1
Damping factor	""	0.5	dataset 2
Damping factor	""	0.5	dataset 3

Table 5.2: Affinity Propagation hyperparameters for datasets 1 - 3

DBSCAN

Parameter	Description	Value	Dataset
Epsilon	Decided using the k-distance plot	0.9 (see figure 8)	Dataset 1
Epsilon	""	TODO	Dataset 2
Epsilon	""	TODO	Dataset 3
Minimum points	Decided using the formula $minPts = n * 2$, where n is the number of features (2.2.1)	4	Dataset 1
Minimum points	""	6	Dataset 2
Minimum points	""	10	Dataset 3

Table 5.3: DBSCAN hyperparameters for datasets 1 - 3

5.3.2. Evaluation

With differential privacy, it is a trade-off of utility versus privacy. Therefore, for the evaluation of the 2D/3D-Laplace algorithms, we compare both criteria to achieve a consensus between utility and privacy. To reduce the measurement bias of results we executed them 10 times for multiple privacy budgets and report the average for each [Huang et al., 2021].

1. All experiments are performed 10 times and the average is reported.

2. All experiments are performed per privacy budget (epsilon). We have a fixed list for this: $0.05, 0.1, 0.5, 1, 2, 3, 5, 7, 9$.

Utility

Based on section 2.2.2, we can conclude that the corresponding literature mainly evaluates one clustering algorithm and not multiple ones. Furthermore, it can be concluded that if we only want to measure the coherence of the clusters, we can use an internal validation method. If we want a concrete measurement compared to the non-private version, an external validation method can be used. Both measurements are important to evaluate, so we use both external and internal validation.

External validation: We will use both ARI and AMI different strengths we evaluate both. It is not likely that we will use many data points for research questions 1 and 2. To compensate for this, the adjusted version is used for both Rand Index and Mutual Information. The implementation for these metrics is provided by the Scikit-learn package. With the underlying formulas:

$$AMI(U,V) = \frac{MI(U,V) - E(MI(U,V))}{avg(H(U),H(V)) - E(MI(U,V))} \tag{5.4}$$

Adjusted Mutual Information formula [Hubert and Arabie, 1985; Vinh et al.]

$$RI = \frac{a+b}{C_2^n} \tag{5.5}$$

$$ARI = \frac{RI - E(RI)}{max(RI) - E(RI)} \tag{5.6}$$

(Adjusted) Rand Index formula [Hubert and Arabie, 1985; Rand, 1971]

Internal validation: To evaluate the cluster algorithms, we use CHI and silhouette score. The expectation is that both will give a similar result, but they measure the cluster coherence in different ways.

Include formulas

PRIVACY

For this reason, we evaluate the privacy provided by the differential privacy algorithm by calculating the average (Euclidean) distance difference in comparison to the non-private data. Then, we evaluate the privacy method by simulating a membership inference attack and calculating the adversary advantage.

Privacy distance: We calculate the Euclidean average difference between non-perturbed data and the perturbed data. This is measured for each epsilon.

Membership inference attack (adversary advantage): We used the MIA that was proposed by Shokri et al. with the implementation that was provided by Adversarial Robustness Toolkit (ART) [noa, 2023]. An earlier study also explored this attack with as goal to evaluate differential privacy. Similar to this attack, we train a classifier with perturbed data and evaluate it using non-perturbed data (test-data / shadow-data) [Zhao et al., 2020]. In this way, we can demonstrate privacy compared to if we had not applied differential privacy. For the classifier, we use a RandomForestClassifier, as this classifier is not yet applied for shadow model attacks [Rigaki and Garcia, 2021]. We consider a semi-surpervised setup, so we generate the labels while using the K-Means algorithm. The 50 as the number of data samples are not enough to measure MIA. This is modified to 2000 samples.

Finally, we evaluate the adversary advantage (percentage) using $TPR - FPR$.

1. TPR: The amount the attacker inferred the member data correctly (the private training data).

2. FPR: The amount the attacker inferred the non-member data correctly (the non-private test data).

A visual setup of the experiment is given in figure: 5.1

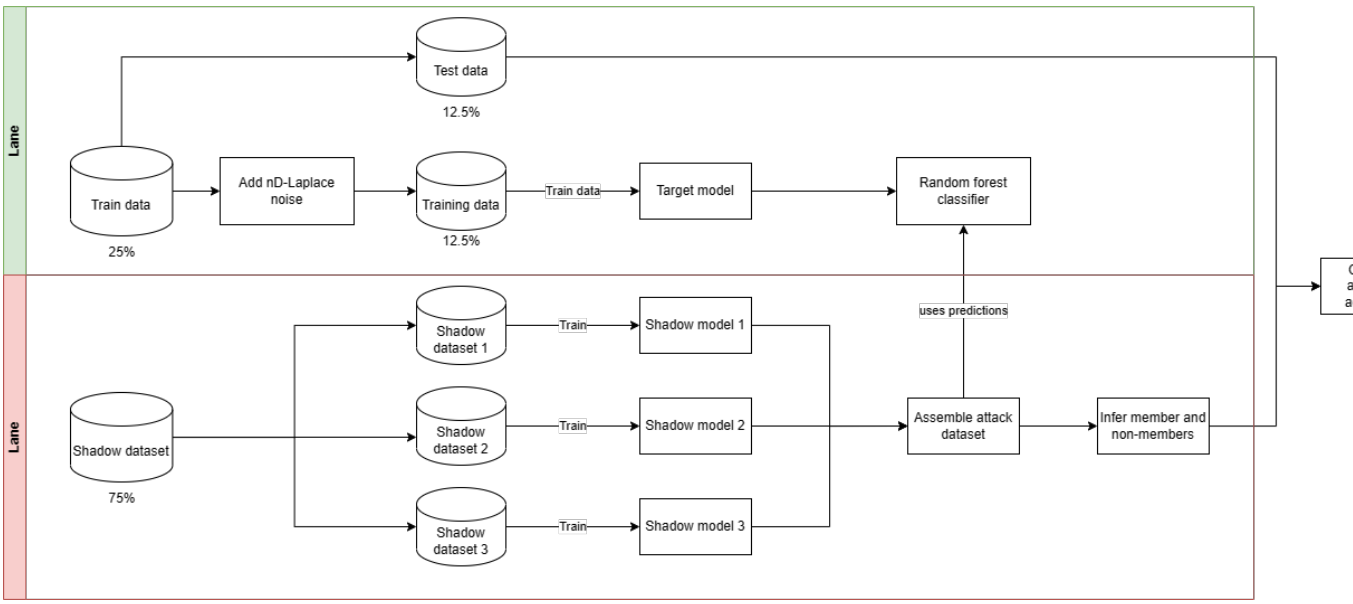

Figure 5.1: Member inference attack using shadow models. The green swim lane illustrates the normal setup and the red swim lane projects the adversary steps.

5.3.3. SCALING

Because we use a distance metric, we need to apply some data standardization. For this purpose, we use standard scaling provided by the Scikit-learn package [4]. This is only for clustering, so it is applied after all the perturbation algorithms.

5.3.4. RESEARCH QUESTION 1

TRUNCATION:

We explained the theory for truncation earlier in paragraph 3.1.2. The methods proposed work correctly for a geographic map where other (historic) locations for remapping are available.

However, it is difficult to apply this to data clustering. The number of data points is not known beforehand, so we may remap to a location that is too far away. This way we lose important distance information, which hurts the clustering. Also, the truncation threshold is so clear (the points are outside the known 2D domain), that we do not have to rely on historical data for remapping. Our algorithm can be much simpler by re-calculating the noise until it will be within the domain:

Algorithm 1 Truncation algorithm ($T(\min, \max, x_0, z)$) for clustering with planar Laplace

Ensure: z
  $x_1, y_1 \leftarrow x_{min}$
  $x_2, y_2 \leftarrow x_{max}$
  $z_x, z_y \leftarrow z$
  if $x_1 < z_x < x_2$ and $y_1 < z_y < y_2$ **then**
    return z
  else
    $x, y \leftarrow x_0$
    $z_2 \leftarrow LP(\epsilon, x, y)$ ▷ See formula 3.3.
    return $T(x_{min}, x_{max}, x_0, z_2)$ ▷ Rerun recursively
  end if

This algorithm uses x_{min} and x_{max} to re-calculate the points within the domain using respectively the minimum X/Y and maximum X/Y. An example of this is visualized:

[4] https://scikit-learn.org/stable/modules/preprocessing.html

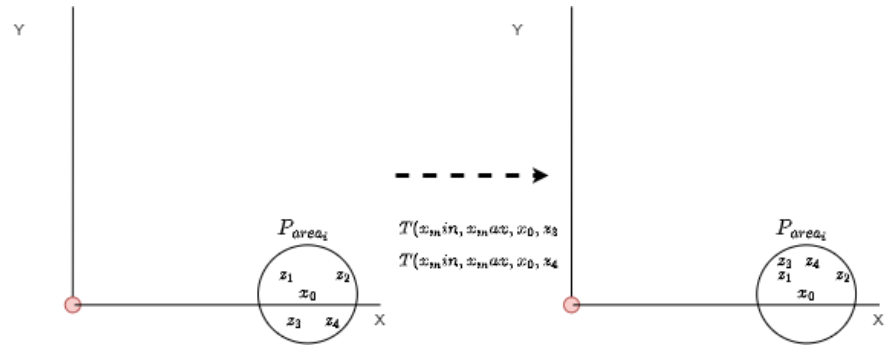

Figure 5.2: Representation of the remapping algorithm for clustering for points z_3 and z_4

ALGORITHM
The full algorithm for the perturbation:

Algorithm 2 Full algorithm for perturbing cluster data based on planar/2D-Laplace [Andrés et al., 2012]

Require: $x \in X$ ▷ 2D array of points
Require: $l \in R^+$
Ensure: $z \in Z$ ▷ 2D array of perturbed points
  $r = \frac{\sigma}{2}$ ▷ formula 4.1
  $\epsilon = \frac{l}{r}$ ▷ Calculating privacy budget [Andrés et al., 2012]
  $x_{min} \leftarrow min(X)$
  $x_{max} \leftarrow max(X)$
  $Z \leftarrow []$
  for $point_i \in X$ **do**
    $\theta \leftarrow [0, \pi 2]$ ▷ Random noise for θ
    $p \leftarrow [0, 1]$
    $z_i \leftarrow {C_\epsilon}^{-1}(p)$ ▷ formula 3.2
    $z_i \leftarrow T(x_{min}, x_{max}, point_i, z_i)$ ▷ algorithm 1.
    $x_{perturbed} \leftarrow point_{i_x} + (z_{i_x} * \cos(\theta))$ ▷ add noise to x-coordinate
    $y_{perturbed} \leftarrow point_{i_y} + (z_{i_y} * \sin(\theta))$ ▷ add noise to y-coordinate
    append $x_{perturbed}, y_{perturbed}$ to Z
  end for
  return Z

5.3.5. RESEARCH QUESTION 2

Starts after RQ1

5.3.6. RESEARCH QUESTION 3

Starts after RQ2

5.4. Results

5.4.1. Research question 1

For research question 1 the results are 2-dimensional plotted using a line diagram.

Utility

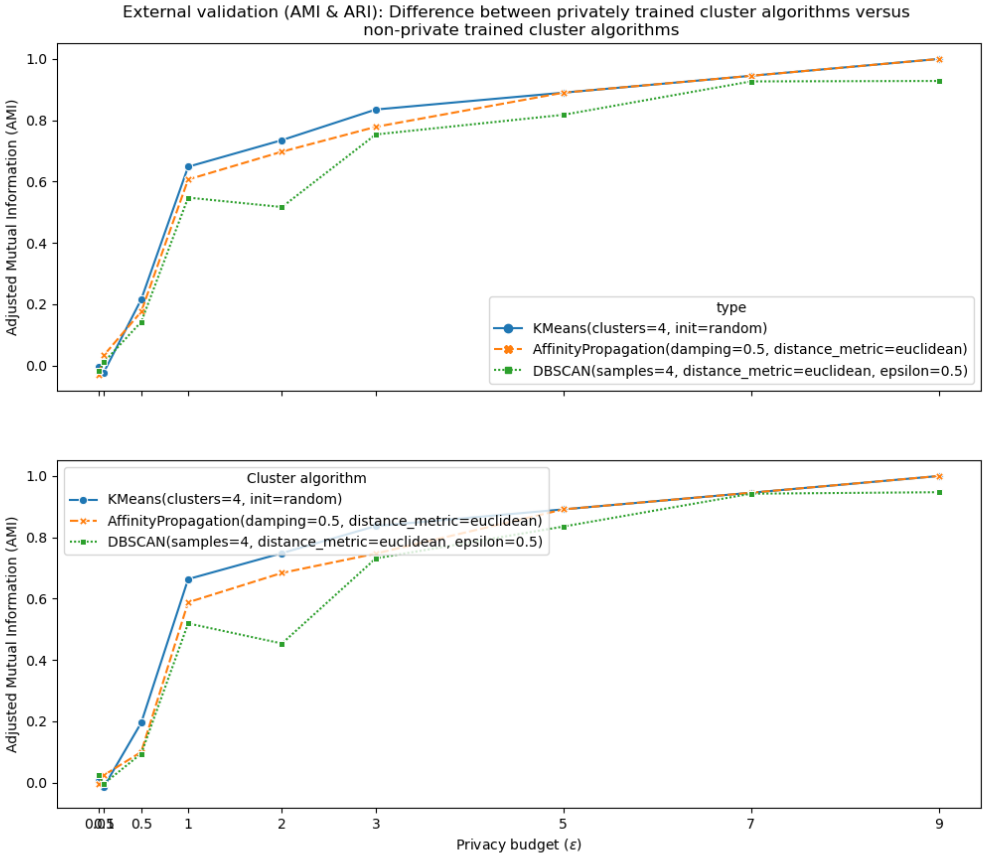

Figure 5.3: ARI and AMI evaluation for cluster algorithms 2D-Laplace for a dataset with shape (50, 2)

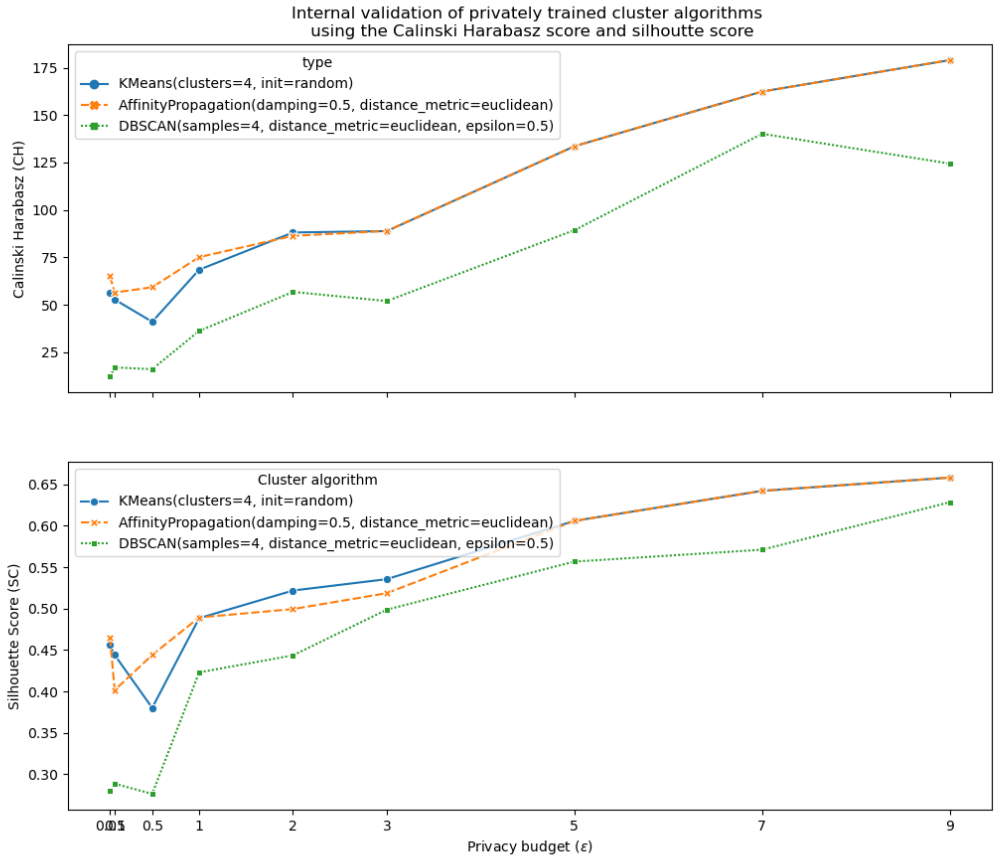

Figure 5.4: SH and CH for cluster algorithms trained with 2D-Laplace for a dataset with shape (50, 2)

PRIVACY

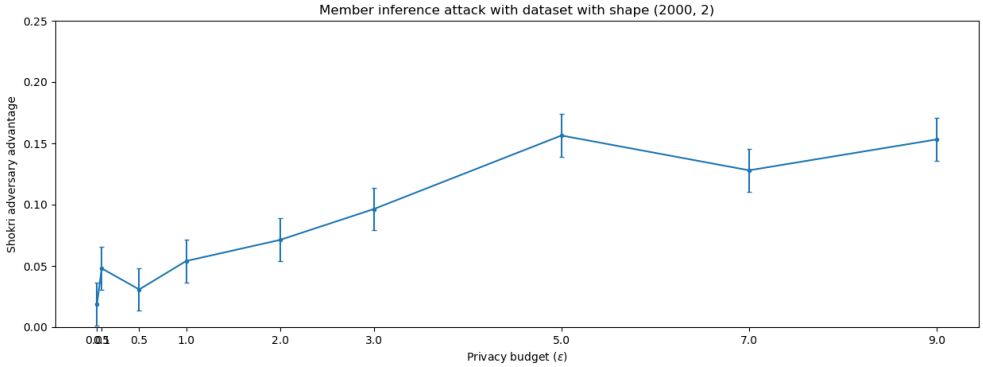

Figure 5.5: Shokri et al. attack mechanism using 3 shadow models and Yeom et al's adversary advantage ($TPR - FPR$) calculated per epsilon. (The error bar illustrates the fluctuation in the results using the standard deviation).

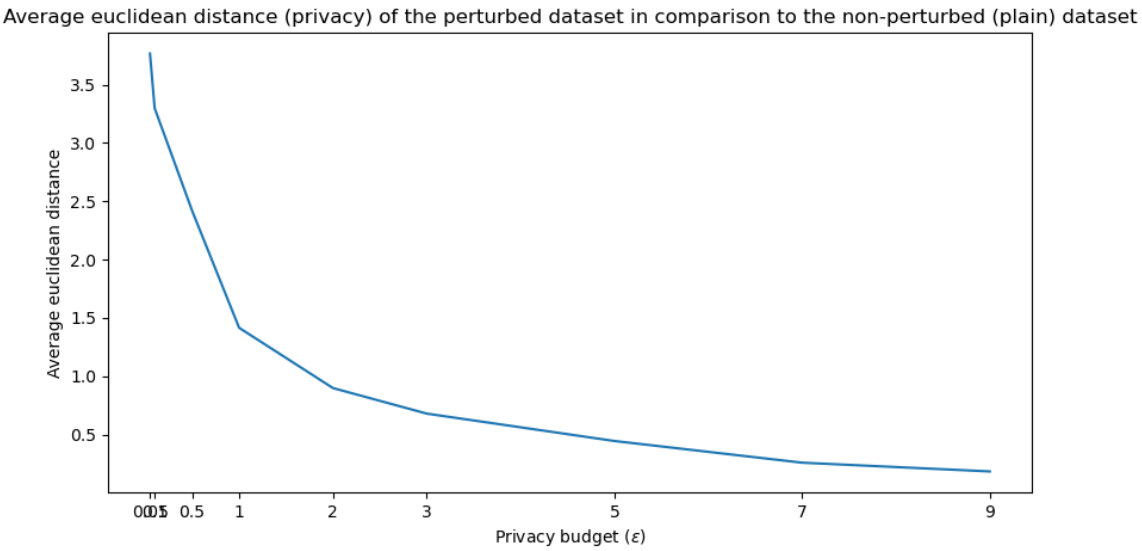

Figure 5.6: Average (euclidean) distance protection provided per epsilon.

5.4.2. RESEARCH QUESTION 2

5.4.3. RESEARCH QUESTION 3

BIBLIOGRAPHY

Adversarial Robustness Toolbox (ART) v1.14. Trusted-AI, April 2023. 21

Mohiuddin Ahmed, Raihan Seraj, and Syed Mohammed Shamsul Islam. The k-means algorithm: A comprehensive survey and performance evaluation. *Electronics*, 9(8):1295, 2020. ISSN 2079-9292. 5

Miguel E. Andrés, Nicolás Emilio Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. Geo-indistinguishability: Differential privacy for location-based systems. *CoRR*, abs/1212.1984, 2012. 10, 11, 12, 23

Daniel Bashir, George D. Montanez, Sonia Sehra, Pedro Sandoval Segura, and Julius Lauw. An Information-Theoretic Perspective on Overfitting and Underfitting, November 2020. 7

Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974. ISSN 0090-3272. 8

Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Marco Stronati. Constructing elastic distinguishability metrics for location privacy. *Proceedings on Privacy Enhancing Technologies*, 2015(2):156–170, June 2015. ISSN 2299-0984. doi: 10.1515/popets-2015-0023. 10, 12

Jianbo Chen, Michael I. Jordan, and Martin J. Wainwright. HopSkipJumpAttack: A Query-Efficient Decision-Based Attack, April 2020. 15

Davide Chicco. Ten quick tips for machine learning in computational biology. *BioData Mining*, 10:35, December 2017. ISSN 1756-0381. doi: 10.1186/s13040-017-0155-3. 7

Christopher A. Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. Label-Only Membership Inference Attacks, December 2021. 15

Toon Van Craenendonck and Hendrik Blockeel. Using Internal Validity Measures to Compare Clustering Algorithms. 8

David L Davies and Donald W Bouldin. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227, 1979. ISSN 0162-8828. 8

Janez Demšar and Blaž Zupan. Hands-on training about overfitting. *PLOS Computational Biology*, 17:e1008671, March 2021. doi: 10.1371/journal.pcbi.1008671. 7

Cynthia Dwork. Differential privacy. In *Automata, Languages and Programming: 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II 33*, pages 1–12. Springer, 2006. ISBN 3-540-35907-9. 3

Mohammed Elbatta and Wesam Ashour. A dynamic Method for Discovering Density Varied Clusters. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 6:123–134, February 2013. 6

Martin Ester, Hans-Peter Kriegel, and Xiaowei Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. 6

Pasi Fränti, Mohammad Rezaei, and Qinpei Zhao. Centroid index: Cluster level similarity measure. *Pattern Recognition*, 47(9):3034–3045, September 2014. ISSN 00313203. doi: 10.1016/j.patcog.2014.03.017. 7

Brendan J. Frey and Delbert Dueck. Clustering by Passing Messages Between Data Points. *Science*, 315(5814):972–976, February 2007. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.1136800. 6

Arik Friedman and Assaf Schuster. Data mining with differential privacy. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 493–502, Washington DC USA, July 2010. ACM. ISBN 978-1-4503-0055-1. doi: 10.1145/1835804.1835868. 3

Marwan Hassani and Thomas Seidl. Using internal evaluation measures to validate the quality of diverse stream clustering algorithms. *Vietnam Journal of Computer Science*, 4 (3):171–183, August 2017. ISSN 2196-8896. doi: 10.1007/s40595-016-0086-9. 8

Hongsheng Hu, Zoran Salcic, Lichao Sun, Gillian Dobbie, Philip S. Yu, and Xuyun Zhang. Membership Inference Attacks on Machine Learning: A Survey, February 2022. 14, 16

D. Huang, X. Yao, S. An, and S. Ren. Private distributed K-means clustering on interval data. In *2021 IEEE International Performance, Computing, and Communications Conference (IPCCC)*, pages 1–9, Los Alamitos, CA, USA, October 2021. IEEE Computer Society. doi: 10.1109/IPCCC51483.2021.9679364. 8, 20

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2: 193–218, 1985. ISSN 0176-4268. 7, 20, 21

Hannah Keller, Helen Möllering, Thomas Schneider, and Hossein Yalame. Balancing Quality and Efficiency in Private Clustering with Affinity Propagation:. In *Proceedings of the 18th International Conference on Security and Cryptography*, pages 173–184, Online Streaming, — Select a Country —, 2021. SCITEPRESS - Science and Technology Publications. ISBN 978-989-758-524-1. doi: 10.5220/0010547801730184. 6

Trupti M Kodinariya and Prashant R Makwana. Review on determining number of Cluster in K-Means Clustering. *International Journal*, 1(6):90–95, 2013. 5

Jussi Lehtonen. The Lambert W function in ecological and evolutionary models. *Methods in Ecology and Evolution*, 7(9):1110–1118, 2016. ISSN 2041-210X. doi: 10.1111/2041-210X.12568. 11

Zheng Li and Yang Zhang. Membership Leakage in Label-Only Exposures, September 2021. 15

Jinfei Liu, Joshua Huang, Jun Luo, and Li Xiong. Privacy preserving distributed DBSCAN clustering. *Transactions on Data Privacy*, 6, March 2012. doi: 10.1145/2320765.2320819. 6

André Fenias Moiane and Álvaro Muriel Lima Machado. EVALUATION OF THE CLUSTERING PERFORMANCE OF AFFINITY PROPAGATION ALGORITHM CONSIDERING THE INFLUENCE OF PREFERENCE PARAMETER AND DAMPING FACTOR. *Boletim de Ciências Geodésicas*, 24(4):426–441, December 2018. ISSN 1982-2170, 1413-4853. doi: 10.1590/s1982-21702018000400027. 6

Simon Oya, Carmela Troncoso, and Fernando Pérez-González. Is Geo-Indistinguishability What You Are Looking for? In *Proceedings of the 2017 on Workshop on Privacy in the Electronic Society*, pages 137–140, Dallas Texas USA, October 2017. ACM. ISBN 978-1-4503-5175-1. doi: 10.1145/3139550.3139555. 4

Yuefeng Peng, Bo Zhao, and Hui Liu. Unsupervised Membership Inference Attacks Against Machine Learning Models. 15

William M Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971. ISSN 0162-1459. 7, 21

Maria Rigaki and Sebastian Garcia. A Survey of Privacy Attacks in Machine Learning, April 2021. 14, 16, 21

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. 8

Jörg Sander, Martin Ester, Hans-Peter Kriegel, and Xiaowei Xu. Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications. *Data Mining and Knowledge Discovery*, 2(2):169–194, June 1998. ISSN 1573-756X. doi: 10.1023/A:1009745219419. 7

Danny Matthew SAPUTRA, Daniel SAPUTRA, and Liniyanti D OSWARI. Effect of distance metrics in determining k-value in k-means clustering using elbow and silhouette method. In *Sriwijaya International Conference on Information Technology and Its Applications (SICONIAN 2019)*, pages 341–346. Atlantis Press, 2020. ISBN 94-6252-963-9. 5

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, September 2017. ISSN 0362-5915, 1557-4644. doi: 10.1145/3068335. 6, 7

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership Inference Attacks against Machine Learning Models, March 2017. 14

Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, 3(Dec):583–617, 2002. 8

Lin Sun, Guolou Ping, and Xiaojun Ye. PrivBV: Distance-aware encoding for distributed data with local differential privacy. *Tsinghua Science and Technology*, 27(2):412–421, April 2022. ISSN 1007-0214. doi: 10.26599/TST.2021.9010027. 8

Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001. ISSN 1369-7412. 5

Matias Valdenegro-Toro and Matthia Sabatelli. Machine Learning Students Overfit to Overfitting, September 2022. 7

Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance. 7, 8, 20

Silke Wagner and Dorothea Wagner. Comparing Clusterings - An Overview. 7, 8

Kaijun Wang, Junying Zhang, Dan Li, Xinna Zhang, and Tao Guo. Adaptive Affinity Propagation Clustering. 2007. 6

Matthijs J. Warrens and Hanneke van der Hoef. Understanding the Adjusted Rand Index and Other Partition Comparison Indices Based on Counting Object Pairs. *Journal of Classification*, 39(3):487–509, November 2022. ISSN 1432-1343. doi: 10.1007/s00357-022-09413-z. 7

Chang Xia, Jingyu Hua, Wei Tong, and Sheng Zhong. Distributed K-Means clustering guaranteeing local differential privacy. *Computers & Security*, 90:101699, 2020. ISSN 0167-4048. 8

Xingxing Xiong, Shubo Liu, Dan Li, Zhaohui Cai, and Xiaoguang Niu. A Comprehensive Survey on Local Differential Privacy. *Security and Communication Networks*, 2020:8829523, October 2020. ISSN 1939-0114. doi: 10.1155/2020/8829523. 4

Yan Yan, Fei Xu, Adnan Mahmood, Zhuoyue Dong, and Quan Z. Sheng. Perturb and optimize users' location privacy using geo-indistinguishability and location semantics. *Scientific Reports*, 12(1):20445, November 2022. ISSN 2045-2322. doi: 10.1038/s41598-022-24893-0. 12

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pages 268–282, Oxford, July 2018. IEEE. ISBN 978-1-5386-6680-7. doi: 10.1109/CSF.2018.00027. 15, 16

Chunhui Yuan and Haitao Yang. Research on K-value selection method of K-means clustering algorithm. *J*, 2(2):226–235, 2019. ISSN 2571-8800. 5

Benjamin Zi Hao Zhao, Mohamed Ali Kaafar, and Nicolas Kourtellis. Not one but many Tradeoffs: Privacy Vs. Utility in Differentially Private Machine Learning. In *Proceedings of the 2020 ACM SIGSAC Conference on Cloud Computing Security Workshop*, pages 15–26, November 2020. doi: 10.1145/3411495.3421352. 21

GLOSSARY

Adjusted Mutual Information Comparable with Adjusted Rand Index this algorithm is modified to account to chance. This means it accounts for a higher MI for a higher amount of clusters between two cluster algorithms. Therefore, the calculations are strongly influenced by that of Adjusted Rand Index [**?**]. . 3, 8, 10

Adjusted Rand Index The Rand Index is improved and adjusted for chance [**?**]. This algorithm takes also into consideration the number of clusters and can be used to also compare different cluster algorithms [**?**].. iv, 3, 8, 10

Average Estimation Error This is the difference between an estimated value and the real value.. 3, 10

Bit Vector List or array to store several bits.. 3, 10

Calinski-Harabasz Index This is a way to measure the similarity of clusters [**?**]. It tells how well the clusters are separated from each other and how well the points are grouped.. 3, 8, 10

Mutual Information This metric can be used to explain the amount of information about a random variable if compared to another random variable. Therefore, it can also be used to compare two cluster similarities.. iv, 3, 7, 10

Normalized Mutual Information The normalized version is a scaled version of Mutual Information to always be a value between 0 (no correlation) and 1 (perfect correlation). This version of Mutual Information is not adjusted and therefore highly influenced by cluster amount [**?**]. So it suffers the same issue as with Mutual Information.. 3, 8, 10

Rand Index Compares the similarity between two clusters by comparing all pairs. It can therefore be used to measure the performance between two clustering algorithms [**?**]. . 3, 10

Acronyms

AEE Average Estimated Error. 3, 10, *Glossary:* Average Estimation Error

AMI Adjusted Mutual Information. 3, 8, 10, 20, *Glossary:* Adjusted Mutual Information

AP Affinity Propogation. 3, 6–8, 10, 19

ARI Adjusted Rank Index. 3, 8, 10, 20, *Glossary:* Adjusted Rand Index

BIRCH Balanced Iterative Reducing and Clustering using Hierarchies. 3, 10

BV Bit Vector. 3, 10

CHI Calinski-Harabasz Index. 3, 8, 10, 21, *Glossary:* Calinski-Harabasz Index

DBSCAN Density-based spatial clustering of applications with noise. 3, 6, 7, 10, 19

DP Differential Privacy. 3, 8, 10

DPC Density Peaks Clustering. 3, 10

GI Geo-indistinguishability. 3, 10

LDP Local Differential Privacy. 3, 10

MI Mutual Information. 3, 7, 10, *Glossary:* Mutual Information

NMI Normalized Mutual Information. 3, 8, 10, *Glossary:* Normalized Mutual Information

Math symbols

θ Angle. 3, 10

l Privacy level. 3, 10

r Radius. 3, 10

Math symbols

$K(x)(Z)$ Randomization method for $x \in X$ and output $z \in Z$.. 3, 10

$Pr(K(x_i) \in (Z))$ Probability of reporting $x \in X$ for $z \in Z$. 3, 10

X Set of locations for a user. (R^2). 3, 10

Z For every $x \in X$ a perturbed location $z \in Z$ is reported.. 3, 10

ϵ Privacy budget. 3, 10

HYPERPARAMETERS

.1. K-MEANS

For selecting the appropriate amount of clusters, we used an "elbow" plot in combination with the silhouette score.

1. Dataset 1: 4 clusters (see figure: 7)
2. Dataset 2: TODO
3. Dataset 3: TODO

.2. DBSCAN

For the selection of the appropriate epsilon, we used the k-distance plot.

1. Dataset 1: 0.9 epsilon (see figure: 8)
2. Dataset 2: TODO
3. Dataset 3: TODO

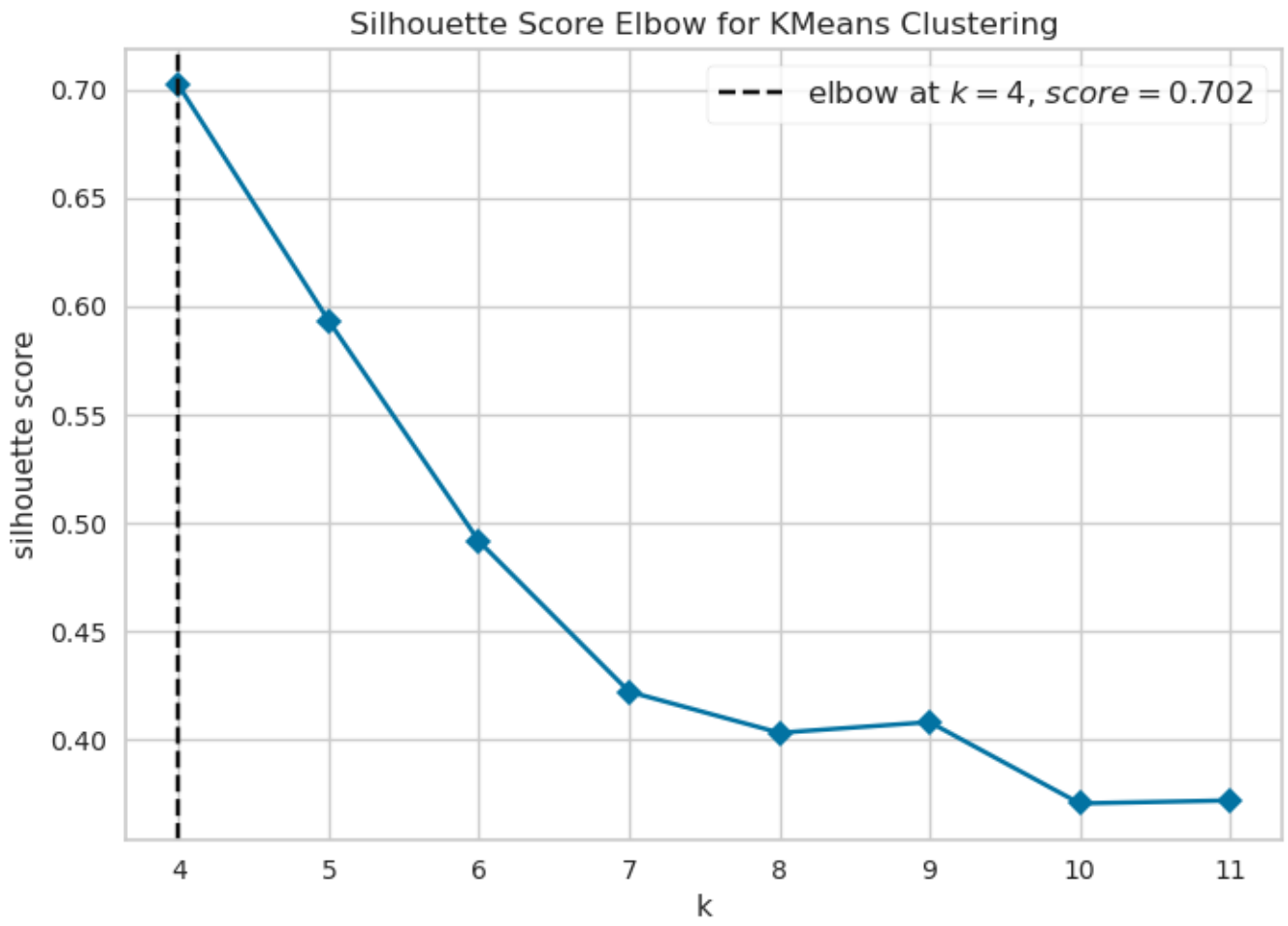

Figure 7: Selecting the k for K-Means for dataset 1 using the "elbow plot" using section 2.2.1

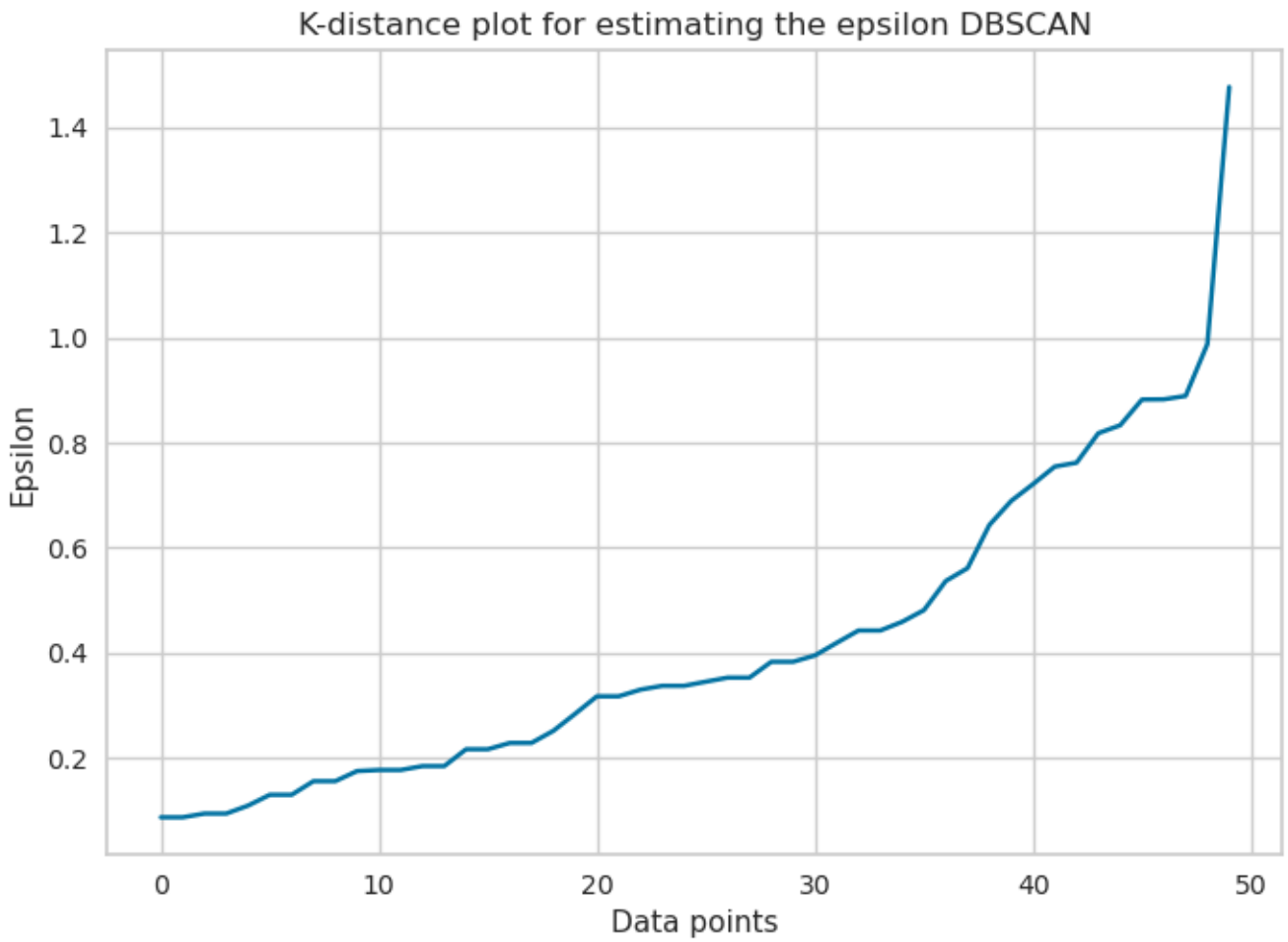

Figure 8: Selecting the ϵ for DBSCAN for dataset 1 using the "k-distance plot" using section 2.2.1