Instance Weighted Clustering: Local Outlier Factor and K-Means*

Paul Moggridge[0000−0003−1004−1298], Na Helian[0000−0001−6687−0306], Yi Sun, Mariana Lilley, and Vito Veneziano

The University of Hertfordshire, Hatfield, UK
p.moggridge@herts.ac.uk

Abstract. Clustering is an established unsupervised learning method. Substantial research has been carried out in the area of feature weighting, as well instance selection for clustering. Some work has paid attention to instance weighted clustering algorithms using various instance weighting metrics based on distance information, geometric information and entropy information. However, little research has made use of instance density information to weight instances. In this paper we use density to define instance weights. We propose two novel instance weighted clustering algorithms based on Local Outlier Factor and compare them against plain k-means and traditional instance selection.

Keywords: Machine Learning · Unsupervised Learning · Instance Weighting.

1 Introduction

In the area of Data Mining, clustering is one type of unsupervised learning that involves finding groups of similar instances in data. Arguably the most popular clustering algorithm is k-means [11]. This algorithm partitions instances into a given number of clusters k. K-means iteratively assigns instances to clusters based on their distance to the centroids of the clusters, the centroids' positions are then recalculated to be the means of instances in their respective clusters.

Instance selection is a well established technique. It is often used for removing instances that are outliers. Feature weighting (also referred to as "attribute weighting") is an ongoing area of research. In Feature weighting the features of a dataset are weighted based on their various metrics typically related to how much they enhance the accuracy of the main data mining activity. Inspired by instance selection and feature weighting, Instance weighting assigns a weight to each of the instances in a dataset. Considering outliers for example, from a statistics' perspective, outlierness is a scale rather a boolean property, so it makes sense to use weighting rather than selection in response.

* Thanks you to Innovate UK for your support

Hawkins defines an outlier as "an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism" [9]. Outlier accommodation is enabling algorithms to accommodate outliers, it is the opposite of outlier diagnosis, where outliers are identified and removed before processing. Instance Weighting can provide a way for clustering algorithms to accommodate outliers, by adjusting how much to learn from outlying instances. This is important since clustering algorithms, such as k-means can be adversely effected by the presence of outliers in a dataset. Whilst it is true that some types and severities of outlier should be fully discarded, some types and severities of outliers may be best partially retained for the clustering process to learn from. This is especially important when the total number of instances is low.

This paper is structured as follows. Section 2 presents the related work. Section 3 describes our two novel instance weighted clustering algorithms. Section 4 is the methodology, experimental results and discussion of our findings. Finally, section 5 draws conclusions from our findings and presents our recommendations for future work.

2 Related Work

Nock and Neilsen's [12] research is inspired by boosting algorithms (from supervised learning) and k harmonic means clustering [15]. They are the first to formalise a boosting based approach, their solution penalises bad clustering accuracy by updating the instance weights. Their algorithm gives more weight to data points that are not well modelled. Their approach could be described as a statistics based approach. Their paper investigates, for which scenarios, instance weighting improves the accuracy of clustering and if instance weighting can reduce initialisation sensitivity. They investigate applying instance weighting on multiple algorithms including k-means, fuzzy k-means, harmonic k-means and Exception Maximisation and prove the applicability of instance weighting to a range of algorithms. Their research shows that instance weighting could speed up the clustering algorithms. They highlight the growing attention around weighted iterative clustering algorithms in unsupervised learning. In our research we have applied a simpler method, but used a density based technique. We also investigate the benefit of instance weighting and how instance weighting can address the presence of outliers in a dataset.

Sample Weighted Clustering by Jian Yu et al. weights instances using a probability distribution derived from an information theory approach [14]. They point out that there is little research on sample (another name of instance) weighted clustering compared to feature weighted clustering. Like our work they investigate the benefit instance weighting for datasets with outliers wrapping the popular k-means algorithm. They highlight that just one outlier can adversely effect the clustering output of k-means, fuzzy c-means and expectation maximisation. Their information theory based approach produces promising results which are robust to outliers across a variety of datasets. They also found their weighting also made their algorithm less sensitive to initialisation.

Lei Gu's research uses geometric based weighting that also takes local neighbour information into account. [7] Their approach uses two weighting schemes per cluster. One scheme for points close to the center of the clusters and another scheme for ambiguous points near the clusters boundaries. Their algorithm outperforms Jain Yu et al.'s algorithm (described in the previous paragraph) for accuracy. Lei Gu's research also considers non image segmentation based clustering problems.

Hammerly and Elkan's research [8] investigates the k harmonic mean algorithm. [15] They found that it produces better accuracy than k-means and show that having a non-constant (iterative) weight function is also useful . They point out many wrapper based solutions have been proposed, such as random restart, optimising the initialisation and optimising k-selection around clustering, but less research has been put into wrappers which iteratively effect the clustering. Hammerly and Elkan point out the benefit of wrapper methods is that they can often be simultaneously applied.

In Adaptive Nonparametric Clustering by Efimov et al. [6] the weightings are assigned to both the instances and features w_{ij} rather than just w_i (instance weighting) or just w_j (feature weighting). The idea of their algorithm is to look for structures in the clustering, for example, slopes away from local homogeneity. Their approach has several strengths, their algorithm supports manifold clustering and is robust against outliers. Another useful property of their algorithm is the lack of a tunable parameter, which many algorithms has. Their paper does not attempt generalise or suggest the possibility of applying their method as a wrapper method.

Jain provides an overview of clustering discussing the key issues in designing clustering algorithms, and points out some of the emerging and useful research directions. [10] Jain's paper outlines six problems / research areas, one of which is "A fundamental issue related to clustering is its stability or consistency. A good clustering principle should result in a data partitioning that is stable with respect to perturbations in the data. We need to develop clustering methods that lead to stable solutions.". This is the problem our research considers solving through instance weighting. Their review paper also points out challenges related semi-supervised clustering (however, we are not considering semi-supervised clustering in this paper), one challenge in the area of semi-supervised clustering is "how to provide the side information". Instance weighting is one possible solution to this problem, our algorithm could be adapted to work in a hybrid mode. Also, with regard to semi-supervised learning it is highlighted that it is desirable to have approach which avoids changing clustering existing algorithms, and instead wrap around them.

Instance weighting is an established technique but there is much less research compared feature weighting. For instance, in recent and comprehensive literature, for example, Data Clutering [1] instance weighting is not mentioned. However, instance weighting is a promising technique and can provide several enhancements to several existing clustering algorithms. Instance weighting is also an increasingly important technique, on the popular dataset website UCI [5], the average size of the datasets in terms of instances is increasing. Instance weighting like ours makes clustering more robust leading towards an increasingly automated knowledge discovery process by reducing the requirement for preprocessing of data.

Some work has paid attention to instance weighted clustering algorithms using various instance weighting metrics based on distance information, geometric information and entropy information. However, little research has made use of instance density information to weight instances. In this paper we use density to define instance weights, develops clustering methods that lead to stable solutions by using instance density information to weight instances.

2.1 Local Outlier Factor

Local Outlier Factor (LOF) is an outlier detection algorithm which provides a measure of outlierness. LOF works by comparing the density of an instance to that of its neighbours [3]. Equations (1), (2) and (3)[1] show how to calculate the LOF of a point A. A represents the point we are calculating the local density of. k represents the number of neighbours to consider. $k-distance$ is the distance from a given point to its k^{th} furthest point. $N_K(A)$ is the set of k nearest neighbours to A.

$$reachability - distance_k(A, B) = \max\{k - distance(B), d(A, B)\} \tag{1}$$

$$\text{lrd}_k(A) := 1/\left(\frac{\sum_{B\in N_k(A)} \text{reachability-distance}_k(A, B)}{|N_k(A)|}\right) \tag{2}$$

$$\text{LOF}_k(A) := \frac{\sum_{B\in N_k(A)} \frac{\text{lrd}(B)}{\text{lrd}(A)}}{|N_k(A)|} = \frac{\sum_{B\in N_k(A)} \text{lrd}(B)}{|N_k(A)|}/\text{lrd}(A) \tag{3}$$

Consider the example dataset in Figure 1 (left), the data point at location (5,5) labelled a is moderately outlying. *k-distance* is the distance to the k^{th} furthest point, so if $k = 3$, then k^{th} nearest neighbour of a would be the point at location (1,1) labelled b. If point a is within the k neighbours of point b (See

[1] `https://en.wikipedia.org/wiki/Local_outlier_factor`

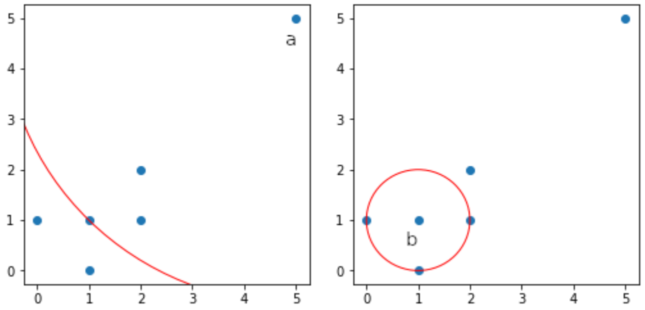

Fig. 1. Calculating the reachability distance.

Figure 1 (right)) the $reachability - distance_k(a, b)$ will be the $k - distance$ of b, the distance to the k^{th} further point (2,1) from b. Otherwise, it will be the real distance of a and b. So in Figure 1 it is **not** within the k neighbours of point b so in this case it is the real distance between a and b.

To get the lrd (local reachability density) for the point a, we will first calculate the reachability distance of a to all its k nearest neighbours and take the average of that number. The lrd is then simply the inverse of that average. Since a is not the third nearest point to b see Figure 1 (right), the reachability distance in this case is always the actual distance. A value greater than one indicates a lower density (thus the instance is outlier). A value one indicates similar density to neighbours. Less than one indicates a higher density. So low density becomes a high LOF score highlighting a instance as an outlier.

One of properties that makes LOF ideal is that the LOF algorithm can work on datasets with clusters of different densities and instance count. As long as the number of k neighbours is below the number of instances in the smallest cluster. This is advantageous since it places little restriction on the dataset to which the weighted clustering algorithm can be applied to. However, one possible drawback to the LOF algorithm is its time complexity of $O(n^2)$, where n is the data size. However, there is existing work speeding up LOF using GPU acceleration. [2]

3 Proposed Methods

We have proposed two novel algorithms based on k-means, Local Outlier Factor Instance Weighted K-Means (LOFIWKM) and Iterative Local Outlier Factor Instance Weighted K-Means (ILOFIWKM). LOFIWKM calculates the weights over the **whole dataset once upon initialisation**, whereas ILOFIWKM calculates the weights **for each cluster upon each iteration**. The weights generated by executing the LOF algorithm are used when calculating means for

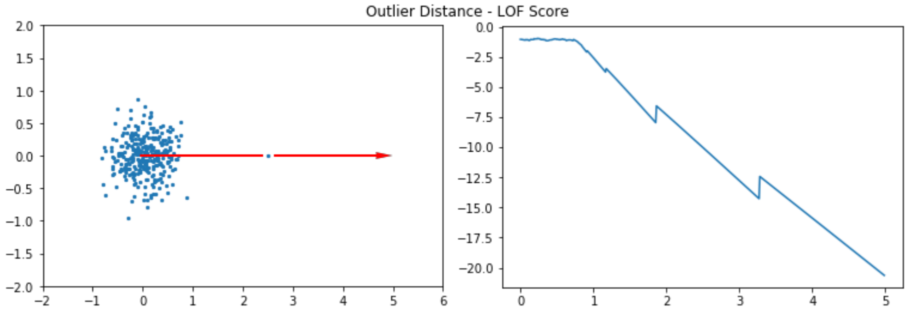

Fig. 2. Demonstrating the LOF scores.

the positions of the new centroids in the k-means algorithm. In Figure 3 the weights are represented by black circles, where the smaller the circle the higher the weight.

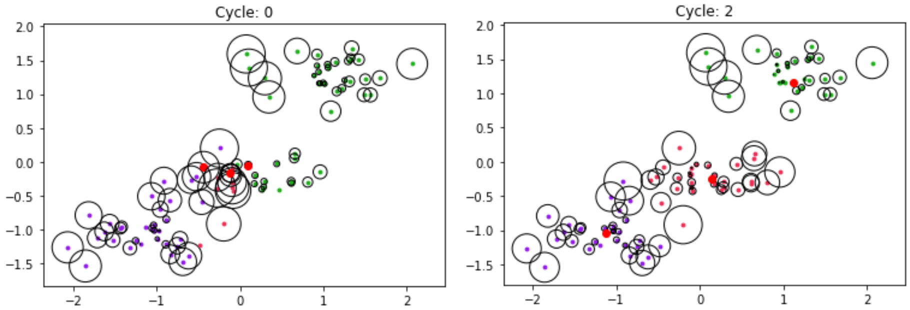

Fig. 3. The I LOF IW K-Means showing different how weights change as the algortihm executes.

More formally, LOFIWKM, starts by calculates the LOF score of every instance considering the whole dataset. Taking the whole dataset into consideration, we highlight outliers relative to the whole dataset. Then as per k-means, centroids are initialised. However, our algorithm uses a weighted random initialisation based on LOF scores and instance positions. Then as per k-means, instances are assigned to the centroids they are closest to. Then as per k-means, the algorithm iterates until converged (there is no more reassignments of instance between clusters) or a max allowed iterations is met. Then, the algorithm calculates the new positions of the centroids based on its' instances, taking a weighted average using normalised LOF scores as weights to moderate the impact of the instance positions on the mean. Then as per k-means instances are assigned the new centroid they are nearest to. Figure 1 shows a formal description of the algorithm where, Dataset of instances = D_i , D = $\{D_1$, D_2 ... D_i $D_N\}$. LOF Scores for each instance in the dataset = LOF_i , LOF = $\{LOF_1$, LOF_2, ... LOF_i, $LOF_N\}$.Clusters corresponding the K value entered = C_k , C = $\{C_1$, C_2 ... $C_k, C_K\}$ a centroid has a position and collection of instances. The number of iterations / k-means cycles = c.

Algorithm 1 LOFIWKM

```
Calculate $LOF$ for $D$
for all $w$ in $LOF$ do
    Assign $\frac{w-min(LOF)}{max(LOF)-min(LOF)}$ to $w^*$
end for
Assign $LOF^*$ to $LOF$
Use $LOF$ weighted random to select $K$ positions from $D$ assign to $C$
for all $i$ in $D$ do
    Assign $i$ to $k$ according to min($dist(i, C)$)
end for
Assign 0 to $c$
while $C$ not converged or $c \leq c^{max}$ do
    for all $k$ in $C$ do
        Assign $\frac{\sum_{k_N}^{k_0} i \cdot w}{\sum_{k_N}^{k_0} w}$ to $k$
    end for
    for all $i$ in $D$ do
        Assign $i$ to $k$ where min($dist(i, C)$)
    end for
    Assign $c + 1$ to $c$
end while
```

ILOFIWKM operates the same as LOFIWKM upto the end of iteration step. Then the algorithm LOF score of every instance running the LOF algorithm per cluster and normalising the LOF scores per cluster. Figure 2 shows a formal description of the algorithm.

4 Experimentation

The purpose of the proposed algorithms is to improve k-means ability to handle outliers. Two variables, count of outliers and range of outliers are experimented with, furthermore the two new algorithms were compared against plain k-means. All experiments are repeated 175 times as the algorithms and the synthetic dataset generation are both stochastic. The outliers are generated using a uniform distribution over a given range, and appended to the dataset. Both synthetic and real world datasets are experimented on. All datasets used included their ground truths and this was used assess clustering accuracy using ARI (Adjusted Random Index). The ARI computes a similarity measure between clusterings by considering all pairs of instances and counting pairs that are assigned in the same or different clusters in the predicted and true clusterings.

The experiments use the scikit-learn libraries where possible [13] to speed development and aid repeatability. Most notably scikit-learn's LOF implementation was used for calculating the measures of outlyingness. Furthermore, scikit-

Algorithm 2 ILOFIWKM

```
Calculate $LOF$ for $D$
for all $w$ in $LOF$ do
    Assign $\frac{w-min(LOF)}{max(LOF)-min(LOF)}$ to $w^*$
end for
Use $LOF$ weighted random to select $K$ positions from $D$ assign to $C$
for all $i$ in $D$ do
    Assign $i$ to $k$ according to min$(dist(i, C))$
end for
Assign 0 to $c$
while $C$ not converged or $c \leq c^{max}$ do
    for all $k$ in $C$ do
        Assign $\frac{\sum_{k_N}^{k_0} i \cdot w}{\sum_{k_N}^{k_0} w}$ to $k$
    end for
    for all $i$ in $D$ do
        Assign $i$ to $k$ where min$(dist(i, C))$
    end for
    for all $C$ do
        Partially recalculate $LOF$ for $i$ in $k$
        for all $w$ in $k$ do
            Assign $\frac{w-min(LOF)}{max(LOF)-min(LOF)}$ to $w^*$
        end for
    end for
    Assign $c + 1$ to $c$
end while
```

learn's Blobs Dataset Generator, Standard Scaler, PCA and Adjusted Random Index were utilised. For the k-means algorithm, our own python implementation was used and updated to create the novel algorithms. This ensures that the only difference between k-means and our instance weighted k-means algorithms was changes described in this paper.

For each run of the experiments the dataset was regenerated. The synthetic blob datasets (noise = 0.3) are generated with 90 instances, 2 features and 3 clusters of equal sizes. For the outlier count various amounts of outliers were tested: 5, 10, 15, 20, 25. For the outlier range experiment, various ranges were tested: 20, 40, 60, 80, 100, the (dataset with outliers spans a range of 10 in either axis).

Also experiments are conducted on a real world dataset containing 210 instances, 7 features and 3 clusters the measurements are of damaged wheat kernels of 3 different varieties. [4] The dataset was obtained via the UCI Machine Learning Repository [5].

4.1 Outlier Count and Range Synthetic Dataset Results

Figure 6 shows positive results for density based instance weighted clustering. The instance weighted algorithms were able to achieve better clustering accuracy than k-means. In Figure 6 on the left, the impact of increasing the count outliers can be seen. In Figure 6 on the right, the impact of creating increasingly distance outliers is shown. The accuracy of k-means quickly deteriorates as the outliers

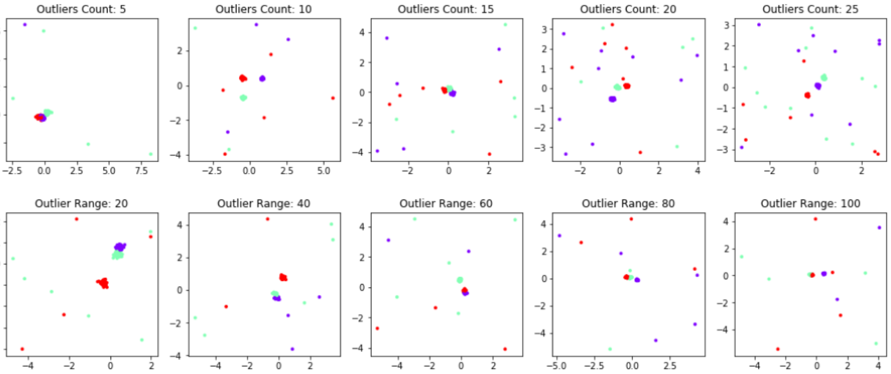

Fig. 4. A sample of the blobs datasets.

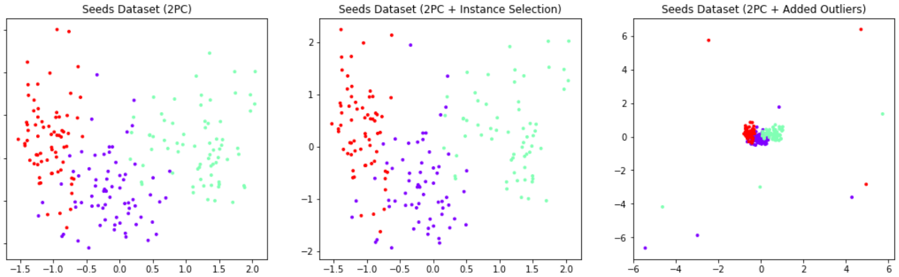

Fig. 5. The Seeds datasets.

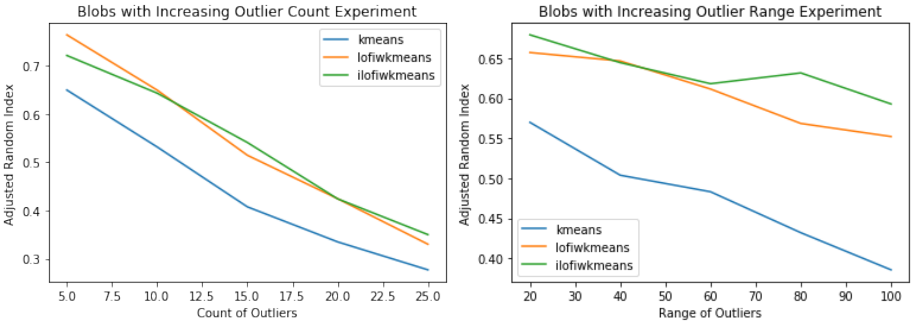

Fig. 6. Adjusted Random Index scores for the Blobs Datasets

get distant. The LOFIWKM and ILOFIWKM algorithms are not as strongly effected by the presence of increasing distant outliers. Across both experiments, a minimal gain can be seen in using the iterative version, ILOFIWKM.

4.2 Real World Dataset Results

In Figure 7 instance weighting is compared with instance selection. The three groups of columns show different conditions of the dataset. Left shows the results with dataset having additional synthetic outliers added. Center shows the results of the algorithms having the outliers removed (i.e. Instance Selection). The outliers were removed using the LOF algorithm with the same neighbours count as in LOFIWKM algorithms (neighbours = 5). The outlier contamination value was set to 0.1 to remove the most outlying 10% of the dataset. Finally right shows results for the original dataset. We can compare instance weighting to instance selection by comparing the right group's LOFIWKM result to the central group's k-means result. It can be seen that instance weighting slightly outperformed instance selection in ARI score, however only slightly. It can also been seen that ILOFIWKM and LOFIWKM provided a large benefit on the original dataset and the with additional outliers added compared to k-means. Our results mirror tests on the seeds dataset in Lei Gu's research were weighting also enhance clustering accuracy. [7]

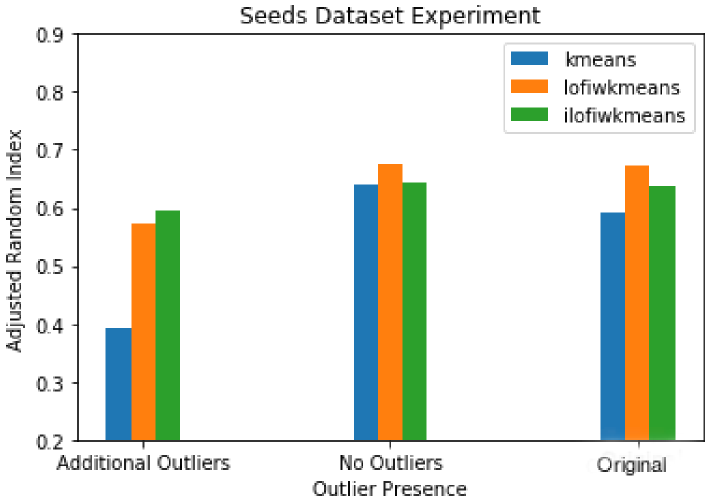

Fig. 7. Adjusted Random Index scores for the Seeds dataset

5 Conclusion and Future Work

In conclusion, this paper has shown that instance weighting can help mitigate the effect of outliers on both a synthetic and a real world dataset.

In this paper we only investigated k-means which has a hard membership function and the LOF algorithm. However, there is likely more useful combinations to be found. Hammerly and Elkan found that varying weights did improve the performance of hard membership function algorithms (i.e. k-means).[8] However, Nock and Neilsen's research confirmed instance weighting to be more advantageous for clustering algorithms with soft membership functions such as fuzzy k-means.[12] A future work of this paper should be to investigate soft membership function algorithms.

Our modifications were made to a basic version of the k-means algorithm. However, it would be possible to combine the LOF instance weighting with a version of k-means which has more optimisations or is being used in conjunction with wrapper functions. Furthermore, with instance weighting there is potential to simultaneously apply multiple instance weights which could prove to increase robustness or accuracy.

The time complexity LOFIWKM is equivalent to LOF instance selection $O(n^2)$ plus k-means $O(n)$, however ILOFIWKM is significantly more costly taking the complexity of k-means plus the execution of the LOF algorithm per cluster (for each clusters instances), further experimentation may prove that ILOFIWKM may be not suitable for large datasets, without optimisation of the LOF algorithm, such as, the research by Alshawabkeh et al.[2]

Future work also includes testing the algorithms with a more thorough outlier generation process. In this paper we added instances from a uniform distribution, centred on the dataset. This had two disadvantages, firstly this method possibly does not highlight one of the advantages of instance weighting. Instance weighting has the potential to retain some of the information an outlier presents, since the "outliers" are uniformly random these benefits are negated. Secondarily, it is possible that when generating the "outliers" that a proportion of fall within a normal range and end up not being outliers.

Currently our algorithm requires parameter selection of k clusters and the size of the LOF neighbourhood. Other algorithms [14][7] require some parameter selection with the exception of the state of the art.[6] It would be clearly better to not require the parameter selection and it does seem possible to automate the selection of these parameters.

References

1. Aggarwal, C.C., Reddy, C.K.: Data Clustering: Algorithms and Applications. Chapman Hall CRC, 1st edn. (2013)

2. Alshawabkeh, M., Jang, B., Kaeli, D.: Accelerating the local outlier factor algorithm on a gpu for intrusion detection systems. In: Proceedings of the 3rd Workshop on General-Purpose Computation on Graphics Processing Units. pp. 104–110 (2010)
3. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: Lof: identifying density-based local outliers. In: ACM sigmod record. vol. 29, pp. 93–104. ACM (2000)
4. Charytanowicz, M., Niewczas, J., Kulczycki, P., Kowalski, P.A., Łukasik, S., Żak, S.: Complete gradient clustering algorithm for features analysis of x-ray images. In: Information technologies in biomedicine, pp. 15–24. Springer (2010)
5. Dua, D., Graff, C.: Uci machine learning repository (2017), `http://archive.ics.uci.edu/ml`
6. Efimov, K., Adamyan, L., Spokoiny, V.: Adaptive nonparametric clustering. IEEE Transactions on Information Theory **65**(8), 4875–4892 (2019)
7. Gu, L.: A novel sample weighting k-means clustering algorithm based on angles information. In: 2016 International Joint Conference on Neural Networks (IJCNN). pp. 3697–3702. IEEE (2016)
8. Hamerly, G., Elkan, C.: Alternatives to the k-means algorithm that find better clusterings. In: Proceedings of the eleventh international conference on Information and knowledge management. pp. 600–607 (2002)
9. Hawkins, D.M.: Identification of outliers, vol. 11. Springer (1980)
10. Jain, A.K.: Data clustering: 50 years beyond k-means. Pattern recognition letters **31**(8), 651–666 (2010)
11. Lloyd, S.: Least squares quantization in pcm. IEEE transactions on information theory **28**(2), 129–137 (1982)
12. Nock, R., Nielsen, F.: On weighting clustering. IEEE transactions on pattern analysis and machine intelligence **28**(8), 1223–1235 (2006)
13. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
14. Yu, J., Yang, M.S., Lee, E.S.: Sample-weighted clustering methods. Computers & Mathematics with Applications **62**(5), 2200–2208 (2011)
15. Zhang, B.: Generalized k-harmonic means. Hewlett-Packard Laboratoris Technical Report (2000)