Concentration Free Outlier Detection

Fabrizio Angiulli[1]

University of Calabria, 87036 Rende (CS), Italy,
fabrizio.angiulli@unical.it,
WWW home page: http://siloe.dimes.unical.it/angiulli

Abstract. We present a novel notion of outlier, called Concentration Free Outlier Factor (CFOF), having the peculiarity to resist concentration phenomena that affect other scores when the dimensionality of the feature space increases. Indeed we formally prove that CFOF does not concentrate in intrinsically high-dimensional spaces. Moreover, CFOF is adaptive to different local density levels and it does not require the computation of exact neighbors in order to be reliably computed. We present a very efficient technique, named *fast*-CFOF, for detecting outliers in very large high-dimensional datasets. The technique is efficiently parallelizable, and we provide a MIMD-SIMD implementation. Experimental results witness for scalability and effectiveness of the technique and highlight that CFOF exhibits state of the art detection performances.

Keywords: Outlier detection, Curse of dimensionality

1 Introduction

Outlier detection is a prominent data mining task, whose goal is to single out anomalous observations, also called outliers [2]. While the other data mining approaches consider outliers as noise that must be eliminated, as pointed out in [11] "one person's noise could be another person's signal", thus outliers themselves are of great interest in different settings (e.g. fraud detection, ecosystem disturbances, intrusion detection, cybersecurity, medical analysis, to cite a few).

Data mining outlier approaches can be supervised, semi-supervised, and unsupervised [13, 8]. Supervised methods take in input data labeled as normal and abnormal and build a classifier. The challenge there is posed by the fact that abnormal data form a rare class. Semi-supervised methods, also called one-class classifiers or domain description techniques, take in input only normal examples and use them to identify anomalies. Unsupervised methods detect outliers in an input dataset by assigning a score or anomaly degree to each object.

Unsupervised outlier detection methods can be categorized in several approaches, each of which assumes a specific concept of outlier. Among the most popular families there are distance-based [16, 23, 5, 4], density-based [7, 15, 20], angle-based [18], isolation-forest [19], subspace methods [1, 14], and others [2, 9, 25].

This work focuses on unsupervised outlier detection problem in the full feature space. In particular, we introduce a novel notion of outlier, the Concentration Free Outlier Factor (CFOF), having the peculiarity to resist concentration phenomena affecting other measures. Informally, the CFOF score measures how many neighbors have to be taken into account in order for the object to be considered close by an appreciable fraction of the population. The term distance concentration refers to the tendency of distances to become almost indiscernible as dimensionality increases, and is part of the so called curse of dimensionality problem [6, 10]. And, indeed, the concentration problem also affects outlier scores of different families due to the specific role played by distances in their formulation [17, 25]. Moreover, a special kind of concentration phenomenon, known as hubness, concerns scores based on reverse nearest neighbor counts [12, 22], that is the concentration of the scores towards the values associated with anomalies, which results in almost all the dataset composed of outliers.

The contributions of the work within this scenario are summarized next:

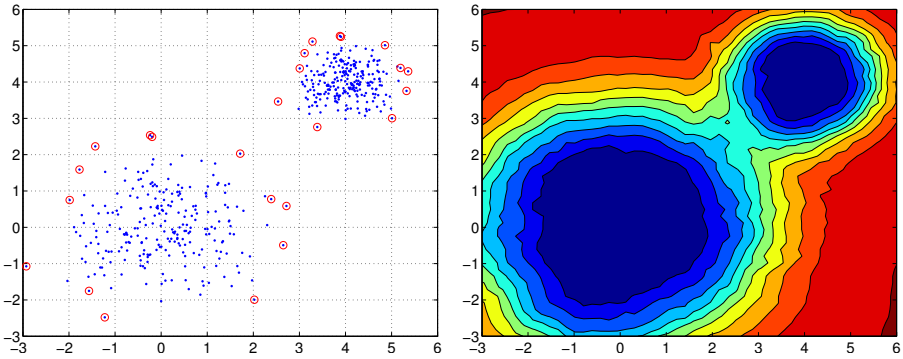
- As a major peculiarity, we formally show that, differently from the practical totality of existing outlier scores, the CFOF score distribution is not affected by concentration phenomena arising when the dimensionality of the space increases.
- The CFOF score is adaptive to different local density levels. Despite local methods usually require to know the exact nearest neighbors in order to compare the neighborhood of each object with the neighborhood of its neighbors, this is not the case for CFOF, which can be reliably computed through sampling. This characteristics is favored by the separation between inliers and outliers guaranteed by the absence of concentration.
- We describe the *fast*-CFOF technique, which from the computational point of view does not suffer of the problems affecting (reverse) nearest neighbor search techniques. The cost of *fast*-CFOF is linear both in the dataset size and dimensionality. Moreover, we provide a multi-core (MIMD) vectorized (SIMD) implementation.
- The applicability of the technique is not limited to Euclidean or vector spaces. It can be applied both in metric and non-metric spaces equipped with a distance function.
- Experimental results highlight that CFOF exhibits state of the art detection performances.

The rest of the work is organized as follows. Section 2 introduces the CFOF score and its properties. Section 3 describes the *fast*-CFOF algorithm. Section 4 presents experiments. Finally, Section draws conclusions.

2 The Concentration Free Outlier Factor

2.1 Definition

We assume that a dataset $\mathbf{DS} = \{x_1, x_2, \ldots, x_n\}$ of n objects belonging to an object space $\mathbb{U}$, on which a distance function dist is defined, is given in input.



(a) The top 25 outliers according to the CFOF definition.

(b) Density estimation performed by means of the CFOF measure.

Fig. 1: Two normal clusters with different standard deviation.

We assume that $\mathbb{U} = \mathbb{D}^d$ (where $\mathbb{D}$ is usually the set $\mathbb{R}$ of real numbers), with $d \in \mathbb{N}^+$, but the method can be applied in any object space equipped with a distance function (not necessarily a metric).

Given an object x and a positive integer k, the *k-th nearest neighbor* of x is the object $nn_k(x)$ such that there exists exactly $k-1$ objects lying at distance smaller than $\text{dist}(x, nn_k(x))$ from x. It always holds that $x = nn_1(x)$. We assume that ties are non-deterministically ordered. The *k nearest neighbors set* $\text{NN}_k(x)$ of x, where k is also called the *neighborhood width*, consists of the objects $\{nn_i(x) \mid 1 \leq i \leq k\}$.

By $\text{N}_k(x)$ we denote the number of objects having x among their k nearest neighbors:

$$\text{N}_k(x) = |\{y : x \in \text{NN}_k(y)\}|,$$

also referred to as *reverse k nearest neighbor count* or *reverse neighborhood size*.

Given a parameter $\varrho \in (0, 1)$ (or equivalently a parameter $k_\varrho \in [1, n]$ such that $k_\varrho = n\varrho$), the *Concentration Free Outlier Score*, also referred to as CFOF, is defined as:

$$\text{CFOF}(x) = \min\left\{k/n : \text{N}_k(x) \geq n\varrho\right\}, \tag{1}$$

that is to say, the score returns the smallest neighborhood width (normalized with respect to n) for which the object x exhibits a reverse neighborhood of size at least $n\varrho$ (or k_ϱ).[1]

Intuitively, the CFOF score measures how many neighbors have to be taken into account in order for the object to be considered close by an appreciable

[1] Notice that k (or k/n), representing a neighborhood width, denotes the output of CFOF, while the other outlier definitions employ k as an input parameter. We warn the reader that, in order to make more intelligible the comparison of CFOF with other outlier techniques, sometimes we will refer to k as an input parameter (the use will be clear from the context). Moreover, in order to avoid confusion and to maintain analogy with the input parameter ϱ, we also refer to ϱ as k_ϱ.

fraction of the dataset objects. We notice that this kind of notion of perceiving the abnormality of an observation is completely different from any other notion so far introduced in the literature.

The CFOF score is adaptive to different density levels. This characteristics is also influenced by the fact that actual distance values are not employed in its computation. Thus, CFOF is invariant to all of the transformations the leave unchanged the nearest neighbor ranking, such as translation or scaling. Also, duplicating the data in a way that avoids to affect the original neighborhood order (e.g. by creating a separate, possibly scaled, cluster from each copy of the original data) will preserve original scores.

Consider Figure 1 showing a dataset consisting of two normally distributed clusters, each consisting of 250 points. The cluster centered in $(4, 4)$ is obtained by translating and scaling (by a factor 0.5) the cluster centered in the origin. The top 25 CFOF outliers for $k_\varrho = 20$ are highlighted (objects within small circles). It can be seen that the outliers are the "same" objects of the two clusters.

2.2 Relationship with the distance concentration phenomenon

The term distance concentration, which is part of the so called curse of dimensionality problem [6], refers to the tendency of distances to become almost indiscernible as dimensionality increases. In a more quantitative way this phenomenon is measured through the ratio between a quantity related to the mean μ and a quantity related to the standard deviation σ of the distance distribution of interest. E.g., in [10] the intrinsic dimensionality ρ of a metric space is defined as $\rho = \mu_d^2/(2\sigma_d^2)$, where μ_d is the mean of the pairwise distance distribution and σ_d the associated standard deviation. The intrinsic dimensionality intends to quantify the expected difficulty of performing a nearest neighbor search: the smaller the ratio the larger the difficulty to search on an arbitrary metric space.

In general, it is said that we have concentration when this kind of ratio tends to zero as dimensionality goes to infinity, as it is the case for objects with i.i.d. attributes.

The concentration problem also affects different families of outlier scores, due to the specific role played by distances in their formulation.

Figure 2 reports the sorted scores of different outlier detection techniques, that are aKNN [5], LOF [7], ABOF [18], and CFOF (the parameters k of aKNN, LOF, and ABOF, and k_ϱ of CFOF, are held fixed to 50 for all the scores), associated with a family of uniformly distributed datasets having fixed size ($n = 1000$) and increasing dimensionality $d \in [10^0, 10^4]$. The figure highlights that, except for CFOF, the other scores exhibit a concentration effect. For aKNN (Figure 2a) the mean score value raises while the spread stay limited. For LOF (Figure 2b) all the values tend to 1 as the dimensionality increases. For ABOF (Figure 2c) both the mean and the standard deviation decrease of various orders of magnitude with the latter term varying at a faster rate than the former one. As for CFOF the score distributions for $d > 100$ are very close and exhibit only slight changes. Notably, the separation between scores associated with outliers and inliers is always ample.

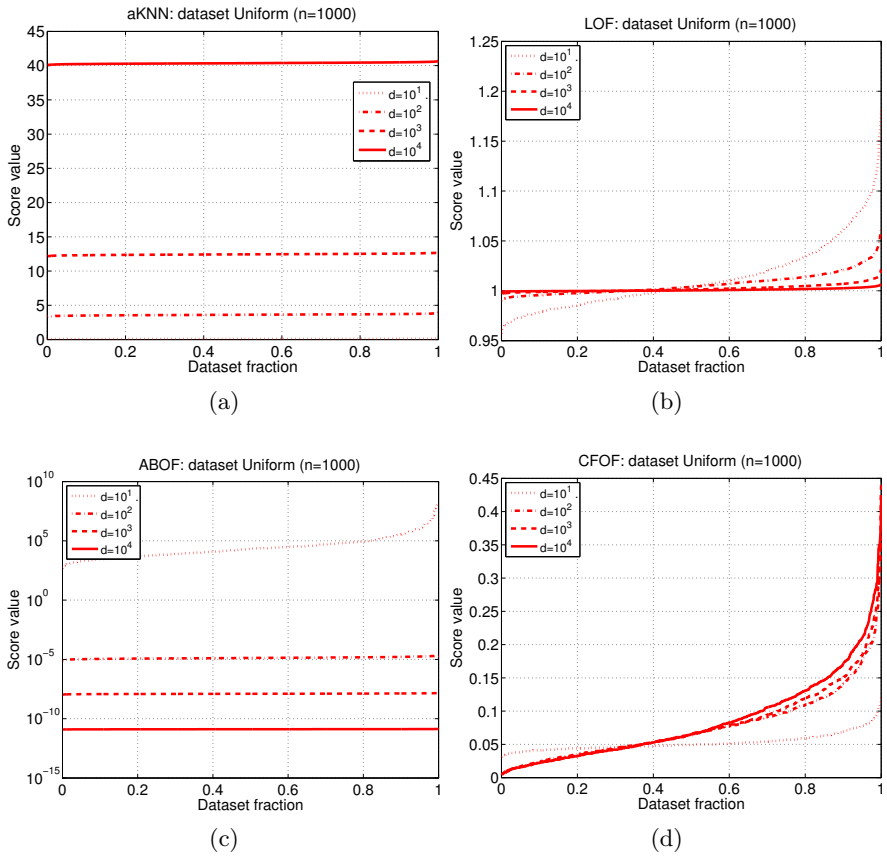
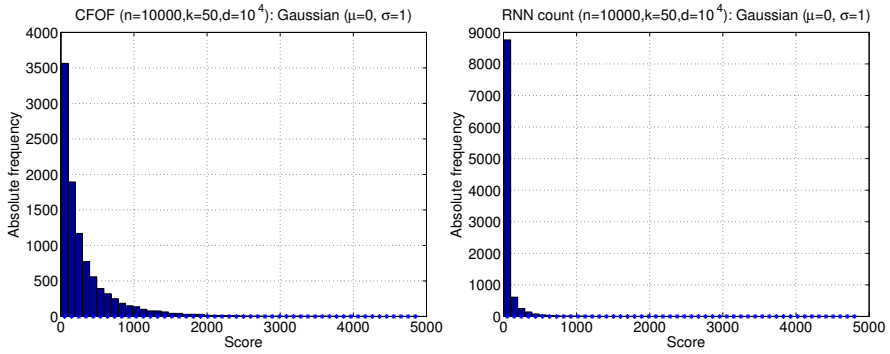


Fig. 2: Sorted outlier scores.

2.3 Relationship with the hubness phenomenon

CFOF has connections with the reverse neighborhood size, a tool which has been also used for characterizing outliers. In [12], the authors proposed the use of the reverse neighborhood size $\mathrm{N}_k(\cdot)$ as an outlier score, which we refer to as RNN count (RNNc for short). Outliers are those objects associated with the lowest RNN counts. However, RNNc suffers of a peculiar problem known as *hubness* [21]. As the dimensionality of the space increases, the number of *antihubs*, that are objects appearing in a much lower number of k nearest neighbors sets (possibly they are neighbors only of themselves), overcomes the number of *hubs*, that are objects that appear in many more k nearest neighbor sets than other points, and, according to the RNNc score, the vast majority of the dataset objects become outliers with identical scores.

We provide evidence that CFOF does not present the hubness problem. Figure 3 reports the distribution of the $N_k(\cdot)$ value and of the CFOF absolute score



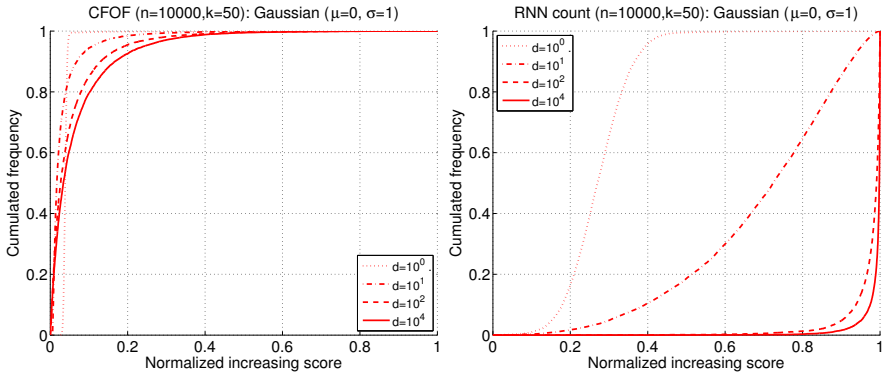
(a) (b)

Fig. 3: Distribution of CFOF and RNN counts.

for a ten thousand dimensional normal dataset (a very similar behavior has been observed also for uniform data). Notice that CFOF outliers are associated with the largest score values, hence to the tails of the corresponding distribution, while RNNc outliers are associated with the smallest score values, hence with the largely populated region of the associated score distribution, a completely opposite behavior. To illustrate the impact of the hubness problem with the dimensionality, Figure 4 shows the cumulative frequency associated with the normalized, between 0 and 1, increasing score. This transformation has been implemented here in order to make the comparison much more interpretable. Original scores have been mapped to $[0, 1]$. CFOF scores have been divided by their maximum value. The mapping for $\mathrm{N}_k(\cdot)$ has been obtained as $1 - \frac{\mathrm{N}_k(x)}{\max_y \mathrm{N}_k(y)}$, since outliers are those objects associated with the lowest counts. The plots make evident the deep difference between the two approaches. Here both n and k for RNNc (k_ϱ for CFOF, resp.) are held fixed, while d is increased. As for RNNc, the hubness problem is already evident for $d = 10$ (where objects with a normalized score ≥ 0.8 corresponds to about the 40% of the dataset), while the curve for $d = 10^2$ closely resembles that for $d = 10^4$ (where almost all the dataset objects have a normalized score ≥ 0.8). As far as CFOF is concerned, the two curves for $d = 10^4$ closely resemble each other and the number of objects associated with a large score value always correspond to a very small fraction of the dataset population.

2.4 Concentration free property of CFOF

In this section we formally prove that the CFOF score is concentration-free. Specifically, the following theorem shows that the separation between the scores associated with outliers and the rest of the scores is guaranteed in any arbitrary large dimensionality.



(a) (b)

Fig. 4: Comparison between CFOF and RNN counts.

Before going into the details, we recall that the concept of intrinsic dimensionality of a space is identified as the minimum number of variables needed to represent the data, which corresponds in a linear space to the number of linearly independent vectors needed to describe each point.

Theorem 1. *Let $\boldsymbol{DS}^{(d)}$ be a d-dimensional dataset consisting of realizations of a d-dimensional independent and (non-necessarily) identically distributed random vector $\boldsymbol{X}$ having distribution function f. Then, as $d \to \infty$, the CFOF scores of the points of $\boldsymbol{DS}^{(d)}$ do not concentrate.*

Proof. Consider the squared norm $\|\mathbf{X}\|^2 = \sum_{i=1}^{d} X_i^2$ of the random vector $\mathbf{X}$. As $d \to \infty$, by the Central Limit Theorem, the standard score of $\sum_{i=1}^{d} X_i^2$ tends to a standard normal distribution. This implies that $\|\mathbf{X}\|^2$ approaches a normal distribution with mean $\mu_{\|\mathbf{X}\|^2} = \mathbf{E}[X_i^2] = d\mu_2$ and variance $\sigma^2_{\|\mathbf{X}\|^2} = d(\mathbf{E}[(X_i^2)^2]) - \mathbf{E}[X_i^2]) = d(\mu_4 - \mu_2^2)$, where μ_2 and μ_4 are the 2nd- and 4th-order central moments of the univariate probability distribution f.

In the case that the components X_i of $\mathbf{X}$ are non-identically distributed according to the distributions f_i $(1 \leq i \leq d)$, the result still holds by considering the average of the central moments of the f_i functions.

Let x be an element of $\mathbf{DS}^{(d)}$ and define the zeta score z_x of the squared norm of x as

$$z_x = \frac{\|x\|^2 - \mu_{\|\mathbf{X}\|^2}}{\sigma_{\|\mathbf{X}\|^2}}.$$

It can be shown [3] that, for large values of d, the number of k-occurrences of x is given by

$$\mathrm{N}_k(x) = n \cdot Pr[x \in \mathrm{NN}_k(\mathbf{X})] \approx n\Phi\left(\frac{\Phi^{-1}(\frac{k}{n})\sqrt{\mu_4 + 3\mu_2^2} - z_x\sqrt{\mu_4 - \mu_2^2}}{2\mu_2}\right).$$

Let $t(z_x)$ denote the smallest integer k such that $\mathrm{N}_k(x) \geq n\varrho$. By exploiting the equation above it can be concluded that

$$t(z_x) \approx n\Phi\left(\frac{z_x\sqrt{\mu_4 - \mu_2^2} + 2\mu_2\Phi^{-1}(\varrho)}{\sqrt{\mu_4 + 3\mu_2^2}}\right).$$

Since $\mathrm{CFOF}(x) = k/n$ implies that k is the smallest integer such that $\mathrm{N}_k(x) \geq n\varrho$, it also follows that $\mathrm{CFOF}(x) \approx t(z_x)/n = \hat{t}(z_x)$.

Moreover, since as stated above the $\|\mathbf{X}\|^2$ random variable is normally distributed, it also holds that for each $z \geq 0$

$$Pr\left[\frac{\|\mathbf{X}\|^2 - \mu_{\|\mathbf{X}\|^2}}{\sigma_{\|\mathbf{X}\|^2}} \leq z\right] = \Phi(z),$$

where $\Phi(\cdot)$ denotes the cdf of the normal distribution.

Thus, for arbitrarily large values of d and for any standard score value $z \geq 0$

$$Pr\left[CFOF(\mathbf{X}) \geq \hat{t}(z)\right] = 1 - \Phi(z),$$

irrespective of the actual data dimensionality value d.

3 Score computation

CFOF scores can be determined in time $O(n^2 d)$, where d denotes the dimensionality of the feature space (or the cost of computing a distance), after computing all pairwise dataset distances.[2] Next we introduce a technique, named *fast*-CFOF which does not require the computation of the exact nearest neighbor sets and, from the computational point of view, does not suffer of the curse of dimensionality affecting nearest neighbor search techniques.

The technique builds on the following probabilistic formulation of the CFOF score. Assume that the dataset consists of n i.i.d. samples drawn according to an unknown probability law $p(\cdot)$. Given a parameter $\varrho \in (0, 1)$, the (*Probabilistic*) *Concentration Free Outlier Factor* CFOF is defined as follows:

$$\mathrm{CFOF}(x) = \min\left\{k/n : \mathbf{E}\left[Pr[x \in \mathrm{NN}_k(y)]\right] \geq \varrho\right\}. \tag{2}$$

To differentiate the two definitions reported in Eqs. (1) and (2), we also refer to the former as *hard*-CFOF and to the latter as *soft*-CFOF. Intuitively, the *soft*-CFOF score measures how many neighbors have to be taken into account in order for the expected number of dataset objects having it among their neighbors correspond to the fraction ϱ of the overall population.

[2] It is generally recognized that this cost can be reduced to $O(dn \log n)$ in low dimensional spaces.

3.1 The *fast*-CFOF technique

Given a dataset **DS** and two objects x and y from **DS**, the building block of the algorithm is the computation of $Pr[x \in \mathrm{NN}_k(y)]$. Consider the boolean function $B_{x,y}(z)$ defined on instances z of **DS** such that $B_{x,y}(z) = 1$ if z lies within the region $\mathcal{I}_{\mathrm{dist}(x,y)}(y)$, and 0 otherwise. We want to estimate the average value $\overline{B}_{x,y}$ of $B_{x,y}$ in **DS**, which corresponds to the probability $p(x, y)$ that a randomly picked dataset object z is at distance not grater than $\mathrm{dist}(x, y)$ from y.

It is enough to compute $\overline{B}_{x,y}$ within a certain error bound. Thus, we resort to *batch sampling*, consisting in picking up s elements of **DS** randomly and estimating $p(x, y) = \overline{B}_{x,y}$ as the fraction $\hat{p}(x, y)$ of the elements of the sample satisfying $B_{x,y}$ [24]. Given $\delta > 0$ (an *error probability*) and ϵ, $0 < \epsilon < 1$ (an *absolute error*), if the sample size s satisfies certain conditions [24] then

$$Pr[|\hat{p}(x, y) - p(x, y)| \leq \epsilon] > 1 - \delta. \tag{3}$$

For large values of n, since the variance of the Binomial distribution becomes negligible with respect to the mean, the cdf *binocdf*$(k; p, n)$ tends to the step function $H\big(k - np\big)$, where $H\big(k\big) = 0$ for $k < 0$ and $H\big(k\big) = 1$ for $k > 0$. Thus, we can approximate the value $Pr[x \in \mathrm{NN}_k(y)] = \mathit{binocdf}(k; p(x, y), n)$ with the boolean function $H\big(k - k_{up}(x, y)\big)$, with $k_{up}(x, y) = n\widehat{p}(x, y)$.[3] It then follows that we can obtain $\mathbf{E}\big[Pr[x \in \mathrm{NN}_k(y)]\big]$ as the average value of the boolean function $H\big(k - n\widehat{p}(x, y)\big)$, whose estimate can be again obtained by exploiting batch sampling. Specifically, *fast*-CFOF exploits the one single sample in order to perform the two estimates above described.

The algorithm *fast*-CFOF receives in input a list $\boldsymbol{\varrho} = \varrho_1, \ldots, \varrho_\ell$ of values for the parameter ϱ, since it is able to perform a *multi-resolution analysis*, that is to compute scores associated with different values of the parameter ϱ with no additional cost. Both $\boldsymbol{\varrho}$ and parameters ϵ, δ can be conveniently left at the default value ($\boldsymbol{\varrho} = 0.001, 0.005, 0.01, 0.05, 0.1$ and $\epsilon, \delta = 0.01$; see later for details).

First, the algorithm determines the size $s = \left\lceil \frac{1}{2\epsilon^2} \log\left(\frac{1}{\delta}\right) \right\rceil$ of the *sample* (or *partition*) of the dataset needed in order to guarantee the bound reported in Eq. (3). We notice that the algorithm does not require the dataset to be entirely loaded in main memory, since only a partition at a time is needed to carry out the computation. Thus, the technique is suitable also for disk resident datasets. We assume that dataset objects are randomly ordered and, hence, partitions can be contiguous. Otherwise, randomization can be done in linear time and constant space by disk-based shuffling. Each partition, consisting of a group of s consecutive objects, is processed by the subroutine *fast*-CFOF*_part* (see Algorithm 1), which estimates CFOF scores of the objects within the partition through batch sampling.

The matrix *hst*, consisting of $s \times B$ counters, is employed by *fast*-CFOF*_part*. The entry $hst(i, k)$ of *hst* is used to estimate how many times the sample object

[3] Alternatively, by exploiting the Normal approximation of the Binomial distribution, a suitable value for $k_{up}(x, y)$ is given by $k_{up}(x, y) = n\widehat{p}(x, y) + c\sqrt{n\widehat{p}(x, y)(1 - \widehat{p}(x, y))}$ with $c \in [0, 3]$.

Algorithm 1: *fast*-CFOF_*part*

Input: Dataset sample $\langle x'_1, \ldots, x'_s \rangle$ of size s, parameters $\varrho_1, \ldots, \varrho_\ell \in (0,1)$, dataset size n

Output: CFOF scores $\langle sc'_{1,\boldsymbol{\varrho}}, \ldots, sc'_{s,\boldsymbol{\varrho}} \rangle$

```
1  initialize matrix hst of s × B elements to 0;
   // Nearest neighbor count estimation
2  foreach i = 1 to s do
       // Distances computation
3      foreach j = 1 to s do
4          dst(j) = dist(x'_i, x'_j);
       // Count update
5      ord = sort(dst);
6      foreach j = 1 to s do
7          p = j/s;
8          k_up = ⌊np + c√(np(1−p)) + 0.5⌋;
9          k_pos = k_bin(k_up);
10         hst(ord(j), k_pos) = hst(ord(j), k_pos) + 1;
   // Scores computation
11 foreach i = 1 to s do
12     count = 0;
13     k_pos = 0;
14     l = 1;
15     while l ≤ ℓ do
16         while count < sϱ_l do
17             k_pos = k_pos + 1;
18             count = count + hst(i, k_pos);
19         sc'_{i,ϱ_l} = k_bin^{-1}(k_pos)/n;
20         l = l + 1;
```

x'_i is the kth nearest neighbor of a generic object dataset. Values of k, ranging from 1 to n, are partitioned into B log-spaced bins. The function k_bin maps original k values to the corresponding bin, while k_bin^{-1} implements the reverse mapping (by returning a certain value within the corresponding bin).

For each sample object x'_i, the distance $dst(j)$ from any other sample object x'_j is computed (lines 3-4) and, then, distances are ordered (line 5) obtaining the list ord of sample identifiers such that $dst(ord(1)) \leq dst(ord(2)) \leq \ldots \leq dst(ord(s))$.

Moreover, for each element $ord(j)$ of ord, the variable p is set to j/s (line 7), representing the probability $p(x'_{ord(j)}, x'_i)$, estimated through the sample, that a randomly picked dataset object is located within the region of radius $dst(ord(j)) = dist(x'_i, x'_{ord(j)})$ centered in x'_i. The value k_{up} (line 8) represents the point of transition from 0 to 1 of the step function $H\big(k - k_{up}\big)$ employed to approximate the probability $Pr[x'_{ord(j)} \in \mathrm{NN}_k(y)]$ when $y = x'_i$. Thus, before

concluding each cycle of the inner loop (lines 6-10), the $k\_bin(k_{up})$-th entry of hst associated with the sample $x'_{ord(j)}$ is incremented.

The last step consists in the computation of the scores. For each sample x'_i the associated counts are accumulated till their sum goes over the value ϱs and the associated value of k is employed to obtain the score.

The temporal cost of the technique is $O(s \cdot n \cdot d)$, where s is independent of the number n of dataset objects and can be considered a constant, and $n \cdot d$ is the size of the input, hence *the temporal cost is linear in the size of the input.* As for the spatial cost, $O(Bs)$ space is needed for storing counters hst, $O(2s)$ for distances dst and the ordering ord, $O(\ell s)$ for storing scores, and $O(sd)$ for the buffer maintaining the sample, hence *the spatial cost is linear in the sample size.*

Before concluding, we notice that *fast*-CFOF is an embarrassingly parallel algorithm, since partition computations do not need to communicate intermediate results. Thus, it is readily suitable for multi-processor/computer system. We implemented a version for multi-core processors (using `gcc`, `OpenMP`, and the `AVX x86-64` instruction set extensions) that elaborates partitions sequentially, but employs both MIMD (cores) and SIMD (vector registers) parallelism to elaborate each single partition.

4 Experimental results

Experiments are performed on a Intel Core i7 2.40GHz CPU (having 4 cores with 8 hardware threads, and SIMD registers accommodating 8 single-precision floating-point numbers) based PC with 8GB of main memory, under the Linux operating system. As for the implementation parameters, the number B of hst bins is set to 100 and the constant c used to compute k_{up} is set to 2. We assume 0.01 as the default value for the parameters ϱ, ϵ, and δ.

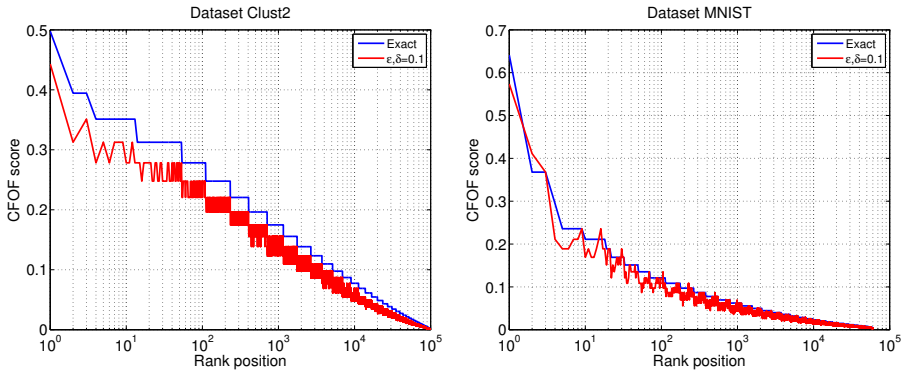
Some of the dataset employed are described next. *Clust2* is a dataset family (with $n \in [10^4, 10^6]$ and $d \in [2, 10^3]$) consisting of two normally distributed clusters centered in the origin and in $(4, \ldots, 4)$, with standard deviation 1.0 and 0.5 along each dimension, respectively. *MNIST* is a dataset consisting of handwritten digits[4] composed of $n = 60000$ vectors and $d = 784$ dimensions.

4.1 Accuracy

The goal of this experiment is to assess the quality of the result of *fast*-CFOF for different sample sizes, that is different combinations of the parameters ϵ and δ. We notice that the default sample size is $s = 26624$. With this aim we first computed the exact dataset scores by setting the sample size s to n.

Figure 5 compares the exact scores with those obtained for the standard sample size on the *Clust2* (for $n = 10^5$ and $d = 100$) and *MNIST* datsets. The blue curve is associated with the exact scores sorted in descending order and the

[4] `http://yann.lecun.com/exdb/mnist/`



(a) (b)

Fig. 5: Accuracy analysis of *fast*-CFOF.

x-axis represents the outlier rank position of the dataset objects. As for the red curve, it shows the approximate scores associated with the objects at each rank position. The curves highlight that the ranking position tends to be preserved and that in both cases top outliers are associated with the largest scores.

We can justify the accuracy of the method by noticing that the larger the CFOF score of x and, for any y, the larger the probability $p(x, y)$ that a dataset object will lie in between x and y and, moreover, the smaller the impact of the error ϵ on the estimated value $\widehat{p}(x, y)$. Intuitively, the objects we are interested in, that are the outliers, are precisely the one least prone to bad estimations.

We employ the Spearman's rank correlation coefficient to assesses relationship between the two rankings. This coefficient is high (close to 1) when observations have a similar rank. Table 1 reports Spearman's coefficients for different combinations of ϵ, δ, and ϱ. The coefficient ameliorates for increasing samples (very high values are reached for the default sample) and larger ϱ values (that exhibit high coefficient values also for small samples).

4.2 Scalability

Figure 6 shows the execution time on the *Clust2* and *MNIST* datasets.

Figure 6a shows the execution time on *Clust2* for the default sample size, $n \in [10^4, 10^6]$ and $d \in [2, 10^3]$. The largest dataset considered ($n = 10^6$ and $d = 10^3$, occupying 4GB of disk space) required about 44 minutes. *fast*-CFOF exhibits a sub-linear dependence from the dimensionality, due to the exploitation of the SIMD parallelism. As for the dashed curves, they are obtained by disabling MIMD parallelism. The performance ratio between the two versions is about 7.6, thus confirming the effectiveness of the parallelization schema.

Figure 6b shows the execution time on *Clust2* ($n = 10^6$, $d = 10^3$) and *MNIST* (180MB of disk space) for different sample sizes. As for *Clust2*, the execution time drops from 44 minutes, for the default sample, to about 24 minutes, for

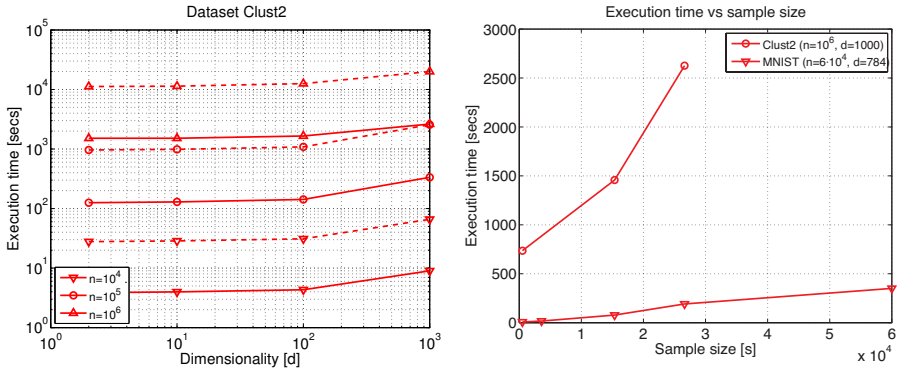
Clust2 ($n = 100000$, $d = 100$)

ϵ	δ	s	ϱ_1 0.001	ϱ_2 0.005	ϱ_3 0.01	ϱ_4 0.05	ϱ_5 0.1
0.1	0.1	512	—	0.874	0.943	0.981	0.986
0.025	0.025	3584	0.933	0.985	0.991	0.996	0.996
0.01	0.1	15360	0.988	0.996	0.997	0.998	0.997
0.01	0.01	26624	0.994	0.998	0.998	0.998	0.997

MNIST ($n = 60000$, $d = 784$)

ϵ	δ	s	ϱ_1 0.001	ϱ_2 0.005	ϱ_3 0.01	ϱ_4 0.05	ϱ_5 0.1
0.1	0.1	512	—	0.526	0.679	0.886	0.939
0.025	0.025	3584	0.683	0.899	0.938	0.979	0.988
0.01	0.1	15360	0.929	0.978	0.985	0.993	0.995
0.01	0.01	26624	0.965	0.989	0.992	0.996	0.997

Table 1: Spearman correlation between the exact and approximate outlier rankings computed by *fast*-CFOF.



(a) (b)

Fig. 6: Scalability analysis of *fast*-CFOF.

$s = 15360$ ($\epsilon = 0.01$, $\delta = 0.1$). Finally, as for *MNIST*, the whole dataset ($s = n$) required less than 6 minutes, while about 3 minutes are required with the default sample.

4.3 Effectiveness

On *Clust2*, we used the distance to cluster centers as the ground truth. Specifically, for each dataset object, the distance R from the closest cluster center has been determined and distances associated with the same cluster have been normalized as $R' = \frac{R-\mu_R}{\sigma_R}$. Table 2 reports the Spearman's correlation between normalized distances R' and CFOF scores. The high correlation values witness for both the meaningfulness of the definition and its behavior as a local outlier measure even in high dimensions.

Clust2 ($n = 100000$, $d = 100$)

ϵ	δ	s	ϱ_1 0.001	ϱ_2 0.005	ϱ_3 0.01	ϱ_4 0.05	ϱ_5 0.1
0.1	0.1	512	—	0.874	0.943	0.981	0.987
0.025	0.025	3584	0.932	0.985	0.992	0.997	0.998
0.01	0.1	15360	0.987	0.997	0.998	0.999	0.999
0.01	0.01	26624	0.993	0.998	0.999	0.999	0.999

Table 2: Spearman correlation between the normalized distance to the object's cluster center and the score computed by *fast*-CFOF.

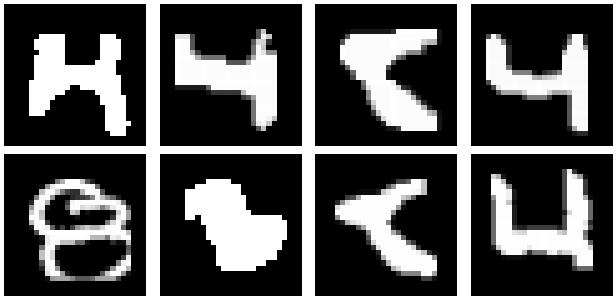


Fig. 7: Top CFOF outliers of *MNIST*.

Figure 7 shows the height top outliers of *MNIST*. It appears that these digits are deformed, quite difficult to recognize, and possibly misaligned within the 28×28 cell grid.

4.4 Comparison with other approaches

We compared CFOF with *a*KNN, LOF, and ABOD, by using some labelled datasets as ground truth. The datasets, randomly selected at the UCI ML Repository[5], are: *Breast Cancer Wisconsin Diagnostic* ($n = 569$, $d = 32$), *Image segmentation* ($n = 2310$, $d = 19$), *Ozone Level Detection* ($n = 2536$, $d = 73$), *Pima indians diabetes* ($n = 768$, $d = 8$), *QSAR biodegradation* ($n = 1055$, $d = 41$), *Yeast* ($n = 1484$, $d = 8$). Each class in turn is marked as abnormal, and a dataset composed of all the objects of the other classes plus 10 randomly selected objects of the abnormal class is considered. Table 3 reports the Area Under the ROC Curve (AUC) obtained by CFOF (*hard*-CFOF has been used), *a*KNN, LOF, and ABOD. As for the parameters k_ϱ and k, for all the methods the corresponding parameter has been varied between 2 and 100, and the best result has been reported in the table. Notice that the wins are 16 for CFOF, 4 for *a*KNN, 2 for LOF, and 4 for ABOD. The comparison points out that CFOF represents an outlier detection definition with its own peculiarities, since the other methods behaved differently, and state of the art detection performances.

[5] `https://archive.ics.uci.edu/ml/index.html`

Dataset	*Class*	CFOF	*a*KNN	LOF	ABOD
Breast	0	0.929	*0.936*	**0.952**	0.914
	1	**0.805**	0.685	*0.780*	0.404
Image	1	**0.942**	0.812	*0.846*	0.649
	2	**0.990**	0.988	0.987	*0.989*
	3	**0.956**	0.817	*0.919*	0.713
	4	0.936	**0.971**	*0.949*	0.941
	5	**0.933**	0.688	*0.884*	0.688
	6	0.979	*0.979*	0.968	**0.982**
	7	**0.993**	0.973	*0.982*	0.976
Ozone	0	**0.728**	0.677	0.662	*0.680*
	1	**0.656**	0.429	*0.591*	0.426
Pima	0	**0.736**	0.509	*0.626*	0.454
	1	*0.677*	**0.700**	0.670	0.626
QSAR	0	**0.692**	0.503	0.444	*0.545*
	1	**0.818**	0.706	0.706	*0.757*
Yeast	0	**0.769**	0.526	*0.568*	0.487
	1	**0.743**	0.678	*0.729*	0.629
	2	**0.788**	0.327	*0.437*	0.313
	3	0.772	**0.832**	0.700	*0.820*
	4	0.721	**0.808**	0.695	*0.803*
	5	**0.735**	*0.728*	*0.728*	**0.735**
	6	*0.766*	0.613	**0.783**	0.636
	7	**0.794**	0.550	*0.587*	0.543
	8	0.814	*0.881*	0.850	**0.892**
	9	0.980	0.993	*0.981*	**1.000**

Table 3: AUCs for the labelled datasets.

5 Conclusions

We presented the Concentration Free Outlier Factor, a novel density estimation measure whose main characteristic is to resist to concentration phenomena usually arising in high dimensional spaces and to allow very efficient and reliable outlier detection through the use of sampling. We are extending the study of the theoretical properties of the definition, assessing guarantees of the *fast*-CFOF algorithm, and extending the experimental activity. We believe that the CFOF score can offer insights also in the context of other data mining tasks. We are currently investigating its application in other classification scenarios.

References

1. C. C. Aggarwal and P.S. Yu. Outlier detection for high dimensional data. In *Proc. Int. Conf. on Managment of Data (SIGMOD)*, 2001.
2. C.C. Aggarwal. *Outlier Analysis*. Springer, 2013.
3. F. Angiulli. On the behavior of intrinsically high dimensional spaces: distance distributions, neighborhood stability and hubness. *Manuscript submitted for publication to an international journal*, 2017. Available at the author's site.
4. F. Angiulli and F. Fassetti. Dolphin: an efficient algorithm for mining distance-based outliers in very large datasets. *ACM Trans. Knowl. Disc. Data*, 3(1):Article 4, 2009.
5. F. Angiulli and C. Pizzuti. Outlier mining in large high-dimensional data sets. *IEEE Trans. Knowl. Data Eng.*, 2(17):203–215, February 2005.

6. K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is "nearest neighbor" meaningful? In *Proc. Int. Conf. on Database Theory*, pages 217–235, 1999.
7. M. M. Breunig, H. Kriegel, R.T. Ng, and J. Sander. Lof: Identifying density-based local outliers. In *Proc. Int. Conf. on Managment of Data (SIGMOD)*, 2000.
8. V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3), 2009.
9. V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection for discrete sequences: A survey. *IEEE Trans. Knowl. Data Eng.*, 24(5):823–839, 2012.
10. E. Chávez, G. Navarro, R. Baeza-Yates, and J.L. Marroquín. Searching in metric spaces. *ACM Comput. Surv.*, 33(3):273–321, 2001.
11. J. Han and M. Kamber. *Data Mining, Concepts and Technique*. Morgan Kaufmann, San Francisco, 2001.
12. V. Hautamaki, I. Karkkainen, and P. Franti. Outlier detection using k-nearest neighbour graph. In *Proc. Int. Conf. on Pattern Recognition (ICPR)*, pages 430–433, 2004.
13. V. Hodge and J. Austin. A survey of outlier detection methodologies. *Artif. Intell. Rev.*, 22(2):85–126, 2004.
14. H.P.Kriegel, P.Kroger, E.Schubert, and A.Zimek. Outlier detection in axis-parallel subspaces of high dimensional data. In *Proc. Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, pages 831–838, 2009.
15. W. Jin, A.K.H. Tung, and J. Han. Mining top-n local outliers in large databases. In *Proc. Int. Conf. on Knowledge Discovery and Data Mining (KDD)*, 2001.
16. E. Knorr and R. Ng. Algorithms for mining distance-based outliers in large datasets. In *Proc. Int. Conf. on Very Large Databases (VLDB)*, pages 392–403, 1998.
17. H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek. Interpreting and unifying outlier scores. In *Proc. SIAM Int. Conf. on Data Mining (SDM)*, pages 13–24, 2011.
18. H.-P. Kriegel, M. Schubert, and A. Zimek. Angle-based outlier detection in high-dimensional data. In *Proc. Int. Conf. on Knowledge Discovery and Data Mining (KDD)*, pages 444–452, 2008.
19. F.T. Liu, K.M. Ting, and Z.-H. Zhou. Isolation-based anomaly detection. *TKDD*, 6(1), 2012.
20. S. Papadimitriou, H. Kitagawa, P.B. Gibbons, and C. Faloutsos. Loci: Fast outlier detection using the local correlation integral. In *Proc. Int. Conf. on Data Enginnering (ICDE)*, pages 315–326, 2003.
21. M. Radovanovic, A. Nanopoulos, and M. Ivanovic. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, 11:2487–2531, 2010.
22. M. Radovanovic, A. Nanopoulos, and M. Ivanovic. Reverse nearest neighbors in unsupervised distance-based outlier detection. *IEEE Trans. Knowl. Data Eng.*, 27(5):1369–1382, 2015.
23. S. Ramaswamy, R. Rastogi, and K. Shim. Efficient algorithms for mining outliers from large data sets. In *Proc. Int. Conf. on Management of Data (SIGMOD)*, pages 427–438, 2000.
24. O. Watanabe. Sequential sampling techniques for algorithmic learning theory. *Theor. Comput. Sci.*, 348(1):3–14, 2005.
25. A. Zimek, E. Schubert, and H.-P. Kriegel. A survey on unsupervised outlier detection in high-dimensional numerical data. *Stat. Anal. Data Min.*, 5(5):363–387, 2012.