Using Nearest Neighbours to Evaluate Overlap in Causal Inference

Author: Jort Vincenti, in affiliation with TU Delft, Rickard Karlsson, Jesse Krijthe

Background

Causal Inference: evaluating if a treatment was the "cause" of the effect observed. [1].

Usually achieved by creating two classes, one with treatment and one with none.

➢ <u>Assumption:</u> there needs to be overlap between those classes

Nearest Neighbours: algorithm that uses proximity to make predictions about the grouping of an individual data point. [2]

➢ K-Nearest Neighbours and Radius Neighbours

Algorithm 1 Nearest Neighbours

```
function ESTIMATE_OVERLAP(X, y, ε, params):
    overlapping_region ← []
    for point p in X do
        all_overlap ← True
        proximityPoints ← get_promixity_points(p, params)
        for class c in y do
            dens ← estimate_density(proximityPoints in c)
            if dens < ε then:
                all_overlap ← False
            end if
        end for
        if all_overlap then:
            overlapping_region ← p
        end if
    end for
    return overlapping_region
end function
```

Figure 1: Overall pseudo code for the Nearest neighbours' methods.

Methodology (1)

Dataset Production

Samples: *numpy.random.normal(mu, sigma, (n, dim))*

True Overlap: *stats.norm.pdf(samples)*

Metrics

Intersection over Union: metric to quantify how much the true and predicted areas overlap.

➢ IoU_{area} = Area of Overlap/Area of Union *(2-Dimensions)*
➢ IoU_{point} = TP/(TP+FP+FN) *(N-Dimensions)*

Methodology (2)

Experiments

How sensible are the models to parameter change?

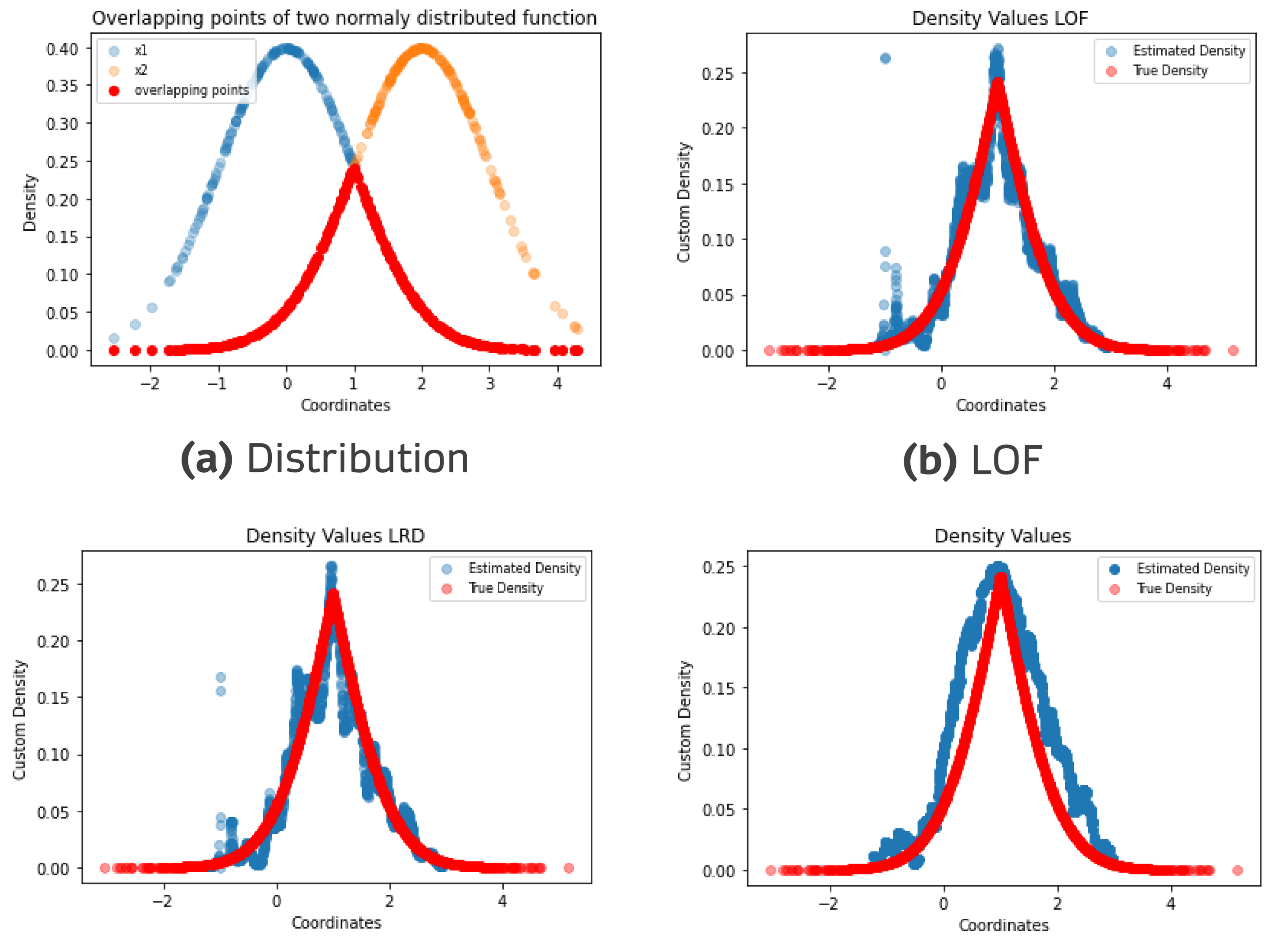
➢ *Graph performance with different parameters*

Compare to well established methods?

➢ *Graph performance for different model on Iris dataset.*

Models

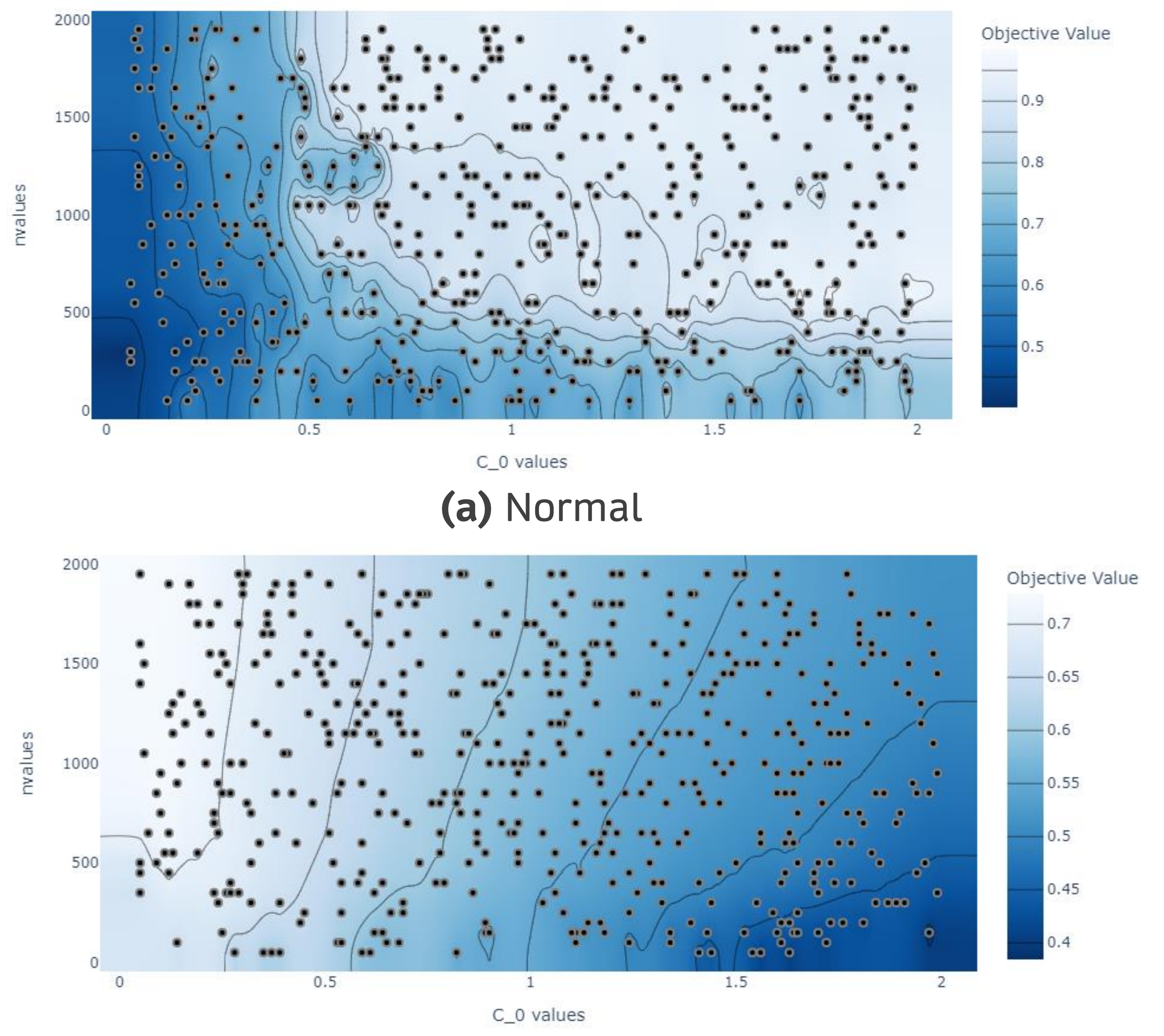
1. ***K-Nearest Neighbours*** [3]: returns *k* nearest points of *p*.
   Parameter: $k = C_0 * n^{4/5}$
   Density Estimators:
   ➢ LOF = $(k_{class}/n_{class}) \times 1/(V_d \times d(p, p_k))$ ¹
   ➢ LRD = $(k_{class}/n_{class}) \times k/(\Sigma d(p, p_{class}) \times V_d \times 2)$ ¹
2. ***Radius Neighbours:*** *returns all points of radius r of point p.*
   Adaptive Radius: point distance to decision boundary
   ➢ Large Distance = Small Radius and vice-versa
   Density Estimator: $(k_{class}/n_{class}) \times (1-\Sigma d(p, p_{class})/\Sigma d(p, p_{neigh})$ ¹



(a) Distribution **(b)** LOF **(c)** LRD **(c)** Radius Neighbours

Figure 2: Density estimators based on distribution **(a)**

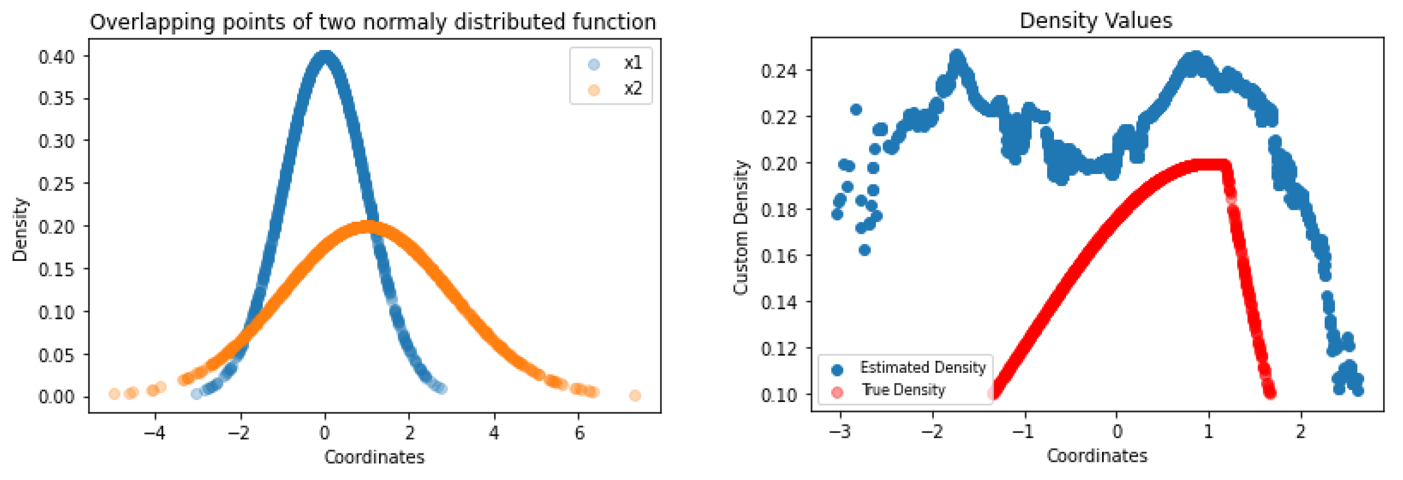
Results



(a) Normal

(b) Uniform

Figure 3: IOU_{point} for different types of distribution



(a) Distribution **(b)** Density estimate

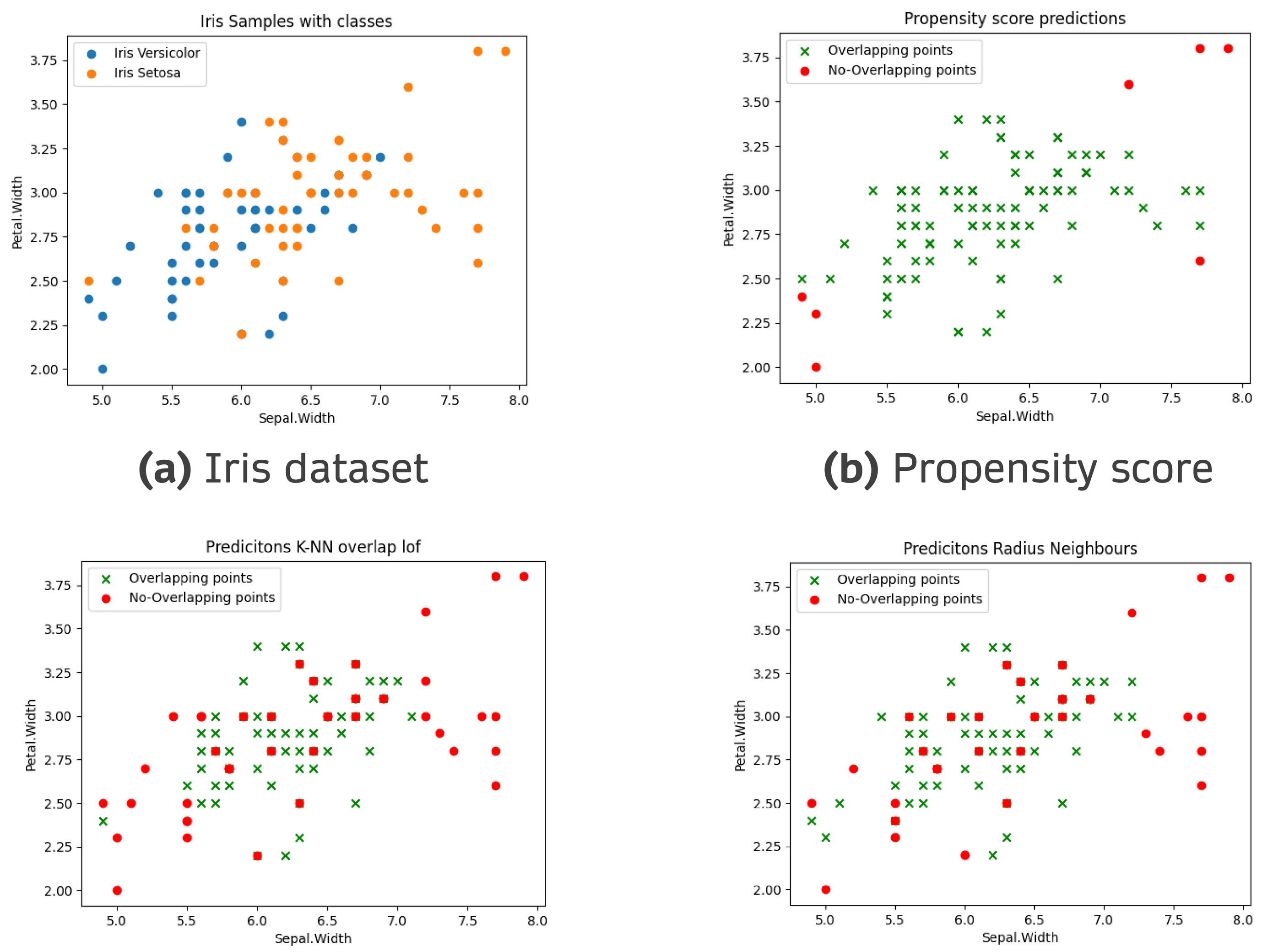
Figure 4: Radius Neighbours density estimator based on **(a)**

Observations (1)

Figure 3: No best C_0 or k can be found as this depends on the distribution and its type

Figure 4: The radius doesn't need to be tuned but fails for edge cases

Comparison



(a) Iris dataset **(b)** Propensity score

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

Figure 5: Predictions of each model on the Iris dataset

Observations (2)

Figure 5: The area of overlap is more varied then **(b)**

➢ *The low k implies high dependency on the neighboring points*

Conclusion

The models can evaluate overlap, however:

➢ Too much dependency on parameters.
➢ Variation from established methods.

The results aren't reliable for sensitive fields (i.e. clinical studies), but can still be used for other purposes (i.e. ML).

References

[1] Rubin, D., & Zell, E. (Eds.) (2018). . (Vols. 1-4). SAGE Publications, Inc., https://doi.org/10.4135/9781506326139
[2] IBM. "What is the k-nearest neighbors' algorithm?" URL: https://www.ibm.com/topics/knn.
[3] 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 Knowledge Disc 30 (2016), pp. 891–927. URL: https://doi.org/10.1007/s10618-015-0444-8.

¹ k_{class}: *number of points from class c within k range*
n_{class}: *total amount of points from class c*
V_d: *d dimensional volume*
p_k: *k'th point p*
p_{class}: *points from the class*
p_{neigh}: *all points in neighbourhood of p*
$d(p, p_{class})$: *distance from the point p to a*