Comparison of Different Calculations of the Density-Based Local Outlier Factor

Vanda Vintrová*, Tomas Vintr†, Hana Řezanková‡
Department of Statistics and Probability
University of Economics, Prague
130 67 Prague, Czech Republic
Email: *vanda.vintrova@vse.cz, †tomas.vintr@vse.cz, ‡hana.rezankova@vse.cz*

***Abstract*—In the paper, we propose several new density-based algorithms for outlier detection. We present the detailed synoptic theoretical analysis of the algorithms that compute the local outlier factor as a function of the densities of the neighborhood of the objects in a set of objects. Based on this analysis we propose a new calculation of the radius of the neighborhood and we create 66 algorithms to compute the outlier factor. All the algorithms are tested in the complex experiments to describe their basic and also specific characteristics. The results are presented and discussed. Intuitively it seems that the way how the radius of the neighborhood is calculated is important. This idea led to numerous modifications of this part of the algorithms, but on the basis of the experiments we demonstrate that these modifications have only little influence, and we describe which part of the algorithms influence the outlier score the most and we recommend three generally applicable algorithms with specific characteristics.**

***Keywords*-local outlier factor; density-based algorithm; outlier detection.**

I. Introduction

In real data files, outliers, as objects considerably inconsistent with other objects from the same dataset, are often present. Some statistical and data mining methods regard these outliers as a noise that should be identified and eliminated as they falsify the analysis. However, outliers can also contain useful information and therefore it is important to investigate outliers in detail.

The outlier detection can be used in clustering methods that applied in the segmentation and typology of students [1], in the text mining and consequently in the information retrieval [2], in the database merge in medicine [3], or in the prediction of inflation [4].

There are different approaches to outlier detection. We focus on local density-based algorithms that can capture not only global outliers, but also local outliers. The original concept to score the local outliers compares a local density of an object with local densities of its k-nearest neighbors [5]. There are several modifications of this approach. In general, we can say that local density-based algorithms for outlier detection assign to every object of a set of objects a value that quantifies the density of the neighborhood of the object, and by comparing the value with the values of the objects in its neighborhood they assign to every object a score representing a degree of being an outlier. The score is mostly computed as a ratio of a density of a neighborhood of an object to an average density of neighborhoods of the objects in the object's neighborhood.

The important subject of the papers concerning about the local density-based algorithms has been the attempt to determine a meaningful neighborhood of an object that should be compared [5], [6], [7]. It is startling that even though these methods have been widely developed, the fundamentals are not synoptically formalized. That leads to the existence of many confusing structures whose idea is difficult to understand.

There exist two basic approaches for determining a density of a neighborhood. The first one is to determine a radius and then to compute how many objects lie in the neighborhood $\mathbf{O}(\mathbf{x}_p)$ of the object $\mathbf{x}_p$, where the radius of the neighborhood R_p is the parameter of the algorithm [8], [7]. As the analysis is usually performed on all the N objects of the set of objects, $p = 1, \ldots, N$. The second approach is determining the number k of the nearest neighbors of the object $\mathbf{x}_p$ and then to find out the radius as a function of the distance between the object $\mathbf{x}_p$ and the k nearest neighbors [5], [6], [9], [10], [11]. In both cases, it is necessary to have a priori knowledge of the set of objects. In the first case we need to know at least a range of clusters in the set of objects and in the second case we need to know at least a minimal number of objects in clusters in the set of objects. Usually, it is more difficult to determine the minimal range correctly; therefore, generally, it is more proper to determine the minimal number of objects in a cluster, as it can also represent a border between clusters that will be considered for the analysis and clusters that are too small for the intended analysis and will be considered as a noise.

In the paper, the computation of the outlier factor is synoptically explained. First, there is a discussion about how to calculate the average radius, then how to calculate the radius of the object‘s neighborhood. On the basis of these mentioned discussions, we propose 66 different combinations of the averages and radii to calculate the outlier factor. Some of these combinations represent the original algorithms, but most of them are newly proposed by us. In the fourth section, the algorithms are applied on synthetic datasets and compared in the complex experiments, the results are presented and discussed and in the last section we propose the recommendation which algorithm to apply for what purpose or dataset type.

II. Related Work

Breunig et al. [5] is the first to introduce the concept of local density outliers and a local outlier factor (LOF). It compares a local density of an object with local densities of its k-nearest neighbors. The local density is estimated by a specific distance at which a point can be reached from its neighbors, called reachability distance, what produces stable results within clusters. LOF value of approximately 1 indicates that the point is located in a region of homogenous density. Higher LOF values signify an outlier, as it is a degree of being an outlier, but the scaling is different for different datasets.

An advantage of the LOF algorithm is that it can detect outliers even if the clusters of a dataset have different density and different size. This algorithm depends only on one parameter k; however this parameter strongly affects the outlierness of an object. If the parameter k is set too low, LOF does not detect outliers which are close to a dense cluster if the parameter is set too high, small clusters are regarded as outliers.

The LOF algorithm was modified several times especially with an aim to speed up the algorithm. An improvement of LOF known as Connectivity Outlier Factor (COF) [11] was proposed to overcome an ineffectiveness of LOF in detecting outliers in sparse datasets. Another modification of the LOF algorithm is LOF', LOF" and GridLOF [6]. LOF' simplifies the formula of LOF for ease of understanding, LOF" distinguishes between a neighborhood for computing the density of an object and a neighborhood for comparing the densities of the neighbors of an object. The GridLOF utilizes grid-based method to prune objects that are not outliers and then compute LOF score.

Another density-based algorithm was proposed by Papadimitriou et al. [7] named Local Correlation Integral (LOCI) based on the idea of a multi-granularity deviation factor (MDEF). The difference between LOF and LOCI is that LOCI uses neighborhood instead of k nearest neighbors. LOCI is less sensitive to input parameters than LOF.

The outlier scores provided by various outlier algorithms differ widely in their scale, range and meaning. For most methods the outlier scores are not comparable from dataset to dataset, for many methods the outlier scores are not comparable even within one dataset. The same outlier score for one object means that this object is an outlier and for another object (even within the same dataset, but from a different cluster) that this object is not extraordinary.

To overcome this problem, a new method has been proposed in [12] to unify outlier scores provided by different outlier algorithms. They propose two types of operations, regularization and normalization. Regularization means that the score is transformed into a range $[0;\infty)$, score equals approximately 0 for inliers, higher values signify outliers. The outlier factor can be regularized only if there exist an unambiguous numerous border between inliers and outliers. Such a border is not common for the existing algorithms. Normalization transforms scores into a range $[0;1]$. These transformations do not change the ordering obtained by the original score. The contribution of this approach is not only unification of outlier scores, but also the fact that these operations can increase a contrast between outlier and inlier scores. A transformed outlier score is a rough probability, if an object is an outlier. Transformed outlier scores are therefore easier to understand and to interpret.

III. Computation of Outlier Factor

The outlier factor (degree of outlierness) is defined as the ratio of the radius of the neighborhood of the object to the average radius of the neighborhoods of the objects in the neighborhood of the object $\mathbf{x}_p$, i. e.,

$$OF = R_p/R_{avg} \ . \tag{1}$$

It means that the more is the object $\mathbf{x}_p$ suspected of being an outlier the higher is the outlier factor.

A. Discussion about the Average Radius

The density of the neighborhood of the selected object $\mathbf{x}_p$ can be defined as

$$d = k/C_n R^n \ , \tag{2}$$

where k is the number of objects in the neighborhood of the object $\mathbf{x}_p$ and $C_n R^n$ is the volume of the neighborhood (n-dimensional hypersphere), $C_n = \frac{\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2}+1)}$.

The outlier factor is calculated according to the formula

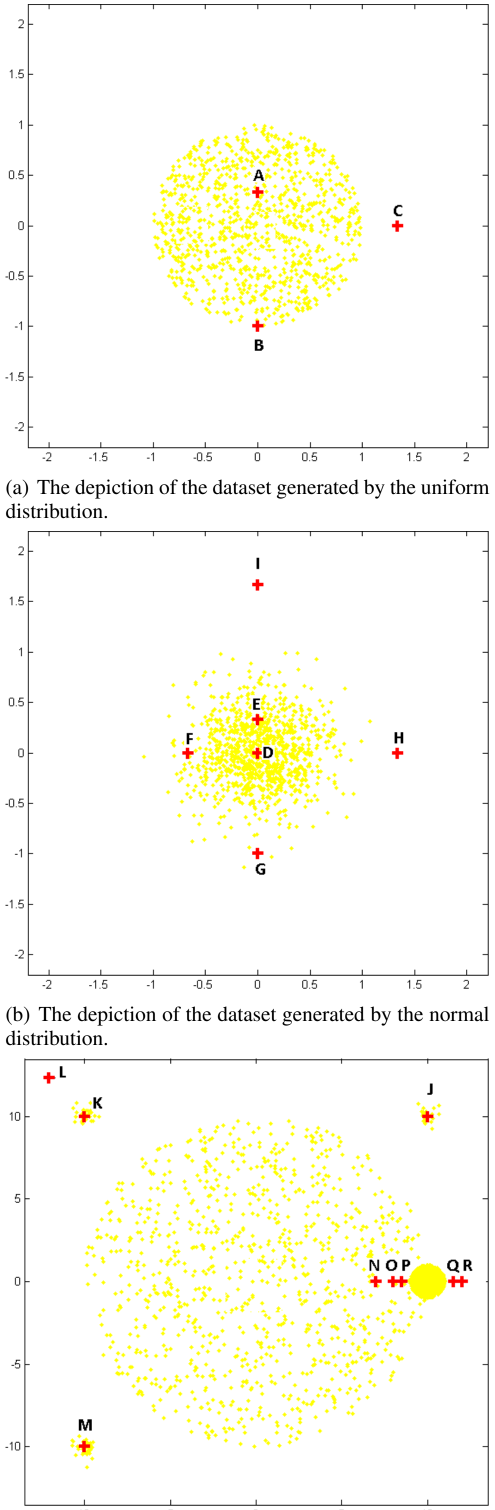
$$OF = \frac{\sqrt[n]{\frac{\sum_{i=1}^{k_p} \frac{k_i}{C_n R_i^n}}{k_p}}}{\sqrt[n]{\frac{k_p}{C_n R_p^n}}} \ , \tag{3}$$

where R_i is the individual radius of the neighborhood $\mathbf{O}(\mathbf{x}_i)$ of the object $\mathbf{x}_i \in \mathbf{O}(\mathbf{x}_p)$, k_i is the number of objects in the individual neighborhoods $\mathbf{O}(\mathbf{x}_i)$, k_p is the number of objects in the neighborhood $\mathbf{O}(\mathbf{x}_p)$ and R_p is the radius of its neighborhood $\mathbf{O}(\mathbf{x}_p)$.

If the radius of the neighborhood of every object contains exactly k objects, the formula can be easily modified as follows:

$$OF = \frac{\sqrt[n]{\frac{\sum_{i=1}^{k} \frac{k}{C_n R_i^n}}{k}}}{\sqrt[n]{\frac{k}{C_n R_p^n}}} = \frac{\sqrt[n]{\frac{\sum_{i=1}^{k} \frac{k}{R_i^n}}{k}}}{\sqrt[n]{\frac{k}{R_p^n}}} = \frac{\sqrt[n]{\frac{\sum_{i=1}^{k} \frac{1}{R_i^n}}{k}}}{\sqrt[n]{\frac{1}{R_p^n}}} =$$

$$= \frac{R_p}{\sqrt[-n]{\frac{\sum_{i=1}^{k} R_i^{-n}}{k}}} = \frac{R_p}{R_{avg}} \ , \tag{4}$$



(a) The depiction of the dataset generated by the uniform distribution.

(b) The depiction of the dataset generated by the normal distribution.

(c) The depiction of the dataset generated for the third experiment.

Figure 1. The illustrative pictures of the objects ($\cdot$) in the datasets and added objects (+) for the outlier factor tests.

where

$$R_{avg} = \sqrt[-n]{\frac{\sum_{i=1}^{k} R_i^{-n}}{k}} \tag{5}$$

is the radius of the average dense neighborhood.

It is important to say that we define the score differently than the authors of LOF [5] and LOF' [6], who define the outlier factor as

$$OF = \frac{R_p}{\sqrt[-1]{\frac{\sum_{i=1}^{k} R_i^{-1}}{k}}} \ . \tag{6}$$

From this formula, it is evident that the authors replace the volume by the radius what we do not consider a felicitous solution.

B. Discussion about the Radius

The radius of the neighborhood of the object $\mathbf{x}_p$ is in the algorithms LOF' and DSNOF [9] an average of a set of distances $\mathbf{D}_p = \{d(\mathbf{x}_p, \mathbf{x}_i)\}_{i=1}^{k}$, where $d(\mathbf{a}, \mathbf{b})$ is the Euclidean distance of an object $\mathbf{a}$ from $\mathbf{b}$. The LOF' algorithm finds the maximum distance and the DSNOF algorithm finds the median distance. Using median in the DSNOF algorithm is a similar idea as using k_1 and k_2, where $k_2 < k_1$, in the LOF" [6] algorithm. The LOF algorithm uses more difficult calculation, the radius is computed as the arithmetic mean of the values that are either the distances $d(\mathbf{x}_p, \mathbf{x}_i)$ or distances $d(\mathbf{x}_i, \mathbf{x}_{ik})$ between an object $\mathbf{x}_i$ and its k-th nearest neighbor $\mathbf{x}_{ik}$. The set $\mathbf{Q}_p = \{q_i\}_{i=1}^{k}$, which is used to calculate the radius R_p in LOF, contains values that fulfill the condition $q_i = max(\{d(\mathbf{x}_p, \mathbf{x}_i,), d(\mathbf{x}_i, \mathbf{x}_{ik})\})$. It is expected that this operation decreases the radius of the neighborhood of the object on the border of the cluster and therefore objects on the border of clusters obtain lower outlier factor OF then in case of the LOF' algorithm. It seems that to calculate R_p as the mean of distances similar to (5) is geometrically meaningful alternative to the computation of the arithmetic mean.

We assume, in the case of one very numerous cluster generated by the uniform or normal distribution (without a noise) and simultaneously if quantile or mean of a set of distances is used as a radius determination, that the radius of the neighborhood of the object on a border of this cluster has to be approximately twice greater than the radius of its k-th most distant neighbor. From this reflection, we propose to determine the radius not only as the average of distances, but also to use some kind of a measure of dispersion. Inspired by the Box's M test that uses the determinants of the sample covariance matrices to test the equality of covariance matrices we propose to determine the radius using the determinants of the sample covariance matrices. As the number of objects in the sample is the same, we do not have to adjust the determinants and we can compare them directly. We compute R_p as a determinant of a covariance matrix $\mathbf{C}_p$ of a set of objects $\mathbf{K}_p = \{\{\mathbf{x}_i\}_{i=1}^{k}, \mathbf{x}_p\}$,

$$R_p = \det \mathbf{C}_p \ . \qquad (7)$$

C. Determining the Outlier Factor

On the basis of the above mentioned reflections, we can compute the outlier factor OF in several ways. The radius R_p of the object $\mathbf{x}_p$ can be calculated from the set $\mathbf{D}_p$ or $\mathbf{Q}_p$ using the following characteristics: first decile, maximum, median, arithmetic mean and the mean similar to (5). Minimum is not an appropriate characteristic because of its low information value about the neighborhood of $\mathbf{x}_p$ and consequent computational instability. The eleventh possibility to calculate the radius R_p is to compute the determinant of a covariance matrix $\mathbf{C}_p$ of a set of objects $\mathbf{K}_p$ proposed by us.

The calculation of R_p can be combined with the calculation of R_{avg}, which can be calculated according to the original algorithms as a harmonic mean of a set of radii R_i of neighborhoods of objects $\mathbf{x}_i$

$$R_{avg} = \sqrt[-1]{\frac{\sum_{i=1}^{k} R_i^{-1}}{k}} \ , \qquad (8)$$

or as an average (5) or as a maximum, minimum, median or ninth decile of a set of a set $\{R_i\}$. We have to mention that a maximum, resp. a minimum of $\{R_i\}$ is equivalent to minimum, resp. maximum of densities, no matter whether the density is computed as a function R^{-1} or R^{-n}. By combining the different averages, radii and sets we create 66 $(11 \cdot 6)$ algorithms to calculate the outlier score.

Because there are many combinations, it is necessary to distinguish individual computations, systematically. For the purpose of this paper we have decided to create the names of individual combinations by combining the shortcuts of the functions included. The first characters represent a shortcut of the chosen average used for the calculation of the radius R_p, the following capital letter represents the set used for the computation of R_p and the characters on the third place represent a shortcut of the average used for the calculation of R_{avg}. Minimum is denoted as *min*, maximum as *max*, median as *med*, arithmetical mean as *mean*, harmonic mean as *hean*, the mean defined in (5) as *nean*, the first decile as 0.1 and the ninth decile as 0.9. The calculation of the radius R_p as a determinant of the covariance matrix will be denoted with the prefix *det*.

For example, the original LOF algorithm will be denoted as *meanQhean*, because it computes R_p as an arithmetic mean of the set $\mathbf{Q}_p$ and R_{avg} is computed as a harmonic mean. LOF' algorithm will be denoted as *maxDhean*, because it computes R_p as a maximum of the set $\mathbf{D}_p$ and R_{avg} is computed as a harmonic mean. An algorithm that will use the determinant of the covariance matrix of the set $\mathbf{K}_p$ to compute R_p and that will compute R_{avg} according to (5) will be denoted as *detKnean*, and so on.

IV. Experiments

We compared the algorithms in three experiments. The first two experiments are very simple, just to show the basic characteristics of the algorithms. We generated two datasets consisting of 1000 vectors. The first dataset was generated by the two–dimensional uniform distribution within the borders of the sphere with radius 1 and center $(0,0)^T$. We added 3 vectors with the coordinates $\mathbf{v}_A = (0, \frac{1}{3})^T$, $\mathbf{v}_B = (0,-1)^T$ and $\mathbf{v}_C = (\frac{4}{3}, 0)^T$ (see Fig. 1 (a)). The second dataset was created by the two–dimensional normal distribution with the mean $(0,0)^T$ and the standard deviation of every variable $\sigma = \frac{1}{3}$. We added 6 vectors to the datasets with the coordinates $\mathbf{v}_D = (0,0)^T$, $\mathbf{v}_E = (0, \frac{1}{3})^T$, $\mathbf{v}_F = (-\frac{2}{3}, 0)^T$, $\mathbf{v}_G = (0,-1)^T$, $\mathbf{v}_H = (\frac{4}{3}, 0)^T$ and $\mathbf{v}_I = (0, \frac{5}{3})^T$ (see Fig. 1 (b)). For both experiments we set $k = 40$. We performed both experiments ten times. The sample mean and the sample standard deviation of outlier factors of the added vectors for the tested algorithms are presented in the Table I, where the double vertical line separates these two experiments and the simple vertical lines highlight the hypothetical boarder of the clusters. We suppose that the mean values of the computed outlier factors should be strongly higher for the vectors to the right from the line to highlight the outliers. The horizontal lines separate the different groups of algorithms.

The third experiment is more complex to show further characteristics of the algorithms. There are 5 clusters in the dataset and we add 9 vectors denoted $\mathbf{v}_J$ to $\mathbf{v}_R$ on which we will demonstrate the behavior of the algorithms. There is one big cluster consisting of 1000 vectors created by the two–dimensional uniform distribution with the center $(0,0)^T$ and radius 10 units. On the border of this cluster is the center $(10,0)^T$ of another cluster with the radius 1 unit consisting also of 1000 vectors created by the two–dimensional uniform distribution. Around the big cluster there are three other clusters consisting of 21, 40 and 40 vectors generated by the two–dimensional normal distribution with mean vectors $\mathbf{v}_J = (10,10)^T$, $\mathbf{v}_K = (-10,10)^T$ and $\mathbf{v}_M = (-10,-10)^T$ respectively. Next to the cluster with the mean vector $\mathbf{v}_K$ is an outlying vector $\mathbf{v}_L = (-12,12)^T$. The vectors $\mathbf{v}_K$ and $\mathbf{v}_L$ have a similar set of k neighbors, but the vector $\mathbf{v}_L$ is an obvious outlier while the vector $\mathbf{v}_K$ is an obvious inlier. Each of the vectors $\mathbf{v}_K$ and $\mathbf{v}_M$ has one significantly distant vector in its neighborhood, for the vector $\mathbf{v}_K$ it is the outlier $\mathbf{v}_L$ and for the vector $\mathbf{v}_M$ it is an inlier from the big sparse cluster in the middle of the dataset, but the vector $\mathbf{v}_M$ does not belong to the set of k neighbors of this vector. The vectors $\mathbf{v}_J$, $\mathbf{v}_Q$ and $\mathbf{v}_M$ are placed in the centers of the small clusters and therefore they are inliers. The vector $\mathbf{v}_N = (7,0)^T$ belongs to the cluster generated by the two–dimensional uniform distribution, within its neighborhood are few or none vectors

Table I
OUTLIER FACTORS ($\bar{x} \pm s^{'}$) OF ADDED VECTORS (1. AND 2. EXPERIMENT).

	A	B	C	D	E	F	G	H	I
0.1Dmin	1.7 ± 0.39	2.6 ± 0.81	9.6 ± 1.25	1.5 ± 0.38	1.7 ± 0.69	2.2 ± 0.77	4.9 ± 1.03	13 ± 2.77	18 ± 4.24
0.1Dnean	1.1 ± 0.18	1.5 ± 0.37	6.4 ± 0.57	1.0 ± 0.15	1.0 ± 0.31	1.3 ± 0.22	2.7 ± 0.38	6.9 ± 1.08	9.7 ± 1.19
0.1Dhean	1.1 ± 0.17	1.5 ± 0.34	6.2 ± 0.54	1.0 ± 0.15	1.0 ± 0.30	1.2 ± 0.21	2.4 ± 0.32	6.3 ± 0.96	8.9 ± 0.95
0.1Dmed	1.1 ± 0.18	1.4 ± 0.31	6.2 ± 0.64	1.0 ± 0.14	0.9 ± 0.26	1.2 ± 0.24	2.3 ± 0.40	6.0 ± 0.91	8.3 ± 1.19
0.1D0.9	0.8 ± 0.13	1.1 ± 0.22	4.5 ± 0.31	0.7 ± 0.11	0.7 ± 0.21	0.7 ± 0.13	1.1 ± 0.26	2.7 ± 0.44	4.2 ± 0.88
0.1Dmax	0.7 ± 0.13	0.9 ± 0.22	3.8 ± 0.36	0.6 ± 0.12	0.6 ± 0.17	0.5 ± 0.15	0.7 ± 0.24	1.6 ± 0.37	2.6 ± 0.65
neanDmin	5.7 ± 8.32	5.7 ± 2.68	16 ± 8.77	3.6 ± 2.82	2.1 ± 0.59	3.0 ± 1.10	7.5 ± 1.94	22 ± 10.9	23 ± 8.85
neanDnean	1.8 ± 1.71	2.2 ± 0.51	7.0 ± 1.71	1.5 ± 0.76	1.2 ± 0.27	1.3 ± 0.38	3.1 ± 0.29	8.2 ± 2.20	10 ± 1.58
neanDhean	1.2 ± 0.49	1.8 ± 0.36	6.2 ± 0.89	1.3 ± 0.46	1.1 ± 0.27	1.2 ± 0.33	2.7 ± 0.28	6.7 ± 1.30	8.7 ± 0.74
neanDmed	0.9 ± 0.26	1.5 ± 0.34	5.3 ± 0.51	1.0 ± 0.21	1.0 ± 0.30	1.1 ± 0.34	2.3 ± 0.42	5.6 ± 0.65	7.6 ± 0.80
neanD0.9	0.7 ± 0.22	1.1 ± 0.23	3.9 ± 0.25	0.8 ± 0.13	0.8 ± 0.19	0.7 ± 0.19	1.2 ± 0.27	2.7 ± 0.41	3.8 ± 0.57
neanDmax	0.7 ± 0.21	0.9 ± 0.23	3.2 ± 0.28	0.7 ± 0.14	0.7 ± 0.16	0.5 ± 0.17	0.8 ± 0.17	1.6 ± 0.38	2.4 ± 0.45
medDmin	1.2 ± 0.15	1.7 ± 0.22	4.1 ± 0.42	1.3 ± 0.15	1.2 ± 0.10	1.8 ± 0.25	3.1 ± 0.29	5.8 ± 0.76	8.1 ± 1.11
medDnean	1.0 ± 0.10	1.4 ± 0.13	3.2 ± 0.30	1.0 ± 0.07	1.0 ± 0.06	1.2 ± 0.14	2.1 ± 0.13	4.0 ± 0.38	5.2 ± 0.39
medDhean	1.0 ± 0.10	1.4 ± 0.12	3.1 ± 0.29	1.0 ± 0.07	1.0 ± 0.06	1.2 ± 0.13	2.0 ± 0.11	3.8 ± 0.34	5.0 ± 0.35
medDmed	1.0 ± 0.09	1.4 ± 0.13	3.1 ± 0.26	1.0 ± 0.07	1.0 ± 0.06	1.2 ± 0.13	2.1 ± 0.17	3.9 ± 0.35	5.0 ± 0.44
medD0.9	0.9 ± 0.10	1.1 ± 0.08	2.5 ± 0.15	0.9 ± 0.07	0.8 ± 0.07	0.8 ± 0.11	1.2 ± 0.13	2.1 ± 0.28	2.8 ± 0.39
medDmax	0.8 ± 0.10	1.0 ± 0.08	2.2 ± 0.17	0.9 ± 0.07	0.8 ± 0.06	0.6 ± 0.08	0.8 ± 0.14	1.4 ± 0.24	2.0 ± 0.28
meanDmin	1.2 ± 0.09	1.6 ± 0.17	4.0 ± 0.51	1.1 ± 0.10	1.1 ± 0.11	1.7 ± 0.23	3.0 ± 0.24	5.8 ± 0.70	8.2 ± 0.99
meanDnean	1.0 ± 0.06	1.4 ± 0.09	3.3 ± 0.30	1.0 ± 0.06	1.0 ± 0.06	1.2 ± 0.12	2.1 ± 0.14	4.0 ± 0.42	5.4 ± 0.36
meanDhean	1.0 ± 0.06	1.4 ± 0.08	3.2 ± 0.29	1.0 ± 0.06	1.0 ± 0.06	1.2 ± 0.11	2.0 ± 0.12	3.9 ± 0.38	5.1 ± 0.33
meanDmed	1.0 ± 0.05	1.4 ± 0.07	3.3 ± 0.31	1.0 ± 0.06	1.0 ± 0.05	1.2 ± 0.12	2.0 ± 0.19	4.0 ± 0.38	5.2 ± 0.49
meanD0.9	0.9 ± 0.07	1.1 ± 0.05	2.6 ± 0.15	0.9 ± 0.06	0.8 ± 0.06	0.8 ± 0.08	1.1 ± 0.12	2.2 ± 0.29	2.9 ± 0.41
meanDmax	0.9 ± 0.07	1.0 ± 0.05	2.4 ± 0.14	0.9 ± 0.07	0.8 ± 0.07	0.6 ± 0.05	0.8 ± 0.14	1.5 ± 0.26	2.1 ± 0.31
maxDmin	1.1 ± 0.05	1.6 ± 0.13	3.0 ± 0.34	1.1 ± 0.07	1.2 ± 0.09	1.7 ± 0.19	2.7 ± 0.16	4.6 ± 0.55	6.3 ± 0.67
maxDnean	1.0 ± 0.03	1.3 ± 0.07	2.5 ± 0.24	1.0 ± 0.04	1.0 ± 0.05	1.2 ± 0.11	1.9 ± 0.11	3.3 ± 0.32	4.2 ± 0.26
maxDhean	1.0 ± 0.03	1.3 ± 0.07	2.4 ± 0.24	1.0 ± 0.04	1.0 ± 0.05	1.2 ± 0.10	1.8 ± 0.10	3.1 ± 0.30	4.0 ± 0.23
maxDmed	1.0 ± 0.03	1.3 ± 0.08	2.4 ± 0.27	1.0 ± 0.04	1.0 ± 0.04	1.2 ± 0.10	1.9 ± 0.14	3.2 ± 0.31	4.1 ± 0.29
maxD0.9	0.9 ± 0.05	1.0 ± 0.04	2.0 ± 0.17	0.9 ± 0.04	0.9 ± 0.08	0.8 ± 0.07	1.1 ± 0.12	1.9 ± 0.23	2.4 ± 0.26
maxDmax	0.9 ± 0.05	1.0 ± 0.04	1.9 ± 0.12	0.9 ± 0.04	0.8 ± 0.08	0.7 ± 0.04	0.8 ± 0.13	1.4 ± 0.21	1.8 ± 0.22
0.1Qmin	1.1 ± 0.05	1.2 ± 0.11	2.2 ± 0.28	1.0 ± 0.03	1.1 ± 0.05	1.5 ± 0.15	2.2 ± 0.17	3.7 ± 0.41	5.9 ± 0.68
0.1Qnean	1.0 ± 0.02	1.1 ± 0.06	2.1 ± 0.25	1.0 ± 0.02	1.0 ± 0.02	1.2 ± 0.08	1.7 ± 0.11	2.9 ± 0.34	4.2 ± 0.32
0.1Qhean	1.0 ± 0.02	1.1 ± 0.05	2.1 ± 0.25	1.0 ± 0.02	1.0 ± 0.02	1.1 ± 0.08	1.6 ± 0.10	2.8 ± 0.33	4.1 ± 0.30
0.1Qmed	1.0 ± 0.02	1.1 ± 0.06	2.1 ± 0.26	1.0 ± 0.01	1.0 ± 0.01	1.1 ± 0.08	1.6 ± 0.15	3.0 ± 0.36	4.2 ± 0.34
0.1Q0.9	1.0 ± 0.04	1.0 ± 0.03	1.9 ± 0.20	1.0 ± 0.03	0.9 ± 0.03	0.9 ± 0.06	1.1 ± 0.09	1.9 ± 0.30	2.8 ± 0.39
0.1Qmax	0.9 ± 0.05	1.0 ± 0.04	1.8 ± 0.17	0.9 ± 0.05	0.9 ± 0.05	0.7 ± 0.05	0.9 ± 0.12	1.4 ± 0.30	2.1 ± 0.38
neanQmin	1.1 ± 0.03	1.3 ± 0.10	2.3 ± 0.25	1.0 ± 0.03	1.1 ± 0.04	1.6 ± 0.15	2.3 ± 0.15	3.8 ± 0.41	5.7 ± 0.64
neanQnean	1.0 ± 0.01	1.1 ± 0.05	2.1 ± 0.21	1.0 ± 0.01	1.0 ± 0.01	1.2 ± 0.08	1.7 ± 0.10	2.9 ± 0.31	4.1 ± 0.26
neanQhean	1.0 ± 0.01	1.1 ± 0.05	2.1 ± 0.21	1.0 ± 0.01	1.0 ± 0.01	1.2 ± 0.07	1.7 ± 0.09	2.9 ± 0.29	4.0 ± 0.24
neanQmed	1.0 ± 0.01	1.1 ± 0.05	2.1 ± 0.22	1.0 ± 0.01	1.0 ± 0.02	1.2 ± 0.06	1.7 ± 0.14	3.0 ± 0.31	4.0 ± 0.28
neanQ0.9	1.0 ± 0.04	1.0 ± 0.02	2.0 ± 0.16	1.0 ± 0.02	0.9 ± 0.03	0.9 ± 0.05	1.1 ± 0.09	1.9 ± 0.25	2.6 ± 0.31
neanQmax	0.9 ± 0.04	1.0 ± 0.03	1.9 ± 0.12	0.9 ± 0.03	0.9 ± 0.03	0.7 ± 0.05	0.9 ± 0.12	1.4 ± 0.24	2.0 ± 0.28
medQmin	1.1 ± 0.04	1.3 ± 0.11	2.4 ± 0.27	1.0 ± 0.04	1.1 ± 0.06	1.6 ± 0.19	2.4 ± 0.15	4.1 ± 0.48	5.9 ± 0.66
medQnean	1.0 ± 0.02	1.2 ± 0.05	2.2 ± 0.23	1.0 ± 0.02	1.0 ± 0.02	1.2 ± 0.11	1.8 ± 0.10	3.1 ± 0.32	4.2 ± 0.26
medQhean	1.0 ± 0.02	1.1 ± 0.05	2.2 ± 0.23	1.0 ± 0.02	1.0 ± 0.02	1.2 ± 0.10	1.7 ± 0.09	3.0 ± 0.30	4.1 ± 0.23
medQmed	1.0 ± 0.02	1.2 ± 0.05	2.2 ± 0.26	1.0 ± 0.01	1.0 ± 0.02	1.2 ± 0.09	1.7 ± 0.15	3.1 ± 0.32	4.1 ± 0.27
medQ0.9	1.0 ± 0.04	1.0 ± 0.04	2.0 ± 0.18	1.0 ± 0.03	0.9 ± 0.04	0.9 ± 0.07	1.1 ± 0.10	2.0 ± 0.25	2.6 ± 0.31
medQmax	0.9 ± 0.05	1.0 ± 0.05	1.9 ± 0.14	0.9 ± 0.04	0.9 ± 0.05	0.7 ± 0.04	0.9 ± 0.13	1.4 ± 0.23	2.0 ± 0.27
meanQmin	1.1 ± 0.03	1.3 ± 0.10	2.3 ± 0.25	1.0 ± 0.03	1.2 ± 0.04	1.6 ± 0.15	2.3 ± 0.14	3.8 ± 0.42	5.6 ± 0.64
meanQnean	1.0 ± 0.01	1.1 ± 0.05	2.1 ± 0.21	1.0 ± 0.01	1.0 ± 0.01	1.2 ± 0.07	1.7 ± 0.11	2.9 ± 0.30	4.0 ± 0.25
meanQhean	1.0 ± 0.01	1.1 ± 0.05	2.1 ± 0.21	1.0 ± 0.01	1.0 ± 0.01	1.2 ± 0.07	1.6 ± 0.09	2.8 ± 0.29	3.9 ± 0.23
meanQmed	1.0 ± 0.01	1.1 ± 0.05	2.1 ± 0.22	1.0 ± 0.01	1.0 ± 0.02	1.2 ± 0.06	1.7 ± 0.14	3.0 ± 0.29	3.9 ± 0.28
meanQ0.9	1.0 ± 0.03	1.0 ± 0.02	2.0 ± 0.15	1.0 ± 0.02	0.9 ± 0.02	0.9 ± 0.04	1.1 ± 0.09	1.9 ± 0.25	2.6 ± 0.29
meanQmax	0.9 ± 0.04	1.0 ± 0.03	1.9 ± 0.11	0.9 ± 0.03	0.8 ± 0.03	0.7 ± 0.05	0.9 ± 0.11	1.4 ± 0.24	2.0 ± 0.27
maxQmin	1.1 ± 0.06	1.3 ± 0.10	2.2 ± 0.28	1.1 ± 0.04	1.2 ± 0.10	2.0 ± 0.21	2.3 ± 0.41	3.2 ± 0.37	4.3 ± 0.48
maxQnean	1.0 ± 0.02	1.1 ± 0.03	1.9 ± 0.12	1.0 ± 0.04	1.0 ± 0.05	1.3 ± 0.12	1.6 ± 0.29	2.2 ± 0.21	2.9 ± 0.16
maxQhean	1.0 ± 0.02	1.1 ± 0.03	1.9 ± 0.12	1.0 ± 0.04	1.0 ± 0.05	1.3 ± 0.11	1.5 ± 0.26	2.1 ± 0.21	2.8 ± 0.15
maxQmed	1.0 ± 0.02	1.0 ± 0.03	1.9 ± 0.12	1.0 ± 0.04	1.0 ± 0.06	1.3 ± 0.13	1.5 ± 0.26	2.1 ± 0.23	2.7 ± 0.21
maxQ0.9	1.0 ± 0.02	1.0 ± 0.02	1.8 ± 0.10	0.9 ± 0.05	0.8 ± 0.04	0.8 ± 0.06	1.0 ± 0.05	1.4 ± 0.28	2.0 ± 0.17
maxQmax	1.0 ± 0.03	1.0 ± 0.02	1.8 ± 0.08	0.9 ± 0.05	0.8 ± 0.02	0.7 ± 0.11	0.9 ± 0.08	1.0 ± 0.10	1.7 ± 0.32
detKmin	1.7 ± 0.56	2.1 ± 1.07	3.9 ± 1.37	1.7 ± 0.49	1.9 ± 0.75	6.1 ± 2.64	14 ± 6.19	24 ± 10.4	51 ± 32.2
detKnean	1.2 ± 0.26	1.3 ± 0.33	2.7 ± 0.79	1.1 ± 0.25	1.1 ± 0.26	2.7 ± 0.87	6.7 ± 3.82	11 ± 4.41	21 ± 9.91
detKhean	1.1 ± 0.24	1.3 ± 0.27	2.6 ± 0.75	1.1 ± 0.21	1.0 ± 0.22	2.2 ± 0.61	5.3 ± 2.84	9.3 ± 3.34	17 ± 6.30
detKmed	1.1 ± 0.20	1.2 ± 0.14	2.5 ± 0.72	1.1 ± 0.21	1.0 ± 0.16	1.7 ± 0.39	3.9 ± 2.01	8.3 ± 2.72	13 ± 5.19
detK0.9	0.8 ± 0.22	0.8 ± 0.18	1.9 ± 0.58	0.7 ± 0.16	0.6 ± 0.14	0.6 ± 0.14	1.1 ± 0.13	2.4 ± 0.76	4.6 ± 1.10
detKmax	0.7 ± 0.20	0.8 ± 0.19	1.7 ± 0.52	0.6 ± 0.16	0.5 ± 0.11	0.4 ± 0.15	0.8 ± 0.16	1.2 ± 0.22	2.8 ± 0.86

Table II
OUTLIER FACTORS ($\bar{x} \pm s^{'}$) OF ADDED VECTORS (3. EXPERIMENT).

	J	K	L	M	N	O	P	Q	R
0.1Dmin	1.3 ± 0.27	1.3 ± 0.36	22 ± 5.54	1.3 ± 0.41	4.3 ± 4.20	12 ± 3.83	12 ± 2.40	14 ± 3.61	25 ± 4.19
0.1Dnean	0.6 ± 0.14	0.7 ± 0.14	12 ± 1.67	0.8 ± 0.18	1.4 ± 0.62	6.5 ± 1.56	7.3 ± 1.02	8.7 ± 1.13	17 ± 1.94
0.1Dhean	0.6 ± 0.13	0.7 ± 0.13	11 ± 1.33	0.7 ± 0.15	1.2 ± 0.29	5.4 ± 1.28	6.8 ± 0.97	8.4 ± 0.97	16 ± 1.71
0.1Dmed	0.4 ± 0.11	0.6 ± 0.12	10 ± 1.13	0.7 ± 0.17	1.0 ± 0.20	5.9 ± 1.81	7.0 ± 0.73	8.5 ± 0.95	16 ± 1.82
0.1D0.9	0.3 ± 0.06	0.3 ± 0.08	5.5 ± 0.81	0.3 ± 0.09	0.8 ± 0.19	0.9 ± 0.10	2.0 ± 1.56	5.2 ± 0.50	10 ± 1.00
0.1Dmax	0.2 ± 0.05	0.1 ± 0.02	4.0 ± 1.01	0.2 ± 0.03	0.6 ± 0.13	0.7 ± 0.08	0.9 ± 0.07	0.5 ± 0.01	1.9 ± 0.03
neanDmin	1.9 ± 0.78	1.8 ± 0.74	27 ± 10.2	2.2 ± 0.79	5.2 ± 3.17	21 ± 16.5	17 ± 9.17	29 ± 21.8	48 ± 28.0
neanDnean	0.7 ± 0.23	0.8 ± 0.28	12 ± 3.01	0.9 ± 0.33	1.7 ± 0.54	7.9 ± 4.14	7.1 ± 1.93	11 ± 4.97	18 ± 5.99
neanDhean	0.6 ± 0.21	0.7 ± 0.24	11 ± 2.00	0.8 ± 0.29	1.4 ± 0.31	5.9 ± 2.62	6.0 ± 1.31	9.0 ± 2.40	15 ± 3.17
neanDmed	0.4 ± 0.19	0.6 ± 0.21	9.0 ± 0.85	0.7 ± 0.27	1.1 ± 0.24	5.5 ± 2.30	5.4 ± 1.00	7.1 ± 0.74	12 ± 1.28
neanD0.9	0.2 ± 0.11	0.3 ± 0.12	4.8 ± 0.53	0.3 ± 0.12	0.8 ± 0.19	0.8 ± 0.27	1.7 ± 1.13	4.9 ± 0.46	8.3 ± 0.78
neanDmax	0.2 ± 0.09	0.1 ± 0.03	3.4 ± 0.63	0.2 ± 0.07	0.7 ± 0.18	0.6 ± 0.20	0.7 ± 0.13	0.6 ± 0.01	1.7 ± 0.03
medDmin	2.1 ± 0.18	1.1 ± 0.05	7.6 ± 0.89	1.0 ± 0.07	3.8 ± 2.39	7.4 ± 0.70	4.7 ± 0.38	5.2 ± 0.31	9.1 ± 0.69
medDnean	1.3 ± 0.06	0.8 ± 0.05	5.5 ± 0.61	0.7 ± 0.05	1.4 ± 0.36	4.8 ± 0.65	3.5 ± 0.27	4.0 ± 0.23	7.0 ± 0.41
medDhean	1.3 ± 0.05	0.7 ± 0.05	5.3 ± 0.58	0.7 ± 0.05	1.2 ± 0.15	4.1 ± 0.65	3.3 ± 0.28	4.0 ± 0.22	6.9 ± 0.39
medDmed	1.1 ± 0.05	0.8 ± 0.05	5.4 ± 0.56	0.7 ± 0.04	1.0 ± 0.07	4.9 ± 1.27	3.6 ± 0.28	4.0 ± 0.25	7.0 ± 0.48
medD0.9	0.9 ± 0.02	0.5 ± 0.05	3.5 ± 0.30	0.5 ± 0.08	0.9 ± 0.04	0.8 ± 0.05	1.4 ± 0.78	3.1 ± 0.20	5.4 ± 0.32
medDmax	0.8 ± 0.03	0.1 ± 0.02	2.8 ± 0.38	0.2 ± 0.04	0.8 ± 0.05	0.8 ± 0.05	0.6 ± 0.02	0.6 ± 0.01	1.8 ± 0.03
meanDmin	2.1 ± 0.17	1.0 ± 0.01	6.0 ± 0.48	1.0 ± 0.01	3.7 ± 2.40	6.7 ± 0.68	4.5 ± 0.25	5.1 ± 0.32	8.9 ± 0.60
meanDnean	1.4 ± 0.07	0.8 ± 0.01	4.7 ± 0.38	0.8 ± 0.01	1.4 ± 0.36	4.5 ± 0.70	3.6 ± 0.26	4.1 ± 0.25	7.2 ± 0.44
meanDhean	1.3 ± 0.06	0.8 ± 0.01	4.6 ± 0.37	0.8 ± 0.01	1.2 ± 0.15	3.9 ± 0.69	3.4 ± 0.28	4.1 ± 0.24	7.1 ± 0.42
meanDmed	1.2 ± 0.04	0.8 ± 0.02	4.7 ± 0.32	0.8 ± 0.03	1.0 ± 0.07	4.8 ± 1.32	3.7 ± 0.25	4.2 ± 0.30	7.2 ± 0.51
meanD0.9	0.9 ± 0.02	0.5 ± 0.02	3.3 ± 0.24	0.5 ± 0.04	0.9 ± 0.04	0.8 ± 0.04	1.5 ± 0.83	3.2 ± 0.18	5.6 ± 0.25
meanDmax	0.9 ± 0.04	0.2 ± 0.01	2.8 ± 0.37	0.3 ± 0.03	0.9 ± 0.04	0.7 ± 0.03	0.6 ± 0.02	0.6 ± 0.01	1.8 ± 0.03
maxDmin	2.7 ± 0.15	1.3 ± 0.07	1.6 ± 0.11	1.6 ± 0.11	4.2 ± 2.13	5.2 ± 0.42	3.6 ± 0.12	3.8 ± 0.28	6.2 ± 0.58
maxDnean	1.7 ± 0.09	1.0 ± 0.01	1.3 ± 0.04	1.0 ± 0.01	1.5 ± 0.34	3.5 ± 0.38	2.7 ± 0.14	3.0 ± 0.15	5.0 ± 0.25
maxDhean	1.6 ± 0.07	1.0 ± 0.01	1.3 ± 0.04	1.0 ± 0.01	1.3 ± 0.16	3.1 ± 0.40	2.6 ± 0.16	2.9 ± 0.14	4.9 ± 0.25
maxDmed	1.4 ± 0.05	1.0 ± 0.02	1.2 ± 0.05	1.0 ± 0.02	1.0 ± 0.05	3.8 ± 0.79	2.8 ± 0.12	3.0 ± 0.19	5.0 ± 0.30
maxD0.9	1.0 ± 0.02	0.9 ± 0.03	1.1 ± 0.03	0.9 ± 0.01	0.9 ± 0.05	0.7 ± 0.05	1.3 ± 0.59	2.4 ± 0.11	4.0 ± 0.17
maxDmax	0.9 ± 0.03	0.8 ± 0.03	1.0 ± 0.03	0.9 ± 0.01	0.9 ± 0.05	0.6 ± 0.04	0.6 ± 0.03	0.6 ± 0.01	1.7 ± 0.03
0.1Qmin	2.5 ± 0.18	1.0 ± 0.00	1.0 ± 0.00	2.0 ± 0.14	4.1 ± 1.86	5.0 ± 0.38	2.9 ± 0.19	3.0 ± 0.22	5.7 ± 0.37
0.1Qnean	1.8 ± 0.10	1.0 ± 0.00	1.0 ± 0.00	1.0 ± 0.01	1.4 ± 0.28	3.8 ± 0.49	2.5 ± 0.14	2.8 ± 0.18	5.2 ± 0.29
0.1Qhean	1.7 ± 0.08	1.0 ± 0.00	1.0 ± 0.00	1.0 ± 0.00	1.2 ± 0.11	3.4 ± 0.50	2.5 ± 0.15	2.7 ± 0.17	5.2 ± 0.29
0.1Qmed	1.5 ± 0.05	1.0 ± 0.00	1.0 ± 0.00	1.0 ± 0.00	1.0 ± 0.08	4.3 ± 1.01	2.7 ± 0.13	2.8 ± 0.19	5.3 ± 0.35
0.1Q0.9	0.9 ± 0.02	1.0 ± 0.01	1.0 ± 0.01	1.0 ± 0.00	0.8 ± 0.05	0.8 ± 0.06	1.4 ± 0.69	2.5 ± 0.16	4.8 ± 0.26
0.1Qmax	0.9 ± 0.04	1.0 ± 0.01	1.0 ± 0.01	1.0 ± 0.00	0.7 ± 0.04	0.7 ± 0.05	0.5 ± 0.02	0.5 ± 0.01	1.9 ± 0.04
neanQmin	2.5 ± 0.16	1.0 ± 0.00	1.0 ± 0.00	1.9 ± 0.12	4.3 ± 2.05	5.0 ± 0.24	3.0 ± 0.14	3.1 ± 0.18	5.3 ± 0.32
neanQnean	1.8 ± 0.08	1.0 ± 0.00	1.0 ± 0.00	1.0 ± 0.01	1.5 ± 0.33	3.7 ± 0.34	2.6 ± 0.11	2.8 ± 0.15	4.8 ± 0.25
neanQhean	1.6 ± 0.06	1.0 ± 0.00	1.0 ± 0.00	1.0 ± 0.00	1.3 ± 0.13	3.3 ± 0.38	2.5 ± 0.11	2.8 ± 0.15	4.8 ± 0.24
neanQmed	1.4 ± 0.04	1.0 ± 0.00	1.0 ± 0.00	1.0 ± 0.00	1.0 ± 0.04	4.2 ± 0.90	2.7 ± 0.10	2.8 ± 0.17	4.9 ± 0.27
neanQ0.9	1.0 ± 0.01	1.0 ± 0.00	1.0 ± 0.00	1.0 ± 0.00	0.9 ± 0.02	0.7 ± 0.04	1.4 ± 0.66	2.5 ± 0.15	4.4 ± 0.23
neanQmax	0.9 ± 0.02	1.0 ± 0.00	1.0 ± 0.00	1.0 ± 0.00	0.9 ± 0.03	0.6 ± 0.03	0.6 ± 0.02	0.6 ± 0.01	1.7 ± 0.03
medQmin	2.6 ± 0.16	1.0 ± 0.01	1.0 ± 0.00	1.8 ± 0.12	4.5 ± 2.20	5.1 ± 0.25	3.1 ± 0.17	3.2 ± 0.13	5.5 ± 0.27
medQnean	1.8 ± 0.09	1.0 ± 0.00	1.0 ± 0.00	1.0 ± 0.01	1.6 ± 0.37	3.7 ± 0.34	2.6 ± 0.13	2.8 ± 0.14	4.9 ± 0.24
medQhean	1.7 ± 0.07	1.0 ± 0.00	1.0 ± 0.00	1.0 ± 0.00	1.3 ± 0.16	3.3 ± 0.38	2.5 ± 0.13	2.8 ± 0.14	4.9 ± 0.24
medQmed	1.4 ± 0.03	1.0 ± 0.01	1.0 ± 0.00	1.0 ± 0.00	1.0 ± 0.02	4.1 ± 0.85	2.7 ± 0.13	2.9 ± 0.16	5.0 ± 0.28
medQ0.9	1.0 ± 0.01	1.0 ± 0.00	1.0 ± 0.00	1.0 ± 0.00	0.9 ± 0.03	0.7 ± 0.05	1.4 ± 0.64	2.5 ± 0.15	4.3 ± 0.25
medQmax	1.0 ± 0.02	1.0 ± 0.00	1.0 ± 0.00	1.0 ± 0.00	0.9 ± 0.04	0.6 ± 0.04	0.6 ± 0.02	0.6 ± 0.01	1.7 ± 0.02
meanQmin	2.5 ± 0.16	1.0 ± 0.00	1.0 ± 0.00	1.8 ± 0.11	4.3 ± 2.14	5.2 ± 0.26	3.1 ± 0.16	3.1 ± 0.19	5.3 ± 0.33
meanQnean	1.8 ± 0.08	1.0 ± 0.00	1.0 ± 0.00	1.0 ± 0.00	1.5 ± 0.35	3.9 ± 0.36	2.6 ± 0.11	2.8 ± 0.15	4.8 ± 0.25
meanQhean	1.6 ± 0.06	1.0 ± 0.00	1.0 ± 0.00	1.0 ± 0.00	1.3 ± 0.14	3.4 ± 0.39	2.5 ± 0.11	2.8 ± 0.15	4.7 ± 0.24
meanQmed	1.4 ± 0.03	1.0 ± 0.00	1.0 ± 0.00	1.0 ± 0.00	1.0 ± 0.03	4.4 ± 0.96	2.7 ± 0.12	2.8 ± 0.17	4.8 ± 0.27
meanQ0.9	1.0 ± 0.01	1.0 ± 0.00	1.0 ± 0.00	1.0 ± 0.00	0.9 ± 0.02	0.7 ± 0.03	1.4 ± 0.67	2.5 ± 0.15	4.3 ± 0.22
meanQmax	0.9 ± 0.01	1.0 ± 0.00	1.0 ± 0.00	1.0 ± 0.00	0.9 ± 0.03	0.7 ± 0.03	0.5 ± 0.02	0.6 ± 0.01	1.7 ± 0.03
maxQmin	2.2 ± 0.13	1.0 ± 0.03	1.0 ± 0.03	1.7 ± 0.08	3.8 ± 2.57	6.6 ± 0.63	4.2 ± 0.18	4.6 ± 0.38	4.5 ± 0.36
maxQnean	1.6 ± 0.07	1.0 ± 0.00	1.0 ± 0.00	1.0 ± 0.00	1.4 ± 0.36	5.0 ± 0.88	3.7 ± 0.29	4.0 ± 0.20	4.0 ± 0.20
maxQhean	1.5 ± 0.05	1.0 ± 0.00	1.0 ± 0.00	1.0 ± 0.00	1.2 ± 0.13	4.3 ± 0.85	3.5 ± 0.31	4.0 ± 0.19	4.0 ± 0.19
maxQmed	1.3 ± 0.06	1.0 ± 0.00	1.0 ± 0.00	1.0 ± 0.00	1.0 ± 0.04	5.6 ± 1.84	3.9 ± 0.31	4.0 ± 0.25	4.0 ± 0.23
maxQ0.9	1.0 ± 0.00	1.0 ± 0.00	1.0 ± 0.00	1.0 ± 0.00	0.9 ± 0.03	0.9 ± 0.03	1.7 ± 1.14	3.8 ± 0.16	3.7 ± 0.18
maxQmax	1.0 ± 0.00	1.0 ± 0.00	1.0 ± 0.00	1.0 ± 0.00	0.9 ± 0.04	0.9 ± 0.04	0.6 ± 0.06	1.0 ± 0.00	1.0 ± 0.00
detKmin	9.0 ± 2.39	1.0 ± 0.00	1.0 ± 0.00	1.0 ± 0.03	3080 ± 4071	679 ± 227	51 ± 32.2	23 ± 7.36	26 ± 8.95
detKnean	4.8 ± 1.03	1.0 ± 0.01	1.0 ± 0.01	1.0 ± 0.02	592 ± 814	389 ± 104	32 ± 15.5	15 ± 4.13	18 ± 5.37
detKhean	3.7 ± 0.69	1.0 ± 0.01	1.0 ± 0.01	1.0 ± 0.02	137 ± 197	302 ± 82	29 ± 12.5	14 ± 3.80	17 ± 4.96
detKmed	1.7 ± 0.14	1.0 ± 0.00	1.0 ± 0.00	1.0 ± 0.00	1.2 ± 0.27	365 ± 174	30 ± 11.1	15 ± 3.64	17 ± 4.66
detK0.9	1.0 ± 0.05	1.0 ± 0.00	1.0 ± 0.00	1.0 ± 0.04	0.7 ± 0.10	0.1 ± 0.04	4.8 ± 6.28	8.9 ± 3.05	10 ± 3.83
detKmax	0.9 ± 0.15	0.9 ± 0.22	0.9 ± 0.22	0.1 ± 0.03	0.6 ± 0.10	0.1 ± 0.05	0.1 ± 0.02	0.9 ± 0.05	1.2 ± 0.07

from the much denser cluster. The vectors $\mathbf{v}_O = (8, 0)^T$, $\mathbf{v}_R = (11, 0)^T$ and $\mathbf{v}_P = (8.5, 0)^T$, $\mathbf{v}_Q = (10.5, 0)^T$ are in the same distance from the center of the dense cluster, so we can compare how can the algorithms detect the outliers in the dataset with noise ($\mathbf{v}_O$, $\mathbf{v}_P$) and without noise ($\mathbf{v}_Q$, $\mathbf{v}_R$). The vectors $\mathbf{v}_Q$ and $\mathbf{v}_R$ are obvious outliers, each of them lies within the set of k neighbors of the other vector, but $\mathbf{v}_R$ is the most distant vector in the neighborhood of the vector $\mathbf{v}_Q$ and also in its own neighborhood. Any other vector, except for the vector $\mathbf{v}_Q$ and $\mathbf{v}_R$, does not have the vector $\mathbf{v}_Q$ or $\mathbf{v}_R$ within its set of k neighbors (see Fig. 1 (c)). The parameter k is set on 40 in all the algorithms. The experiment was performed ten times. One can see the sample mean and the sample standard deviation of outlier factors of the added vectors for the tested algorithm in the Table II, where the vertical lines highlights the different groups of selected vectors. The horizontal lines separate the different groups of algorithms.

On the basis of the two first experiments, we can state that the usage of the set **Q** considerably decreases the deviation of the OF values from 1 compared to the set **D**. The shortcoming of the usage of the set **Q** is clearly demonstrated on the vector $\mathbf{v}_L$, which is in all the cases labeled as an inlier. It is possible to say that if there is k considerably detached vectors and the set **Q** is applied, then they will be all labeled as inliers, no matter what is their mutual position. The set **Q** considerably suppress the effect of the combination of low quantiles for the calculation of both R_p and R_{avg} simultaneously in the case that the cluster consists of less than k vectors. The algorithm *maxQmax* is the only algorithm that labeled the vector $\mathbf{v}_R$ as an inlier.

The results of the algorithms applying *detK* are very similar to the algorithms applying **Q**, both label the vector $\mathbf{v}_L$ as an inlier, but the OF value of outliers strongly increases. The vector $\mathbf{v}_L$ is labeled as an inlier, because the identical set **K** with the identical distribution was used for the calculation of the OF value of all the vectors within the cluster around the vector $\mathbf{v}_K$, and also of the vector $\mathbf{v}_L$. Unlike the algorithm $\ldots$*Qmin* the algorithm *detKmin* labeled the vector $\mathbf{v}_M$ as an inlier.

Low quantiles applied for the calculation of R_p increase the deviation of the OF values from 1, high quantiles decrease the deviation of the OF values from 1. The average *nean* is similar to a very low quantile, whereas the average *mean* is similar to the median or a little higher quantile. The average *nean* is computationally unstable when applied on the set **D**, especially when combined with a low quantile for the calculation of R_{avg}. Vectors of the clusters consisting of less than k vectors, but simultaneously denser than their neighborhood, can be labeled as inliers, if a quantile low enough is applied for the calculation of R_p.

The OF values are influenced the most by the way how the R_{avg} is calculated. There are two extremes. When the minimum is applied, only the vector with the smallest R_p from all the vectors within its neighborhood is labeled as an inlier. In other words, only the vector with the densest neighborhood from all the vectors in its neighborhood is labeled as an inlier. On the other extreme, when the maximum is applied, only the vector with the biggest R_p from all the vectors within its neighborhood is labeled as an outlier. In other words, the vector is labeled as an outlier only when it is the most outlying vector within its neighborhood. The averages *nean* and *hean* in most cases generate similar results as the median, but in the case demonstrated by the vector $\mathbf{v}_N$ they generate the results similar to lower quantiles. It means that they are influenced by the presence of a small number of vectors with strongly higher density of their neighborhood, in the neighborhood of the examined vector. The average *nean* is influenced more.

V. Conclusion and Future Work

The original algorithms *meanQhean* (LOF) and *maxDhean* (LOF') are comparable, *maxDhean* is little bit faster and *meanQhean* has better results for the vectors on the border of clusters generated by the uniform distribution. We are convinced that LOF should be defined as *neanQnean* not only because it is geometrically much more elegant, but also because *neanQnean* increases the OF values for outliers and therefore highlights them, what is described as convenient in [12].

As demonstrated by the results of the experiments, the OF values are only very little influenced by the way how the R_p is calculated. Therefore we recommend that the researchers apply the individual quantiles of the set **D**, which is easier to calculate, according to whether they want to detect even denser regions smaller than k. The parameter k can be set relatively high, it means much more than generally recommended $k = 20$.

Especially, if we suppose that the dataset contains a lot of noise and relatively sparse clusters, it is essential to set the parameter k high and to apply a low quantile of the set **D** what cannot be replaced by the original LOF algorithm. The similar idea is proposed by the LOF" algorithm.

It is much more important how the R_{avg} is calculated. If a researcher wants to find only strong outliers with a low probability to label an inlier wrongly as an outlier, then it is important to compute R_{avg} as a high quantile of the set of all R_i in the neighborhood of the examined vector. In the extreme case, it is possible to apply max R_i.

If a researcher wants to be sure that only the vectors with considerably denser neighborhood will be labeled as inliers, or if a researcher wants to minimize the probability to label an outlier wrongly as an inlier, then it is important to compute R_{avg} as a low quantile of the set of all R_i in the neighborhood of the examined vector.

In general, the following algorithms are recommended: the algorithm $0.1D0.1$ with high parameter k for the detection of the centers of the clusters or in case of a dataset with a lot of noise, the algorithm $maxD0.9$ for the detection of the most distant outliers or the clusters smaller than k in the dataset with relatively low portion of noise, and the algorithm $medDmed$ if a researcher wants to eliminate as many vectors as possible without the loss of the information.

In the future work, we would like to focus on preparing the datasets for the clustering, where we would like to use outlier factors as applicable weights for clustering algorithms. We would like to extend the method experimental evaluation using large real world datasets such as the one described in [13]. Thus, we would be able to evaluate impact of the outlier detection on robustness of AI algorithms used in the mobile robotics domain [14], [15], especially for an UAV [16].

ACKNOWLEDGMENTS.

This work was supported by the project IGA F4/6/2012.

REFERENCES

[1] M. Žambochová, “Typology of foreign students interested in studying at czech universities,” *E+M Ekonomika a Management*, no. 2, pp. 141–154, 2012.

[2] A. Frolov, D. Husek, and P. Polyakov, “Recurrent-neural-network-based boolean factor analysis and its application to word clustering,” *Neural Networks, IEEE Transactions on*, vol. 20, no. 7, pp. 1073–1086, 2009.

[3] K. Wegrzyn-Wolska, G. Dziczkowski, and L. Bougueroua, “Linking the drugs and pharmaceutical databases,” in *Next Generation Web Services Practices, 2009. NWESP’09. Fifth International Conference on*. IEEE, 2009, pp. 3–8.

[4] E. Zimková and V. Úradníček, “Inflačné cielenie a možnosti predikovania inflácie v podmienkach slovenska,” *Ekonomický časopis*, no. 06, p. 658, 2004.

[5] M. Breunig, H. Kriegel, R. Ng, J. Sander *et al.*, “Lof: identifying density-based local outliers,” *Sigmod Record*, vol. 29, no. 2, pp. 93–104, 2000.

[6] A. Chiu and A. Fu, “Enhancements on local outlier detection,” in *Seventh International Database Engineering and Applications Symposium, 2003. Proceedings.* IEEE, 2003, pp. 298–307.

[7] S. Papadimitriou, H. Kitagawa, P. Gibbons, and C. Faloutsos, “Loci: Fast outlier detection using the local correlation integral,” in *19th International Conference on Data Engineering, 2003. Proceedings.* IEEE, 2003, pp. 315–326.

[8] E. Knorr and R. Ng, “Algorithms for mining distance-based outliers in large datasets,” in *Proceedings of the International Conference on Very Large Data Bases.* Citeseer, 1998, pp. 392–403.

[9] H. Cao, G. Si, W. Zhu, and Y. Zhang, “Enhancing effectiveness of density-based outlier mining,” in *International Symposiums on Information Processing (ISIP), 2008.* IEEE, 2008, pp. 149–154.

[10] W. Jin, A. Tung, J. Han, and W. Wang, “Ranking outliers using symmetric neighborhood relationship,” *Advances in Knowledge Discovery and Data Mining*, pp. 577–593, 2006.

[11] J. Tang, Z. Chen, A. Fu, and D. Cheung, “Enhancing effectiveness of outlier detections for low density patterns,” *Advances in Knowledge Discovery and Data Mining*, pp. 535–548, 2002.

[12] H. Kriegel, P. Kröger, J. Sander, and A. Zimek, “Density-based clustering,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, no. 3, pp. 231–240, 2011.

[13] T. Vintr, L. Pastorek, and H. Rezankova, “Autonomous robot navigation based on clustering across images,” *Research and Education in Robotics-EUROBOT 2011*, pp. 310–320, 2011.

[14] T. Krajník and L. Přeučil, “A Simple Visual Navigation System with Convergence Property,” in *European Robotics Symposium 2008.* Heidelberg: Springer, 2008, pp. 283–292.

[15] T. Krajník, J. Faigl, M. Vonásek, V. Kulich, K. Košnar, and L. Přeučil, “Simple yet stable bearing-only navigation,” *J. Field Robot.*, 2010.

[16] T. Krajník, M. Nitsche, S. Pedre, L. Přeučil, and M. Mejail, “A Simple Visual Navigation System for an UAV,” in *International Multi-Conference on Systems, Signals and Devices*. Piscataway: IEEE, 2012, p. 34.