Fuzzy Systems and Data Minin[g]
A.J. Tallón-Ballesteros and K. Li (Ed[s.])
IOS Press, 20[18]

doi:10.3233/978-1-61499-927-0-2[08]

Mean-Shift Outlier Detection

Jiawei YANG[a], Susanto RAHARDJA[b] and Pasi FRÄNTI [a,1]
[a] *School of Computing, University of Eastern Finland, Joensuu 80101, Finland,*
[b] *School of Marine Science and Technology, Northwestern Polytechnical Universi[ty],*
127 West Youyi Raod, Xi'an, Shaanxi 710072, China

Abstract. We propose mean-shift to detect outlier points. The method processed every point by calculating its *k-nearest neighbors* (k-NN), and then shifting the point to the mean of its neighborhood. This is repeated three times. The bigger the movement, the more likely the point is an outlier. Boundary points are expected to move more than inner points; outliers more than boundary. The outlier detection is then a simple thresholding based on standard deviation of all movements. Points that move more than that are detected as outliers. The method outperforms all compared outlier detection methods.

Keywords. Boundary point, noise removal, outlier removal, mean-shift

Introduction

Outliers are points that deviate from the typical data. They can represent signifi[cant] information that are wanted to be detected such as fraud detection, public health, [and] network intrusion [1], and they can affect statistical conclusions based on significan[ce] tests [2]. Outliers can also be noise points who might harm the data analysis proces[s. In] any case, it is desired that the outliers can be detected.

Outlier detection approaches fall roughly into *global* and *local* outlier models [1]. The global methods make a binary decision whether an observation is outlier wh[ile] local methods assign a score to each point. This score indicates how likely a point is [an] outlier. The method then retrieves the top-n rankings as outliers, which gives m[ore] flexibility how to interpret the data.

However, even the local outlier models need to select the top-n parameter; h[ow] many points are chosen as outliers. In this paper, we propose a new local outl[ier] detection method in which such parameter is not needed. We first calculate *outl[ier] score* for all data points, and then calculate the standard deviation from the distribu[tion] of all outlier scores to serve as a global threshold. This can be automati[cally] determined.

The way how we calculate the outlier scores is based on the idea presented in [4]. The effect of the noise is reduced by applying few iterations of medoid-shifting [as] follows. Firstly, we find *k-nearest neighbors* (k-NN) for every data point. Then [we] replace the original points by the mean (or medoid) value of its neighbors. This pro[cess] is iterated few times. This iterative process is demonstrated in Figure 1. Howe[ver,]

[Ins]tead of replacing the original point by its shifted version as in [4], we merely [c]alculate how much it was moved, i.e. the distance between the original point and its [sh]ifted location. This distance defines the outlier score. Example of the outlier scores is [sho]wn in Figure 2.

The accuracy of the proposed method is far better than any other we tested. The [m]ethod is robust as it provides the same result with the same global threshold [pa]rameter whereas all comparative methods use the a priori knowledge of the amount [o]f noise. Unlike the other methods, it has only one parameter to set (the size of [ne]ighborhood *k*). The proposed method is also easy to understand and simple to [im]plement.

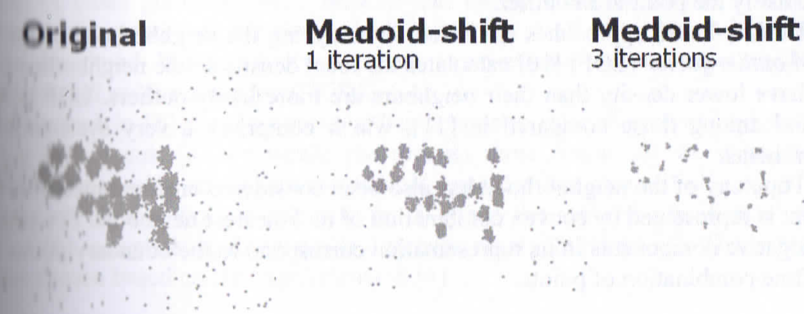
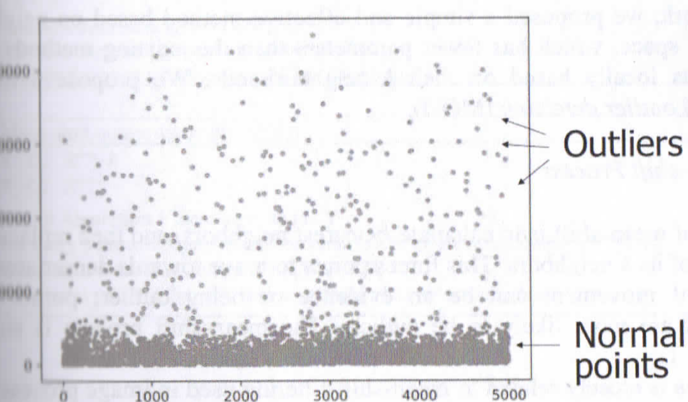


Figure 1. Example of the iterative process of the medoid-shift [4].



[Fig]ure 2. Outlier scores (y-axis) for S1 dataset: true data points (gray) and noise points (red). [The re]sults are for the proposed DOD method (using medoid).

[1] Existing Work and Their Limitations

[Min]imum *covariance determinant* (MCD) [5] is based on statistical test with the [ass]umption that the true data objects follow a (known) distribution and occur in a high [prob]ability region of this model. Outliers are expected to deviate strongly from this

Distance-based approaches [6, 7, 8] are based on the assumption that true data objects have a dense neighborhood whereas outliers lie far apart from their neighbors. For example, in [9] a data point is marked as an outlier if there are at most k points within a given distance. The method in [6] calculates the k-nearest neighbors (k-NN) and use the distance to the k^{th} neighbor. A slightly modified variant in [7] uses the average distance to all the k neighbors. Points with largest distances are considered as outliers.

However, distance-based outlier detection models have problems if the data has areas of varying densities. Instead of using the distance values directly, a method called ODIN [7] analyzes the relationship of the point. For a given point, it calculates how many other points consider it as their k-nearest neighbor. The smaller the value, the more likely the point is an outlier.

Density-based approaches are based on analyzing the neighborhood of the points. *Local outlier factor* (LOF) [10] calculates the local density of the neighborhood. Points that have lower density than their neighbors are more likely outliers. LOF is the best method among those compared in [11], which comprises a very systematic set of experiments.

Topology of the neighborhood has also been considered in recent methods. In [12], a point is represented by convex combination of its k-nearest neighbors. For each point, the negative components in its representation correspond to the boundary points among its affine combination of points.

2 Mean-shift Outlier Detection

In this work, we proposed a simple and effective method based on neighborhoods in Euclidean space, which has fewer parameters than the existing methods. We analyse data points locally based on their k-neighborhoods. We propose a method called *mean-shift outlier detection* (MOD).

2.1 Mean-shift Process

The idea of mean-shift is to calculate k-nearest neighbors, and then replace the point by the mean of its k neighbors. This forces points to move towards denser areas. Hence the distance of movement can be an evidence of being outlier; points with greater movement are more likely to be outliers. The mean-shift process is summarized in Algorithm 1.

The idea is closely related to mean-shift filtering used in image processing [13], and mean-shift clustering algorithm [14]. The first one takes pixel 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 [15]. The idea resembles also low-pass and median filtering used for image denoising.

2.2 Mean-shift for Outlier Detection

Mean-shift clustering [14] iterates the process until convergence. However, since we are not clustering the data but aim at finding outliers, we use the processing result

merely for analysis purpose. In specific, we calculate the location of the points before and after the shifting. This difference is used as the outlier score, called *mean-shift score*.

The final step is to detect the outliers. The key idea is to analyze the distribution of the obtained mean-shift scores. In specific, we calculate the standard deviation (SD) of all the scores in the dataset. This value is then used as global threshold; any point with bigger outlier score than SD is marked as an outlier. The pseudo code of the algorithm is summarized in Algorithm 2.

Both *mean* and *median* have been used in the mean-shift clustering concept [14, 16]. The benefit of using mean is that it is trivial to calculate for numeric data. However, it can cause blurring because a very noisy point can bias the calculation of the mean of clean points as well. *Medoid* can be more robust in this sense. It is calculated as the point that has minimal total distance to all other points in the same k-NN neighborhood. We call the two variants as *mean-shift outlier detection* (MOD) and *medoid-shift outlier detection* (DOD).

The method in [6] can be considered as a special case of our method with the following differences: (a) we iterate the process three times, (b) we calculate the threshold automatically, and (c) we use medoid instead of the mean. If we iterated only once, used mean, and did not calculate the SD then the method would equal to the variant of [6] except using the average distance to neighborhood [7]. The parameter three was chosen based on the experiments in [4].

Algorithm 1: Mean-shift process

Input:	$\mathbf{X} \subset R^{d\times n}$, $\boldsymbol{k}$
Output:	$\mathbf{Y} \subset R^{d\times n}$
For every point $x_i \in \mathbf{X}$:	
Step 1:	Find its k-nearest neighbors kNN(x_i)
Step 2:	Calculate the mean M of the neighbors kNN(x_i)
Step 3:	Replace the point x by the mean M and save it to **Y**

Algorithm 2: Mean-shift outlier detection (MOD)

Input:	$\mathbf{X} \subset R^{d\times n}$, $\boldsymbol{k}$
Output:	$\mathrm{N} \subset R^{d\times n}$
Step 1:	Repeat **Algorithm 1** three times to get **Y**
Step 2:	For every point $x_i \in \mathbf{X}$ and its shifted version $y_i \in \mathbf{Y}$ calculate distance $D_i = \lvert x_i - y_i \rvert$
Step 3:	Calculate the standard deviation (*SD*) of all D_i
Step 4:	For a point $x_i \in \mathbf{X}$ if $D_i > SD$, then x_i is detected as an outlier; save it to **N**.

3 Experiments

We test the proposed method based on Algorithm 2 with four existing outlier detection algorithms summarized in Table 1. We use the 9 benchmark datasets in Table 2 and visualized in Figure 3. The S sets have varying level of cluster overlap. A sets have varying number of clusters; unbalance and XOR datasets [17] have clusters with different densities. We evaluate the methods by F1-measure, which is basically the average of precision and recall. *Precision* is the ability of the classifier not to label as positive a sample that is negative, and *recall* is the ability of the classifier to find all the positive samples.

Table 1. Compared outlier detection algorithms.

Algorithm	Ref	Type	Parameter	Year	Publication
MCD	[5]	Statistical testing	top-N	1984	J. Am. Stat. Assoc.
ODIN	[7]	Distance-based	k, top-N	2004	Int. Conf. on Pattern Recognition
LOF	[9]	Density-based	k, top-N	2000	ACM SIGMOD
NC	[12]	Math. optimization	k, top-N	2018	IEEE-TNNLS
DOD	new	Shifting-based	k	2018	Int. Conf. Fuzzy Syst. Data Mining
MOD	new	Shifting-based	k	2018	Int. Conf. Fuzzy Syst. Data Mining

Table 2. Datasets used in the experiments. (http://cs.uef.fi/sipu/datasets/)

Dataset	Size	Outlier	Clusters
S1-S4	5000	7%	15
A1-A3	3000, 5250, 7500	7%	20, 35, 50
Unbalance	6500	7%	8
XOR	2000	7%	4

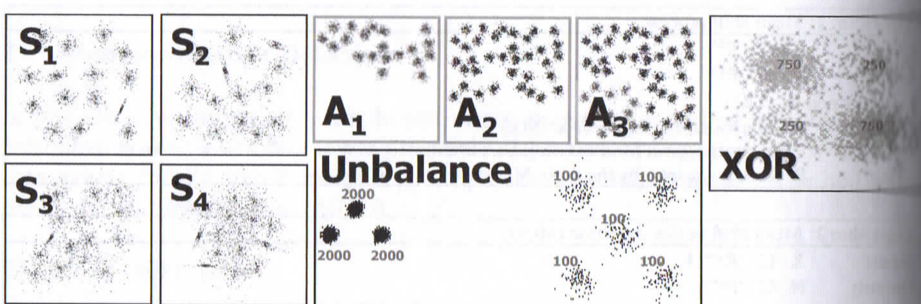


Figure 3. Datasets used in the experiments.

3.1 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_{mean}-2·*range*, x_{mean}+2·*range*], where x_{mean} is the mean of all data points, and *range* is the maximum distance of any point from the mean: *range* = max($|x_{max}$- $x_{mean}|$, $|x_{mean}$- $x_{min}|$). 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 Figure 4.

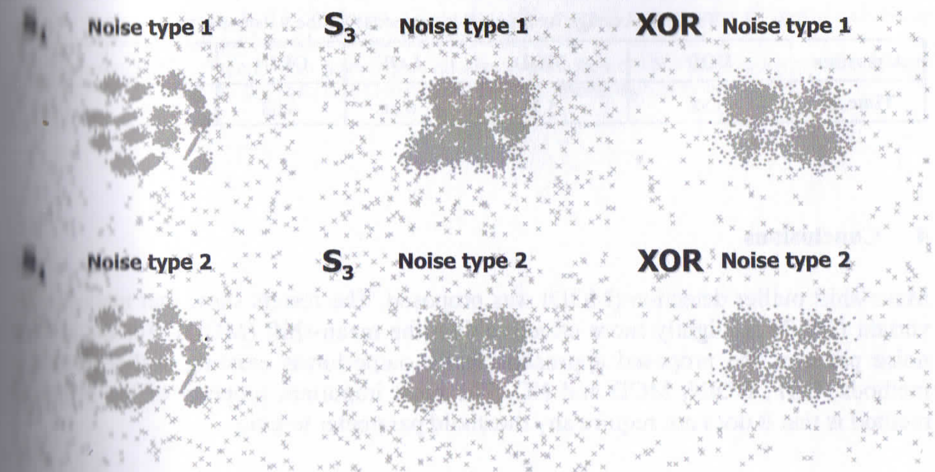


Figure 4. Noisy datasets S1, S3 and XOR with noise type 1 (up) and noise type 2 (down).

Results

The detection results are summarized in Figure 6. The first results (gray) are obtained using a priori knowledge of the amount of noise (7%). Each method selects exactly the same amount of outliers. The second results (red) are when no a priori parameter is allowed but the algorithm is forced to solve the thresholding by its own.

In Figure 5, both proposed methods are clearly better than the other tested methods (gap more than 4%). We can also see that the proposed methods perform almost equally well when no threshold parameter (red results). No automatic thresholding were developed for the other methods, so comparative results are missing.

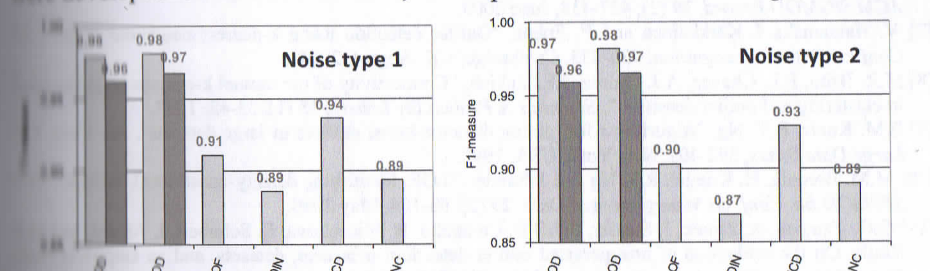


Figure 5. Average detection results for the 9 datasets in Table 2 and Figure 4. We used k=30 in all tests. The gray results are obtained using a priori (7%) number of outliers, and the red results using the number of outliers automatically determined by the method.

All methods based on k-NN require O(N^2) calculations. To address this slowness of the brute-force approach, we use the KD-tree technique [18] with all algorithms. It works fast in low-dimensional but can become inefficient with higher dimensions (D>20). In this case, faster approximate like *NNDES* [19] or the *Random pair divisive* (RP-div) [20] can become more efficient. Running times of the 9 datasets with noise type 1 are summarized in Table 3. The proposed method is somewhat slower than the other methods because of the iterations.

Table 3: Average running times (seconds) for the 9 datasets.

Algorithm	DOD	MOD	LOF	ODIN	MCD	NC
Time (s)	5.58	5.31	0.68	0.08	1.88	1.44

4 Conclusions

Mean-shift outlier detection (MOD) was proposed. The results show that medoid-shift variant (DOD) is slightly more effective than the mean-shift (MOD). For the studied noise patterns, the proposed approach clearly outperforms existing outlier detection methods: LOF, ODIN, MCD and NC. The most important property of the proposed method is that it does not require any threshold parameter to tune.

References

[1] A.M. Ali, P. Angelov, "Anomalous behaviour detection based on heterogeneous data and data fusion", *Soft Computing*, 2018.

[2] T.V. Pollet, L. van der Meij, "To remove or not to remove: the impact of outlier handling on significance testing in testosterone data", *Adaptive Human Behavior and Physiology*, **3** (1), 43-60, March 2017.

[3] H.-P. Kriegel, P. Kröger, and A. Zimek, "Outlier detection techniques," *13th Pacific-Asia Conf. Knowledge Discovery Data Mining*, 1–73, 2009.

[4] P. Fränti and J.W.Yang, "Medoid-shift noise removal to improve clustering", *Int. Conf. Artificial Intelligence and Soft Computing* (ICAISC), Zakopane, Poland, June 2018.

[5] P.J. Rousseeuw. Least median of squares regression. *J. Am Stat Ass*, **79**:871, 1984

[6] S. Ramaswamy, R. Rastogi and K. Shim, "Efficient algorithms for mining outliers from large data sets", *ACM SIGMOD Record*, **29** (2), 427-438, June 2000.

[7] V. Hautamäki, I. Kärkkäinen and P. Fränti, "Outlier detection using k-nearest neighbour graph", *Int. Conf. on Pattern Recognition*, 430-433, Cambridge, UK, August, 2004.

[8] M.R. Brito, E.L. Chavez, A.J. Quiroz, J.E. Yukich, "Connectivity of the mutual k-nearest-neighbor graph in clustering and outlier detection", *Statistics & Probability Letters*, **35** (1), 33-42, 1997.

[9] E.M. Knorr, R.T. Ng, "Algorithms for mining distance-based outliers in large datasets", *Int. Conf. Very Large Data Bases*, 392-403, New York, USA, 1998.

[10] M.M. Breunig, H. Kriegel, R.T. Ng and J. Sander, "LOF: Identifying density-based local outliers", *ACM SIGMOD Int. Conf. on Management of Data*, **29** (2), 93-104, May 2000.

[11] G.O. Campos, A. Zimek, J. Sander, R.J.G.B. Campello, B. Micenkova, E. Schubert, I. Assent, and M.E. Houle. On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *Data Mining and Knowledge Discovery*, **30** (4), 891–927, 2016.

[12] X. Li, J. Lv and Z. Yi, "An efficient representation-based method for boundary point and outlier detection", *IEEE Trans. on Neural Networks and Learning Systems*, **29** (1), January, 2018.

[13] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis", *IEEE Trans. Pattern Analysis and Machine Intelligence*, **24** (5), 603–619, May 2002.

[14] Y. Cheng, "Mean shift, mode seeking, and clustering", *IEEE Trans. Pattern Analysis and Machine Intelligence*, **17** (8), 790–799, August 1995.

[15] D.-M. Tsai and J.-Y. Luo, "Mean shift-based defect detection in multicrystalline solar wafer surfaces", *IEEE Trans. on Industrial Informatics*, **7** (1), 125-135, February 2011.

[16] Y.A. Sheikh, E.A. Khan, T. Kanade, "Mode-seeking by Medoidshifts", *IEEE Int. Conf. on Computer Vision* (ICCV), Rio de Janeiro, Brazil, October 2007.

[17] Y. Li and L. P. Maguire, "Selecting critical patterns based on local geometrical and statistical information," *IEEE Trans. Pattern Analysis Machine Intelligence*, 33 (6), 1189–1201, June 2011.

[18] Bentley, J.L., "Multidimensional binary search trees used for associative searching", *Communications of the ACM*,1975

[19] W. Dong, C. Moses, K. Li, "Efficient k-nearest neighbor graph construction for generic similarity measures". *ACM Int. Conf. on World Wide Web*, 577–586, 2011.

[20] S. Sieranoja and P. Fränti, "Fast random pair divisive construction of kNN graph using generic distance measures", *Int. Conf. on Big Data and Computing* (ICBDC), Shenzhen, China, April 2018.