Synchronization Based Outlier Detection

Junming Shao[1], Christian Böhm[1], Qinli Yang[2], and Claudia Plant[3]

[1] Institute of Computer Science, University of Munich, Germany
[2] School of Engineering, University of Edinburgh, UK
[3] Department of Scientific Computing, Florida State University, USA

Abstract. The study of extraordinary observations is of great interest in a large variety of applications, such as criminal activities detection, athlete performance analysis, and rare events or exceptions identification. The question is: how can we naturally flag these outliers in a real complex data set? In this paper, we study outlier detection based on a novel powerful concept: synchronization. The basic idea is to regard each data object as a phase oscillator and simulate its dynamical behavior over time according to an extensive Kuramoto model. During the process towards synchronization, regular objects and outliers exhibit different interaction patterns. Outlier objects are naturally detected by local synchronization factor (LSF). An extensive experimental evaluation on synthetic and real world data demonstrates the benefits of our method.

Keywords: Outlier Detection, Synchronization, Kuramoto model.

1 Introduction

> "*An outlying observation, or outlier, is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism.*" [1]

Such irregular observations often contain useful information on abnormal behavior of the system described by the data. The detection of these irregular data is thus equally or even more interesting and useful than finding regular patterns applicable to a considerable portion of objects in a data set. For example, the identification of criminal activities, such as credit card fraud, is crucial in electronic commerce applications. The detection of potential outstanding players is critical for athlete performance analysis and management. The wide range of applications also include clinical trials, voting irregularity analysis, data cleansing, network intrusion, gene expression analysis, severe weather prediction, geographic information systems, and may more.

Currently, outlier detection has attracted increasing attention and many algorithms have been proposed (e.g. [2] [3] [4] [5]). However, they suffer from one or more of the following drawbacks: They explicitly or implicitly assume the data to follow a given distribution model, such as Gaussian, uniform or Exponential Power Distribution. The results of many methods strongly depend on suitable parametrization and/or their results are difficult to interpret. In addition, most

J.L. Balcázar et al. (Eds.): ECML PKDD 2010, Part III, LNAI 6323, pp. 245–260, 2010.

approaches are restricted to a flat data structure and do not support complex hierarchical data. A more detailed discussion on these studies will be given in Section 2.

In this paper, we consider outlier detection from a novel different point of view: *synchronization*. Synchronization is the phenomenon that a group of events spontaneously comes into co-occurrence with a common rhythm, despite of the differences between individual rhythms of the events. It is a powerful concept in nature regulating a large variety of complex processes ranging from the metabolism in the cell to social behavior in groups [6]. For example, the effect of synchrony has been described in experiments of people conversation, song or rhythm, or of groups of children interacting to an unconscious beat. In all cases the purpose of the common wave length or rhythm is to strengthen the group bond. The members lacking of synchrony are called "out of synchronization".

To illustrate synchronization, consider for example opinion formation. In the beginning, each person usually has their own view about the problem. After mutual influence by conversation or discussion, people with similar educational background, age span, hobby, career or experience will easily talk together and finally form a common opinion (synchronization). Over time groups with different opinions emerge. For some people (outliers) with significantly different educational background, experience or other characteristics, it is not easy to join any group for discussion. Therefore, they tend to isolate from other people and keep their own opinions over time (out of synchronization). Inspired by such natural synchronization phenomena, we propose a novel technique for outlier detection. Our approach robustly identifies outliers based on their completely different behaviors in comparison to regular objects during the process towards synchronization.

The remainder of this paper is organized as follows: in the following section, we briefly survey the related work. Section 3 presents our algorithm in detail. Section 4 contains an extensive experimental evaluation and Section 5 concludes the paper.

2 Related Work

Currently, most existing approaches to outlier detection can be mainly classified into three categories: distribution-, distance-, and density-based outlier detection. In addition, a brief survey of the application of Kuramoto Model and synchronization is given.

Distribution-based Outlier Detection. Methods in this category are mainly developed in the field of statistics. Generally, they assume a known distribution model (Gaussian, Poisson, Exponential Power Distribution, etc.) for the observations based on statistical analysis of a given data set. Outliers are defined as those objects that deviate considerably from the model assumptions [7] [1] [8]. In [7], numerous discordancy tests are discussed for different scenarios. In [9] [10], authors propose SmartSifter (SS), which is an on-line real-time outlier detection

algorithm. The basic principle of SS is to use a probabilistic model (a finite mixture model) to represent the underlying distribution of a given data set. Each time a datum is input and SS employs an on-line learning algorithm to adjust the probability model. An anomaly score is calculated for each datum based on the learned model. However, the method relies on histograms and requires preparing as many Gaussian mixture models as cells in the histogram density. Moreover, in real-world applications, it is not trivial to find an appropriate model to fit an arbitrary data distribution without prior knowledge. Recently, CoCo, an information-theoretic outlier detection approach has been proposed by Böhm, et al. [5]. Based on the MDL principle, outliers are flagged as those objects which need more coding cost than regular objects. For coding each object, the optimal neighborhood size is heuristically determined. Independent Component Analysis and Exponential power distribution (EPD) are used to estimate the probability and the corresponding coding cost. Like most distribution-based methods, CoCo tends to fail if the estimated distribution does not fit the data model well. It is also time consuming to find the optimal neighborhood to estimate the coding cost for each object by screening for suitable neighborhood sizes.

Distance-based Outlier Detection. The concept of distance-based outlier detection is proposed by E.M. Knorr and R.T. Ng [3] [4]. These techniques identify potential outliers from ordinary points based on the number of points in the specified neighborhood. It defines a point in a data set T to be an outlier if at least p fraction of points in T have greater distance than d from it. The basic notion is extended in [11] by computing the distances to the k nearest neighbors and then ranking the objects based on their proximity to their k-th nearest neighbors. Consequently top n outliers are obtained using a partition-based algorithm. However, it is difficult to accurately determine the parameters p and d for a arbitrary data set.

Density-based Outlier Detection. M. Breunig, et al. [2], introduce a notion of local outlier from a density-based perspective. An object is regarded as an outlier if its local density does not fit well into the density of its neighboring objects. The local outlier factor (LOF) is then proposed to capture the degree to which the object is an outlier. It is defined as the average of the ratio of the local reachability density of the object and those of the objects in its neighborhood. A LOF value of approximately 1 indicates the object is located inside a cluster, while the objects with higher LOF values are more rather considered as outliers. In [12], a connectivity-based outlier factor (COF) scheme is proposed to improve the effectiveness of LOF scheme when a pattern itself has a similar neighborhood density as an outlier. Although these approaches to outlier detection are useful, their performances are sensitive to the parameter $Minpts$ which can be very difficult to determine. The Local Outlier Integral (LOCI) [13] flags outliers, based on probabilistic reasoning and motivated from the concept of a multi-granularity deviation factor (MDEF). Similar to LOF, the LOCI outlier model takes the local object density into account, but differently, the MDEF of LOCI uses ϵ-neighborhoods rather than $MinPts$ nearest neighbors. The local

neighborhood in LOCI model is defined by two parameters: the counting and the sampling neighborhood. The counting neighborhood specifies some volume of the feature space which is used to estimate the local object density. The sampling neighborhood is larger than the counting neighborhood and contains all points which are used to compute the average object density in the neighborhood. Objects which deviate in their local object density more than three times of the standard deviation are regarded as outliers. The flagging scheme of LOCI thus assumes the object densities follow a Gaussian distribution.

Kuramoto Model and Synchronization. Currently, the study of synchronization phenomena have widely used in physical, biological, chemical, and social systems. The Kuramoto model [14] [15] is one the most famous models to explore collective synchronization. Arenas et al. [16] apply the Kuramoto model for network analysis, and study the relationship between topological scales and dynamic time scales in complex networks. This analysis provides a useful connection between synchronization dynamics, network topology and spectral graph analysis. Recently, the Kuramoto model has attracted some attention in clustering [17] [18]. Aeyels et. al [19] introduce a mathematical model for the dynamics of chaos system. They characterize the data structure by a set of inequalities in the parameters of the model and apply it to a system of interconnected water basins. In summary, previous approaches mainly focus on the synchronization phenomena of a dynamic system from a global perspective. Inspired by ideas from the synchronization phenomena and existing dynamical system analysis, we propose a novel outlier detection technique based on synchronization.

3 Synchronization-Based Outlier Detection

In this section, we introduce SOD, to detect outliers based on synchronization principle. We first illustrate the basic idea and then propose an extensive Kuramoto Model for outlier detection. In Section 3.3 we discuss the algorithm SOD and its properties in detail.

3.1 Basic Idea

The concept of synchronization provides a natural way to outlier detection. The basic idea is to flag outliers by distinguishing the object dynamical behaviors during the process towards synchronization. In our work, each data object is regarded as a phase oscillator and interacts dynamically with other objects according to an Extensive Kuramoto model (EKM), which we will introduce in Section 3.2.

To give an intuition of the synchronization-based outlier detection, let's consider a simple data set as illustrated in Figure 1. For an object within a cluster, such as $P1$, many similar objects are located around it and P1 starts interacting with these similar objects. Through non-linear dynamical interaction, the object changes its initial phase and moves towards the main direction of its interaction

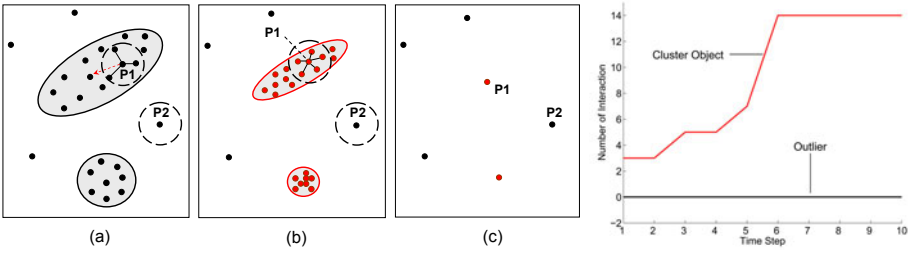


Fig. 1. Illustration of synchronization-based outlier detection. (a)-(c): The dynamics of objects towards synchronization. (d): Interaction Plot.

partners (Figure 1(a)). The situation is like the mutual influence of people in discussion. As time evolves, regular objects move gradually closer together through mutual interaction and thus more and more objects can interact with them. Figure 1(b) displays the new positions of the objects for comparison. The objects with similar attributes gradually synchronize together. The initial points (black color) are replaced by the red points after one time stamp. Then, in a sequential process, all these similar objects synchronize together, which finally have the same phase (Figure 1(c)). Outliers, such as $P2$, due to the significantly different attributes in comparison with regular objects, have difficulties to interact with other objects and tend to keep their own phases. Therefore, the dynamics of the objects show two different patterns during the process towards synchronization. For each regular object, as time evolves, it interacts with more and more objects and finally synchronized together with other objects. For outliers, there is none or only very minor interaction. Strong outliers keep their unique phase over the whole time during the process towards synchronization. The two different patterns can be easily visualized: Figure 1(d) shows the *Interaction Plot*, which displays for each object the number of interactions on the time scale.

3.2 Extensive Kuramoto Model

One of the most successful attempts to understand collective synchronization phenomena is due to Kuramoto [14] [15], who analyzes a model of phase oscillators which are coupled through the sine of their phase differences. The *Kuramoto model* (KM) consists of a population of N coupled phase oscillators where the phase of the *i-th* unit, denoted by θ_i, evolves in time according to the following dynamics:

$$\frac{d\theta_i}{dt} = \omega_i + \frac{K}{N}\sum_{j=1}^{N}\sin(\theta_j - \theta_i), (i = 1, \ldots, N), \tag{1}$$

where ω_i stands for its natural frequency and K describes the coupling strength between units. $\frac{d\theta_i}{dt}$ denotes the instantaneous frequency. θ_i describes the phase of i-th unit.

The Kuramoto model well describes the global synchronization behavior of all coupled phase oscillators. In real-life, this situation rarely occurs. Partial

synchronization is observed more frequently, which is the case when a local ensemble of oscillators are synchronized together. It is also observed that the sets of oscillators with high similarity synchronize more easily than those with large variance. Therefore, partial synchronization provides rich information about the object behaviors. In order to explore the different dynamic behaviors between regular objects and outliers, we extensively reformulate $Eq.(1)$. Formally, we first need to define the notion of ϵ-neighborhood.

Definition 1: (ϵ-neighborhood of an object x) The ϵ- neighborhood of object x, which denoted by $Nb_\epsilon(x)$, is defined as:

$$Nb_\epsilon(x) = \{y \in \mathcal{D} | dist(y, x) \leq \epsilon\}, \tag{2}$$

where $dist(y, x)$ is metric distance function.

Definition 2: (Extensive Kuramoto model) Let $x \in \mathbb{R}^d$ be an object in the data set $\mathcal{D}$ and x_i be the i-th dimension of the data object x respectively. We regard each object x as a phase oscillator, according to Eq.(1), with a ϵ-neighborhood interaction. The dynamics of each dimension x_i of the object x is governed by:

$$\frac{dx_i}{dt} = \omega_i + \frac{K}{|Nb_\epsilon(x)|} \sum_{y \in Nb_\epsilon(x)} \sin(y_i - x_i). \tag{3}$$

Let $dt = \Delta t$, then:

$$x_i(t+1) = x_i(t) + \Delta t \cdot \omega_i + \frac{\Delta t \cdot K}{|Nb_\epsilon(x(t))|} \cdot \sum_{y \in Nb_\epsilon(x(t))} \sin(y_i(t) - x_i(t)). \tag{4}$$

For unsupervised outlier detection we assume that all objects having the same frequency w, since we have no external knowledge on the data. Thus the term $\Delta t \cdot \omega_i$ is the same for each object and can be ignored. Similarly, $\Delta t \cdot K$ is a constant which we set to 1. Finally the dynamics of each dimension x_i of the object x over time is provided by:

$$x_i(t+1) = x_i(t) + \frac{1}{|Nb_\epsilon(x(t))|} \cdot \sum_{y \in Nb_\epsilon(x(t))} \sin(y_i(t) - x_i(t)). \tag{5}$$

The object x at time step $t = 0$: $x(0)$ $(x_1(0), ..., x_d(0))$ represents the initial phase of the object (the original location of object x). The $x_i(t+1)$ describes the renewal phase value of i-th dimension of object x at the $t = (0, \ldots, T)$ time evolution.

To characterize the level of synchronization between oscillators during the process, an order parameter needs be defined. Instead of considering a global observable, we define a local order parameter r, measuring the coherence of local oscillator population.

Definition 3: (Local Order Parameter) The local order parameter r characterizing the degree of local synchronization is provided by:

$$r = \frac{1}{N}\sum_{i=1}^{N}\Big(\sum_{y\in Nb_\epsilon(x)} e^{-||y-x||}\Big|x \in \mathcal{D}\Big). \tag{6}$$

The value of r increases as more neighbors synchronize together with time evolution. The process toward synchronization will terminate when r converges, which indicates local similar objects achieve phase coherence. At this moment, all local similar objects have the same phase (location).

3.3 The SOD Algorithm

In this section, we elaborate the *SOD* algorithm based on our extensive Kuramoto model.

First, without any interaction, all objects in a data set have their own phases. As time evolves, each object starts to interact with its ϵ-neighborhood, cf. Definition 1. The traces of all objects are in line with the main direction of their neighborhoods. Gradually, regular objects with similar attributes synchronize together following the intrinsic structure of a data set. In contrast, outliers are difficult to interact with other objects due to the large variance. Finally, the local regular objects with similar attributes synchronize together with same phase while outliers tend to keep their original phases. The objects synchronization process is terminated when the local order parameter converges.

To simply illustrate the objects dynamical movement, the Figure 2 (a)-(d) shows the detailed dynamics of 2-dimensional points at time steps: $t = 0, 1, 3, 5$. $t = 0$ indicates the original data set at the initial time. From that moment on, all objects with similar attributes start to synchronize together through the dynamical interaction according to Eq.(5) and finally, all objects in the data set synchronize at two different phases after 5 time steps. A more intuitive visualization of the objects movement is illustrated in Figure 2(e). Figure 2 (f) demonstrates the local order parameter of the data set with time evolution.

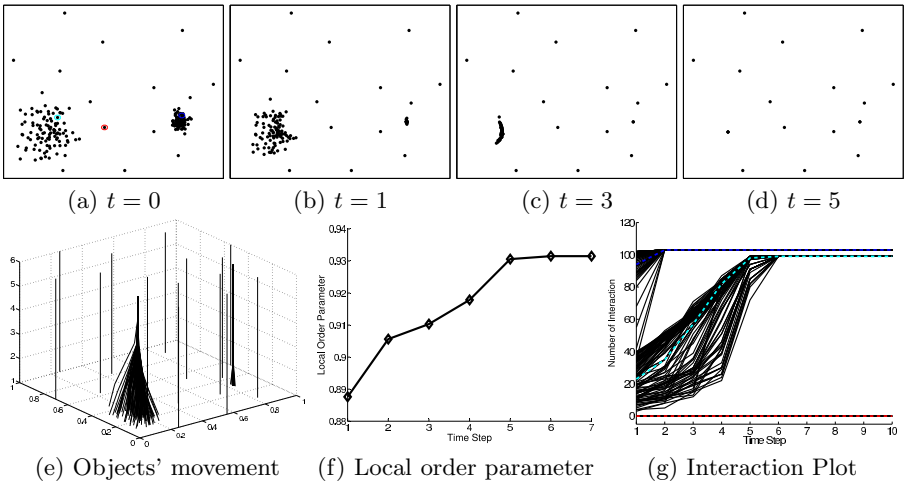
Definition 4: (Local Synchronization Factor of an object x) The local synchronization factor $LSF(x)$ of object x is defined as:

$$LSF(x) = \frac{1}{T}\sum_{t=0}^{T}\Big(\frac{1}{|Nb_\epsilon(x(t))|}\sum_{y(t)\in Nb_\epsilon(x(t))} cos(||y(t)-x(t)||)\Big). \tag{7}$$

where T is the whole time steps for the process of synchronization. The synchronization factor captures the degree to which the object of being an outlier. The smaller the LSF value, the higher the probability of being an outlier.

According to the definition, LSF shows major desirable properties:

1. *Intuitive.* Since the LSF value indicates each object synchronization factor, it provides an intuitive way to summarize its dynamical interaction behavior



(a) $t = 0$ (b) $t = 1$ (c) $t = 3$ (d) $t = 5$

(e) Objects' movement (f) Local order parameter (g) Interaction Plot

Fig. 2. The dynamics of objects toward to synchronization. (a)-(d): The detailed objects' movement during the time resolution. (e): Objects movement. (f): Local order parameter, (g): Interaction Plot for three objects circled in (a).

with other objects during the process towards synchronization. The easier an object synchronizes with other objects, the higher is its LSF value. Outliers are objects which are out of synchronization.

2. *Tightness.* The range of LSF is restricted to [0 1). The lower bound of LSF value is 0, which means that the object does not interact with any other object during the synchronization process. For cluster points which easily synchronize the LSF value is close to 1. The LSF value can thus be easily interpreted as the probability of each object of being an outlier, e.g. Probability$(p) = 1 - LSF(p)$.
3. *Distinguishable.* Due to the different dynamics between regular objects and outliers during the synchronization process, the values of LSF are fairly distinguishable. For outliers, the LSF value is around 0 while the regular objects nearly to 1. It can easily discern them.

In addition, in order to explore each object dynamics during the process towards synchronization, the *Interaction Plot* is defined to characterize the interaction behavior pattern between objects over time.

Definition 5: (*Interaction Plot*) For any object x, the plot of the number of objects involved in mutual interaction with x versus the time step, is called *Interaction Plot*.

As a valuable addition to LSF, the *Interaction Plot* provides a detailed visualization of the dynamic behavior of each object according to the Extensive Kuramoto model. With the same data set above , the interaction plot is illustrated in Figure 2(g). From this plot, two distinct interaction patterns become evident. For regular objects, during the synchronization process, more and more

objects interact together along with the time steps. Outliers often fail to interact with other objects (maybe a few at the beginning). As time evolves, the number of objects for interaction tend to keep the same. For example, two regular objects and one outlier are visualized with dash color lines to illustrate the different interaction patterns in Figure 2 (a),(g).

Outliers Flagging. After LSF is obtained for each object, all outliers exhibit usually low values in comparison to the regular objects. The denser of the local region of an object, the higher its value of LSF. Therefore, selecting a suitable threshold for flagging outliers could be easily selected since the LSF value is distinct for outliers and regular objects. However, for automatically flagging, in this work, the K-Means algorithm are applied on the LSF values to split the data into two clusters: outliers and regular objects. Finally, the Pseudocode of the *SOD* is illustrated in Algorithm 1.

Algorithm 1. $SOD(D, \epsilon)$

```
LSF := {}; // Synchronization Factors
while loopFlag=true do
  for each object p ∈ D do
    Compute Nb_ε(p);
    Obtain new value of object p using Eq.(5); // Update value
  end for
  Compute local order r using Eq.(6);
  if r converges then
    loopFlag=false;
    for each object p ∈ D do
      Compute local synchronization factor LSF(p) using Eq.(7);
    end for
  end if
end while

Flagging outliers by K-Means based on LSF values.
```

Runtime Complexity. For SOD, to detect outliers based on synchronization, the runtime complexity with respect to the number of data objects is $O(T \cdot N^2)$, where N is the number of objects and T is the time evolution. In most cases, T is small with $5 \leq T \leq 20$. If there exists an efficient index, the complexity reduces to $O(T \cdot N \log N)$.

Parameter Setting. In order to flag outliers based on synchronization principle, an interaction range (ϵ) needs to be specified for EKM. The question is: how to determine the ϵ value and how does the LSF value change when the ϵ value is adjusted?

Given a data set, theoretically, the ϵ value can be 0, which means there is no interaction at all among the objects. In order to generate object interaction, there should be a lower bound of ϵ. To generate a stable interaction for most

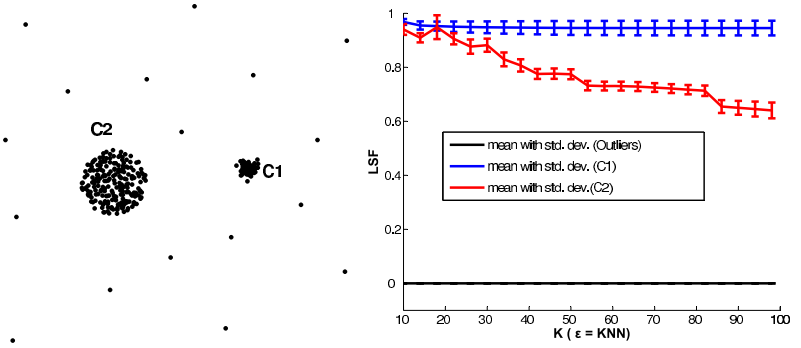


Fig. 3. Influence of ϵ on LSF

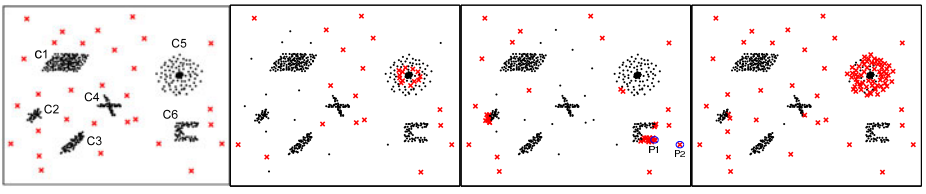
objects, a heuristic way is to use the average value of the k-nearest neighbor distance determined by a sample from the data set for a small k. The ϵ should not be very large that enclose all data objects for interaction. In that way, we can not detect the local object dynamical behaviors. However, the range of suitable values for ϵ is very large. In all experiments on different data sets, we set ϵ the the average value of the k-nearest neighbor distances for k ranging from 10 to 50.

To asses the impact of ϵ value on LSF value and outlier detection, Figure 3 shows a simple data set which consists of 2 clusters with different size (C1:50, C2:200) and density. 17 outliers are added to the data. For each ϵ value, the mean and standard deviation of LSF values are calculated for each cluster as well as for the outliers. Figure 3 displays the LSF value with respect to ϵ. For all settings of ϵ, the mean LSF value for the points in cluster C1 and C2 are clearly much larger than 0, in most cases close to 1 while the mean and standard deviation of LSF value for outliers remain stable at 0. With the increase of ϵ, the LSF value of cluster objects begin to decrease. The reason behind it is that more and more objects are enclosed to interact with each other at each time step and thus more difficult to synchronize together. The situation is like two persons are much easier to agree with one thing than a larger group of people. Moreover, since objects in denser cluster are easier to synchronize, the LSF values are much more closer to 1 (e.g. the mean LSF value of objects in C1 are larger than that in C2). For different parameters, outliers and regular objects show distinct LSF values and can be discerned easily. For further evaluation on the robustness of SOD w.r.t. parameter settings please refer to the experimental section.

4 Experimental Evaluation

In the following we evaluate our outlier detection *SOD* in comparison to LOF [2], LOCI [13], CoCo [5] on synthetic data set as well as NBA data. We implemented SOD and LOF in Java and obtained the implementation of LOCI and CoCo from the authors.

Synthetic Data. We start the evaluation with two-dimensional synthetic data sets to facilitate presentation. The data set displayed in Figure 4 consists of six clusters (C1-C6): one Gaussian cluster (C2:39), two correlation clusters (C1:167 and C3:73), two arbitrarily shaped clusters (C4:60 and C6:67) and one a spherical hierarchical cluster (C5:131), including a nested cluster. 30 outliers are added to the data set. Figure 4 provides the results of outlier detection by using SOD, LOF, LOCI and CoCo for the same synthetic data set.



(a) SOD (b) LOF (c) LOCI (d) CoCo

Fig. 4. Outlier detection results with different methods. (a): SOD ($k = 10 - 40$), (b): LOF (MinPts = 20, selecting only the top 30 outliers), (c): LOCI (α= 1, $rmin$ = 10 and $rmax$ =50), (d): CoCo. Detected outliers are highlighted with red crosses.

Without any prior knowledge, SOD successfully detects all 30 outlier points (Figure 4 (a)) from the complex data. The outliers are highlighted with red crosses and cluster objects are shown in black. Moreover, SOD obtains the same result with the parameter ϵ set to the average k-nearest neighbor distance for k ranging from 10 to 40.

For LOF, we try a wide range of different settings for the parameters $MinPts$ from 10 to 50, which is suggested by authors. The top 30 outliers are obtained according to the LOF value. For different parameters, there are 17, 17, 9, 8 and 10 out of 30 correctly assigned with $MinPts = \{10, 20, 30, 40, 50\}$, respectively. Most cluster objects, especially for the objects of hierarchical cluster are wrongly flagged as outliers. The best result is presented in Figure 4 (b) obtained with parameter $MinPts$ =20. Obviously, the result of LOF is very much influenced by the parameter $MinPts$. In addition, we often have no a priori information about the number of desired outliers.

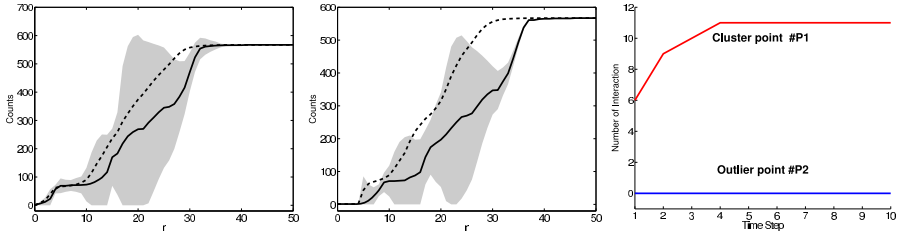


Fig. 5. LOCI Plot of cluster point P1 (left) and outlier P2(middle); Interaction Plot of P1 and P2 (right)

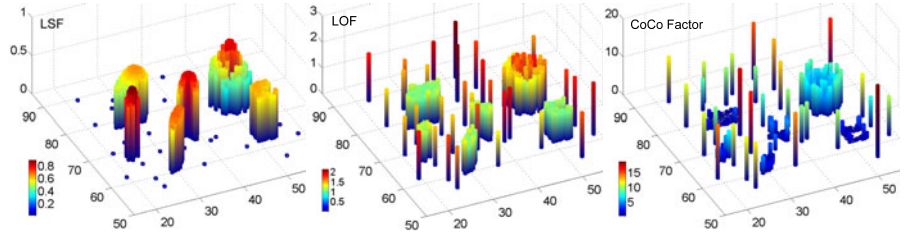


Fig. 6. Visualization of the LSF, LOF and CoCo for the synthetic data set

LOCI is applied to our synthetic data set with $\alpha = 1$, $rmin = 10$ and $rmax = 50$ (Figure 4 (c)). 46 outlier points are detected based on the suggested outlier flagging criteria and 16 true outliers are correctly detected. Many cluster points of C2, C4 and C6 are wrongly flagged. With the decrease of $rmin$ value, more true outlier points are found, but at the same time more cluster points are mislabeled as outliers. As the LOCI plot provides the information for each object, we thus have a closer look at the cluster point (Fig. 4 (c): $P1$) and outlier point ($P2$). It is noticeably that the two LOCI plots look very similar (Figure 5(a)-(b)). For comparison, *Interaction Plot* is displayed for the same two points (Figure 5(c)). It is very clear that the cluster point $P1$ and outlier point $P2$ have totally different characteristics.

CoCo identifies all 95 outliers for the synthetic data and only one outlier is missing. However, 66 cluster points are wrongly flagged as outliers. CoCo relies on the assumption that the data follows an Exponential Power Distribution, which is not the case for our example and therefore CoCo yields many false-positives.

To visualize the degree of "outlierness" for each object, the Local Synchronization Factor (LSF) is further compared to the outlier factor of LOF and CoCo in Figure 6. For LSF, the local synchronization factor of all outlier points are nearly 0 and all cluster points are close to 1. Due to the desirable properties of LSF, such as tightness and distinguishable, cluster points and outliers are easily to differentiate. As to LOF and CoCo, the range of values is very wide and the gap between outliers and cluster points is not clear. For example, the range of LOF is from 0.85 to 2.15, which makes it difficult to determine a suitable threshold for outlier flagging.

NBA Performance Statistics. After extensive evaluation of SOD on synthetic data sets, we apply our novel outlier detection method to the real data. We use the NBA data available at the NBA website `http://www.nba.com`. In the Season 2008/09, the performance of 444 players are described with four attributes: the number of games played (GP), the number of points (PPG), the rebounds (RPG), and assists (APG) per game.SOD is applied to this NBA data detecting 18 outliers with $k = 30$. Figure 7 displays scatter plots of the data with different attributes and the histogram of each attribute in the diagonal respectively. Obviously, the data distribution is non-Gaussian. All 18 outliers are

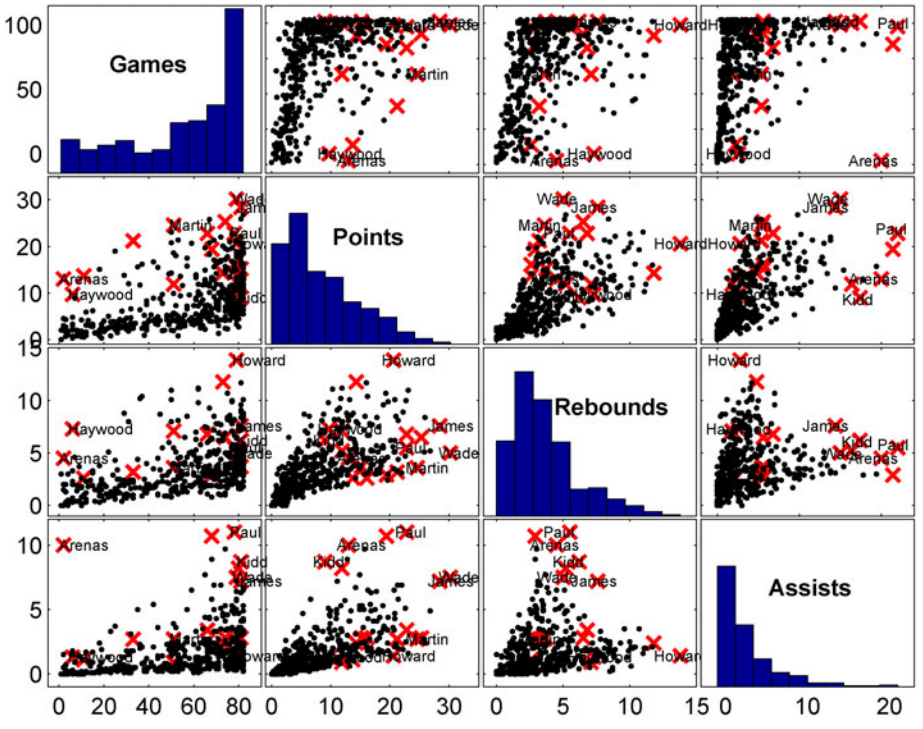


Fig. 7. Outlier detection with SOD for the NBA data set. All 18 outliers are marked in red. The strongest outliers with LSF = 0 are also marked with the name.

highlighted in red. For the most outstanding players with LSF = 0 also the names are provided. For comparison with other techniques, Table 1 lists the top 10 outliers for various settings of k. For different parametrization (k=30,40,50) the same players are among the top 10 outliers of SOD. Eight of 10 players are strongest outliers with a LSF of 0 for all parameterizations. For comparison, Table 2 lists the top 10 outliers identified by LOF with MinPts =50. Only seven players are reproducibly detected as top 10 for MinPts =40. LOCI ($\alpha = 1$, $rmin = 10$ and $rmax$ =50) and CoCo detect the top 10 outliers listed in Table 3 and Table 4, respectively. For the comparison methods, the intersection with SOD is marked in bold. Gilbert Arenas with LSF = 0 is a strong outlier which is among the top 10 for all methods. Having played only 2 games, he has shown outstanding performance in terms of rebounds, points and especially in terms of assists. Most other players with an LSF of zero are also among the top 10 of at least one of the comparison methods, e. g. the well-known and truly outstanding players Brendan Haywood, Dwyane Wade and LeBron James. Interestingly, Chris Paul is not among the top 10 of any comparison methods although he is especially outstanding in the number of points and assists per game. In the 2006-07 season he has been ranked fourth in the overall NBA in assists (`http://www.nba.com/playerfile/chris_paul/bio.html`). Another strong outlier missed by the comparison methods is Dwight

Table 1. Top 10 outliers identified by SOD on NBA data

LSF(k=30)	(k=40)	(k=50)	Name	GP	PPG	RPG	APG
0	0	0	Chris Paul (NOH)	78	22.8	5.5	11
0	0	0	Jason Kidd (DAL)	81	9	6.2	8.7
0	0	0	Dwyane Wade (MIA)	79	30.2	5	7.5
0	0	0	LeBron James (CLE)	81	28.4	7.6	7.2
0	0	0	Kevin Martin (SAC)	51	24.6	3.6	2.7
0	0	0	Dwight Howard (ORL)	79	20.6	13.8	1.4
0	0	0	Gilbert Arenas (WAS)	2	13	4.5	10
0	0	0	Brendan Haywood (WAS)	6	9.7	7.3	1.3
0	0.036	0	Cuttino Mobley (LAC)	11	13.7	2.6	1.1
0	0.260	0.035	Deron Williams (UTA)	68	19.4	2.9	10.7

Table 2. Top 10 outliers identified by LOF on NBA data

LOF	Name	GP	PPG	RPG	APG
1.5058	**Gilbert Arenas (WAS)***	2	13	4.5	10
1.4938	**Brendan Haywood (WAS)***	6	9.7	7.3	1.3
1.4463	DJ White (OKC)*	7	8.9	4.6	0.9
1.4069	Michael Redd (MIL)*	33	21.2	3.2	2.7
1.4011	**Cuttino Mobley (LAC)**	11	13.7	2.6	1.1
1.3623	Carlos Boozer (UTA)*	37	16.2	10.4	2.1
1.3532	Monta Ellis (GSW)*	25	19	4.3	3.7
1.3492	Elton Brand (PHI)*	29	13.8	8.8	1.3
1.3365	Chris Kaman (LAC)	31	12	8	1.5
1.3294	Tracy McGrady (HOU)	35	15.6	4.4	5

Table 3. Top 10 outliers identified by LOCI on NBA data

Name	GP	PPG	RPG	APG
Dwyane Wade (MIA)	79	30.2	5	7.5
LeBron James (CLE)	81	28.4	7.6	7.2
Ben Wallace (CLE)	56	2.9	6.5	0.8
Monta Ellis (GSW)	25	19	4.3	3.7
Pops Mensah-Bonsu (TOR-SAS)	22	5	5.1	0.3
Corey Brewer (MIN)	15	6.2	3.3	1.7
Gilbert Arenas (WAS)	2	13	4.5	10
Kobe Bryant (LAL)	82	26.8	5.2	4.9
Jason Kidd (DAL)	81	9	6.2	8.7
Michael Redd (MIL)	33	21.2	3.2	2.7

Howard who has been named the NBA Defensive Player of the Year in 2008-2009 season (`http://www.nba.com/playerfile/dwight_howard/bio.html`). Dwight Howard is especially characterized by an outstanding number of 13.8 rebounds per game.

Table 4. Top 10 outliers identified by CoCo on NBA data

CoCo o.f.	Name	GP	PPG	RPG	APG
20.76	Michael Redd (MIL)	33	21.2	3.2	2.7
17.39	Andrew Bogut (MIL)	36	11.7	10.2	2
17.31	**Kevin Martin (SAC)**	51	24.6	3.6	2.7
16.55	Monta Ellis (GSW)	25	19	4.3	3.7
14.42	Elton Brand (PHI)	29	13.8	8.8	1.3
13.84	**Gilbert Arenas (WAS)**	2	13	4.5	10
13.80	Tracy McGrady (HOU)	35	15.6	4.4	5
12.12	Kobe Bryant (LAL)	57	17.5	3	5
11.85	Cuttino Mobley (LAC)	11	13.7	2.6	1.1
11.45	Tyson Chandler (NOH)	45	8.8	8.7	0.5

5 Conclusions

In this paper, we propose SOD, a novel outlier detection algorithm inspired by synchronization phenomenon. The major benefits of SOD can be summarized as follows:

1. *Natural outlier detection.* Based on the synchronization principle, outliers are naturally flagged from a data set due to their unique dynamical interaction pattern during the process towards synchronization.
2. *Intuitive to interpret.* Outliers are represented as those objects which hardly interact with other objects since they have been generated by a different mechanism. Outlier objects are the members of the system which are out of synchronization. The probability for each object of being an outlier can be characterized by a numerical value: The Local Synchronization Factor (LSF). The *Interaction Plot* provides a detailed view of the dynamical behavior pattern of each object.
3. *Without any data distribution assumption.* Without any data distribution assumption or any prior knowledge, outliers are easily flagged by investigating the different interact patterns, which are driven by the intrinsic data structure.
4. *Complex data handling.* The SOD allows to detect outliers from a complex data set including clusters with arbitrary number, shape, size and densities as well as hierarchical data structures.
5. *Robustness to parametrization.* The outlier detection result is insensitive to parameter settings.

In ongoing and future work, we will focus on fully automatic outlier detection based on the powerful concept of synchronization and study online algorithms for outlier detection in data streams, which is essential in many applications.

References

1. Hawkins, D.: Identification of Outliers. Chapman and Hall, London (1980)
2. Breunig, M.M., Kriegel, H.-P., Ng, R.T., Sander, J.: Lof: identifying density-based local outliers. In: Proceedings of the ACM SIGMOD Conference (2000)
3. Knorr, E.M., Ng, R.T.: Algorithms for mining distance-based outliers in large datasets. In: VLDB, pp. 392–403 (1998)
4. Knorr, E.M., Ng, R.T.: Finding intensional knowledge of distance-based outliers. In: VLDB, pp. 211–222 (1999)
5. Boehm, C., Haegler, K., Mueller, N.S., Plant, C.: CoCo: Coding Cost For Parameter-Free Outlier Detection. In: Proc. ACM SIGKDD 2009, pp. 149–158 (2009)
6. Arenas, J.K.Y.M.A., Guilera, A.D., Zhou, C.S.: Synchronization in complex networks. Phys. Rep. 469, 93–1535 (2008)
7. Barnett, V., Lewis, T.: Outliers in Statistical Data. John Wiley, Chichester (1994)
8. Rousseeuw, P.J., Leroy, A.M.: Robust Regression and Outlier Detection. John Wiley and Sons, Chichester (1987)
9. Yamanishi, K., Takeuchi, J., Williams, G., Milne, P.: On-line unsupervised outlier detection using finite mixtures with discounting learning algorithm. In: Proceedings of KDD 2000, pp. 320–324 (2000)
10. Yamanishi, K., Takeuchi, J.: Discovering Outlier Filtering Rules from Unlabeled Data. In: Proc. ACM SIGKDD 2001, pp. 389–394 (2001)
11. Ramaswamy, S., Rastogi, R., Kyuseok, S.: Efficient Algorithms for Mining Outliers from Large Data Sets. In: Proc. ACM SIDMOD Int. Conf. on Management of Data (2000)
12. Tang, J., Chen, Z., Fu, A.W.-C., Cheung, D.W.: Enhancing effectiveness of outlier detections for low density patterns. In: Chen, M.-S., Yu, P.S., Liu, B. (eds.) PAKDD 2002. LNCS (LNAI), vol. 2336, p. 535. Springer, Heidelberg (2002)
13. Papadimitriou, S., Kitagawa, H., Gibbons, P.B., Faloutsos, C.: LOCI: Fast Outlier Detection Using the Local Correlation Integral. In: Proceedings of IEEE International Conference on Data engineering, Bangalore, India (2003)
14. Kuramoto, Y.: In: Araki, H. (ed.) Proceedings of the International Symposium on Mathematical Problems in Theoretical Physics. Lecture Notes in Physics, pp. 420–422. Springer, New York (1975)
15. Kuramoto, Y.: Chemical oscillations, waves, and turbulence. Springer, New York (1984)
16. Arenas, A., Diaz-Guilera, A., Perez-Vicente, C.J.: Plasticity and learning in a network of coupled phase oscillators. Phys. Rev. Lett. 96 (2006)
17. Kim, C.S., Bae, C.S., Tcha, H.J.: A phase synchronization clustering algorithm for identifying interesting groups of genes from cell cycle expression data. BMC Bioinformatics 9(56) (2008)
18. Böhm, C., Plant, C., Shao, J., Yang, Q.: Clustering by Synchronization. In: Proc. of the 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA (2010)
19. Aeyels, D., Smet, F.D.: A mathematical model for the dynamics of clustering. Physica D: Nonlinear Phenomena 273(19), 2517–2530 (2008)