Medoid-Shift for Noise Removal to Improve Clustering

Pasi Fränti(✉) and Jiawei Yang

School of Computing, University of Eastern Finland, Joensuu, Finland
pasi.franti@uef.fi

Abstract. We propose to use medoid-shift to reduce the noise in data prior to clustering. The method processes every point by calculating its *k-nearest neighbors* (k-NN), and then replacing the point by the medoid of its neighborhood. The process can be iterated. After the data cleaning process, any clustering algorithm can be applied that is suitable for the data.

Keywords: Clustering · Noise removal · Outlier detection

1 Introduction

Outliers are points that deviate from the typical data. They can be anomalies that represent significant information that are wanted to be detected such as credit card frauds [1], and they can affect statistical conclusions based on significance tests [14]. In *clustering*, outliers are usually considered harmful as they can affect the clustering quality. It is therefore desired that the outliers can be detected and removed.

Detection of outliers is typically considered as a separate pre-processing step prior to clustering. Another approach is to perform the clustering first, and then label those points as outliers which did not fit into any cluster, see Fig. 1. DBSCAN is an example of this kind of approach [5]. However, it is a kind of *chicken-or-egg problem*: removing outliers first would improve the clustering, but on the other hand, performing clustering first would make the outlier detection easier.

Outlier removal can also be integrated in the clustering directly by modifying the cost function [13]. The benefit is that the clustering can provide more information to make better decision whether a point is an outlier, and yet, better clustering can be obtained as the outliers can be removed before the final clustering. In [9], outlier removal was integrated within the k-means algorithms so that the farthest points from their centroid were removed, and k-means is then repeated for the modified dataset. This iterative process removes not only outliers but also border points with the hope of obtaining more stable cluster centroids.

In this paper, we propose a new approach where we do not remove the outliers at all. Instead, we pre-process the data so that the effect of potential outliers will be reduced. The key idea is similar to smoothing in image processing, where the pixel values are altered based on the local window. We do the same in the feature space by

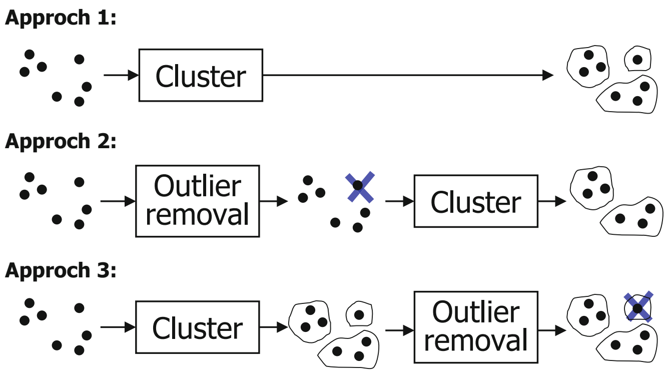

Fig. 1. Three approaches how to deal with outliers: (1) ignore; (2) remove outliers by pre-processing; (3) detect outliers from the clustering result.

finding *k-nearest neighbors* (k-NN) for every pixel, and then processing the pixel value in this neighborhood. We apply replacing the original value by the medoid value of its neighbors.

2 Existing Work and Their Limitations

Density-based approaches analyze the neighborhood of the points, and consider low density points as outliers. For example, a data point is marked as an outlier if there are at most k points are within a given distance d [11]. Another method calculates the k-nearest neighbors (k-NN), and use the distance to the k^{th} neighbor [15], or the average distance to all k neighbors [10] as the outlier indicator. Points with the largest distance are considered as outliers. However, these approaches do not take into account that clusters may have significantly different densities.

Topology of the neighborhood has also been considered. In [3], undirected graph is constructed from the k-NN pointers. A link between two points is created only if they are mutual neighbors of each others. Connected components are considered as data points, and isolated points as outliers. In ODIN [10], the *indegree* of the nodes in the k-NN graph is used as the indicator of outliers because isolated points appear less frequently in the k-neighborhood of others.

However, if the number of outliers is large, traditional outlier removal techniques might not be efficient. In case when the number of outliers is of the same order of magnitude than the size of data, or even larger, it becomes significantly more challenging to differentiate outliers from normal points, see Fig. 2. Removing low density points might work into certain extent but setting the correct threshold is not trivial, especially if the data points have varying densities.

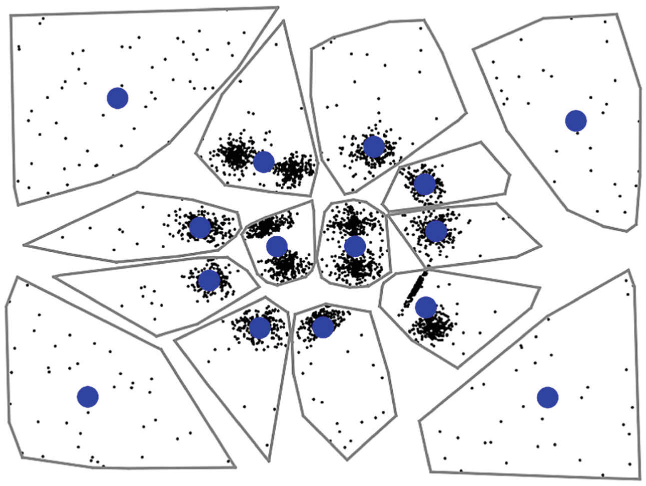

Fig. 2. Example of noisy data and how it affects clustering.

3 Medoid-Shift Noise Removal

In this work, we consider an alternative approach by formulating the outlier detection as noise removal problem. Instead of trying to detect whether a point is outlier, we process the data points locally based on the neighborhood. We propose a method called *medoid-shift noise removal*. The idea is to find k-nearest neighbors, and then replace the point by the medoid of its *k* neighbors. This will effectively remove the effect of isolated points and reduce the effect of abnormal points in the border areas of the clusters.

The idea is closely related to mean-shift filtering used in image processing [4], and mean-shift clustering algorithm [19]. The first one takes pixels value and its coordinates as the feature vector, and transforms each feature towards the mean of its neighbors. It has been used for detecting fingerprint and contamination defects in multicrystalline solar wafers [17]. The idea resembles also low-pass and median filtering used for image denoising.

The difference to mean-shift clustering is that we apply the shift merely for noise removal, and that we do not iterate the process until convergence. In other words, we do not force to use mean-shift clustering algorithm but the mean-shift process is merely a pre-processing stage. The choice of the algorithm is left as a separate question. This is because the mean-shift process itself can be good for removing noisy point, but its clustering performance is far from perfect.

3.1 Method Description

The core algorithm applies the following three steps for every point *x*:

1. Calculate the *k*-nearest neighbors *kNN*(*x*) of the point *x*.
2. Calculate the mean of the neighbors.
3. Replace point *x* by the medoid *M*.

The algorithm modifies the data so that the effect of the noisy points is minimized. It is implemented as separate pre-processing step so that it is independent on the clustering method applied. The process can be iterated several times to have stronger noise removal effect. The number of iterations is the parameter.

The iterative variant is similar to the CERES algorithm in [9] where most remote points are iteratively removed in each cluster. The difference is that the outliers are chosen based on the distance to their cluster centroid in the intermediate clustering solution. It is therefore possible that points can be falsely removed if the cluster is not correctly determined. In our method, we remove the points based on their local neighborhood regardless of the clustering process. Figure 3 demonstrates the process.

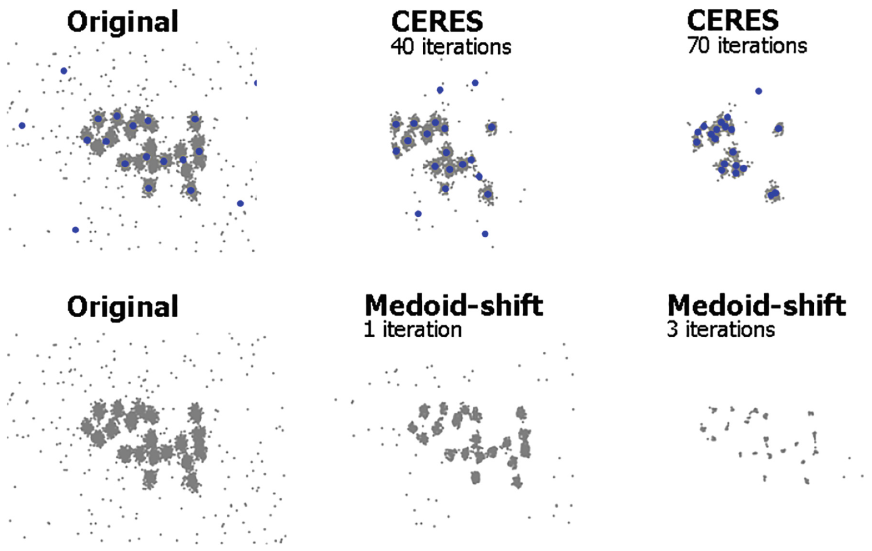

Fig. 3. Example of the iterative processed of CERES [9] and the proposed medoid-shift.

3.2 Mean or Median?

Both *mean* and *median* have been used in the mean-shift clustering concept [16, 19]. The benefit of using mean is that it is trivial to calculate for numeric data. Its main disadvantage is that it can cause blurring effect where very noisy points bias the calculation of the mean of clean points as well. Median is less sensitive to this because single noisy point in the neighborhood may not affect the calculation of the median at all, and it has therefore less effect on the clean points. Median can also reach its *root signal* [18] when repeated until converge.

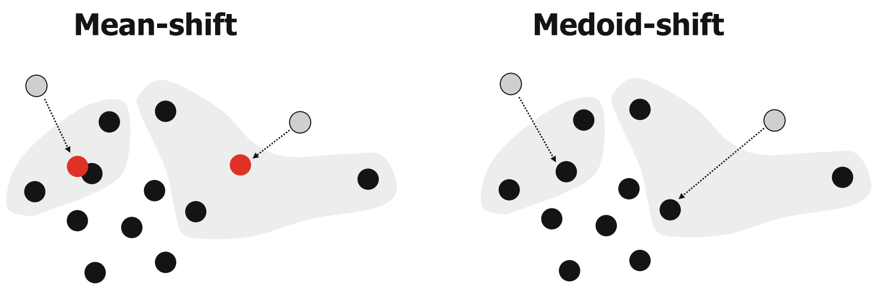

Fig. 4. Effect of mean (left) and median (right) of two sample points in a noisy neighborhood.

However, with multivariate data it is not obvious how median is defined. We use the *medoid*, which is the point in the set that has minimal total distance to all other points in the set. A disadvantage of medoid is that its calculation can be more time-consuming. A toy example of using mean and median is shown in Fig. 4.

4 Experiments

We test the proposed method with two clustering algorithms:

- K-means [6]
- Random swap [7].

Both algorithms aim at minimizing sum-of-squared errors. The first one is commonly used but does not always find the correct clustering solution even with the clean data. Random swap, however, finds the correct clustering result with all our datasets when no noise added. Therefore, after noise added, any errors the clustering algorithm makes can be addressed to the noise. For the effect of the noise removal, we compare the following methods:

- LOF [2]
- ODIN [10]
- Mean-shift (proposed)
- Medoid-shift (proposed).

Table 1. Datasets used in the experiments. (http://cs.uef.fi/sipu/datasets/)

Dataset	Size	Clusters
S1	5000	15
S2	5000	15
S3	5000	15
S4	5000	15
A1	3000	20
A2	5250	35
A3	7500	50
Unbalance	6500	8

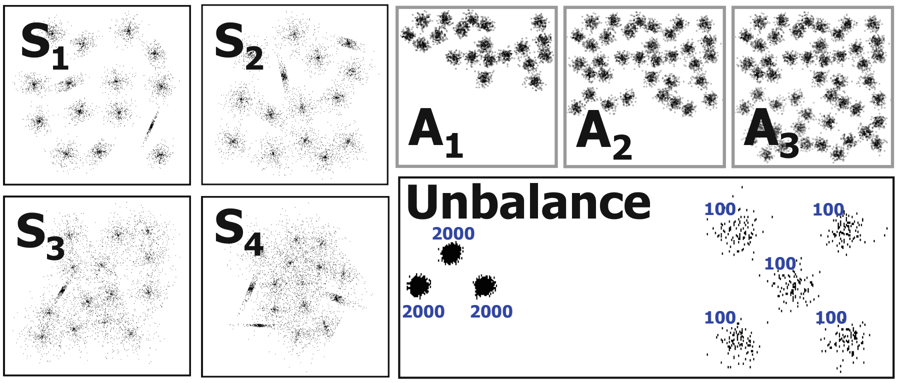

Fig. 5. Datasets used in the experiments.

We evaluate the clustering quality by the following two measures:

- CI = Centroid index [8]
- NMI = Normalized mutual information [12].

Centroid index is cluster level measure, which counts how many cluster centroids are wrongly located. Value CI = 0 indicates that all cluster centroids are correct with

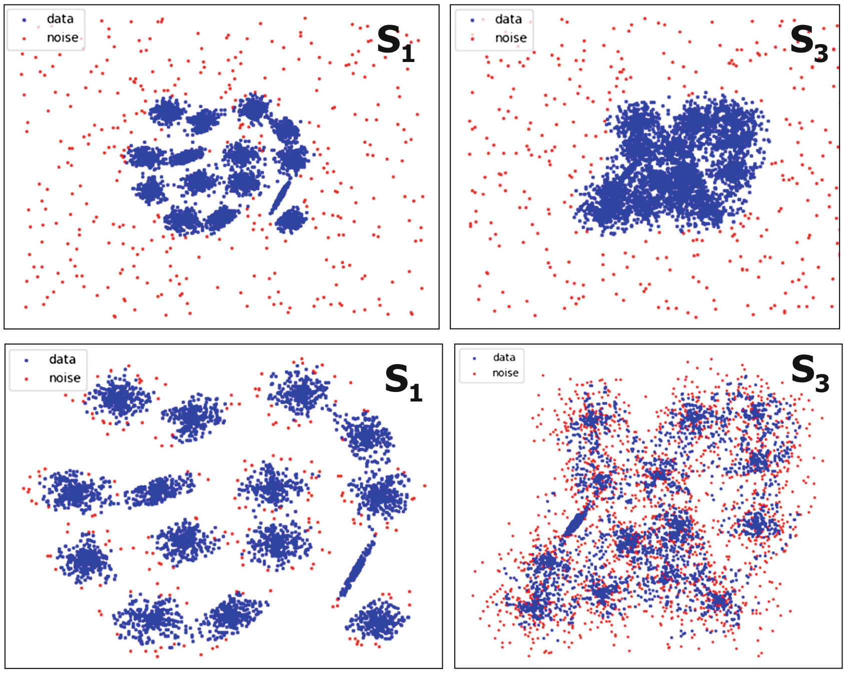

Fig. 6. Noisy datasets S1 (left) and S3 (right) with 7% noise level.

respect to the ground truth. *Normalized mutual information* is a point-level measure, which calculates the amount of information the clustering shares with the ground truth. Value 1 indicates perfect match to that of the ground truth.

4.1 Datasets

We use the eight benchmark datasets summarized in Table 1. The datasets are also shown in Fig. 5. The S sets have varying level of cluster overlap, A sets have varying number of clusters, and unbalance set have clusters with different densities.

Table 2. Summary of the point-level clustering results (NMI).

Noise type 1

Prepro-cess:	Combination	S1	S2	S3	S4	A1	A2	A3	Un	Av.
none	RS	0.87	0.82	0.78	0.67	0.84	0.87	0.88	0.61	0.75
	KM	0.85	0.82	0.78	0.67	0.83	0.87	0.89	0.90	0.82
noise removal	LOF+RS	0.85	0.84	0.76	0.69	0.88	0.90	0.91	0.70	0.81
	LOF+KM	0.90	0.84	0.81	0.67	0.89	0.90	0.91	**0.91**	0.85
	ODIN+RS	0.85	0.80	0.76	0.66	0.81	0.85	0.86	0.70	0.78
	ODIN+KM	0.85	0.79	0.76	0.62	0.80	0.85	0.86	0.84	0.80
medoid-shift	medoid+RS	**0.93**	**0.89**	0.83	0.68	**0.91**	**0.92**	**0.92**	0.90	**0.87**
	medoid+KM	0.91	0.88	**0.84**	**0.69**	0.90	**0.92**	**0.92**	0.89	**0.87**
	mean+RS	0.87	0.83	0.77	0.67	0.87	0.89	0.90	0.76	0.82
	mean+KM	0.88	0.83	0.78	0.66	0.87	0.89	0.89	0.90	0.84

Noise type 2

Prepro-cess:	Combination	S1	S2	S3	S4	A1	A2	A3	Un	Av.
none	RS	0.80	0.76	0.70	0.70	0.77	0.82	0.84	**0.81**	0.77
	KM	0.80	0.76	0.70	0.66	0.78	0.81	0.83	0.79	0.77
noise removal	LOF+RS	0.79	0.74	0.68	0.68	0.75	0.82	0.83	0.64	0.74
	LOF+KM	0.79	0.74	0.67	0.65	0.77	0.82	0.83	0.82	0.76
	ODIN+RS	0.78	0.74	0.68	0.67	0.68	0.67	0.63	0.77	0.70
	ODIN+KM	0.78	0.71	0.68	0.64	0.75	0.79	0.81	0.75	0.74
medoid-shift	medoid+RS	**0.86**	**0.84**	**0.77**	0.72	**0.85**	**0.86**	**0.87**	0.80	**0.82**
	medoid+KM	**0.86**	**0.84**	**0.77**	**0.73**	**0.85**	**0.86**	**0.87**	0.80	**0.82**
	mean+RS	0.80	0.76	0.68	0.69	0.81	0.83	0.86	0.63	0.76
	mean+KM	0.83	0.76	0.68	0.66	0.82	0.85	0.85	0.78	0.78

4.2 Noise Models

We consider two types of noise:

1. Random noise
2. Data-dependent noise.

In the first case, uniformly distributed random noise is added to the data. Random values are generated in each dimension between $[x_{\text{mean}} - 2 \cdot \textit{range}, x_{\text{mean}} + 2 \cdot \textit{range}]$, where x_{mean} is the mean of all data points, and *range* is the maximum distance of any point from the mean: $\textit{range} = \max(|x_{\max} - x_{\text{mean}}|, |x_{\text{mean}} - x_{\min}|)$. The amount of noise is 7% of data size. In the second case, 7% of the original points are copied and moved to random direction. Noisy datasets are shown in Fig. 6.

Table 3. Summary of the cluster-level results (CI).

Noise type 1

Preprocess:	Combination	S1	S2	S3	S4	A1	A2	A3	Un	Av.
none	RS	4	4	4	3	6	7	14	4	5.8
	KM	4	3	3	3	5	8	13	**2**	5.1
noise removal	LOF+RS	3	4	3	2	5	5	9	**2**	4.1
	LOF+KM	2	4	3	3	4	6	10	3	4.4
	ODIN+RS	4	4	4	3	5	7	14	3	5.5
	ODIN+KM	4	4	4	4	6	8	11	**2**	3.5
medoid-shift	medoid+RS	**0**	**0**	2	2	**0**	**2**	4	**2**	1.5
	medoid+KM	1	1	**1**	**1**	**0**	**2**	**3**	**2**	**1.4**
	mean+RS	4	3	6	3	3	7	13	3	5.3
	mean+KM	4	4	4	2	4	7	14	**2**	5.1

Noise type 2

Preprocess:	Combination	S1	S2	S3	S4	A1	A2	A3	Un	Av.
none	RS	4	4	4	3	5	7	13	3	5.4
	KM	4	4	4	3	4	6	12	3	5.0
noise removal	LOF+RS	4	4	4	3	5	7	13	**2**	5.3
	LOF+KM	4	4	4	3	5	7	11	**2**	5.0
	ODIN+RS	4	4	4	4	5	8	13	3	5.6
	ODIN+KM	4	4	4	3	4	7	11	**2**	4.8
medoid-shift	medoid+RS	**0**	**0**	**1**	2	**0**	**1**	5	3	1.5
	medoid+KM	**0**	**0**	**1**	**1**	**0**	**1**	**4**	**2**	**1.1**
	mean+RS	4	4	4	3	4	7	11	3	5.0
	mean+KM	2	4	4	3	3	5	12	3	4.5

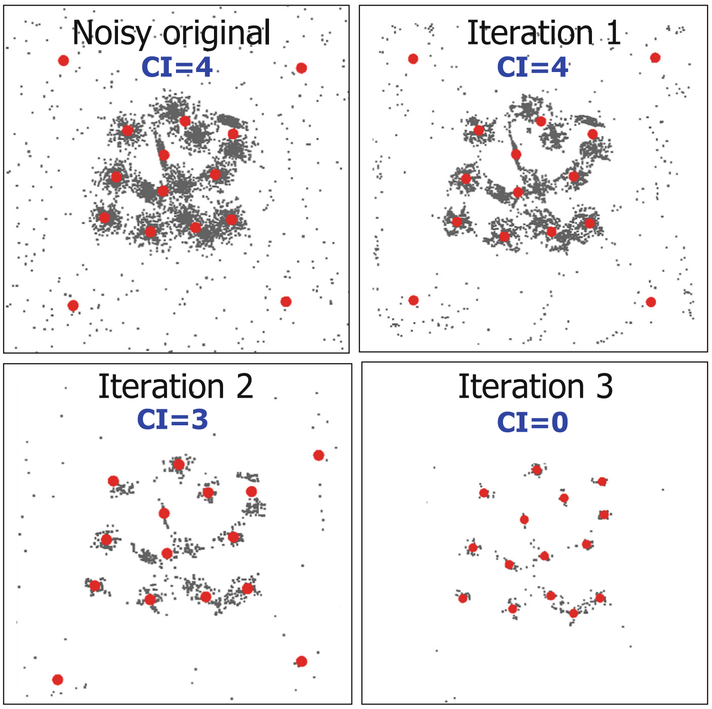

Fig. 7. Visualization of the medoid-shift noise removal on the noisy S1 dataset.

4.3 Results

The clustering results are summarized in Tables 2 and 3, both at the point level (NMI) and at the cluster level (CI). The point level results show that the medoid-shift improves both k-means and random swap clustering result from about NMI = 0.82 to 0.87. The medoid-shift is significantly better than the mean-shift, which is not really effective with this amount of noise. Visual examples are shown in Fig. 7.

The cluster-level results reveal that the task is challenging. The medoid-shift finds the correct clustering (CI = 0) only with the three easier datasets: S1, S2 and A1. The competitive outlier removal methods fail almost in all cases. The density-based method (LOF+KM) manages to improve the k-means clustering result in case of A2 and A3 datasets.

The choice of the clustering algorithm also matters but less with the noisy data. Random swap optimizes the cluster centroids better whereas k-means makes fails to put centroids in the low density areas. However, since noise has lower density, it mostly cancels the inferior optimization capability of k-means, and overall, both algorithms perform equally in practice. To sum up, the noise reduction is more significant than the optimization capability of the clustering algorithm.

5 Conclusions

Mean-shift and medoid-shift were proposed as a separate noise removal process before clustering. The results show that medoid-shift is more effective than mean-shift, and that they both improves k-means clustering. Best results were achieved using three iterations of the process. The proposed approach outperforms two existing outlier removal methods: LOF and ODIN.

All methods have parameters to tune. The parameter *k* is needed in all methods. In addition, mean-shift and medoid-shift have the number of iterations (our recommended value is 3). LOF has the top-*N* parameter, which we set to 3% of the data size. ODIN has the threshold parameter, which we set to 1. Our future work includes studying the effect of parameters, and how their effect could be reduced.

References

1. Ali, A.M., Angelov, P.: Anomalous behaviour detection based on heterogeneous data and data fusion. Soft Comput. 1–15 (2018). https://doi.org/10.1007/s00500-017-2989-5
2. Breunig, M.M., Kriegel, H., Ng, R.T., Sander, J.: LOF: identifying density-based local outliers. In: ACM SIGMOD International Conference on Management of Data, vol. 29, no. 2, pp. 93–104, May 2000
3. Brito, M.R., Chavez, E.L., Quiroz, A.J., Yukich, J.E.: Connectivity of the mutual k-nearest-neighbor graph in clustering and outlier detection. Stat. Prob. Lett. **35**(1), 33–42 (1997)
4. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. IEEE Trans. Pattern Anal. Mach. Intell. **24**(5), 603–619 (2002)
5. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: International Conference on Knowledge Discovery and Data Mining, KDD, pp. 226–231 (1996)
6. Forgy, E.: Cluster analysis of multivariate data: efficiency vs. interpretability of classification. Biometrics **21**, 768–780 (1965)
7. Fränti, P.: Efficiency of random swap clustering. J. Big Data **5**(13), 1–29 (2018)
8. Fränti, P., Rezaei, M., Zhao, Q.: Centroid index: cluster level similarity measure. Pattern Recognit. **47**(9), 3034–3045 (2014)
9. Hautamäki, V., Cherednichenko, S., Kärkkäinen, I., Kinnunen, T., Fränti, P.: Improving k-means by outlier removal. In: Kalviainen, H., Parkkinen, J., Kaarna, A. (eds.) SCIA 2005. LNCS, vol. 3540, pp. 978–987. Springer, Heidelberg (2005). https://doi.org/10.1007/11499145_99
10. Hautamäki, V., Kärkkäinen, I., Fränti, P.: Outlier detection using k-nearest neighbour graph. In: International Conference on Pattern Recognition, ICPR 2004, Cambridge, UK, pp. 430–433, August, 2004
11. Knorr, E.M., Ng, R.T.: Algorithms for mining distance-based outliers in large datasets. In: International Conference on Very Large Data Bases, New York, USA, pp. 392–403 (1998)
12. Kvålseth, T.O.: Entropy and correlation: some comments. IEEE Trans. Syst. Man Cybern. **17**(3), 517–519 (1987)
13. Ott, L., Pang, L., Ramos, F., Chawla, S.: On integrated clustering and outlier detection. In: Advances in Neural Information Processing Systems, NIPS, pp. 1359–1367 (2014)

14. Pollet, T.V., van der Meij, L.: To remove or not to remove: the impact of outlier handling on significance testing in testosterone data. Adapt. Hum. Behav. Physiol. **3**(1), 43–60 (2017)
15. Ramaswamy, S., Rastogi, R., Shim, K.: Efficient algorithms for mining outliers from large data sets. In: ACM SIGMOD Record, vol. 29, no. 2, pp. 427–438, June 2000
16. Sheikh, Y.A., Khan, E.A., Kanade, T.: Mode-seeking by medoidshifts. In: IEEE International Conference on Computer Vision, ICCV, Rio de Janeiro, Brazil, October 2007
17. Tsai, D.-M., Luo, J.-Y.: Mean shift-based defect detection in multicrystalline solar wafer surfaces. IEEE Trans. Ind. Inf. **7**(1), 125–135 (2011)
18. Yin, L., Yang, R., Gabbouj, M., Neuvo, Y.: Weighted median filters: a tutorial. IEEE Trans. Circ. Syst. II: Analog Digit. Signal Process. **43**(3), 157–192 (1996)
19. Cheng, Y.: Mean shift, mode seeking, and clustering. IEEE Trans. Pattern Anal. Mach. Intell. **17**(8), 790–799 (1995)