Markov Boundary-Based Outlier Mining

Kui Yu and Huanhuan Chen

***Abstract*— It is a grand challenge to identify the outliers existing in subspaces from a high-dimensional data set. A brute-force method is computationally prohibitive since it requires examining an exponential number of subspaces. Current state-of-the-art methods explore various heuristics to significantly prune subspaces, facing the tradeoff between the subspace completeness and search efficiency. In this brief, we discuss a principal type of subspace outliers whose behaviors are different from the others on individual attributes. We formulate such outliers by a novel notion of the Markov boundary-based (MBB) outliers. The central idea is that for each attribute T in a data set, we consider only the subspace representing the knowledge needed to predict the behavior on T, which is captured by the MB of T. Then, the outliers whose behavior is different from others on T can be detected in the subspace of the MB, and thus, our approach reduces the number of possible subspaces from exponential to linear with respect to dimensionality. Using both synthetic and real data sets, we validate the effectiveness and efficiency of our method.**

***Index Terms*— Bayesian network (BN), Markov boundary (MB), MB-based (MBB) outlier, subspace outliers.**

I. Introduction

In many applications, we need to detect the outliers from high-dimensional data [1]. High dimensionality brings two major challenges on outlier detection [28]. One is that the relative contrast (e.g., distance) between data objects will become more and more similar as the dimensionality increases. The other is that outliers can be easily masked by irrelevant (noise) attributes (in this brief, we use the terms "attribute," "variable," and "dimension" interchangeably) since a high-dimensional data set may include many irrelevant (noise) dimensions. Subspace outlier detection approaches can address the problems by identifying the outliers in a low-dimensional subspace. However, a grand challenge is how to effectively identify the outliers from numerous subspaces of the original feature space. Given a data set with high dimensionality, the number of subspaces will be exponential with the dimensionality. An outlier may exist in any of those subspaces. A brute-force method has to examine an exponential number of subspaces and thus often is impractical due to the prohibitive computational costs.

The state-of-the-art methods for subspace outlier detection, such as [11] and [17], explore various heuristics in order to significantly prune subspaces. While we will review some representative, the state-of-the-art methods in Section II, two challenges remain. First, the definitions of subspace outliers in those methods often have to be constrained heavily by the heuristics adopted. For example, the 4S method [19] greedily selects top-k attribute-pairs with the highest correlations for generating subspaces and confines the outlier detection only in those subspaces. Second, to reduce the computational cost, many methods have to trade off completeness in outlier search for efficiency—not all subspaces where outliers may exist are searched. For instance, the popular outlier search algorithms, including adaptive outlierness for subspace outlier ranking (OUTRES) [18], high contrast subspaces (HiCS) [14], and cumulative mutual information (CMI) [20], only search top-k subspaces for outliers.

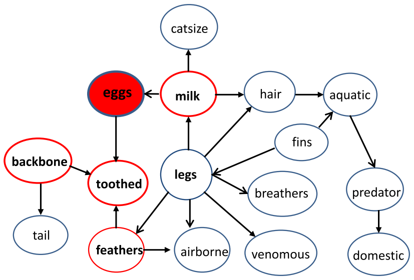

Fig. 1. Using Markov boundaries to detect outliers.

Can we systematically search all the subspaces where outliers may exist without sacrificing efficiency? In this brief, we propose the notion of Markov boundary-based (MBB) outliers based on the fundamental concept of MB in Bayesian networks (BNs) [22]. For any attribute T in a BN, its MB is the set of children, parents, and spouses of T. The MB of T represents the knowledge needed to predict the behavior on T by making T irrelevant to the remaining attributes [22]. A critical idea is that for each attribute T, we only need to consider the MB of T to detect the outliers whose behavior is different from others on T.

Example 1 (Motivation and Ideas): Consider the zoo data set from the University of California at Irvine (UCI) machine learning repository [5], which contains the records about 101 animals on 16 attributes, including "eggs," "milk," and so on. Suppose we are interested in outliers in attribute "eggs" and want to find animals whose behavior of laying eggs is dramatically from the others. An animal's behavior of laying eggs depends on the animal's other attributes. By looking at only the values on the attribute "eggs," we cannot find any meaningful outliers. Instead, we have to also consider the attributes that are related to "eggs." For example, most mammals do not lay eggs at all, whereas all birds lay eggs. Now, the challenge is how we can systematically search subspaces containing attribute "eggs," where an outlier may appear. In the worst cases, a brute-force method has to search the remaining 15 attributes resulting in $2^{15} - 1 = 32\,767$ possible subspaces.

We can learn a BN, as shown in Fig. 1, from the zoo data set using the max–min hill-climbing algorithm [26]. Under a moderate assumption that will be explained later, the MB of "eggs" includes its parent node "milk," child node "toothed," and spouse nodes "backbone" and "feathers." Theoretically, the value of "eggs" depends on only those attributes in its MB. Accordingly, instead of searching 32 767 possible subspaces, we only need to focus on the subspace formed by the MB with respect to "eggs." As will be reported in Section IV, the outliers on attribute "eggs," such as platypus and sea

Manuscript received August 22, 2017; revised February 8, 2018 and May 16, 2018; accepted July 25, 2018. Date of publication August 20, 2018; date of current version March 18, 2019. This work was supported in part by the National Key Research and Development Program of China under Grant YF2150110007, in part by National Science Foundation of China under Grant GG2150110001 and Grant GG2150110029, and in part by the Fundamental Research Funds for Central Universities under Grant PA2018GDQT0011. *(Corresponding author: Kui Yu.)*

K. Yu is with the School of Computer and Information, Hefei University of Technology, Hefei 230009, China (e-mail: yukui@hfut.edu.cn).

H. Chen is with the School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China (e-mail: hchen@ustc.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2018.2861743

snake, can be found within the subspace induced by the MB of the attribute.

While the idea of using MB to detect outliers whose behaviors are different on individual attributes is intuitive, we need to develop a method that works effectively and efficiently. Our contributions include the following.

1) We propose the notion of MBB outliers. By this novel notion of MBB outliers, we reduce the number of possible subspaces from exponential to linear with respect to the number of attributes in data.
2) We develop the MBB outlier mining (MBOM) algorithms to mine MBB outliers. To compare the outliers with different MB subspaces, the MBOM algorithm employs the local outlier factor (LOF) metric to discover the MBB outliers in MB subspaces.
3) Using synthetic and real-world data sets, we compare the proposed method, in this brief, with the state-of-the-art subspace outlier detection methods in terms of the effectiveness and efficiency.

II. Related Work

Outlier detection as an important data mining task has been studied widely in the community [8], [10], [13]. Different outlier detection paradigms have been proposed, such as distance-based methods [4], [16], [23], density-based methods [6], linear methods [25], spectral methods [12], [24], and subspace-based methods [2], [18]. More details can be found from good survey papers [1], [7], [9], [28] and references therein.

High-dimensional data have become ubiquitous in various domains. Due to the curse of dimensionality, relative contrast between the data objects decreases with increasing dimensionality [21]. Then, outliers often exist in low-dimensional subspaces (i.e., small attribute subsets of the original attributes) [18]. The traditional outlier detection methods compute the degrees of deviation between the data objects in the full feature space and cannot detect the outliers existing in subspaces [14]. Subspace-based methods as popular solutions have been developed for tackling outlier detection in high dimensionality [15], [17], which search for a set of feature subspaces and identify outliers in those subspaces to avoid the curse of dimensionality. Limited by space, here, we only focus on several recent representative results on subspace-based outlier detection, which are highly related to this paper.

Müller *et al.* [18] proposed the OUTRES algorithm to select the subspaces that are not distributed uniformly in random using an *a priori*-like, breadth-first style search scheme. Kriegel *et al.* [17] proposed a local model that takes into account correlations among varying subsets of attributes to find outliers in arbitrarily oriented subspaces. Keller *et al.* [14] proposed the concept of high-contrast subspaces and developed the *HiCS* method, and Nguyen *et al.* [20] proposed the *CMI* method, both finding high-contrast subspaces for outlier detection.

Those methods employ a breadth-first search on subspaces. The computational cost is expensive on high-dimensional data. Nguyen *et al.* [19] proposed the 4S algorithm to mitigate the problem of exponential runtime. The 4S algorithm computes the correlation of each pair of dimensions. Then, it greedily selects the top-k attribute-pairs with the largest correlations by pairwise comparisons and transforms them to an undirected correlation graph, where nodes are dimensions. Finally, 4S mines maximal cliques from this graph for subspaces.

The existing methods suffer from two aspects [28]. First, the capability of detecting subspace outliers in those methods is heavily constrained by the heuristics used. Second, due to high computational costs, those methods face the tradeoff between detection accuracy and completeness of subspace search (i.e., not all subspaces where outliers may exist are searched).

III. Markov Boundary-Based Outlier Mining

In this section, we first present the notion of MBB outliers and, then, propose a method to find the MBB outliers. Consider a data set $D = \{D_1, D_2, \ldots, D_N\}$ of N records (data objects) on a set of M dimensions $F = \{F_1, F_2, \ldots, F_M\}$. The value of D_j on attribute F_i is denoted by $D_j \cdot F_i$.

A. MBB Outliers

Let P be the joint probability distribution of the set of random variables F via a directed acyclic graph (DAG) G. We call the triplet $\langle F, G, P\rangle$ a BN if $\langle F, G, P\rangle$ satisfies the Markov condition: every variable is independent of any subset of its nondescendant variables conditioned on its parents in G [22]. By the Markov condition, a BN encodes the joint probability P over F and decomposes it into a product of the conditional probability distributions over each variable given its parents in G. Pearl [22] defined the fundamental notions of faithfulness and MB.

Definition 2 (Faithfulness): Given a BN $\langle F, G, P\rangle$, G is faithful to P over F if and only if every independence present in P is entailed by G and the Markov condition. P is faithful if and only if there exists a DAG G, such that G is faithful to P. A BN is said to satisfy the faithfulness condition if P is faithful to G.

Definition 3 (MB): If a BN satisfies the faithfulness condition, then the MB of a variable T in the BN, denoted by $\mathrm{MB}(T)$, consists of the set of children, parents, and spouses of T.

Theorem 4 [22]: In a faithful BN, the MB of each attribute is unique.

Assuming F is a full feature space and $S \subseteq F$, any attribute subset $S = \{F_1, \ldots, F_j\}$ $(j \leq M)$ is called a j-D subspace. In addition, the projection of S on data set D is defined as the j-D data vector $D.S = \{D.F_1, \ldots, D.F_j\}$, where $D.F_j$ represents the values of attribute F_j in D.

Definition 5 (MB Subspace): For $\forall F_i \in F$, the set $\{\mathrm{MB}(F_i) \cup F_i\}$ represents the MB subspace of F_i, and the projection of $\{\mathrm{MB}(F_i) \cup F_i\}$ on D is $D \cdot \{\mathrm{MB}(F_i) \cup F_i\}$.

Corollary 6: In a faithful BN, for $\forall F_i \in F$, F_i has a unique MB subspace.

Theorem 7 [22]: For $\forall F_i \in F$, $\mathrm{MB}(F_i)$ is a minimal set of attributes and satisfies $\forall S \subseteq F \setminus \{\mathrm{MB}(F_i) \cup \{F_i\}\}$ s.t. $P(F_i|\mathrm{MB}(F_i), S) = P(F_i|\mathrm{MB}(F_i))$.

A distinct benefit coming from Corollary 6 is that in order to find the outliers existing in MB subspaces, the number of subspaces that we need to search equals to the number of MB subspaces and, thus, is linear to the number of dimensions of a data set. Theorem 7 illustrates that $\mathrm{MB}(F_i)$ renders $F_i \in F$ statistically independent of all the remaining attributes, and all information that may influence F_i's value is stored in the values of the attributes of its MB. Thus, $\mathrm{MB}(F_i)$ is the only knowledge needed to predict the behavior on F_i.

Using the above-mentioned notions, in this section, we employ the commonly used LOF metric [6] as the outlier measure to define and identify the MBB outliers in MB subspaces. As for LOF as a seminal work for mining local outlier, any improvement metrics in the area can be used here.

Definition 8 (MBB Outlyingness): For a data object $D_j \in D$, its MBB outlyingness degree with respect to F_i is defined as

$$\deg_{F_i}(D_j) = \mathrm{LOF}(D_j \cdot \{\{F_i\} \cup \mathrm{MB}(F_i)\}).$$

The larger the degree, the more outlying the object.

Definition 9 (MBB Outlier): Given an outlyingness threshold η, D_j is an MBB outlier with respect to F_i if $\deg_{F_i}(D_j) \geq \eta$.

Based on Corollary 6, MBB outliers are tractable without searching an exponential number of subspaces. With Definition 5, any existing MB discovery algorithms can be used here for identifying the MB subspace of an attribute of interest.

B. Mining MBB Outliers

In this section, we first propose how to calculate $\text{LOF}(D_j \cdot \{\{F_i\} \cup \text{MB}(F_i)\})$ and, then, present the MBOM algorithm to discover MBB outliers.

1) Calculating $LOF(D_j \cdot \{\{F_i\} \cup MB(F_i)\})$: The key idea behind LOF is based on a concept of a local density. The local density is given by k nearest neighbors of a data object (k is a positive integer) whose distance is used to estimate the density. By comparing the local density of the data object with the local densities of its neighbors, one can identify the regions of similar density and data objects that have a substantially lower density than their neighbors. These are considered to be outliers. LOF compares the local reachability density (lrd) of the ϵ-distance neighborhood set of a test data instance with those of the neighborhoods of each member within the ϵ-distance neighborhood set. Parameter ϵ is defined as a positive integer for denoting the size of the ϵ-distance neighborhood set [6]. Using the LOF score defined in [6], let $\text{MBS}(F_i)$ represent $\{\text{MB}(F_i) \cup \{F_i\}\}$ and $d(D_j, D_i)$ denote the distance between the objects D_j and D_i. Then, $\text{LOF}(D_j \cdot \{\{F_i\} \cup \text{MB}(F_i)\})$ can be calculated as

$$\text{LOF}(D_j \cdot \{\{F_i\} \cup \text{MB}(F_i)\}) = \frac{\sum_{D_i \in N_\epsilon(D_j \cdot \text{MBS}(F_i))} \frac{\text{lrd}_\epsilon(D_i \cdot \text{MBS}(F_i))}{\text{lrd}_\epsilon(D_j \cdot \text{MBS}(F_i))}}{|N_\epsilon(D_j \cdot \text{MBS}(F_i))|} \quad (1)$$

where $|N_\epsilon(D_j \cdot \text{MBS}(F_i))|$ denotes the number of data objects in the ϵ-distance neighborhood set of D_j in the MB subspace of F_i (i.e., $\text{MBS}(F_i)$), and $\text{lrd}_\epsilon(D_j \cdot \text{MBS}(F_i))$ (lrd) is defined as the inverse average reachability distance from the neighbor set of $N_\epsilon(D_j \cdot \text{MBS}(F_i))$ and is calculated as

$$\text{lrd}_\epsilon(D_j \cdot \text{MBS}(F_i)) = \frac{|N_\epsilon(D_j \cdot \text{MBS}(F_i))|}{\sum_{D_i \in N_\epsilon(D_j \cdot \text{MBS}(F_i))} \text{reach\_dist}_\epsilon(D_j \cdot \text{MBS}(F_i), D_i \cdot \text{MBS}(F_i))}. \quad (2)$$

The lrd denotes that an outlier has low density compared to its local neighborhood, and thus, a high value of $\text{lrd}_\epsilon(D_j \cdot \text{MBS}(F_i))$ indicates the outlierness of D_j. In (2), $\text{reach\_dist}_\epsilon(D_j \cdot \text{MBS}(F_i), D_i \cdot \text{MBS}(F_i))$ is the reachability distance of $D_j \cdot \text{MBS}(F_i)$ with respect to $D_i \cdot \text{MBS}(F_i)$ and is calculated as [6]

$$\text{reach\_dist}_\epsilon(D_j \cdot \text{MBS}(F_i), D_i \cdot \text{MBS}(F_i)) = \max\{\epsilon\text{-}d(D_i \cdot \text{MBS}(F_i)), d(D_j \cdot \text{MBS}(F_i), D_i \cdot \text{MBS}(F_i))\}. \quad (3)$$

In (3), we define $\epsilon\text{-}d(D_j \cdot \text{MBS}(F_i))$ as the distance between $D_j \cdot \text{MBS}(F_i)$ and $D_i \cdot \text{MBS}(F_i)$ in the MB subspace with respect to F_i. According to [6], the distance should satisfy the following condition.

1) For at least ϵ objects $D_h \in \{D - \{D_j\}\}$, the following term holds:

$$d(D_j \cdot \text{MBS}(F_i), D_h \cdot \text{MBS}(F_i)) \leq d(D_j \cdot \text{MBS}(F_i), D_i \cdot \text{MBS}(F_i)).$$

2) For at most $\epsilon - 1$ objects, $D_h \in \{D - \{D_j\}\}$, the following term holds:

$$d(D_j \cdot \text{MBS}(F_i), D_h \cdot \text{MBS}(F_i)) \leq d(D_j \cdot \text{MBS}(F_i), D_i \cdot \text{MBS}(F_i)).$$

Using (1), TMB is the set including the MB subspaces of all attributes on D, we define the outlier score of D_j as

$$\text{Outlierscore\_}N(D_j) = \underset{\text{MB}(F_i) \in \text{TMB}}{\arg\max} \ \text{LOF}(D_j \cdot \{\{F_i\} \cup \text{MB}(F_i)\}) \quad (4)$$

For a data object, (4) uses the maximum score among all MB subspaces as its outlier score. In addition, we can also use the aggregation score and the average score among all subspaces, respectively. But the problem is that both the aggregation score and the average score between data objects will become more and more alike as the number of MB subspaces increases and, thus, may lead to undesirable results. Thus, based on (4), we select the top-κ data objects as potential outliers. For a given outlier, we characterize this outlier using the MB subspace that maximizes (4).

2) MBOM Algorithm: By (1) and (4), we propose the MBOM algorithm to mine MBB outliers, as shown in Algorithm 1. At Steps 2–4, instead of learning an entire BN from a data set, we use a state-of-the-art HITON-MB algorithm [3] to discover the MB subspace of each attribute in the data set independently (any other sound MB mining methods can be used here). HITON-MB can effectively and efficiently find the MB of a given attribute from a data set with thousands of attributes.

Algorithm 1 MBOM Algorithm

```
   Input: $D$ with $N$ data objects and $M$ attributes
   Output: top $\kappa$ candidate outliers
1  /*Identify MB subspaces*/
2  for $i = 1$ to $M$ do
3      $MB(F_i)$=GetMB($F_i$);
4  end
5  /*Mine MBB outliers in MB subspaces*/
6  for $j = 1$ to $N$ do
7      for $k = 1$ to $M$ do
8          Compute the outlier degree of $D_j$ in $\{F_i \cup MB(F_i)\}$
           by (1);
9      end
10     Compute the outlier score of $D_j$ by (4);
11 end
12 Output top $\kappa$ data objects with highest scores
```

The time complexity of MBOM consists of two parts. Steps 2–4 are to discover the MB subspace of each attribute in D. Given an attribute T, HITON-MB first finds parents and children (PC) of T, then spouses of T. Thus, assuming $\text{PC}(T)$ has the largest size among all attributes in D, and $\lambda^{|\text{PC}(T)|}$ denotes all subsets within $\text{PC}(T)$ whose sizes do not exceed λ, the time complexity of Steps 2–4 is $O(M|\text{PC}(T)|^2\lambda^{|\text{PC}(T)|})$, which is mainly determined by the size of $\text{PC}(T)$ [3]. In practice, the size of $\text{PC}(T)$ is always much smaller than M. The worst complexity is $O(M|\text{PC}(T)|^2 2^{|\text{PC}(T)|})$. Steps 7–12 calculate the LOF scores in each MB subspace, and the time complexity is $O(N^2)$. Since the size of each MB subspace is always much smaller than M, the main computations of MBOM lie in Steps 2–4.

IV. Experiment Results

To evaluate the quality of our approaches, we compare MBOM with HiCS [14], 4S [19], and LOF [6]. In all experiments in this section, both MBOM and LOF are implemented in MATLAB while HiCS and 4S in JAVA. The key parameters of these algorithms are set as follows. For MBOM, HITON-MB sets the significance level to 0.01. For 4S, K is set to $K = M \log N$, where K is the number

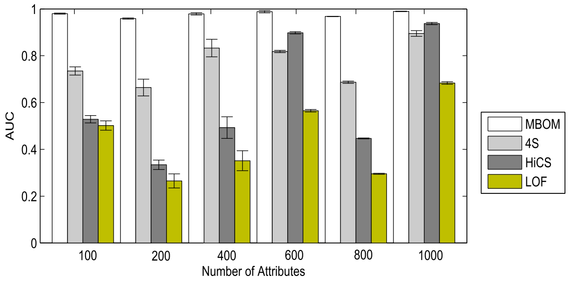

Fig. 2. AUC of MBOM against 4S and HiCS on synthetic data.

of top attribute-pairs with the largest correlations, M is the number of dimensions, and N is the number of data objects [19]. For HiCS, the number of statistical tests m is set to the default value 50. The test statistic size α and the candidate cutoff parameter are set to 0.1 and 400, respectively [14]. MBOM, HiCS, and 4S all use the LOF metric to calculate subspace outlier scores. For LOF, we use Euclidean distance and set parameter ϵ (i.e., the number of nearest neighbors of a data object) to 3.

A. Experiments on Synthetic Data

We generate the synthetic data sets of different sizes and dimensionality for testing MBOM. Each data set contains subspace clusters with dimensionality varying from 4 to 10. We picked 3–6 objects in a subspace and make them to deviate from all clusters in the selected subspace, and thus, this allows an object to be an outlier in multiple subspaces independently. We generate 15–30 outliers in multiple subspaces in a synthetic data set. Using the area under the curve (AUC) metric, we generate data sets of a fixed size of 5000 objects in dimensionality ranging from 100 to 1000. For each dimensionality setting, we generate 10 data sets and calculate the average AUC for each algorithm.

Fig. 2 compares the performance of MBOM against 4S, HiCS, and LOF in AUC using synthetic data. MBOM achieves higher AUC and a lower standard deviation than 4S, LOF, and HiCS. In HiCS, the parameter of candidate cutoff limits the number of candidates in the bottom-up subspace processing, then affects the quality of HiCS. When the dimensionality increases, the number of subspaces becomes huge accordingly. In this case, it is hard to select an appropriate candidate cutoff parameter to guarantee the quality of HiCS. For 4S, it uses the parameter K (i.e., K pairs of attributes with the largest correlations) to trade off completeness in search for efficiency. When the dimensionality increases, it is also hard to set a proper value of K for 4S. MBOM reduces the number of possible subspaces from exponential to linear with respect to the dimensionality and, thus, does not require additional parameters to trade off completeness of subspace search for efficiency. Finally, since LOF was not designed for subspace outlier detection, it achieves the worst detection accuracy. In summary, using synthetic data, MBOM achieves higher performance in AUC.

Since LOF, HiCS, 4S, and MBOM were implemented in the different environments (i.e., MATLAB or JAVA), instead of reporting the running time of these algorithms in synthetic data, we summarize the time complexity of LOF, HiCS, 4S, and MBOM (for MBOM, assuming $|\mathrm{PC}(T)| = \psi$) in Table I. The worst complexity of LOF is $O(N^2)$ and its average complexity is $O(N \log N))$ with a well-established index structure [6]. Assuming $K = M \log N$ (i.e., the number of top attribute-pairs with the largest correlations), the worst complexity of 4S is $O(\zeta(M - \log N)^3 N)$, where ζ is the largest size of subspace and the average complexity is $O(MN)$ [19]. With an *a priori*-like approach, HiCS starts with 2-D subspaces in a levelwise manner. Assume that HiCS reaches level h, its worst complexity is $O(N^2 \sum_{i=2}^{h}(M/i))$ and the average complexity is $O(N^2 M^2)$ when it only selects top ξ subspaces to generate new candidates at each level [14]. Table I shows that LOF may be the fastest algorithm, whereas HiCS could be slowest one. For computational costs, 4S is determined by the user-defined parameter K, whereas MBOM is determined by ψ (ψ is determined by D and always much smaller than M in D). Thus, we cannot tell which of them is more efficient although the complexity of MBOM and 4S seems to be competitive.

TABLE I
TIME COMPLEXITY OF LOF, HiCS, 4S, AND MBOM

Algorithm	Worst case	Average case
LOF	$O(N^2)$	$O(N \log N))$
HiCS	$O(N^2 \sum_{i=2}^{h} \binom{M}{i}))$	$O(N^2 M^2)$
4S	$O(\zeta(M - \log N)^3 N)$	$O(MN)$
MBOM	$O(M\psi^2 \lambda^{\psi})$	$O(M\psi^2 2^{\psi})$

TABLE II
SUMMARY OF REAL-WORLD DATA SETS

Dataset	number of instances	number of dimensions
zoo	101	16
leaf	340	15
breast-cancer	569	30
biodegradation	1,055	41
spectf	267	44
spambase	4,601	57
libras movement	360	90
madelon	2,000	500
isolate	1,560	617

TABLE III
OUTLIERS IN THE ZOO DATA SET

Outlier	Attribute	Outlier detail in MB subspace
platypus	*eggs*	milk $= yes$, egg $= yes$
seasnake	*eggs*	eggs$= no$, milk $= no$, toothed $= yes$
seal	*legs*	hair $= yes$, feathers $= no$, milk $= yes$, legs $= 0$
scorpion	*tail*	backbone$= no$, tail $= yes$
frog	*venomuous*	venomous $= yes$, legs $= 4$
sealion	*fins*	fins $= yes$, legs $= 2$

B. Experiments on Real-World Data

In addition to the synthetic data sets, we use nine real-world data sets from the UCI machine learning repository [5] to evaluate the MBOM, HiCS, and 4S algorithms. The statistics of the data sets are summarized in Table II.

We first use the zoo data set to investigate the relations between the outliers and the MB subspaces since the zoo data set stores information about animals and the outliers identified from it are easily understood. Table III lists the top six outliers in the zoo data set discovered by MBOM and shows the discovered outliers and their corresponding MB subspaces. In Table III, "Attribute" denotes the outlier in the first column found in this attribute's MB subspace. In Table III, for example, the abnormal behaviors associated with attribute *eggs* (animals that lay eggs or not), such as platypus that produces milk but lays eggs ("milk = yes, eggs = yes") and the invertebrate of sea snake that does not lay eggs ("eggs = no, milk = no, toothed = yes"), and so on, are captured using the MB of *eggs*.

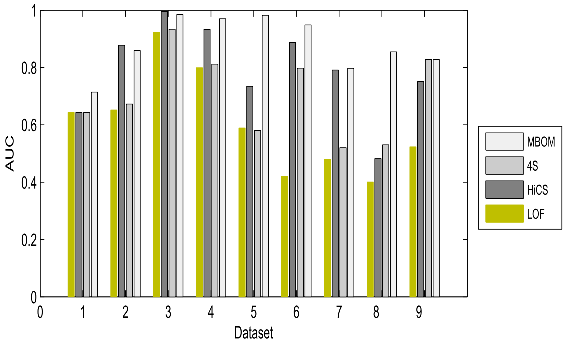

Fig. 3. AUC of MBOM against 4S and HiCS. The labels of the x-axis from 1 to 9 denote the data sets. 1—zoo. 2—leaf. 3—breast-cancer. 4—biodegradation. 5—spambase. 6—spectf. 7—libras movement. 8—madelon. 9—isolate.

For real-world data sets, we do not have a ground truth with explanatory information. To evaluate the algorithms in outlier detection, instead of listing all found outliers, it is a commonly used procedure to consider outlier detection as equivalent to rare class detection, i.e., the rare class as ground truth for the evaluation of the detected outliers, then use the AUC metric to compare outlier algorithms [14], [18]. Recent study illustrates that outliers are bound to be rare, but data objects in a rare class are not necessarily outliers [27] a. For example, in the zoo data set, "platypus," "sea lion," and "seal" are outliers, but they belong to the majority class in the zoo data set. Following the data preprocessing principle in [7] and [27], the data sets in Table II are preprocessed as follows for outlier detection.

The data sets in Table II are preprocessed as follows. On the leaf data set, we set the samples of class of "36" as outliers. On the breast-cancer data set, we randomly select 5% objects from the class of "1" as outliers. On the spectf data set, we sample 30% objects from the class of "0" as outliers. On the madelon data set, all classes have similar population. In this case, we randomly select 2.5% objects from the class of "−1" as outliers. On spambase, 2% samples selected from class of "1" are treated as outliers. On data set isolate, we merge data instances in the classes of "22" and "23" as outliers. On libras movement, we choose 25% samples from class "1" and class "13," respectively, as the outliers. On biodegradation, we downsample 5% samples from class "1" as the outliers. Using the eight data sets, the comparison of MBOM with 4S and HiCS will be conducted as follows.

Fig. 3 shows the AUC results of MBOM, HiCS, LOF, and 4S. Except for the leaf and breast-cancer data sets, MBOM is very competitive with HiCS; while using the remaining data sets, the results demonstrate that MBOM achieves better performance than HiCS, 4S, and LOF. Although MBOM, HICS, LOF, and 4S are implemented in different programming languages, we still show their running time in Table IV for observing the computational costs of these algorithms.

Meanwhile, HiCS shows a high variation of running time on the number data objects or attributes of a data set used. The explanation is that HiCS employs the correlation between attributes as its objective function for computing subspace contrasts. A high-contrast subspace is only the subspace that shows high dependencies between attributes in this subspace. But the number of subspaces will exponentially grow if attributes in a data set are highly correlated with each other. In this case, it is high computational for HiCS to use bottom-up subspace principle. For example, the attributes in the spambase data set for classifying email as spam or nonspam are highly correlated. Although the spambase data set only contains 57 attributes, HiCS spends more time on it than on the remaining data sets. 4S alleviates this problem by only choosing the top K pairs of attributes with the largest correlations using pairwise comparison for generating subspaces.

TABLE IV
RUNNING TIME (IN SECONDS)

Dataset	MBOM	4S	HiCS	LOF
zoo	1	1	5	0.02
leaf	1	1	147	0.1
breast-cancer	2	1	209	1
biodegradation	8	4	1,508	1
spambase	213	105	265,229	13
spectf	3	2	723	0.1
libras movement	10	2	101	0.3
madelon	348	810	887	1
isolate	822	83	4,665	2

MBOM outperforms 4S. The 4S algorithm greedily uses the top K pairs of attributes with the largest correlations to generate subspaces and, thus, may miss outliers due to its heuristic and pairwise comparison. Furthermore, the MB subspaces produced by MBOM are more concise than the subspaces found by 4S. Since 4S employs pairwise correlation between attributes to generate subspaces, the subspaces found may contain redundant relationships between attributes. For example, in the subspace $S \subseteq F$ found by 4S, assuming $F_i \in S$ and $F_j \in S$ are correlated with each other, but this relationship between F_i and F_j may be redundant if there exists a subset $Z \subseteq F - \{F_i \cup F_j\}$ to make F_i and F_j independent. Due to only pairwise comparison employed, 4S cannot detect this redundant relation between F_i and F_j. However, those redundant relationships can be removed by MBOM because of the property of MB defined by Theorem 7.

V. Conclusion

In this brief, we propose the notion of MBB outliers. By our new notion, a significant gain is that we reduce the number of possible subspaces from exponential to linear with respect to the dimensionality of an input data set. We propose the MBOM algorithm to mine the MBB outliers. Using synthetic and real-world data, our experiments have validated the effectiveness and efficiency of MBOM. To the best of our knowledge, this is the first attempt to mine outliers in MB subspaces and a lot of work is worth further exploring. For example, in this brief, we only use the original LOF metric to calculate outlier scores in an MB subspace. Any other outlier score metrics are worth further exploring in MB subspaces. Furthermore, it is also an interesting direction to use association rules for mining outliers in MB subspaces to tackle the outlying aspects mining for explaining a data object of interest in high-dimensional data. In summary, we hope the work in this brief provides a new direction for outlier detection in subspaces.

References

[1] C. C. Aggarwal, *Outlier Analysis*. Cham, Switzerland: Springer, 2016.

[2] C. C. Aggarwal and P. S. Yu, "Outlier detection for high dimensional data," *ACM SIGMOD Rec.*, vol. 30, no. 2, pp. 37–46, 2001.

[3] C. F. Aliferis, A. Statnikov, I. Tsamardinos, S. Mani, and X. D. Koutsoukos, "Local causal and Markov blanket induction for causal discovery and feature selection for classification part I: Algorithms and empirical evaluation," *J. Mach. Learn. Res.*, vol. 11, pp. 171–234, Jan. 2010.

[4] F. Angiulli and C. Pizzuti, "Fast outlier detection in high dimensional spaces," in *Proc. ECML PKDD*, 2002, pp. 15–27.

[5] K. Bache and M. Lichman. (2015). *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml

[6] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, “LOF: Identifying density-based local outliers,” *ACM SIGMOD Rec.*, vol. 29, no. 2, pp. 93–104, 2000.

[7] G. O. Campos *et al.*, “On the evaluation of unsupervised outlier detection: Measures, datasets, and an empirical study,” *Data Mining Knowl. Discovery*, vol. 30, no. 4, pp. 891–927, 2016.

[8] L. Cao, M. Wei, D. Yang, and E. A. Rundensteiner, “Online outlier exploration over large datasets,” in *Proc. KDD*, 2015, pp. 89–98.

[9] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” *ACM Comput. Surv.*, vol. 41, no. 3, p. 15, 2009.

[10] J. Chen, S. Sathe, C. Aggarwal, and D. Turaga, “Outlier detection with autoencoder ensembles,” in *Proc. SDM*, 2017, pp. 90–98.

[11] X. H. Dang, I. Assent, R. T. Ng, A. Zimek, and E. Schubert, “Discriminative features for identifying and interpreting outliers,” in *Proc. ICDE*, Mar./Apr. 2014, pp. 88–99.

[12] X. H. Dang, B. Micenková, I. Assent, and R. T. Ng, “Outlier detection with space transformation and spectral analysis,” in *Proc. SDM*, 2013, pp. 225–233.

[13] L. Duan, G. Tang, J. Pei, J. Bailey, A. Campbell, and C. Tang, “Mining outlying aspects on numeric data,” *Data Mining Knowl. Discovery*, vol. 29, no. 5, pp. 1116–1151, 2015.

[14] F. Keller, E. Müller, and K. Böhm, “HiCS: High contrast subspaces for density-based outlier ranking,” in *Proc. ICDE*, Apr. 2012, pp. 1037–1048.

[15] F. Keller, E. Müller, A. Wixler, and K. Böhm, “Flexible and adaptive subspace search for outlier analysis,” in *Proc. CIKM*, 2013, pp. 1381–1390.

[16] E. M. Knorr, R. T. Ng, and V. Tucakov, “Distance-based outliers: Algorithms and applications,” *Very Large Data Bases J.*, vol. 8, nos. 3–4, pp. 237–253, 2000.

[17] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, “Outlier detection in arbitrarily oriented subspaces,” in *Proc. ICDM*, Dec. 2012, pp. 379–388.

[18] E. Müller, M. Schiffer, and T. Seidl, “Adaptive outlierness for subspace outlier ranking,” in *Proc. CIKM*, 2010, pp. 1629–1632.

[19] H. V. Nguyen, E. Müller, and K. Böhm, “A near-linear time subspace search scheme for unsupervised selection of correlated features,” *Big Data Res.*, vol. 1, pp. 37–51, Aug. 2014.

[20] H. V. Nguyen, E. Müller, J. Vreeken, F. Keller, and K. Böhm, “CMI: An information-theoretic contrast measure for enhancing subspace cluster and outlier detection,” in *Proc. SDM*, 2013, pp. 198–206.

[21] G. Pang, H. Xu, L. Cao, and W. Zhao, “Selective value coupling learning for detecting outliers in high-dimensional categorical data,” in *Proc. CIKM*, 2017, pp. 807–816.

[22] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA, USA: Morgan Kaufmann, 2014.

[23] S. Ramaswamy, R. Rastogi, and K. Shim, “Efficient algorithms for mining outliers from large data sets,” *ACM SIGMOD Rec.*, vol. 29, no. 2, pp. 427–438, 2000.

[24] S. Sathe and C. Aggarwal, “LODES: Local density meets spectral outlier detection,” in *Proc. SDM*, 2016, pp. 171–179.

[25] M.-L. Shyu, S.-C. Chen, K. Sarinnapakorn, and L. Chang, “A novel anomaly detection scheme based on principal component classifier,” in *Proc. ICDMW*, 2003, pp. 171–179.

[26] I. Tsamardinos, L. E. Brown, and C. F. Aliferis, “The max-min hill-climbing Bayesian network structure learning algorithm,” *Mach. Learn.*, vol. 65, no. 1, pp. 31–78, 2006.

[27] A. Zimek, M. Gaudet, R. J. G. B. Campello, and J. Sander, “Subsampling for efficient and effective unsupervised outlier detection ensembles,” in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 428–436.

[28] A. Zimek, E. Schubert, and H.-P. Kriegel, “A survey on unsupervised outlier detection in high-dimensional numerical data,” *Statist. Anal. Data Mining*, vol. 5, no. 5, pp. 363–387, Oct. 2012.