Using Nearest-Neighbors to Evaluate Overlap in Causal Inference

Jort Vincenti[1]

Supervisor(s): Jesse Krijthe[1]**, Rickard Karlsson**[1]

[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 25, 2023

Name of the student: Jort Vincenti
Final project course: CSE3000 Research Project
Thesis committee: Jesse Krijthe, Rickard Karlsson, Frans Oliehoek

Abstract

To validate the results of a medical trial, there must be an overlap between the treatment and control groups. This implies the crucial need for good evaluation methods. This study, therefore, aimed to evaluate the overlap between causal classes using the Nearest Neighbours' methods.

Firstly, a case study was built around the common failures of those methods (i.e. dependencies on hyper-parameters and sensitivity to increasing features, samples, and outliers). Secondly, a comparison of the Nearest Neighbours to other already existing approaches was made, to determine if they vary from the standard solution.

The results demonstrated that the methods can be used to assess overlap but had too much dependency on hyper-parameters, no drastic sensitivity to increasing sample and feature, and varied performance to outliers depending on their position. Additionally, the set of Nearest Neighbours methods predicted a smaller overlapping area compared to the established methods, emphasizing the caution with which the forecasts should be taken into account.

1 Introduction

In many medical disciplines, such as discovering the underlying mechanisms of a disease or estimating the effectiveness of a treatment, causal inference is essential. This process consists of evaluating if a treatment was the "cause" of the effect observed. Usually, this is achieved by creating two groups: one exposed to the treatment (experimental) and one not (controlled). If the effect were to be observed in the experimental group, possible conclusions could be made about the treatment. However, in order for the two testing groups to be comparable, they must be similar enough by sharing multiple characteristics (such as age or gender).

The implications of having overlap between classes are drastic. For example, when researchers publish the findings of a clinical trial, they also share the cohort statistics in order to characterize the similarity of study subjects [1]. These statistics give the writers an assurance that the outcomes are valid, and allow other researchers the means to come up with the same conclusion.

Therefore, elaborating new methods to evaluate overlap would help the researchers assess the validity of the output of clinical studies. Additionally, these new insights may be helpful in other areas besides the medical field, such as machine learning, where the degree of overlap between various classes affects how challenging a classification assignment is [2]. Those new methods could be the set of Nearest Neighbours algorithms, since they have already been used as solutions to density evaluation, a closely related problem [3]. However, no public paper has made this connection yet. For this reason, this paper will attempt to give new insights into this problem and contribute to its solution.

The first goal of this study is to evaluate overlap in causal inference using the Nearest Neighbors algorithms. The second is to compare those algorithms to other approaches (i.e. rule-based categorization, density estimators, and propensity score procedures) and evaluate the common predictions. The following sub-questions will be discussed in further detail:

- Are the Nearest Neighbors algorithms sensitive to their hyper-parameters for a given dataset?
- How do those methods perform as the number of features and samples increases in the dataset?
- Are the Nearest Neighbors algorithms sensitive to outlier data points?

In order to achieve both objectives, a comprehensive description of the issue will be provided in the Problem Description. The Related Works will go into further detail on discoveries in this area. Next, the Methodology of the study will be discussed. The Results and Discussion section will include a debate outlining the advantages and disadvantages of the Nearest Neighbours methods and a comparison with the other algorithms will be presented in section 6. The section about Responsible Research will then begin and, finally, the study's findings, as well as recommendations for future research, will be explained in the Conclusions and Future Work section.

2 Problem Description

The division of the tasks specified in the introduction will enable us to assess the common pitfalls of the Nearest Neighbors algorithms on the specific overlap problem [4]. In addition, we will compare them to well-established methods to determine if they vary from the standard solution. As a result, this division will cover the major theme of this study: how to apply the Nearest-Neighbors methods to evaluate overlap in causal inference.

Overlap describes the extent to which the range of data is the same across treatment groups [5]. In other words, given some data X and group labels y the overlapping criterion becomes:

$$P(X|y) \geq \epsilon \tag{1}$$

The formula (1) introduces the concept of threshold ϵ; in fact, it is not enough to have overlap across treatment groups concerning a single individual, but there must be more than a specific number of people in the overlapping zone. The probability $P(X|y)$ will be tackled by creating density distributions for the two testing groups and determining whether both density functions are greater than the threshold.

The Nearest Neighbors' methods are a class of algorithms that use proximity to make predictions about the grouping of an individual data point [6]. The general pseudo-code that

returns the data points in the overlapping regions using those methods is stated in Algorithm 1.

Algorithm 1 Nearest Neighbours

```
1: function ESTIMATE_OVERLAP(X, y, ϵ, params):
2:     overlapping_region ← []
3:     for point p in X do
4:         all_overlap ← True
5:         proximityPoints ← get_promixity_points(p, params)
6:         for class c in y do
7:             dens ← estimate_density(proximityPoints in c)
8:             if dens < ϵ then:
9:                 all_overlap ← False
10:            end if
11:        end for
12:        if all_overlap then:
13:            overlapping_region ← p
14:        end if
15:    end for
16:    return overlapping_region
17: end function
```

The algorithm requires as input the data points X, their class y, threshold ϵ, and the parameters $params$ specific to each Nearest Neighbours model (line 1). For each point p in X, it gets the set of points that are in the immediate proximity of p (lines 3-5). Those points are then grouped together per class y, and used to estimate the density of each group around point p (line 7). If all the densities are above the threshold ϵ, then point p is added to a list (lines 8-13). This list returns all the points in the overlapping region (line 16).

The main difference between each Nearest Neighbours method is the $estimate\_density$ and the $get\_promixity\_points$ functions. The former will be explained in the Related Works section. For the latter, given a point p from X, the methods differ in the following way:

1. **Radius Neighbours**: Given parameter r and p it returns all the points within a radius r of p.
2. **K-Nearest Neighbours (K-NN)**: Given parameter k and p it returns the k closest amount of points to p.
3. **Local Outlier Probabilities (LoOP)**: Given parameter k and p it uses a K-NN to evaluate the surrounding of p and returns the probability of p being an outlier.

Those two models were selected as the K-NN is by far the most researched Nearest Neighbour model due to its intuitive use and its longevity. As a result, plenty of work relating to our topic has already been made using it. In contrast, research on the radius neighbours has been far less focused. However, as the definition suggests, the radius neighbours can easily be assimilated to the K-NN, we will attempt to bridge this gap in the following sections.

3 Related Works

This section will open a discussion on the density estimators and the hyper-parameters, introducing some foundations behind the intuition used in this paper.

3.1 Density estimate

Regarding the density estimators, in Guilherme O. Campos' work [7] to evaluate outliers (i.e. an observation that deviates from its density distribution [8]), three estimators make usage of the K-NN model:

1. The Local Reachability Density (LRD) [9]:

$$\text{lrd}(p) := 1 \Big/ \frac{\sum_{o \in k\text{NN}(p)} \text{reach-dist}_k(p \leftarrow o)}{|k\text{NN}(p)|} \tag{2}$$

2. A simplified version of the Local Outlier Factor (LOF) [10]:

$$\text{dens}(p) = \frac{1}{k\text{-dist}(p)} \tag{3}$$

3. The Local Outlier Probability (LoOP) estimate [11]:

$$\text{LoOP-dens}(p) = 1 \Big/ \sqrt{\frac{1}{|k\text{NN}(p)|} \sum_{o \in k\text{NN}(p)} d(o,p)^2} \tag{4}$$

Those estimates are closely related to our problem, as they estimate $P(X|y)$ for one distribution ($|y| = 1$), instead of multiple distributions ($|y| > 1$). However, the predictions of estimates (2) and (3) are not normalised between zero and one, but rather from zero to infinity, with a value bigger than one signifying a "somewhat" outlier [12]. Those values become problematic when comparing them to the threshold ϵ mentioned in section two, which is fixed between zero and one.

A normalised formula (5), used several times before [13, 14], can be found in Sanjoy Dasgupta's work [15]. The idea is identical to $dens(p)$ (3) as it only considers the distance to the k'th point. The $lrd(p)$ formula (2) will be changed in a similar way in order to have less fluctuation since it takes the average sum of k points.

$$p_{knn}(p) = (k/n) * 1/(V_d * R_k(p)) \tag{5}$$

where $V_d = \pi^{d/2}/\gamma(d/2+1)$ is the volume of a unit d-dimensional ball, $\gamma(x)$ is the Gamma function, d represents the number of dimensions and, $R_k(p)$ denotes the distance from p to it's k'th nearest neighbour point.

3.2 Hyper-parameters

As mentioned in the Problem Description, this set of methods is particularly sensitive to the parameters selected. A common approach is to perform hyperparameter tuning on a given dataset, resulting in the "best possible choice of parameters" according to a true overlap metric [16]. However, this strategy would not be favoured since, in the majority of cases, the dataset does not contain any true overlap measure, as the true distribution is unknown.

This leads to the use of estimates, which try to approximate the best parameters, given information about the distribution of data (i.e. the size, standard deviation, mean etc.). We will next provide the estimates that were employed in this research.

K-NN: The parameter k

Sanjoy Dasgupta's [15] suggests using the following estimate:

$$k = C_0 * n^{4/5} \tag{6}$$

For some constant C_0, he guarantees an optimal convergence rate. Finding this parameter will be done in the Results and Discussion section. Additionally, regarding the k of the LoOP model, it will be implemented using the module's recommendation: $k = \sqrt{n}$, where n is the number of samples [17].

Radius Neighbours: The radius r

Although this parameter influences the model significantly [18], no public paper has been published to estimate density with this method. Some insights could be taken from the program MatchIT [19], which uses the Radius Neighbours to make matches between causal classes. However, it is not explained why $r = 1$. Having a fixed radius makes this model sensitive to the standard deviation of the distributions. Therefore, to solve those types of edge cases, and due to the lack of information about this model, it will be presented as the most "out of the box" method in the later sections.

4 Methodology

The methodology of this study will be broken down into five sections: the metrics used to assess overlap, an overview of the experiments, the dataset production, parameters used for the experiment and the final estimates per model.

4.1 Metrics

Concerning the metrics, Intersection over Union (IoU) is going to be the main focus. This is a metric used to assess the overlap between two areas ranging between zero (no overlap) and one (full overlap) [20], of which two variants exist:

$$IoU_{area} = Area\ of\ Overlap / Area\ of\ Union \tag{7}$$

where $Area\ of\ Overlap$ represents the estimated overlap intersected with the true overlapping region, and $Area\ of\ Union$ represents both areas combined.

$$IoU_{point} = True\ Positives / (True\ Positives + False\ Positives + False\ Negatives) \tag{8}$$

A new boolean feature is created for this. If a point is in the overlapping region, it will have a ***True*** value and vice-versa. Accordingly, the True Positives, False Positives and False Negatives can then be evaluated by comparing the prediction of the model and this new feature.

Note that as the number of dimensions increases, the overlapping area transforms into an N-dimensional space, making it difficult to determine the shape of the boundary between the overlapping and non-overlapping region. As a result, it gets exponentially harder to find IoU_{area}. A solution concerns using IoU_{point}; it is less accurate but, when enough data points are selected, it can approximate IoU_{area} as some of the points will lie close to this overlapping boundary.

4.2 Experiments

In order to evaluate the reliability of the estimators and answer each sub-question, the following processes are built:

1. ***Optimisations:*** A graph representing an increase of ϵ with respect to the IOU will assess how those estimates perform with high density. This process will be performed for multiple different overlapping cases (fully, partially and no overlap regions), but also with different distributions (uniformly, normal and low variance-distributed data).
2. ***Increase of samples and dimensions:*** A graph showing the time spent and a graph showing the IOU of a partially overlapping distribution will be constructed to achieve both the sample and the dimension case. The sample case will also graph a scenario in which the classes are imbalanced.
3. ***Outliers:*** A zero to five percent outlier data sample will be added to establish when the estimators fail to ignore those points. This will be plotted by representing the density estimators compared to the true overlapping region.
4. ***Comparison:*** The Iris Dataset will be used to compare the Nearest Neighbors models against more conventional approaches, and a plot for each model's prediction will be shown.

4.3 Dataset Production

Regarding the creation of the datasets, they will be produced utilizing *numpy.random.normal()* function, an n-dimensional Numpy distribution technique returning a specifiable amount of random samples from a specific distribution [21]. This can be tuned to implement different Normal, Poisson and Uniform distributions. Additionally, those distributions will contain outliers, skewness and imbalance [22] to address some of the sub-questions. The true overlap will then be evaluated using *stats.norm.pdf()* from the SciPy library [23], from which given the distribution and the samples from Numpy, it will return the $P(X|y)$ (probability density) per point which is displayed in Figure 1.

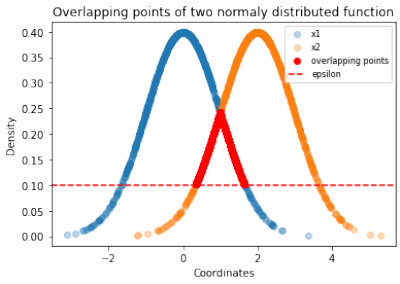


Figure 1: Density values of a set of points sampled from two normally distributed functions: *x1* (blue) and *x2* (orange)[1]. The overlapping points, which represent the minimum density of both distributions, are established by taking the density of *x1* and *x2* and asserting if both are above epsilon.

4.4 Parameters

Regarding the parameters, the k from the K-NN will be selected according to Related Works section 3.2, while the choice of the radius r will be presented next.

Radius Neighbour: Parameter r

The estimate used to approximate r will variate per point and will be based on the distance to the decision boundary [24]. In terms of overlapping regions, this is the region where the overlap between classes is the highest according to the Radius Neighbours model. As a result, the radius we would like to take into account is higher for a point that is close to this border, since it is more likely to be in the overlapping area. This is due to the fact that, if fewer points were taken into account, a more varied (dependent on the neighbouring points) density estimate would be produced. For points further away from the decision boundary, the radius should not be constant, as this would result in considering points that are too far away and potentially close to the overlapping region, ineffectively increasing the density estimate. The radius should, instead, be reduced as follows:

$$\begin{aligned} radius &= min(a, b) \\ a &= (\min_{d_1,\dots,d_T} X + \overline{d_X})/2 \\ b &= 1/(\sqrt{\#X * |d_{boundary}|}) \end{aligned} \tag{9}$$

Factor a is a constant that ensures a certain maximum radius and it represents the average between the minimum ($\min_{d_1,\dots,d_T} X$) and the average distance ($\overline{d_X}$) between all points. Factor b represents the inverse of the distance to the decision boundary, it is multiplied by the amount of data points ($\#X$), to ensure that a limited amount of points are considered (the more points the smaller the radius).

[1] x1 has mu=0, sigma=1 and n=1000, and x2 has mu=2, sigma=1 and n=1000

4.5 Density estimates

Refering back to Algorithm 1, per model, the details of the estimates function (*estimate_density(proximityPoints)*) are the following:

$$p_{knn}(p) = (k_c/n_c) * 1/(V_d * R_k(p)) \tag{10}$$

1. **K-NN**: The equation (10) is taken from the related work section (5) and displayed in Appendix Figure 11. The $(V_d * R_k(p))$ indicates an n-dimensional sphere (V_d) centered around point p, with a radius equal to the k'th nearest neighbour's distance ($R_k(p)$). The intuition behind this formula is that the furthest away this k'th point is, the lower the density, as this implies that the points are further apart. Additionally, the spherical part of the formula and the k_c/n_c, which is common to all estimates and explained later on, guarantee that the overall value is lowered to approximate $P(X|y)$ [13].

$$adapted\_lrd(p) := k_c/n_c * 1 \Big/ \frac{\sum_{i=0}^{points_c} d(p,i) * V_d * 2}{k} \tag{11}$$

2. **K-NN**: In contrast to formula (10), which only takes into account the k'th distance for the radius, formula (11), shown in Appendix Figure 11, considers, for a class, the average all distances of points in the vicinity of p ($\sum_{i=0}^{points_c} d(p,i)/k$). Then, using the factor 2 (assuming that the average distance is roughly half the maximum distance), this average distance is scaled up to the maximum distance matching (10). Since this formula depends on all sets of points within k rather than only the k'th point, the intention is to reduce variability.

$$\begin{aligned} &radius\_density(p) = \\ &k_c * (1 - \sum_{i=0}^{points_c} d(p,i) / \sum_{i=0}^{points} d(p,i))/n_c \end{aligned} \tag{12}$$

3. **Radius Neighbours**: Equation (12), shown in Appendix Figure 12, evaluates the sum of distances from a class ($\sum_{i=0}^{points_c} d(p,i)$) compared to the total sum of distances ($\sum_{i=0}^{points} d(p,i)$). This is preferred over the k'th distance as it is fixed by the radius itself. In other terms, the smaller the sum between the points from a class, the higher the density.
4. **LoOP**: As mentioned in Problem Description, this model returns a probability between 0 and 1 of each point being an outlier. This is scaled according to the maximum histograms density estimate from the Freedman-Diaconis rule [25]:

$$Binwidth = 2 * IQR(x)/\sqrt[3]{N} \tag{13}$$

where *IRQ(x)* is the interquartile range of the data and N is the number of observations in sample x.

The scaling k_c/n_c factor in all formulas denotes the proportion of points from a class that is close to point p (k_c), divided by the total number of points from the class (n_c). There is no requirement for class separation in section 3 (k/n from (5)), since this is done for one distribution. However, in our case, we specifically differentiate between the two because we wish to estimate each class's density individually (assuming they are independent).

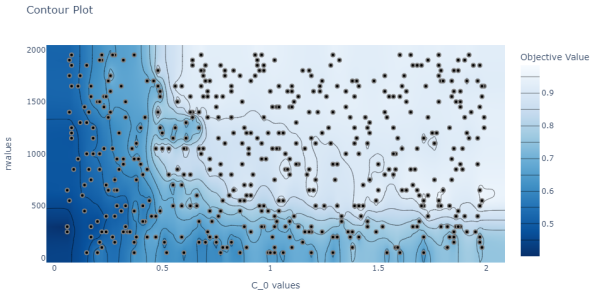
5 Results and Discussion

This section will display the findings for the three following sub-questions: first, the sensitivity to parameters, second, the analysis of whether the models scale with an increase of samples and features and, third, the sensitivity to outliers.

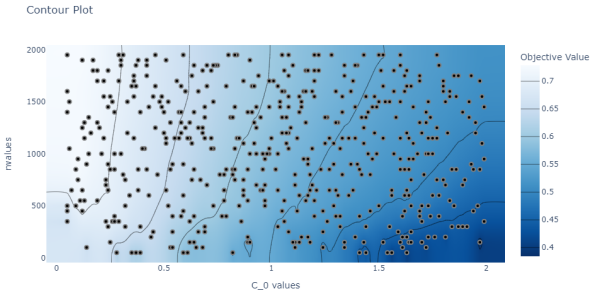
5.1 Parameter Sensitivity

The hyper-parameter k from the K-NN

As noted in Section 3, this parameter is anticipated to be the most important regarding the estimators. Figure 15 and Figure 16 show how k affected the density estimate. Additionally, as anticipated, the lrd formula exhibits less volatility than the $p_k nn$ formula because fewer data points are far from the red points (Appendix Figure 11), supporting the assertion from the Related Works.



(a) Normal Distribution[2]



(b) Uniform Distribution[3]

Figure 2: Contour plot for two different distributions. The x-axis displays the C_0 factor while the y-axis shows the size of the distribution. The lighter areas represent values that have a better IOU_{point} metric, showing a different trend for both distributions: the normal has it for high C_0, while the uniform for low C_0. The black points are the data points tested to showcase the graph.

Regarding the best C_0 (Related Works section: $k = C_0 * n^{4/5}$), none has been found to tackle the overlapping problem. As displayed in Figure 2, the best IOU score depends on the type of distribution, as its tendency switches between types. Furthermore, both show that, regardless of the C_0, the IOU improves as the number of samples rises. This is in contrast to what is claimed in the study, although it is not clear what "$|x|$ is very large" implies [15].

This results highlight the major flaw of this family of methods: despite occasionally producing decent results, it is extremely difficult to anticipate C_0, or more generally, k, in advance [15].

The radius parameter from the Radius Neighbours

As explained in section four, the radius parameter is already tuned to be adaptive with regard to the decision boundary. It is interesting to note what the implications are on the performance of the model.

Firstly, this model performs well in evaluating the points that are next to the boundary line. In other words, for specific cases where the true overlap is close to this boundary the resulting performance will be high (Figure 3).

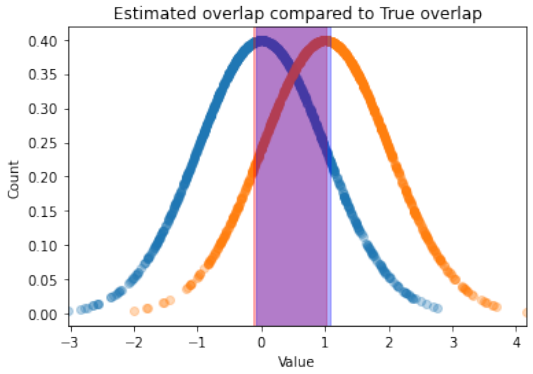
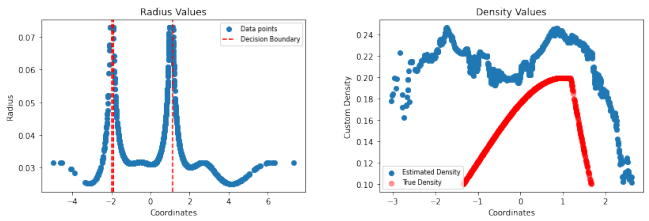


Figure 3: Radius Neighbours Performance on two overlapping distributions: *x1* (blue) and *x2* (orange)[4]. The true Overlap (Blue Area) and the Estimated Overlap (Red Area) combine to form the (Purple Area), from which the IOU_{area} is evaluated $IOU_{area} = 0.92$

[2] x1 has mu=0 and sigma=1, and x2 has mu=2 and sigma=1 with $\epsilon = 0.1$

[3] x1 has mu=0 and sigma=3, and x2 has mu=1 and sigma=2 with $\epsilon = 0.1$

[4] x1 has mu=0, sigma=1 and n=1000, and x2 has mu=1, sigma=1 and n=1000 with $\epsilon = 0.22$



(a) Radius Values **(b)** Density Values

Figure 4: Radius Neighbours performance on an overlapping distribution (Figure 14) where two decision boundaries can be observed. This results in the density values being wrong for the left side of the predictions, as the blue points (estimated density) are deviating much more from the red points (true density), resulting in a drop of the IOU_{point}
$IOU_{point} = 0.77$

Secondly, however, some edge cases can be drawn in which this model would perform poorly. In the case where one distribution is contained within the range of the other distribution (Figure 14), then, as displayed in Figure 4, there exist two decision boundaries. As a result, the radius values are not representative of the points within the two boundaries. This has the consequence of bringing down the IOU metric (per point), as all the points within those 2 boundaries are inaccurate. Additionally, if the epsilon is low, the decision boundary is too far away from the distribution to have a positive impact.

Performance

The graphs shown in Appendix B compare the performance of all three Nearest Neighbours models to three different types of cases (fully, partially, and no overlap regions), but also divided into separate distributions (evenly, normally, and light-tailed distributed data).

In general, the increase in threshold causes the IOU to drop drastically. This is mostly due to the density estimators, as they have more variability as this epsilon increases, leading to erroneous predictions. However, this highlights a limitation of the metrics. As the normal distribution has the shape of a bell curve, there are fewer representative points as the density increases, which results in limited True Positives to evaluate.

It can also be noted that the IOU_{area} has greater variability than the IOU_{point}, since it only considers the minimum and maximum points, while IOU_{point} considers all overlapping points. This is particularly true when there is only one data point (the highest density possible), in which case the area is zero because the min and max values are the same.

For the set of uniformly distributed data, the LoOP algorithm performs best, primarily since its estimator has a flat top (Figure 13) which fits the uniform distributions. The LoOP method does poorly at detecting non-overlapping data, which may be because the outlier score is not discriminate enough when taking distance into account.

For the set of low standard deviation data, the IOU decreases dramatically. This is primarily due to the fact that when the points come closer to one another, their estimations of density rise significantly, leading to misspecification. This could be solved by switching to the Mahalanobis distance metric, which considers the variation of the datasets [26].

5.2 Scalability

The LoOP algorithm has not been used in relation to the growth in dimensions because of the run time taken and its failure to multi-dimensional distributions [27].

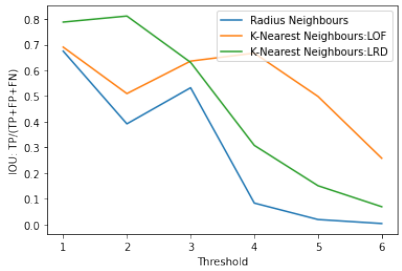
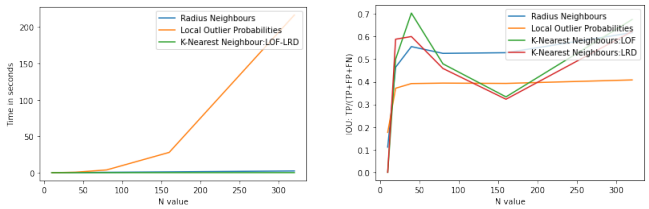


Figure 5: Performance of the models (y-axis representing IOU_{point}) as the number of dimensions increase (x-axis). The distributions were adapted for each dimension such that 30% of the points were always in the overlapping region[5]

.

Looking at Figure 5, due to the high dimensions, multiple decision boundaries are created by the radius neighbours algorithm, resulting in poor performance for this problem (similarly to Figure 4). The p_{knn} (LOF) performs well, although it should be noted that the k requires a better tunning for multi-dimensions. The lrd on the other hand seems to perform quite poorly. Since the standard deviation was increased per dimension to keep a similar overlap between distributions, it could have resulted in the "assuming that the average distance is approximately half of the maximum distance" from section 4 being false. A possible solution could be adding the standard deviation to this factor.



(a) Time taken in seconds **(b)** Performance IOU_{point}

Figure 6: Performance and time taken with respect to a progressive increase of samples (y-axis) for two overlapping distributions[6].

[5] x1 has a constant mean=0 and x2 has an adapted mean of 2.5, 1, 0.65, 0.5, 0.43 and 0.39 for each dimension. Both have a standard deviation of 1, 1, 0.5, 0.33, 0.25 and 0.2 as the dimensions increase, this was evaluated on a total of 1000 samples (500 each) and epsilon=0.05

With regards to Figure 6, most techniques function effectively as the number of data points increases. However, for a small number of data points (between 1 and 30), the performance suffers due to the lack of nearby points to analyze, yielding a decrease in graph *b*. Regarding the time taken, in graph a, the Radius Neighbors and K-NN algorithms perform fast compared to the input size $\mathcal{O}(n * log(n))$ [28], while the LoOP methods take an $\mathcal{O}(n^3)$ amount of time because they require comparing each point to all distributions.

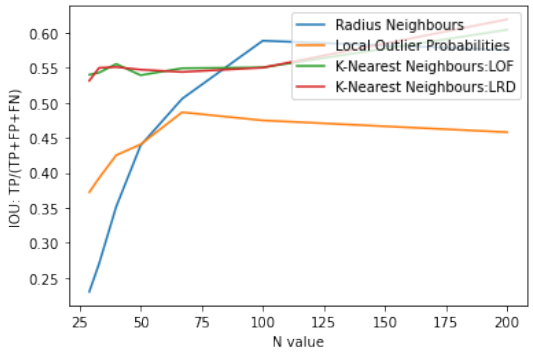


Figure 7: Performance IOU_{point} with respect to a decrease in imbalance samples, in other terms, *x1* was always kept at 200 samples while *x2* was progressively increased (from 25 to 200) representing the y-axis[7].

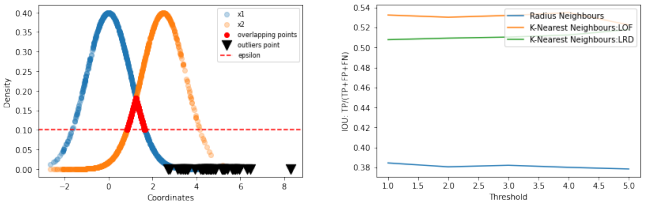
Dealing with imbalanced samples, in Figure 7, all algorithms perform at a constant rate, because the k_c/n_c factor present in all density estimators takes into account the total quantity of samples from that class (n_c). If the amount of samples is low, it would increase the k_c/n_c factor. This further explains why, the LoOP not having this factor, has generally worse performance than the other models.

5.3 Outliers

Regarding outliers, mentioned in section 3, an increasing amount (from 1% to 5%) is added to the tail of distribution *x1*, then the true overlap is evaluated ignoring those point to assess if compared to the prediction, the outliers have an impact on the models. Two edge cases can be identified: a high and a low epsilon. The reason for this distinction is that the outliers influence the extent to which the density varies. As a result, it is expected that the models are less sensitive when epsilon is high than when it is low.

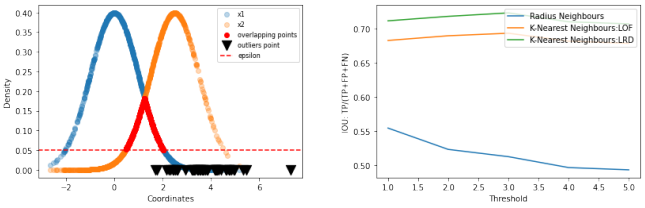
[6] x1 has a mean=0 and an std=1 while x2 has a mean=2 and an std=1. The size was increased from 10 data points to 360 and kept equal for both distributions, $\epsilon = 0.125$

[7] x1 has a mean=0 and and std=1 while x2 has a mean=2 and an std=1.$\epsilon = 0.125$



(a) Distribution **(b)** Performance: IOU_{point} for 1 to 5% outliers added (x-axis)

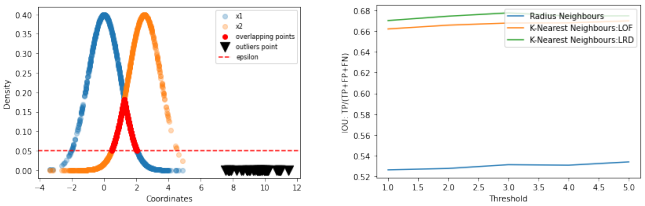
Figure 8: Performance of Nearest Neighbours models on distribution **a** for $\epsilon = 0.1$



(a) Distribution **(b)** Performance: IOU_{point} for 1 to 5% outliers added (x-axis)

Figure 9: Performance of Nearest Neighbours models on distribution **a** for $\epsilon = 0.05$

Figure 8 and Figure 9 demonstrate this expectation, as the performance of Figure 8 is less affected by the increase of outlier samples than the performance graph in Figure 9.



(a) Distribution **(b)** Performance: IOU_{point} for 1 to 5% outliers added (x-axis)

Figure 10: Performance of Nearest Neighbours models on distribution **a** for $\epsilon = 0.05$

Additionally, referring to Figure 10, it is interesting to note that, even when epsilon is low, if the outliers are too far from both distributions, the algorithms are not impacted by them. This is because the model's predictions are based on the proximity of points.

6 Comparison

The efficiency of the Nearest Neighbours models can be evaluated by contrasting them with commonly employed techniques, including rule-based classification, density

estimators and propensity score methods. To make this comparison, the well-known Iris dataset [29] was used because its unknown distribution makes it impossible to favor any model over others. Since *stats.norm.pdf()* requires selecting the distribution type, there is no true overlap. To compare the differences between the models, it is still interesting to plot all of the predictions, but it should be remembered that no conclusion about the "best model" can be drawn from this.

Appendix C displays how the models perform with a low epsilon (Figure 26-27) and high epsilon (Figure 28-29). This distinction is necessary since the Nearest Neighbors models perform worse as the epsilon grows (as explained in Performance section from subsection 5.1).

It can generally be noticed that, compared to the other models, the Nearest Neighbours models predict fewer points in the overlapping area. This is most likely due to the few amount of samples (50 per class) linking to the observation from Figure 6, in which the performance significantly dropped for a sample size below 50.

Additionally, compared to Figure 28, the Nearest Neighbours models in Figure 29 have no precise area of overlapping points showcasing the variability of the models. This is most likely because the estimate $k = \sqrt{N} = 7$ implies that each point is too dependent on its neighbouring points, resulting in such unpredictable results.

Overall, this showcases the limitation of this study, as no similar results can be found yet to more commonly used methods. This is mostly caused by the variability of the estimates and dependence on parameters. Perhaps, a better approach would be to, first, tune the hyperparameters on one of the established models (assuming it returns true overlap). Then use the Nearest Neighbours models to output the results leading to faster run time.

7 Responsible Research

Responsible research is essential to uphold ethical standards, ensure data integrity, and promote unbiased analysis, thus fostering trust and credibility in research findings. Regarding this study, the responsible research can be divided into two parts: reproducibility and applications to the medical sector.

7.1 Reproducibility

In order to confirm the results displayed in this paper, ensuring that the data generation and the algorithms yield the same outcomes is crucial. To this effect, the numpy.seed(0) was utilised for the generation of the data. Regarding the models, only deterministic ones were employed, meaning that given an input from the distribution the output would be identical. Additionally, the code base is available here[8] in order to assist with the model creation and plotting functions.

[8]https://github.com/JortVincenti/Using-Nearest-Neighbors-to-Evaluate-Overlap

The time taken in Figure 6 depends on the hardware of the computer but, the conclusions regarding the time complexity remain the same.

7.2 Application to the medical sector

Although this study displays valid results, it is still incomplete regarding real-life cases. The data is randomly generated within ranges that do not represent any feature (i.e. age, gender, etc.) nor does it represent actual groups of patients. This proves that the presented models are not tuned correctly to evaluate overlap for clinical studies. Instead, we recommend taking this as a basis to improve further before applying it to real-life problems.

8 Conclusions and Future Work

Overall, this research paper aimed to analyse the usage of the nearest Neighbours' methods to evaluate the overlap between causal classes. This was further broken down into examining the impact of the hyper-parameters on the overall outcomes and attempting to develop estimators for them, by testing an increased number of samples and features on the output, and investigating the impact of outliers.

This demonstrated that the Nearest Neighbours' methods can be used to assess overlap, but their predictions are too reliant on the hyper-parameters. Although estimated solutions can be obtained, optimality could not be achieved for all circumstances since this fluctuates too much depending on the distribution. Regarding the samples, the Radius Neighbors and K-NN showed some good time performance. The K-NN performed the best as the number of features expanded but, as they did, it became clear that k needed to be adjusted. Outliers don't appear to have an impact on the models for high epsilon but given specific edge cases, the density estimates would be greatly impacted.

Concluding results for the particular causal inference problem were not obtained as they differed too much from standard solutions. However, those models could be applied to fields with fewer implications. For instance, removing the overlapping region in order to improve machine learning models. In that scenario, their performance metrics could be checked and, if no improvement can be distinguished, the process is reversed.

The following are the primary areas that need improvement in this study, ranked by estimated difficulty:

- To evaluate the effectiveness of the density, additional metrics like $RMSE$ could be considered. This could be achieved by retrieving each point's density and comparing it to the true density.
- Perform a majority voting procedure per point for all models. In other terms, run each model on a given distribution and combine their predictions together.
- Regarding $adapted\_lrd$ formula (11), the 2 factor could be replaced by a factor representing the standard deviation. This would effectively resolve cases in which distributions are skewed.

- Although the density estimators from the K-NN already contain a factor to account for an increase in features (the V_d factor in (10) and (11)), this was concluded to be insufficient. As a result, this factor could be further improved and perhaps made more discriminatory.
- To resolve misspecification, points that are predicted in the overlapping region but too far away from other predicted points can be removed using outlier detection. This would open a broader discussion on how to detect and deal with model misspecification when using those methods.
- Finding solutions to evaluate the IOU_{area} in multiple dimensions.
- An SVC from Sklearn evaluates the Radius Neighbors algorithm's distance from each point to the decision boundary. To further enhance the predictions, this may be modified to take into account the real Radius Neighbors' decision boundary.

References

[1] Michael Oberst et al. "Characterization of Overlap in Observational Studies". In: *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*. Ed. by Silvia Chiappa and Roberto Calandra. Vol. 108. Proceedings of Machine Learning Research. PMLR, 26–28 Aug 2020, pp. 788–798. URL: https://proceedings.mlr.press/v108/oberst20a.html.

[2] Miriam Seoane Santos et al. "A unifying view of class overlap and imbalance: Key concepts, multi-view panorama, and open avenues for research". In: *Information Fusion* 89 (2023), pp. 228–253. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2022.08.017. URL: https://www.sciencedirect.com/science/article/pii/S1566253522001099.

[3] Yu Liu. *Introduction to local density estimation methods*. 2019. URL: https://www.projectrhea.org/rhea/index.php/Introduction_to_local_density_estimation_methods (visited on 06/06/2023).

[4] Kashvi Taunk et al. "A Brief Review of Nearest Neighbor Algorithm for Learning and Classification". In: May 2019, pp. 1255–1260. DOI: 10.1109/ICCS45141.2019.9065747.

[5] Andrew Gelman. *Causal inference using more advanced models - Department of Statistics*. URL: http://www.stat.columbia.edu/~gelman/arm/chap10.pdf.

[6] IBM. *What is the k-nearest neighbors algorithm?* URL: https://www.ibm.com/topics/knn.

[7] Campos G.O. Zimek A. Sander J. et al. "On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study." In: *Data Min Knowl Disc 30* (2016), pp. 891–927. URL: https://doi.org/10.1007/s10618-015-0444-8.

[8] D. M. Hawkins. *Identification of Outliers. The Science of Microfabrication*. Springer Dordrecht, 1980. ISBN: 978-94-015-3996-8.

[9] Breunig MM Kriegel HP Ng R Sander J. "LOF: identifying density-based local outliers". In: (2000), pp. 93–104. DOI: 10.1145/342009.335388.

[10] Arthur Kriegel Hans-Peter Schubert Erich Zimek. "Local outlier detection reconsidered: a generalized view on locality with applications to spatial, video, and network outlier detection". In: (2014). DOI: 10.1007/s10618-012-0300-z. URL: https://doi.org/10.1007/s10618-012-0300-z.

[11] Hans-Peter Kriegel et al. "LoOP: Local Outlier Probabilities". In: *Proceedings of the 18th ACM Conference on Information and Knowledge Management*. CIKM '09. Hong Kong, China: Association for Computing Machinery, 2009, pp. 1649–1652. ISBN: 9781605585123. DOI: 10.1145/1645953.1646195. URL: https://doi.org/10.1145/1645953.1646195.

[12] Vaibhav Jayaswal. *Local Outlier Factor (LOF)-Algorithm for outlier identification*. URL: https://towardsdatascience.com/local-outlier-factor-lof-algorithm-for-outlier-identification-8efb887d9843.

[13] Yen-Chi Chenl. "Lecture 7: Density Estimation: k-Nearest Neighbor and Basis Approach". In: (2018). URL: https://faculty.washington.edu/yenchic/18W_425/Lec7_knn_basis.pdf.

[14] Raj Praveen Selvaraj. *KnnDensityEstimation*. 2019. URL: https://www.projectrhea.org/rhea/index.php/KnnDensityEstimation (visited on 06/06/2023).

[15] Z Ghahramani et al. "Advances in neural information processing systems 27: 28th Annual Conference on Neural Information Processing Systems 2014 (NIPS); December 8-13, 2014, Montreal, Canada; proceedings of the 2014 conference". In: (2014). URL: https://papers.nips.cc/paper_files/paper/2014.

[16] Wikipedia contributors. *Hyperparameter optimization — Wikipedia, The Free Encyclopedia*. [Online; accessed 20-June-2023]. 2023. URL: https://en.wikipedia.org/w/index.php?title=Hyperparameter_optimization&oldid=1160392495.

[17] Valentino Constantinou. "PyNomaly: Anomaly detection using Local Outlier Probabilities (LoOP)." In: *Journal of Open Source Software* 3.30 (Oct. 2018), p. 845. DOI: 10.21105/joss.00845. URL: https://doi.org/10.21105/joss.00845.

[18] Hongxia Zhang, Yanhui Dong, and Yongjin Yang. *Mobility-Aware Personalized Service Recommendation in Mobile Edge Computing*. Nov. 2020. DOI: 10.21203/rs.3.rs-117144/v1.

[19] Daniel Stuart Elizabeth A. King Gary Imai Kosuke Ho. "MatchIt: Nonparametric Preprocessing for Parametric Causal Inference". In: *Journal of Statistical Software* (1970). URL: http://nrs.harvard.edu/urn-3:HUL.InstRepos:11130519.

[20] Vineeth S Subramanyam. *IOU (Intersection over Union)*. 2021. URL: https://medium.com/analytics-vidhya/iou-intersection-over-union-705a39e7acef (visited on 04/25/2023).

[21] *Random sampling (numpy.random) - NumPy v1.16 Manual*. URL: https://numpy.org/doc/1.16/reference/routines.random.html (visited on 05/09/2023).

[22] Tom Rosenström. *Distribution-Based Causal Inference : A Review and Practical Guidance for Epidemiologists*. 2020. URL: https://helda.helsinki.fi/bitstream/handle/10138/338562/Rosenstrom_GarciaVelazquez_2020_author_MS_of_Chapter_11_in_Wiedermann_etal.pdf?sequence=1 (visited on 04/28/2023).

[23] Pauli Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[24] Decision boundary. *Decision boundary — Wikipedia, The Free Encyclopedia*. 2010. URL: https://en.wikipedia.org/wiki/Decision_boundary (visited on 06/06/2023).

[25] David Freedman and Persi Diaconis. "On the histogram as a density estimator:L2 theory". In: *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete* 57.4 (Dec. 1981), pp. 453–476. ISSN: 1432-2064. DOI: 10.1007/BF01025868. URL: https://doi.org/10.1007/BF01025868.

[26] R. De Maesschalck. *The Mahalanobis distance*. 2000. URL: https://www.sciencedirect.com/science/article/abs/pii/S0169743999000477 (visited on 04/28/2023).

[27] Kevin H. Knuth. "Optimal Data-Based Binning for Histograms". In: (2013). URL: https://arxiv.org/pdf/physics/0605197.pdf.

[28] Kriegel HP Schubert E Zimek A. "Fast and scalable outlier detection with approximate nearest neighbor ensembles." In: *Proceedings of the 20th international conference on database systems for advanced applications (DASFAA)* (2015). URL: https://link.springer.com/chapter/10.1007/978-3-319-18123-3_2.

[29] R. A. Fisher. *Iris*. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C56C76. 1988.

A Graph of Density Estimators

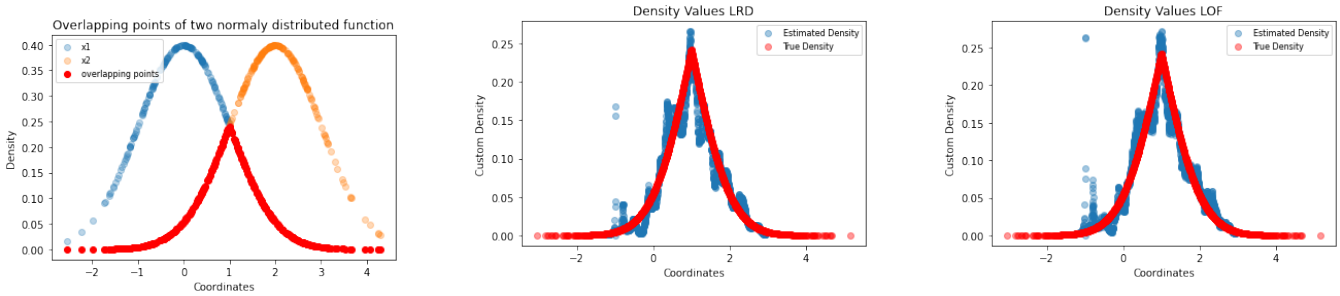


Figure 11: K-NN density estimates for the $p_{knn}(p)$ (LOF) and the $adapted\_lrd(p)$ (LRD) formula for the distribution of the leftmost Figure [9]

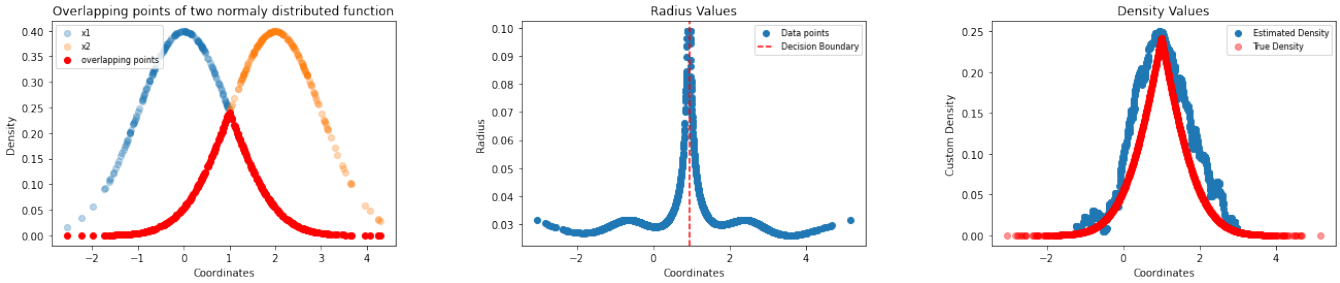


Figure 12: Radius and density estimates for the Radius Neighbours model based on the distributions of the leftmost Figure [9]

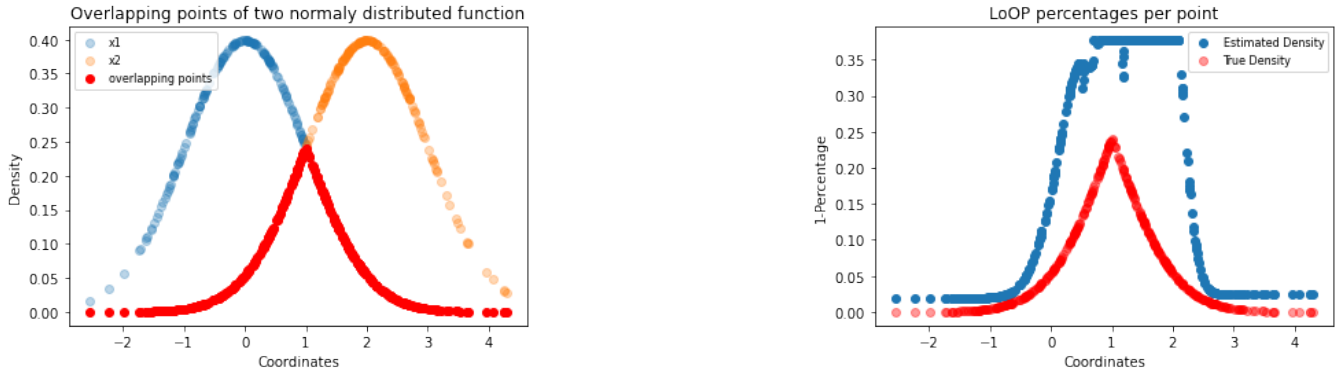


Figure 13: Density Estimate for the $LoOP - dens(p)$ estimate for the leftmost Figure [9]

[9] x1 (np.random.normal(mu=0, sigma=1, n=200)) and x2 (np.random.normal(mu=2, sigma=1, n=200))

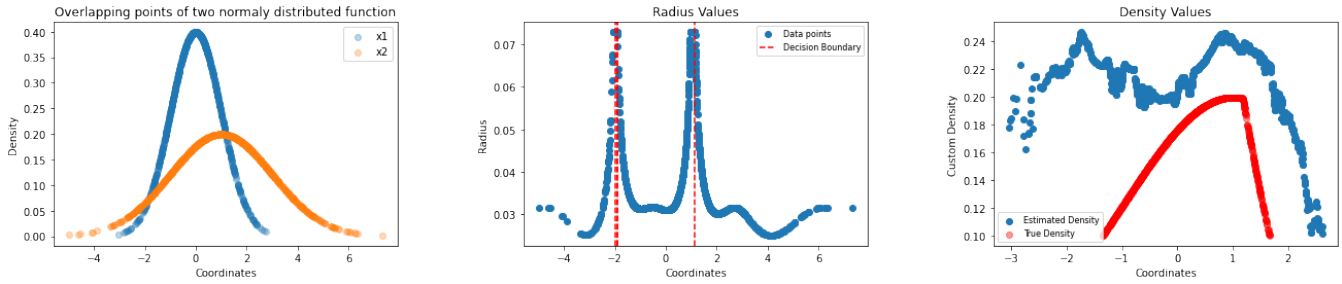


Figure 14: Radius Neighbours performance on the overlapping distribution in the leftmost Figure [10]
$IOU_{point} = 0.77$

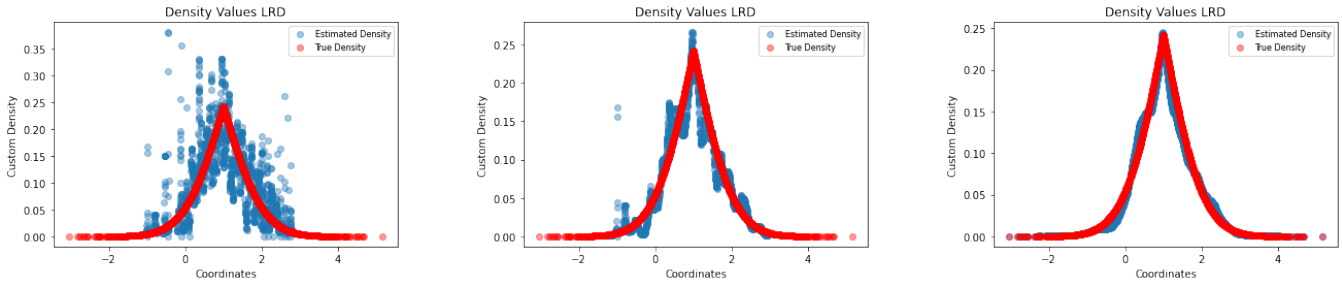


Figure 15: $adaptive\_lrd$ density estimates of an overlapping distribution [11] with k=25,100 and 275

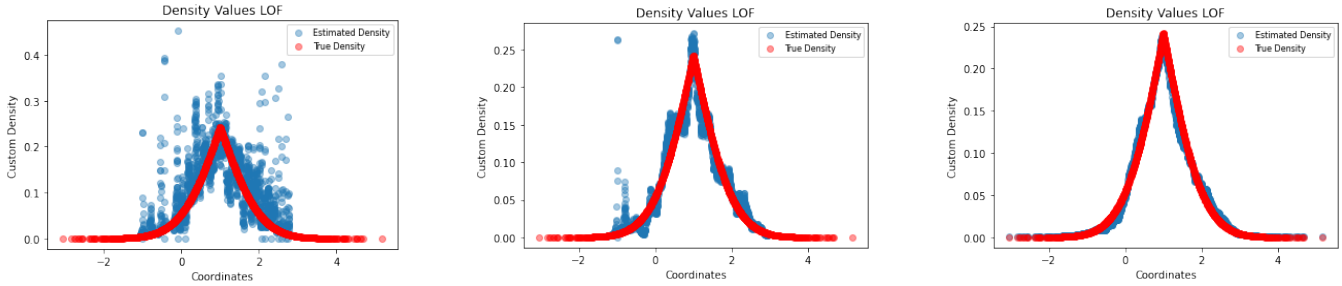


Figure 16: $p_{knn}(p)$ density estimate of an overlapping distribution [11] with k=25,100 and 275

[10] x1 = (mu=0, sigma=1, n=1000) and x2 (mu=0, sigma=2, n=1000) with $\epsilon = 0.1$
[11] x1 (np.random.normal(mu=0, sigma=1, n=200)) and x2 (np.random.normal(mu=2, sigma=1, n=200))

B Overall Performance

For all graphs presented below, the leftmost figure represents the density distribution, the middle figure represents the IOU_{area}, and the rightmost figure represents the IOU_{point}.

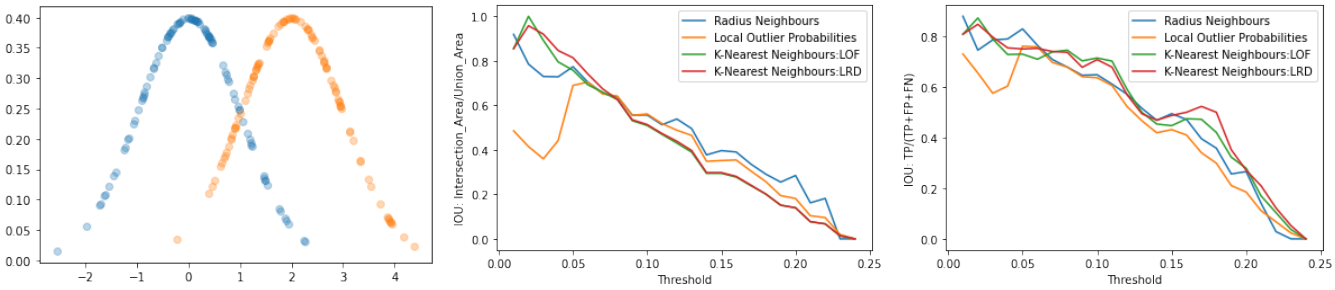


Figure 17: Performance of all models for the density distribution in the left-most figure (a normal distribution with overlap)

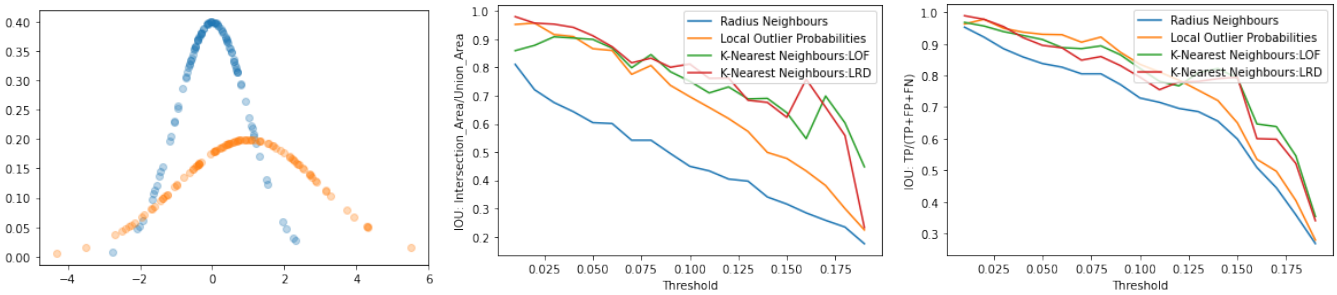


Figure 18: Performance of all models for the density distribution in the left-most figure (a normal distribution with two overlapping boundaries)

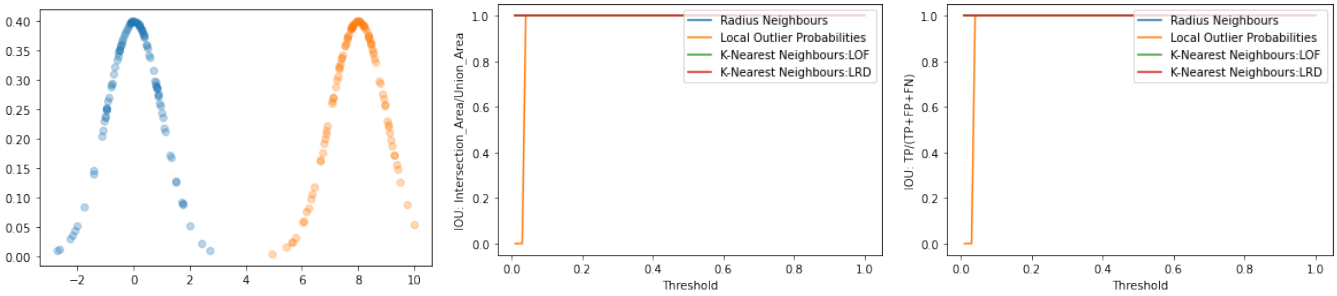


Figure 19: Performance of all models for the density distribution in the left-most figure (a normal distribution with no overlap)

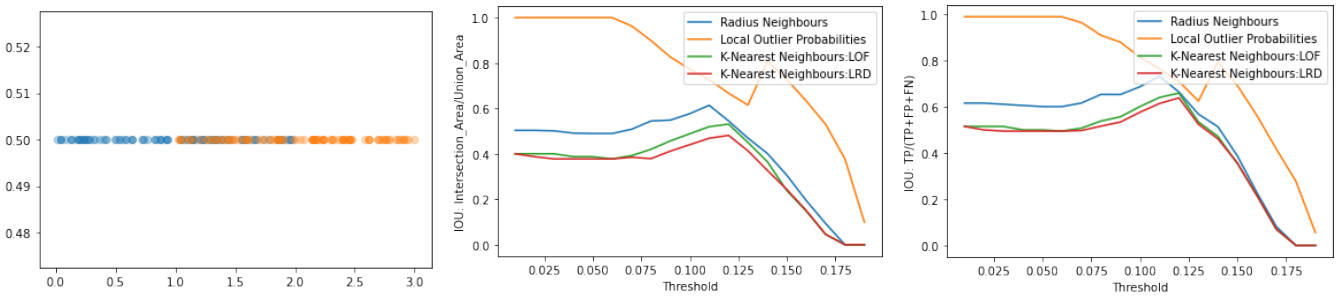


Figure 20: Performance of all models for the density distribution in the left-most figure (a uniform distribution with overlap)

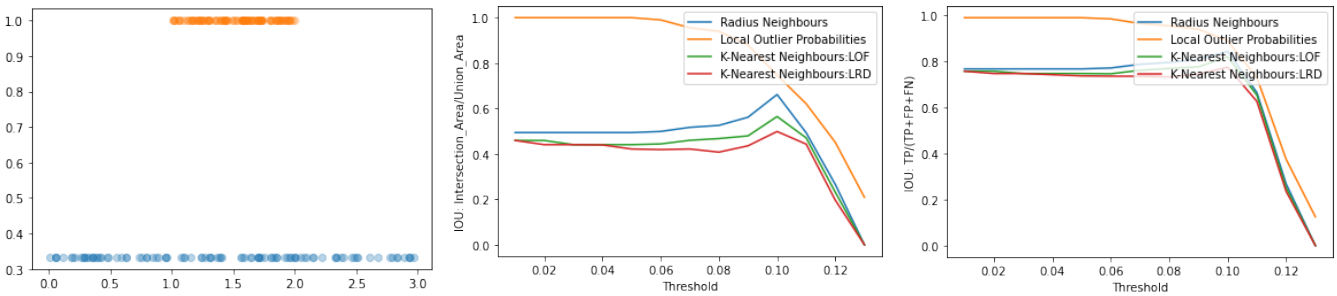


Figure 21: Performance of all models for the density distribution in the left-most figure (a uniform distribution with 2 overlapping boundaries)

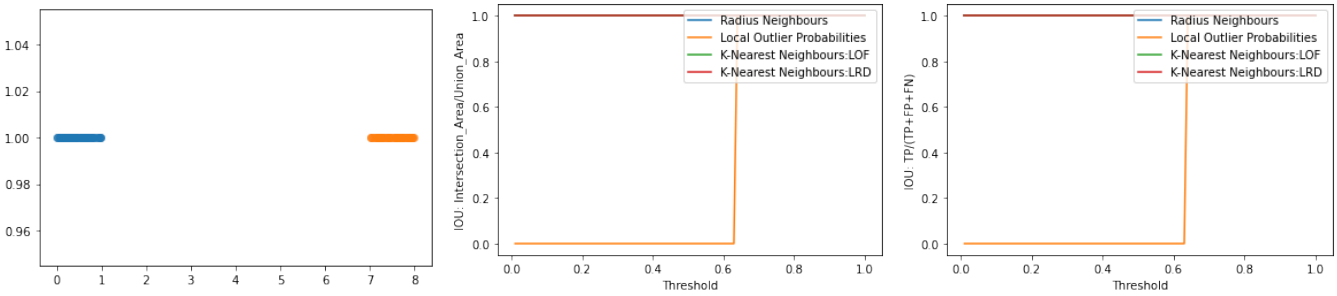


Figure 22: Performance of all models for the density distribution in the left-most figure (two uniform distributions with no overlap)

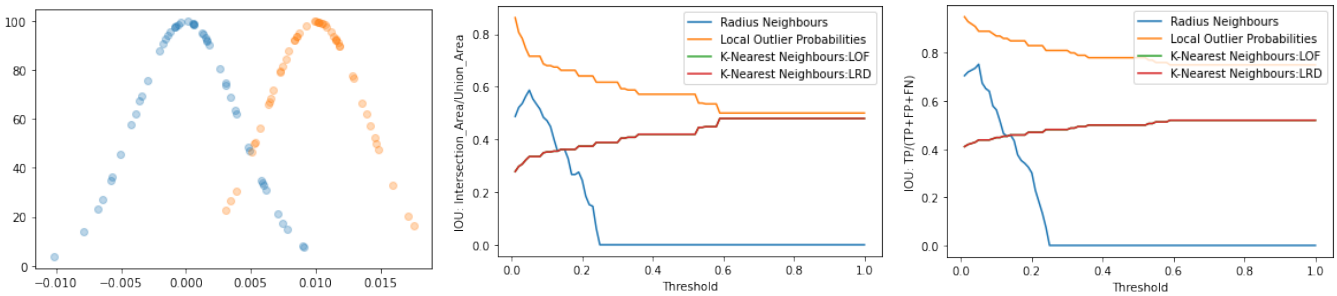


Figure 23: Performance of all models for the density distribution in the left-most figure (two normal distributions with a low standard deviation and with overlap)

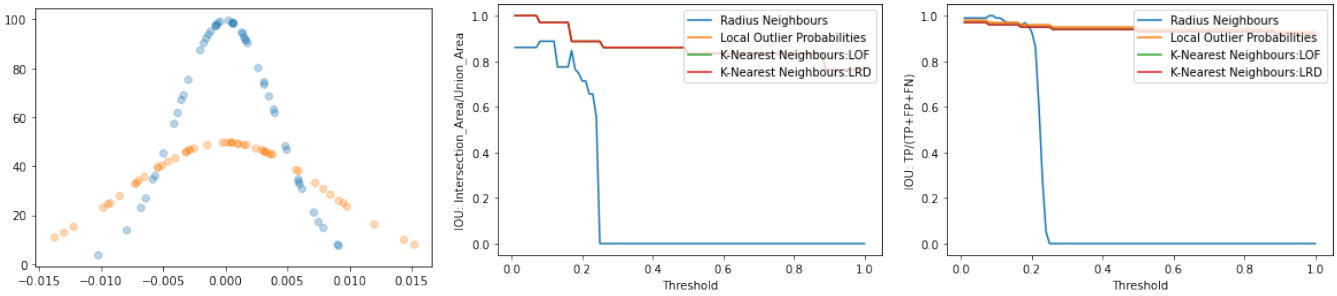


Figure 24: Performance of all models for the density distribution in the left-most figure (two normal distributions with a low standard deviation and with two overlapping boundaries)

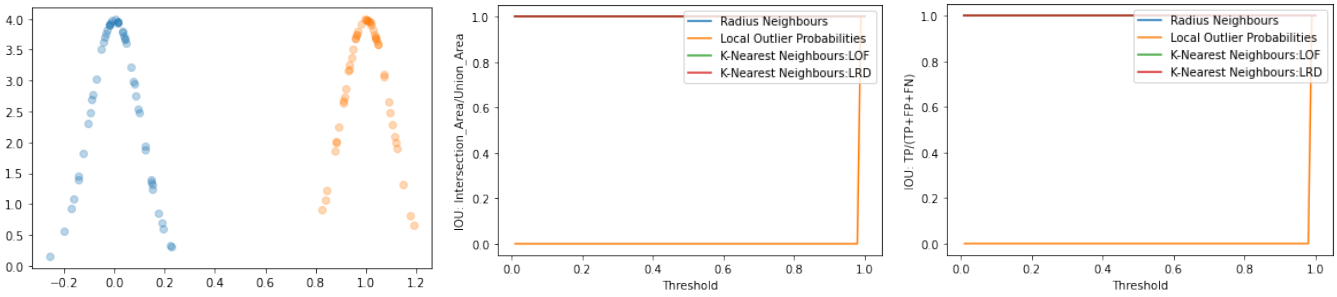
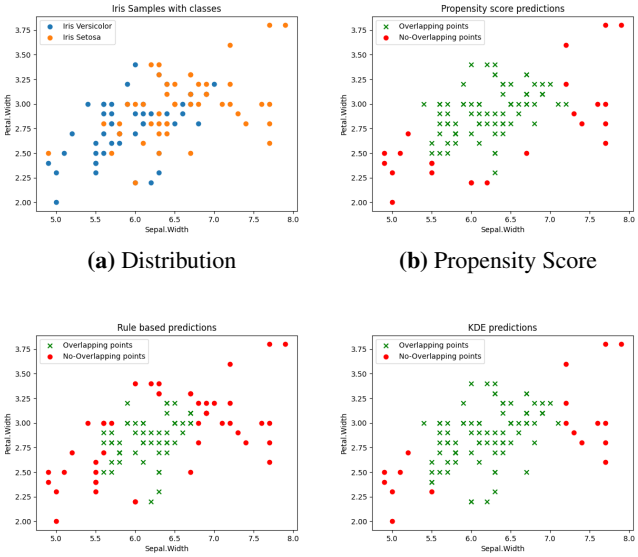


Figure 25: Performance of all models for the density distribution in the left-most figure (two normal distributions with a low standard deviation and with no overlap)

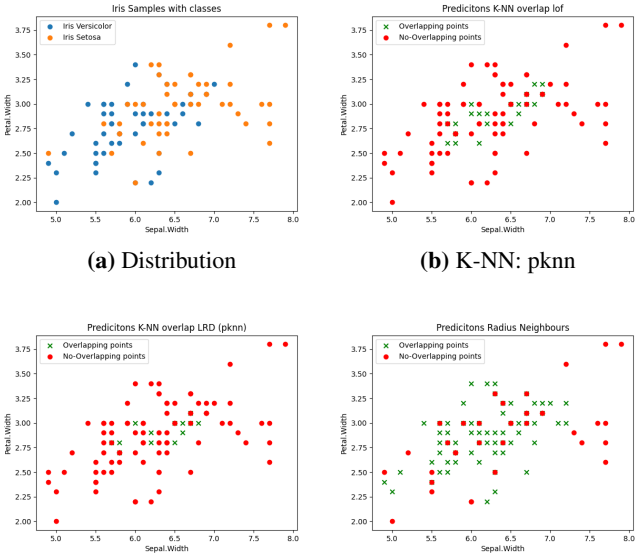
C Comparison



(a) Distribution **(b)** Propensity Score

(c) Rule based **(d)** Kernel Density Estimate

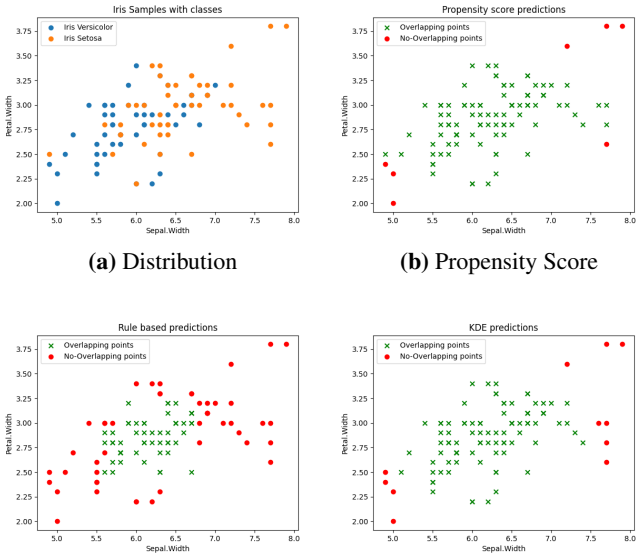
Figure 26: Comparison of established models on the Iris Dataset with epsilon=0.1



(a) Distribution **(b)** K-NN: pknn

(c) K-NN: lrd **(d)** Radius Neighbours

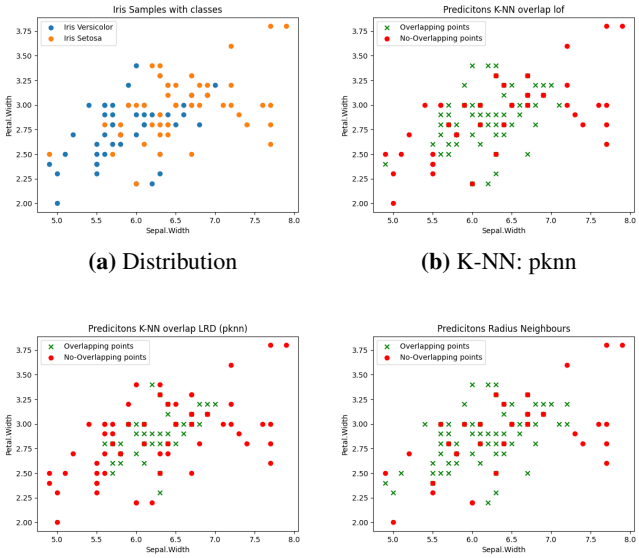
Figure 27: Comparison of the Nearest Neighbours models on the Iris Dataset with epsilon=0.1



(a) Distribution **(b)** Propensity Score

(c) Rule based **(d)** Kernel Density Estimate

Figure 28: Comparison of established models on the Iris Dataset with epsilon=0.05



(a) Distribution **(b)** K-NN: pknn

(c) K-NN: lrd **(d)** Radius Neighbours

Figure 29: Comparison of the Nearest Neighbours models on the Iris Dataset with epsilon=0.05