A Fuzzy Constraint Based Outlier Detection Method

Vasudev Sharma[1(✉)], Abhinav Nagpal[1], and Balakrushna Tripathy[2(✉)]

[1] School of Computer Science and Engineering, VIT University, Vellore 632014, India
vasudevsharma74@yahoo.com
[2] School of Information Technology and Engineering, VIT University, Vellore 632014, India
tripathybk@vit.ac.in

Abstract. With a huge amount of data generated every second, it has become important to remove data anomalies. Outliers are the extreme value that deviates from other observations in data. We propose a novel outlier detection method; FCBODM (Fuzzy Constraint based Outlier Detection Method) that takes into account of fuzzy constraint and background knowledge to discover the outliers in a dataset. Our key idea is to use fuzzy constraint technology wherein we used nearness measure theory in fuzzy mathematics for finding similarities between data objects and background information. It helps in finding more meaningful outliers. Our novel approach can be integrated with traditional outlier detection methods to improve the outlier ranking. In order to validate and demonstrate the effectiveness and scalability of our method we experimented it on real and semantically meaningful datasets.

Keywords: FCBODM · Fuzzy constraints · Outlier detection · Data mining · Background knowledge · Nearness measure

1 Introduction

An outlier or an anomaly is an observation which deviates from the rest of observations significantly based on some measure. They are usually present due errors in measurements or different system conditions and thus, does not abide the with common properties of the system. With the increase in the amount of data, outlier detection has recently become an important data mining job. It is almost impossible to analyze a large dataset manually to detect outliers present in it. Hence, a mechanism that can identify outliers present in the data is essential. Outlier detection finds usage in many applications like fraud detection in credit cards, network security, medicine and public health etc. For high dimensional data, locating the correct outliers is not an easy job as the traditional outlier detection methods are not efficient. Traditional outlier detection techniques are based on a full dimension space, and incapable of detecting outliers hidden in partial dimensions because of the dimensionality curse. The outliers present in a high dimensional dataset remains unidentified due to the presence noise effects of

D.-S. Huang et al. (Eds.): ICIC 2019, LNAI 11645, pp. 515–527, 2019.
https://doi.org/10.1007/978-3-030-26766-7_47

many dimensions in it. However, the number of subspaces increases exponentially when the number of dimension increases, and exhausting all subspaces in high-dimensional data is impossible.

There are many existing clustering algorithms that detect outliers apart from clustering [1]. However, these algorithms detect only those points that are not present in any of the major cluster and call them as an outlier. Thus, the algorithms indirectly believe that outliers are the background noise with clusters embedded in them. [2] defines that outliers are those points that are not a member of any cluster and background noise. They are points which do not follow similar patterns or behavior compared to the other points in the dataset. Distance-based outlier methods take note of the outlier of a data object by its distance distances to other nearby objects and by the number of objects nearby [3, 4]. The angle-based outlier method detects an outlier by checking the difference in the angles formed by the distance vectors of all pair of points with the query point suspected to be an outlier in the dataset. A good example of such an algorithm is ABOD [5].

In Density-based outlier detection methods, density of each point in the dataset is compared w.r.t. the nearby neighborhood [6]. Breunig et al. assigned local outlier factor (LOF), a score, to all objects in the dataset [7]. In this method, similarities of a candidate outlier and its density is calculated according to its distance from the surrounding points. The LOF method has been modified many times. Some example of its other versions are uncertain local outlier factor [8], the flexible kernel density estimates [9], and natural outlier factor [10]. Eskin [11] proposed a statistical method that uses statistical tests and machine learning methods for finding anomalies. Chen et al. [12] presented robust estimation and outlier detection approaches based on their proposed generalized local statistical framework. However, all the above method follows the assumption that some set of fixed features are important for the detecting outliers.

There are methods that try to find outliers in an arbitrarily oriented subspace. Searching for an outlier using all the dimensions is less complex than dealing with a subset of the dimensions. To solve this problem, an algorithm was proposed by Aggarwal et al. [13] based on outlier detection in subspace, which can find outliers in any subspace. Kriegel et al. [14] formulated a local outlier method to find exceptional outliers by the subspace method. Müller et al. [15] propose an outlier ranking, which computes local density deviation by searching relevant subspaces for objects deviating in subspace projections.

The remaining paper is divided into the following sections – Sect. 2 explains the design of the fuzzy constraint based outlier detection method in detail. It elaborates on extension of fuzzy constraint method on the traditional methods of detecting outliers. Section 3 discusses the experimental evaluation of FCBODM on datasets along with the various evaluation measures used for detecting the quality of outlier results. Section 4 provides conclusion and the possibility of future scope towards FCBODM.

2 Design of the Proposed Algorithm

The following section introduces the fuzzy set and fuzzy similarity scale. Figure 1 shows the complete flowchart of the algorithm.

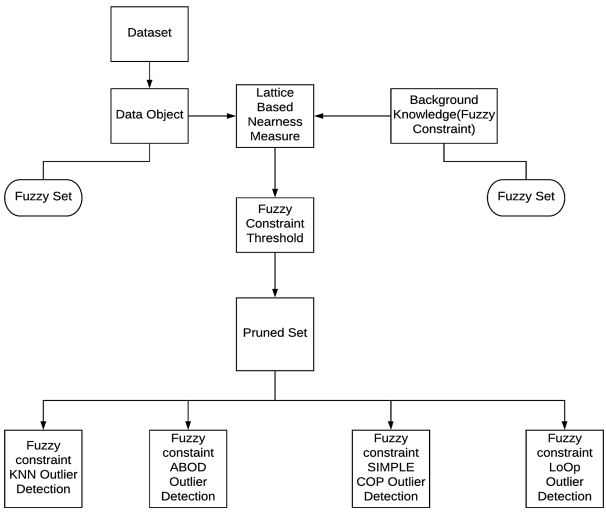

Fig. 1. Flowchart of the proposed algorithm

2.1 Fuzzy Set and Similarity Scale

The notion of fuzzy set was introduced by Zadeh [17] as an extension of the notion of crisp sets in order to model uncertain data. In a crisp set, an element is either a member of the set or not. Fuzzy sets, on the other hand, allow elements to be partially in a set. Each element is given a degree of membership in a set. This membership value can range from 0 (not an element of the set) to 1 (a member of the set). Formally, fuzzy sets can be defined as follows:

Definition 1. Let U be a universe of discourse. F is a fuzzy subset of U if there is a membership function $\mu_F : U \rightarrow [0, 1]$, which associates with each element u belonging to U a membership value $\mu_F : U$ in the interval $[0, 1]$. The membership value $\mu_F(u)$ for each u belonging U represents the grade of membership of the element u in the fuzzy set F. Equation 1 gives the notation for a fuzzy set F as proposed by Zadeh.

$$F(u) = \{(u, \mu_F(u)) : u \in U\} \tag{1}$$

Given $\{s_{1,}, s_{2,}, s_{3,}, s_{4,} \ldots, s_{n,}\}$ be n fuzzy sets on n standard classes on X. Given $S \in F(X)$, we need to know which class s should belong. To solve this problem, we

need to measure the closeness of fuzzy set using nearness measure which is formally defined as follows:

Definition 2. If $N : F(X) \times F(X) \rightarrow [0, 1]$ satisfies that

1. $\mathrm{N}(\emptyset, \mathrm{X}) = 0$ and $\mathrm{N}(S, \mathrm{S}) = 1$ whenever $S \in F(X)$,
2. $\mathrm{N}(\mathrm{S}, \mathrm{T}) = \mathrm{N}(\mathrm{T}, \mathrm{S})$ whenever $S, T \in F(X)$,
3. $N(S, P) \leq \min(N(S, T), N(T, P))$ whenever $\mathrm{S} \subseteq T \subseteq P$ then N is called a nearness measure.

2.2 Fuzzy Constraint Based on Nearness Measure

There are many types of nearness measures for numerical data such as Euclidean distance-related, Hamming distance-related, lattice-based methods and Minkowski distance-related. These distance measures are independent of the underlying data distribution. In cases where the values along the x-dimension is much larger than the y-dimension, normalization such as z-transform or min-max normalization of each data object is performed.

Definition 3. Given a dataset DS consisting of n attributes $A = \{A_1, A_2, A_3, \ldots, A_n\}$. Let $D = \{D_1, D_2, D_3, \ldots, D_k\}$ be the set of k data objects in DS where $D_i = \{D_{i1}, D_{I2}, D_{i3} \ldots D_{in}\}$. Therefore, D_{ij} represent the value of the j^{th} attribute of the i^{th} object. Let $M = \{M_1, M_2, M_3, \ldots, M_k\}$ represent the priori information given by the users, where M_i is the priori value on attribute i.

Let $G(X)$ be a fuzzy set, where X is a subset of the attributes and $D_i, M \in F(X)$. Equations 3 and 4 represents inner and outer product of D_i and M_i respectively.

$$D_i \oplus M_i = \vee_{x \in X}(D_i(x) \wedge M_i(x)) \tag{3}$$

$$D_i \otimes M_i = \wedge_{x \in X}(D_i(x) \vee M_i(x)) \tag{4}$$

The lattice based nearness measures Z_L can be defined using x and y as follows:

$$Z_L(O_i, U) = (D_i(x) \oplus M_i(x)) \wedge (1 - D_i(x) \otimes M_i(x)) \tag{5}$$

Let us now describe how lattice based nearness can be used to prune the outliers from the dataset. Given an object D_i, priori knowledge M, and threshold value σ, if $Z_L(D_i, M) \geq \sigma$, then object D_i is called a required object, which matches the constraint condition given by the user. This means the data object is of user's interest. If $Z_L(D_i, M) < \sigma$, then object D_i needs to be pruned as it does not match the constraint condition given by the user. This means the data object is not of user's interest. In this algorithm, threshold value σ also known as nearness-threshold is provided by users. Thus, we calculate the nearness measure between each object in dataset DS and priori knowledge M. This prunes the dataset DS removing data objects of disinterest from DS. This reduced dataset helps in improving the efficiency of outlier detection when further steps are applied on it.

2.3 Pseudocode

```
program FCBODM (Outliers)
  var   DS: given dataset
  const M: priori information
  const di: i^th data object
  const σ: pruning threshold
  const n: number of data objects in DS
  begin:
    normalized_DS:= min_max_normalization(DS)
    repeat:
    inner:= inner_product(di,Mi)
    outer:= outer_product(di,Mi)
    nearness_di:= min(inner,1-outer)
    if nearness_di< σ then
       pruned_DS:= prune(di,normalized_DS)
    until i=n
    outliers:= method(pruned_DS)#method:ABOD,KNN,LoOP,COP
   end
```

2.4 Outlier Detection Methods

FCBODM ABOD. It has been seen that comparing distances between points to identify outliers is not efficient if the dimensionality of the dataset is large. ABOD algorithm proposes uses the distance between points and the direction of the distance vectors. Comparison of the angles between two distance vectors to other points is carried out in the algorithm. This helps to identify outliers in the dataset. If the angles between the distance vectors of an object are relatively large, then the object is inside the cluster. However, if the angles between the distance vectors of an object are relatively small, then the object is outside the cluster. The difference in the direction of the distance vectors of objects is calculated using ABOF - angle based outlier factor [7]. The ABOF(A) is the variance over the angles between difference vectors of all pairs of points in dataset D to a data point A weighted by the distance of the points:

$$ABOF(\overrightarrow{A}) = VAR_{\overrightarrow{B},\overrightarrow{C}\in D}\left(\frac{\langle \overline{AB}, \overline{AC}\rangle}{\|\overline{AB}\|^2 \cdot \|\overline{AC}\|^2}\right) \tag{6}$$

For every object in the dataset, the ABOF value is calculated and the points are ranked on their basis. ABOD algorithm has the advantage of not requiring any parameter.

FCBODM KNN. Developed by Ramaswamy and Shim [4], KNN is a distance-based outlier that calculates the distance of a point from its neighboring points. All points in the dataset are ranked according to their distances from their nearest neighbor. The points with the largest distance are declared as the outlier. Let there be a point p, then DK(p) denotes the distance from the kth nearest neighbor. Let n be the number of outliers that need to be removed. Here, DK(p) also describes the degree of how much outlier is the point p. Points with larger DK(p) value have sparse surroundings and have more chances of being an outlier than points inside a dense cluster that have a lower value of DK(p). Let us say we need to find n outliers. Then, the n points with the maximum DK(p) values are declared as outliers. An advantage of this method is that the user need not have to provide a distance variable to qualify a point as an outlier.

FCBODM Simple COP. In order to find an outlier present in the subspaces of the original attribute space, COP algorithm was made. It considers many combinations of subsets of attributes, to find outliers deviating from their values. The points detected as outliers do not relate to any major correlation in the data. The local correlations within the outlier detection method are considered first. Then, outliers present in the subset of the actual dataset are identified. It then chooses the relevant correlation of attributes to detect the corresponding outlier. An object is considered as an outlier if it does not match the correlations. The objects that are present on a δ-dimensional hyperplane, called correlation hyperplane, show local correlation. Here, d is the dimensionality of the dataset and $\delta < d$. Thus, outliers are the objects that are not present and do not show any correlation in such hyperplanes.

FCBDOM LoOP. Local outlier probabilities (LoOP) is an outlier detection method which is based on local density. It evaluates whether a point is an outlier or not by giving a score from 0 to 1. Let us take a set P containing k objects with d as the distance function. Let $o \in D$ be the probabilistic distance to a context set $S \subseteq P$, referred to as pdist(o,S). This distance has the following property: $\forall s \in S$: P[d(o,s) $\leq$ pdist(o,S)] φ. A sphere around o with a radius of pdist covers objects with a probability of φ in set S. The probabilistic distance pdist(o, S) between o and S can be calculated as the statistical extent of set S. Based on this, density around an object w.r.t. a context set, the Probabilistic Local Outlier Factor (PLOF) of an object $o \in P$ w.r.t. a significance λ and a context set $S(o) \subseteq P$, is defined as:

$$PLOF_{\lambda,S(o)} = \frac{pdist(\lambda, o, S(o))}{E_{s \in S(O)}[pdsit(\lambda, s, S(s))]} \tag{7}$$

For every object o in the dataset, the PLOF value is calculated which is the ratio of the estimation for the density around o and the expected value of the estimations for the densities around all objects in the context set S(o). The points with the minimum sorted PLOF values are declared outliers.

3 Experimental Evaluation

3.1 Datasets

We evaluated our algorithm on two types of datasets; real datasets, which have been used in the research literature and semantically meaningful datasets, where the semantic interpretation of outliers can be given from the datasets. The algorithm was tested on six UCI datasets. Prior to the evaluation, we performed dataset preparation for the evaluation of outlier detection algorithms. The classification datasets have been used where we assumed class having minority labels as outlier class since outlier detection is tantamount to detect objects belonging to a rare class. The classification datasets need to transformed to outlier datasets hence we performed down-sampling of a class, duplicates removal, min-max normalization and cleaning to deal with categorical and missing attributes.

Real Datasets. To evaluate our outlier detection we selected three UCI machine learning repository datasets [20] which are real world benchmark datasets namely - Shuttle, WPBC, and Ionosphere. Outlier mining is nothing but conceptually similar to detecting objects belonging to a rare class hence we focus on datasets where the class labels feature a clear minority class. We assume this class to contain the outliers in these datasets. In addition, these datasets are pre-processed as illustrated in the above section. Table 1 summarizes the characteristics of these three datasets.

Table 1. Characteristics of real datasets used in literature

Name	Instances	Outliers	Attributes
Shuttle	351	126	32
WPBC	198	47	33
Ionosphere	148	6	19

Semantically Meaningful Datasets. These datasets have certain classes that can be identified with real world scenarios. Data points containing outliers are both rare and digressing, for instance, consider patients suffering from 'Hutchinson-Gilford Progeria'; a rare disease; among a population of patients. But for some scenarios, there might be a possibility that outliers are dominated within a discrimination dataset. To overcome this problem, we down-sampled outlier class (2, 5, 10, 20% of outliers). UCI repository datasets [20] were selected and processed for evaluation of outlier results. These datasets are Cardiotocography, Arrhythmia and Heart Disease as illustrated in Table 2.

Table 2. Characteristics of semantically meaningful datasets.

Name	Instances	Outliers	Attributes
Cardiotocography	2126	471	21
Heart Disease	270	120	13
Arrhythmia	450	206	259

3.2 Evaluation Measures

Outlier detection methods used here yields a complete ranking of the database objects. Data points are given an outlier score upon evaluation by the outlier detections methods. Not every data object is relevant as the user is only interested in finding out the outlier score of say topmost ranked objects of the whole set. One such evaluation criteria are Precision at n (P@n) [16] where the target number of data objects; n is specified well in advance. Precision at n signifies ratio of correct results amid top n ranks [16]. Consider a database(DB) of size N consisting of outliers O ⊆ DB and inliers I ⊆ DB where DB = I ∪ O. P@n can be formulated as

$$P@n = \frac{|\{o \in O | rank(o) \leq n\}|}{n} \tag{8}$$

While P@n is a measure to evaluate the robustness of the outlier detection algorithm, it is unclear on what value of parameter n to choose. When the number of the outlier(n = |O|) is low in comparison to large N, P@n value is marginally small hence not useful enough whereas when the number of outliers(n = |O|) is large enough with respect to N, P@n would he high as small fraction of inliers exist. For an unambiguous measure, P@n should be adjusted for a chance to compare different measures where there is variation in an expected score. Since the maximum number of outliers are O, P@n maximum value is |O| / n provided that O > n else it is 1. The expected value of a completely random ranking is given by |O| / n. Henceforth Adjusted P@n is formulated by the given formula.

$$Adjusted\ P@n = \frac{P@n - |O|/N}{1 - |O|/N} \tag{9}$$

An anomaly with both these measures is the trade-off between the number of outlier and inliers. Generally for an outlier dataset, |I| ≫ |O| and |I| = N. P@n and Adjusted P@n measures are highly sensitive to n. The same issue of sensitivity toward n occurs with Adjusted P@n. Nonetheless, the other evaluation measures solve this problem by averaging over values of n. On such measure is average precision (AP) used in information retrieval evaluation methods. Instead of evaluation over single n, the values are averaged over ranks of outlier objects.

$$AP = \frac{1}{|O|} \sum\nolimits_{o \in O} P@rank(o) \tag{10}$$

Similar to Adjusted P@n, an adjusted form of average precision is used for comparing different datasets, having the expected value of random ranking as |O| / n and maximum value as 1.

Another evaluation measure used widely in unsupervised learning is the Receiver Operating Characteristic (ROC). It's obtained by plotting across all n the true positive rate, and the false positive rate. If a ROC curve is close to the diagonal then it may be probably due to random outlier ranking, whereas a perfect ranking would result in a

curve where a vertical line is at false positive rate 0 and a horizontal line is at the top of the plot. ROC adjusts for a chance as the normalization of false positives rate by false positive and the normalization of true positive rates by true positive is carried inherently. Therefore, ROC is insensitive to adjustment for a chance. ROC AUC (value varies between 0 and 1), a measure which summarizes a ROC curve by a single value. It can be thought as the average of the recall at n, with n taken over the all the ranks of inlier data objects in |I|. External ground truth labels that are inliers and outlier are required in all the above evaluation measures.

3.3 Evaluation on the Datasets

To assess and validate the quality of our outlier detection algorithm we have used 3 different curves namely PR AUC, ROC AUC and P@n (where we took n = |O|) as indicated in Sect. 3.2. For each of the evaluation measure, 3 plots were produced on the ionosphere (35.9% outliers) and Heart disease (44% outliers) datasets.

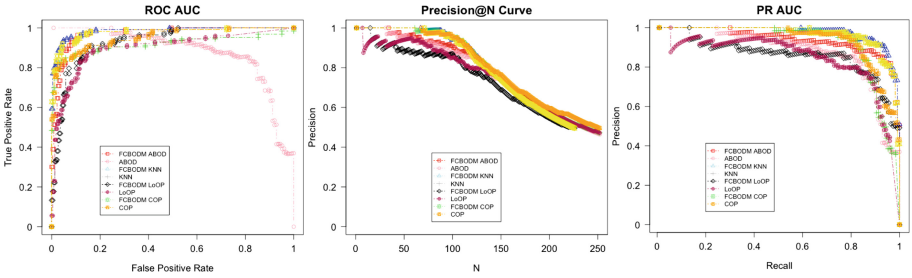

Fig. 2. Results on real dataset (IONOSPHERE), comparing ROC AUC, Precision@n and PR AUC with existing outlier algorithms

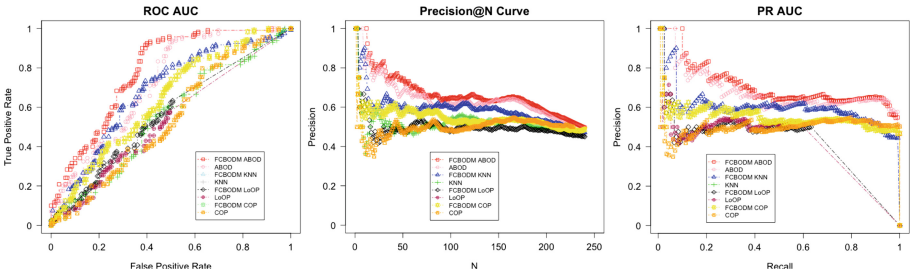

Fig. 3. Results on semantically meaningful dataset (HEART DISEASE), comparing ROC AUC, Precision@n and PR AUC with existing outlier algorithms

On these two groups of datasets we can clearly see the improvement in quality when compared FCBODM with traditional outlier methods. Figure 2 illustrates Precision@n, PR AUC and ROC AUC measures on the ionosphere dataset; a real dataset, with 4 traditional outlier methods as indicated in Sect. 2.3 and four improvised

FCBODM. An equivalent analysis is depicted in Fig. 3 on the Heart Disease dataset; a semantically meaningful dataset. Table 3 enumerates all the evaluation measures suggested in Sect. 3.2 on two real (Shuttle and WPBC) and two semantically meaningful (Heart disease and Arrhythmia) datasets.

We evaluated the runtimes of our algorithm FCBODM with existing outlier detection methods used in literature. Run time evaluation was carried on a real and a semantically meaningful UCI dataset. To our observation, FCBODM proved to excel at runtime when evaluated against earlier outlier methods. Figure 4 (left bar chart) shows the runtime percentage change in performance due to FCBODM when tested against traditional methods on SHUTTLE dataset constituting 1.38% of outliers. Computationally expensive algorithms such as ABOD and simple COP reported a change in the runtime of about 39.31% and 69.6% respectively while algorithms such KNN and LoOP suggested an improvement of about 21.05% and 2% respectively.

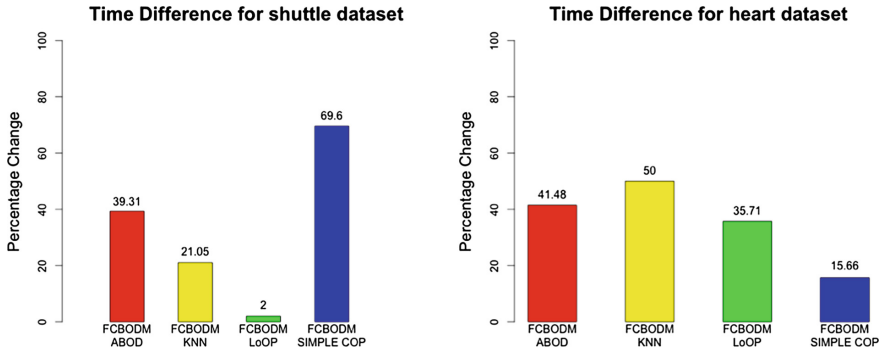

Fig. 4. Measure of percentage change in runtime milliseconds when compared with existing outlier detection techniques on shuttle (left bar chart) and heart dataset (right bar chart).

Table 3. Algorithm results on various evaluation measures.

Datasets	Algorithm	ROC AUC	AP	Max. F1	NDCG	PR AUC	Adj. AP	Adj. max F1	Adj. DCG
SHUTTLE	ABOD FCBODM	0.97	0.94	0.89	0.99	0.94	0.91	0.84	0.95
	ABOD	0.90	0.88	0.83	0.98	0.88	0.81	0.74	0.90
	KNN FCBODM	0.98	0.97	0.92	1.00	0.97	0.96	0.88	0.98
	KNN	0.92	0.92	0.88	0.99	0.92	0.88	0.81	0.94
	LOOP FCBODM	0.98	0.34	0.54	0.62	0.33	0.33	0.53	0.46
	LOOP	0.96	0.20	0.36	0.52	0.19	0.19	0.35	0.32
	SIMPLE COP FCBODM	0.97	0.27	0.37	0.62	0.25	0.26	0.37	0.45
	SIMPLE COP	0.85	0.22	0.39	0.56	0.20	0.21	0.38	0.37

(*continued*)

Table 3. (*continued*)

Datasets	Algorithm	ROC AUC	AP	Max. F1	NDCG	PR AUC	Adj. AP	Adj. max F1	Adj. DCG
HEART DISEASE	ABOD FCBODM	0.78	0.71	0.76	0.93	0.71	0.48	0.57	0.65
	ABOD	0.72	0.69	0.76	0.92	0.68	0.38	0.53	0.56
	KNN FCBODM	0.68	0.61	0.67	0.90	0.60	0.29	0.40	0.44
	KNN	0.55	0.53	0.66	0.86	0.52	0.08	0.34	0.17
	LOOP FCBODM	0.56	0.50	0.62	0.86	0.42	0.09	0.31	0.23
	LOOP	0.53	0.51	0.66	0.85	0.41	0.05	0.33	0.14
	SIMPLE COP FCBODM	0.64	0.56	0.65	0.87	0.55	0.20	0.38	0.31
	SIMPLE COP	0.54	0.51	0.67	0.84	0.50	0.04	0.36	0.06
WPBC	ABOD FCBODM	0.49	0.29	0.40	0.72	0.26	0.05	0.20	0.16
	ABOD	0.48	0.25	0.39	0.69	0.23	0.02	0.20	0.08
	KNN FCBODM	0.53	0.26	0.40	0.66	0.26	0.03	0.21	-0.01
	KNN	0.47	0.23	0.39	0.63	0.22	-0.02	0.19	-0.09
	LOOP FCBODM	0.58	0.31	0.44	0.71	0.27	0.08	0.25	0.13
	LOOP	0.59	0.28	0.43	0.69	0.25	0.06	0.25	0.08
	SIMPLE COP FCBODM	0.59	0.32	0.47	0.67	0.32	0.09	0.29	0.03
	SIMPLE COP	0.59	0.29	0.42	0.67	0.29	0.08	0.25	0.01
ARRHYTHMIA	ABOD FCBODM	0.74	0.74	0.68	0.95	0.74	0.53	0.40	0.70
	ABOD	0.72	0.71	0.64	0.94	0.71	0.50	0.37	0.69
	KNN FCBODM	0.75	0.76	0.68	0.95	0.75	0.55	0.41	0.72
	KNN	0.73	0.73	0.64	0.94	0.72	0.52	0.38	0.71
	LOOP FCBODM	0.74	0.73	0.68	0.95	0.72	0.51	0.40	0.69
	LOOP	0.72	0.71	0.65	0.94	0.69	0.49	0.39	0.69
	SIMPLE COP FCBODM	0.93	0.86	0.82	0.97	0.86	0.79	0.73	0.88
	SIMPLE COP	0.90	0.85	0.82	0.97	0.84	0.77	0.72	0.86

We tested FCBODM on semantically meaningful Heart Disease dataset which comprised 22% of outliers. Figure 4 (right bar chart) reveals that computationally expensive detection methods resulted in a great improvement in run time; ABOD and Simple COP, when evaluated on FCBODM, indicated an increase in runtime of 41.48% and 15.66% respectively, whereas other outlier detection methods like KNN and LoOP led to improvement of 50% and 35.71% respectively.

4 Conclusion and Future Work

We formulated a novel fuzzy constraint-based outlier detection which can be extended on top of existing outlier detection algorithms to improve not only the various evaluation parameters (ROC AUC, PR AUC, NDCG, F1 score) but also helps us to fathom pertinence of outlier mining results. For improving the relevance of outlier results we relied on lattice-based nearness measure in fuzzy mathematics is where we pruned some existing data objects that do not adhere to constraint condition (background knowledge). Nearness measure technique coupled with constraint condition drastically

reduced the size of the dataset. Our algorithm, when combined with existing outlier detection methods (ABOD, Simple COP, LoOP, and KNN), yields an improvement in the performance measures. We validated our results on three real and three semantically meaningful UCI datasets and our novelty proved to be better than existing outlier methods. Due to computational constraint, we were unable to evaluate our results on a large number of datasets and large varieties of algorithms. We aim to use parallel and distributed environments to improve our results and extend to approach on a greater number of datasets and existing outlier detection algorithms.

References

1. Aggarwal, C.C., Yu, P.: Finding generalized projected clusters in high dimensional spaces. In: Proceedings of ACM SIGMOD, pp. 70–81 (2000)
2. Arning, A., Agrawal, R., Raghavan, P.: A linear method for deviation detection in large databases. In: Proceedings of KDD, pp. 164–169 (1996)
3. Knorr, E.M., Ng, R.T.: Algorithms for mining distance-based outliers in large datasets. In: Proceedings of VLDB (1998)
4. Ramaswamy, S., Rastogi, R., Shim, K.: Efficient algorithms for mining outliers from large data sets. In: Proceedings of SIGMOD, pp. 427–438 (2000)
5. Kriegel, H.P., Schubert, M., Zimek, A.: Angle-based outlier detection in high-dimensional data. In: Proceedings of KDD, pp. 444–452 (2008)
6. Jin, W., Tung, A., Han, J.: Mining top-n local outliers in large databases. In: Proceedings of KDD, pp. 293–298 (2001)
7. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: LOF: identifying density-based local outliers. In: ACM SIGMOD Record, vol. 29, no. 2, pp. 93–104. ACM (2000)
8. Liu, B., Xiao, Y., Yu, P.S., Hao, Z., Cao, L.: An efficient approach for outlier detection with imperfect data labels. IEEE Trans. Knowl. Data Eng. **26**(7), 1602–1616 (2014)
9. Schubert, E., Zimek, A., Kriegel, H.P.: Generalized outlier detection with flexible kernel density estimates. In: Proceedings of the 14th SIAM International Conference on Data Mining (SDM), Philadelphia, PA, pp. 542–550 (2014)
10. Huang, J., Zhu, Q., Yang, L., Feng, J.: A non-parameter outlier detection algorithm based on natural neighbor. Knowl.-Based Syst. **92**, 71–77 (2016)
11. Eskin, E.: Anomaly detection over noisy data using learned probability distributions. In: International Conference on Machine Learning, pp. 255–262 (2000)
12. Chen, F., Lu, C.T., Boedihardjo, A.P.: GLS-SOD: a generalized local statistical approach for spatial outlier detection. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2010)
13. Aggarwal, C.C., Philip, S.Y.: An effective and efficient algorithm for high-dimensional outlier detection. Int. J. Very Large Data Bases **14**(2), 211–221 (2005)
14. Kriegel, H.P., Kröger, P., Schubert, E., Zimek, A.: Outlier detection in arbitrarily oriented subspaces. In: IEEE International Conference on Data Mining, pp. 379–388 (2012)
15. Müller, E., Schiffer, M., Seidl, T.: Statistical selection of relevant subspace projections for outlier ranking. In: 2011 IEEE 27th International Conference on Data Engineering (ICDE), pp. 434–445. IEEE (2011)
16. Campos, G.O., et al.: On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. Data Mining Knowl. Discov. **30**(4), 891–927 (2016)
17. Zadeh, L.: Fuzzy sets. Inf. Control **8**(3), 338–353 (1965)

18. Wu, D., Mendel, J.M.: A vector similarity measure for linguistic approximation: interval type-2 and type-1 fuzzy sets. Inf. Sci. **178**(2), 381–402 (2008)
19. Wu, D., Mendel, J.M.: A comparative study of ranking methods, similarity measures and uncertainty measures for interval type-2 fuzzy sets. Inf. Sci. **179**(8), 1169–1192 (2009)
20. UCI machine learning repository. http://archive.ics.uci.edu/ml