Revisiting Attribute Independence Assumption in Probabilistic Unsupervised Anomaly Detection

Sunil Aryal[1,2(✉)], Kai Ming Ting[2], and Gholamreza Haffari[1]

[1] Clayton School of Information Technology, Monash University, Victoria, Australia
{sunil.aryal,gholamreza.haffari}@monash.edu
[2] School of Engineering and Information Technology, Federation University, Victoria, Australia
{sunil.aryal,kaiming.ting}@federation.edu.au

Abstract. In this paper, we revisit the simple probabilistic approach of unsupervised anomaly detection by estimating multivariate probability as a product of univariate probabilities, assuming attributes are generated independently. We show that this simple traditional approach performs competitively to or better than five state-of-the-art unsupervised anomaly detection methods across a wide range of data sets from categorical, numeric or mixed domains. It is arguably the fastest anomaly detector. It is one order of magnitude faster than the fastest state-of-the-art method in high dimensional data sets.

Keywords: Fast anomaly detection · Independence assumption · Big data

1 Introduction

Let database D be a collection of n data instances $(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n)$. Each instance $\mathbf{x}$ is a m-dimensional vector $\langle x_1, x_2, \cdots, x_m \rangle$ where each component is either a numeric attribute $x_i \in \mathcal{R}$ ($\mathcal{R}$ is a real domain) or a categorical attribute $x_i \in \{v_{i_1}, \cdots, v_{i_w}\}$ (where v_{i_j} is a label out of w possible labels for attribute x_i). The problem of anomaly detection is to identify anomalous instances which are significantly different from the majority of instances in the database.

In the literature, anomaly detection has two main approaches [1]: (i) the supervised approach classifies an instance in either anomaly or normal class by using a classification model trained on a labelled training set; (ii) the unsupervised approach trains an anomaly detector from unlabelled training data, and identifies anomalies based on their anomaly scores. In many real-world applications, labelled training data are difficult to obtain; and thus anomalies have to be identified using an unsupervised approach. In this paper, we focus on the unsupervised approach to anomaly detection.

M. Chau et al. (Eds.): PAISI 2016, LNCS 9650, pp. 73–86, 2016.
DOI: 10.1007/978-3-319-31863-9_6

As databases are currently growing rapidly in terms of volume and dimensionality, identifying anomalous patterns in massive databases is a challenging task. Traditional distance or density based unsupervised methods using nearest neighbours such as kNN [2] and LOF [3] are not applicable in large databases because of their high time complexity in the order of $O(n^2m)$. Recently, simpler and more efficient methods such as iForest [4] and the nearest neighbour in a small subsamples (Sp) [5] are proposed for numeric domains. In categorical domains, anomaly detection methods have been largely based on frequent patterns (e.g., FPOF [6] and COMPREX [7]) which do not scale up to high dimensionality and large data size.

In probabilistic approach, instances in low density region are considered as anomalies, i.e., anomalies have a low probability to be generated from the distribution of normal instances. The simplest efficient probabilistic approach of estimating multivariate probability as the product of univariate probabilities has been used for anomaly detection in some domains [1,8]. The intuition behind this Simple univariate Probabilistic Anomaly Detector (we call it SPAD) is that an anomalous instance is significantly different from normal instances in a few attributes where it has low probability.

Despite its simplicity, effectiveness and efficiency, SPAD is not considered as a benchmark to compare the performance of recently proposed efficient unsupervised anomaly detectors [4–7]. They are shown to run orders of magnitude faster than traditional kNN based methods and produce better or competitive detection accuracy. But, it is not clear if they are more effective and efficient than the simplest traditional probabilistic method SPAD.

In this paper, we show that SPAD performs competitively to or better than all the state-of-the-art anomaly detection methods mentioned above in a wide range of 25 data sets from categorical only, numeric only, and mixed domains. It runs one order of magnitude faster than the simplest nearest neighbour anomaly detector Sp [5] in data sets with high dimensionality. In categorical domains, it runs up to five orders of magnitude faster than the existing state-of-the-art categorical based methods [6,7].

The rest of the paper is organised as follows. SPAD and five widely used state-of-the-art anomaly detection methods are discussed in Sect. 2, followed by experimental evaluations in Sect. 3. The parametrized version of SPAD is described in Sect. 4, and the conclusions are provided in the last section.

2 Related Work

In this section, we review six unsupervised anomaly detectors including five widely used state-of-the-art anomaly detection methods and SPAD. The first three are designed primarily for numeric domains and the last three are mainly for categorical domains. The pertinent details of these methods are described in the following six subsections.

2.1 Local Outlier Factor (LOF)

Breunig et al. (2000) [3] proposed a method based on relative density of an instance with respect to its k-neighbourhood. Let $N^k(\mathbf{x})$ be the set of k-nearest neighbours of $\mathbf{x}$, $d(\mathbf{x}, \mathbf{y})$ be the distance between $\mathbf{x}$ and $\mathbf{y}$ and $d^k(\mathbf{x}, D)$ is the distance between $\mathbf{x}$ and its k^{th}-NN in D. The anomaly score of $\mathbf{x}$ is defined as follows:

$$s_{lof}(\mathbf{x}) = \frac{\sum_{\mathbf{y} \in N^k(\mathbf{x})} lrd(\mathbf{y})}{|N^k(\mathbf{x})| \times lrd(\mathbf{x})} \tag{1}$$

where $lrd(\mathbf{x}) = \frac{|N^k(\mathbf{x})|}{\sum_{\mathbf{y} \in N^k(\mathbf{x})} \max(d^k(\mathbf{y}, D), d(\mathbf{x}, \mathbf{y}))}$.

It requires a distance measure to compute all pairwise distances between instances in D. Euclidean distance $\left(d_{euc}(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{a=1}^{m} (x_a - y_a)^2}\right)$ is the most widely used distance measure. In the case of categorical attribute a, $x_a - y_a = 0$ if $x_a = y_a$; and 1 otherwise. An alternative measure in categorical domains advocated by Boriah et al. [9] is Occurrence Frequency (OF): $d_{of}(\mathbf{x}, \mathbf{y}) = \frac{1}{m} \sum_{a=1}^{m} (x_a - y_a)$ where $x_a - y_a = 0$ if $x_a = y_a$; and $x_a - y_a = \log \frac{n}{f(x_a)} \times \log \frac{n}{f(y_a)}$ otherwise; where $f(x_a)$ is the frequency of the categorical label x_a in D.

2.2 Isolation Forest (*i*Forest)

Instead of density, *i*Forest [4] employs an isolation mechanism to isolate every instance in the given training set. This is done efficiently by random axis-parallel partitioning of the data space in a tree structure until every instance is isolated. A set of t trees is constructed, each tree T_i is built using a subsample randomly selected from D. The anomaly score of an instance $\mathbf{x}$ is measured as the average path length over t trees as follows:

$$s_{iforest}(\mathbf{x}) = \frac{1}{t} \sum_{i=1}^{t} \ell_i(\mathbf{x}) \tag{2}$$

where $\ell_i(\mathbf{x})$ is the path length of $\mathbf{x}$ in tree T_i.

The intuition is that anomalies are more susceptible to isolation. Isolation using trees yields that anomalies have shorter average path lengths than normal instances. *i*Forest is designed for numeric domains only. In this paper, we show that it can be effective in categorical and mixed domains as well, by simply converting each categorical label into a binary {0, 1} attributes and treating them as numeric attributes (see the empirical evaluation in Sect. 3).

2.3 Sampling (Sp)

Instead of searching k-nearest neighbour in D, Sugiyama and Borgwardt (2013) [5] proposed to search the nearest neighbour (i.e., $k = 1$) in a small random

subsample $\mathcal{D} \subset D$ to detect anomalies. The anomaly score is the distance to the nearest neighbour in $\mathcal{D}$, defined as follows:

$$s_{sp}(\mathbf{x}) = \min_{\mathbf{y} \in \mathcal{D}} d(\mathbf{x}, \mathbf{y}) \tag{3}$$

They have shown that the method called Sp with a very small subsample ($\psi = 20$) performs better than or competitive to LOF; but it runs several orders of magnitude faster. In [5], Sp is used only in numeric domains with the euclidean distance. In this paper, we evaluated Sp in categorical and mixed domains as well with the euclidean and Occurrence Frequency (OF) distance measures [9].

2.4 Frequent Pattern Outlier Factor (FPOF)

He et al. (2005) [6] proposed an anomaly detection method for categorical domains based on frequent patterns. It uses Apriori algorithm [10] to generate frequent itemsets of maximum size η with minimum support threshold δ in D, denoted as $FPS(D, \eta, \delta)$. The score of an instance $\mathbf{x}$ is estimated as follows:

$$s_{fpof}(\mathbf{x}) = \frac{\sum_{z \subseteq \mathbf{x} \ \wedge \ z \in FPS(D,\eta,\delta)} support(z)}{|FPS(D, \eta, \delta)|} \tag{4}$$

where z is a frequent itemset, $z \subseteq \mathbf{x}$ denotes that z is contained in $\mathbf{x}$, and $support(z)$ is the support of z.

The intuition of Eq. 4 is that $\mathbf{x}$ is more likely to be an anomaly if it has a few or none of the frequent itemsets, i.e., the lower the score, the more likely $\mathbf{x}$ is an anomaly.

2.5 Pattern Based Compression Technique (COMPREX)

Recently, Akoglu et al. (2012) proposed a pattern-based compression technique called COMPREX for anomaly detection in categorical data [7]. It builds a collection of dictionaries (code tables) $CT_1, CT_2, \cdots, CT_k$ which are learnt from data using disjoint subsets of highly correlated features based on information gain. The anomaly score of $\mathbf{x}$ is defined as the cost of encoding $\mathbf{x}$ using the code tables.

$$s_{cmprx}(\mathbf{x}) = \sum_{F \in \mathcal{P}} \sum_{p \subseteq \pi_F(\mathbf{x})} L(code(p)|CT_F) \tag{5}$$

where $\mathcal{P} = \{F_1, F_2, \cdots, F_k\}$ is a set of disjoint partitions of the feature set, p is a pattern, $\pi_F(\mathbf{x})$ is a projection of $\mathbf{x}$ into feature subset F, $code(p)$ is a code word corresponding to p and $L(\cdot)$ is the length of a code word.

The intuition is that the higher the cost of encoding $\mathbf{x}$, the more likely it is to be an anomaly. Even though it produces better detection accuracy than other categorical methods, it is limited to low dimensional data sets because of its high time complexity.

2.6 Simple Probabilistic Method (SPAD)

In probabilistic approach, instance $\mathbf{x}$ is an anomaly if the probability of $\mathbf{x}$, $P(\mathbf{x})$, is low. An estimate of $P(\mathbf{x})$ requires a large amount of data (even in a moderate number of dimensions) which is usually infeasible in many applications. Assuming the attributes are independent of each other (e.g. naive Bayes [11]), it can be decomposed as:

$$\hat{P}(\mathbf{x}) = \prod_{i=1}^{m} \hat{P}(x_i) \tag{6}$$

The one-dimensional $\hat{P}(x_i)$ can be estimated from D using the Laplace-corrected estimate as: $\hat{P}(x_i) = \frac{f(x_i)+1}{n+w_i}$, where $f(x_i)$ is the occurrence frequency of x_i in D, and w_i is the number of possible values of x_i. The same estimation can be used in numeric domains by converting numeric attributes into categorical attributes through discretisation[1].

In order to avoid floating point overflow, instances are ranked using logarithm of $P(\mathbf{x})$ in Eq. 6, effectively using the summation of the logarithm of univariate probabilities $P(x_i)$ as:

$$s_{spad}(\mathbf{x}) = \sum_{i=1}^{m} \log \hat{P}(x_i) \tag{7}$$

Instances are ranked in ascending order based on their probabilities — instances having low probability, which are ranked at the top, are likely to be anomalies. Note that SPAD is parameter-free, like COMPREX. Goldstein and Dengel (2012) used a similar score as defined in Eq. 7, though they have called it a histogram-based method and evaluated it using three numeric data sets only [8].

Compared with the commonly used frequent pattern based approach, SPAD shares two common features: they are all based on categorical domains and employ probability or frequency as the basis for detecting anomalies. Yet, SPAD is significantly simpler and requires no search; whereas the frequent pattern approach requires an expensive search.

SPAD is scalable to both high dimension and large data sets as it just needs to store the frequency count of every categorical label which can be done in a single pass of data in $O(nm)$ time and requires $O(mw)$ space (where w is the average number of labels in each dimension). Having constructed the frequency count table (in a preprocessing step), calculating the score of an instance just needs a table look-up which costs $O(m)$. Hence, the total runtime complexity of ranking n instances is $O(nm)$ which is cheaper than iForest (if $m < t \log_2 \psi$ which is generally the case unless the dimensionality of the data is very high in the order of thousands); and it is ψ times faster than Sp.

The time and space complexities of the above six anomaly detectors are presented in Table 1.

[1] Though $\hat{P}(\cdot)$ can be estimated directly in numeric domains, it is a lot easier and faster to do it in categorical domains.

Table 1. Time and space complexities

Anomaly detector	Time complexity	Space complexity
LOF	$O(n^2m)$	$O(nm)$
iForest	$O(nt \log_2 \psi)$	$O(t\psi)$
Sp	$O(nm\psi)$	$O(m\psi)$
FPOF	$O(n2^m)$	$O(2^m)$
COMPREX	$O(nm^2)$	$O(m^2)$
SPAD	$O(nm)$	$O(mw)$

w: The average number of categorical labels in each dimension.

3 Empirical Evaluation

In this section, we present the evaluation results of the six anomaly detection methods discussed in Sect. 2: LOF-L2 (LOF using the euclidean distance), LOF-OF (LOF using the occurrence frequency based distance), iForest, Sp-L2 (Sp with the euclidean distance), Sp-OF (Sp with the occurrence frequency based distance), FPOF, COMPREX and SPAD.

We used 25 benchmark data sets from UCI machine learning data repository [12] with categorical only, numeric only and mixed attributes. The data sets were from different domains ranging from health and medicine, text, email filtering to digit recognition. Many of the data sets used were from classification problems. We converted them to anomaly detection problems by considering some larger classes (1 or more) as normal and some smaller classes (1 or more) as anomalies. In the case where classes were more or less uniformly distributed, anomalies were random samples from classes which are not used as normal. They have different data characteristics in terms of data size ($366 \leq n \leq 5$ million), dimensionality ($5 \leq m \leq 4670$) and anomaly proportion ($0.5\,\% \leq p \leq 35\,\%$). The characteristics of data sets used are given in Table 2.

Some preprocessing is required for some algorithms. For iForest which can handle numeric attributes only, each categorical label was converted into a binary attribute [13] (where 0 represents the absence of the label and 1 represents the presence); and all converted binary attributes are treated as numeric. For algorithms that can handle categorical attributes only, numeric attributes were converted into categorical using a modified version of equal width bins which divides the range $[\mu_j - 3\sigma_j, \mu_j + 3\sigma_j]$ into b equal width bins (where μ_j and σ_j are the mean and standard deviation of data values in dimension j). This version is used to reduce the distortion due to outliers. This discretisation technique with $b = 5$ was used to discretise numeric attributes for all categorical methods: SPAD, LOF-OF, Sp-OF, FPOF and COMPREX.

Parameters in all the algorithms were set to the suggested values in respective papers. For LOF, k was set to a commonly used value of 10 [3,5]. The parameter ψ in Sp was set to 20 as suggested in [5]. Parameters η and δ in FPOF were set

Table 2. Characteristics of data sets. n: data size, m: # attributes, m_{num}: # numeric attributes, m_{cat}: # categorical attributes

Name	n	m	m_{num}	m_{cat}	anomaly %
Chess	4580	6	0	6	0.5
Nursery	4648	8	0	8	7
Solar	1066	9	0	9	1
Mushroom	4429	22	0	22	5
Dermatology	366	33	0	33	5.5
Reuters	4297	4134	0	4134	9
Newsgroup	5668	4670	0	4670	14
Advertise	3279	1558	3	1555	14
Arrhythmia	452	279	206	73	14.5
Kddcup99	64759	41	34	7	6.5
U2r	60821	41	34	7	0.5
Census	299285	40	7	33	6
Hypothyroid	3772	29	7	22	7.5
Sick	3772	27	6	21	6
Annthyroid	7200	21	6	15	7.5
Lymph	148	18	3	15	4
Covertype	287128	12	10	2	1
Linkage	5749132	7	2	5	0.5
Breastw	683	9	9	0	35
Spambase	2964	57	57	0	6
Mnist	20444	96	96	0	3.5
Har	7032	561	561	0	20
Secom	1567	590	590	0	6
Isolete	730	617	617	0	1.5
Mfeat	410	649	649	0	2.5

to 5 and 0.1, respectively, as suggested in [6]. Parameters $t = 100$ and $\psi = 256$ were used in iForest, as suggested in [4].

All the algorithms were implemented in JAVA using the WEKA [13] platform, except COMPREX for which we used the MATLAB implementation provided by the authors [7][2]. All the experiments were conducted in a Linux machine with 2.27 GHz processor and 8 GB memory.

We used area under the receiver operating curve (AUC) as the measure of anomaly detection performance. AUC = 1 if all the anomalies are at the top of the ranked list; and a randomly ranked list will yield AUC = 0.5. We conducted

[2] http://www3.cs.stonybrook.edu/~leman/pubs.html.

10 runs for each of the randomised methods: *i*Forest, Sp-L2 and Sp-OF; and reported the average AUC (and time). A significance test was conducted using confidence interval based on two standard errors over 10 runs. A win or loss between two algorithms is counted only if the difference is significant; otherwise it is a draw.

The AUC of all methods in the 25 data sets is provided in Table 3. Note that FPOF and COMPREX did not complete in data sets with high number of dimensions and/or large data size in 24 h.

The overall performance is summarised using the average rank, shown in the bottom three rows in Table 3. SPAD is the best method having an average rank of 1.8; and the closest contenders are COMPREX which has a rank of 2.3 (based on the result of 15 data sets only), and *i*Forest which has a rank of 2.9.

In the last row of Table 3, the pairwise win:loss:draw (w:l:d) counts of contenders against SPAD clearly shows that LOF-L2, LOF-OF, FPOF, *i*Forest, Sp-L2 and Sp-OF produced significantly worse AUC than SPAD (i.e., they have more losses than wins). COMPREX produced competitive detection accuracy in comparison with SPAD, where they have the same number of wins and losses on 15 data sets in which COMPREX completed.

SPAD produced the best (or equivalent to the best) AUC in 13 data sets followed by *i*Forest (7), COMPREX (5), Sp-OF (3), Sp-L2 (2), FPOF (2), LOF-L2 (1) and LOF-OF (0). Note that SPAD is one of the top three performers in almost every data set we have used, shown in the second last row in Table 3. The only exception is Mnist, where SPAD is the fourth best performer. The two closest contenders are *i*Forest and COMPREX which are one of the top three performers in 17 out of 25 data sets and 11 out of 15 data sets, respectively.

The total runtime, including preprocessing (discretisation or nominal to binary conversion), model building (if required), ranking n instances and computing AUC, is provided in Table 4. Note that the direct comparison of the runtime of COMPREX with other methods is not fair as it was implemented in MATLAB and others were implemented in JAVA. It is included in the table to provide an idea about its runtime.

SPAD was faster than all the contenders in all data sets, except in a few small data sets where it ran slight slower than Sp. Note that SPAD ran one order of magnitude faster than its closest contender Sp and *i*Forest in the two highest dimensional data sets (Reuters and Newsgroup) which have more than 4000 attributes; and SPAD ran one to five orders of magnitude faster than LOF, FPOF and COMPREX in all data sets. The only exception is Lymph, compared with LOF-L2. Note that the runtime of LOF-L2 presented here are without using indexing scheme to expedite the k nearest neighbour search [2,14,15]. We did not use indexing because they are not effective as the number of dimensions increases and they do not work with non-metric distance such as OF.

It is interesting to note the longest time a method took in all these data sets: SPAD took less than 20 s to complete in Linkage, the largest data set having more than 5 million instances. Sp and *i*Forest took about 50 s and 256 s, respectively, in the same data set. Ignoring the data sets in which they could not complete

Table 3. Anomaly detection performance (AUC). The average rank, the number of data sets on which a method is among the top three performers, and win:loss:draw (w:l:d) counts of a method against SPAD are included in the last three rows.

Data set	LOF-L2	LOF-OF	FPOF	CMPRX	*i*Forest	Sp-L2	Sp-OF	SPAD
Chess	0.890	0.730	0.912	**0.995**	0.945	0.707	0.901	0.994
Nursery	0.446	0.365	**1.000**	**1.000**	**1.000**	0.759	0.480	**1.000**
Solar	0.588	0.651	0.979	0.968	0.943	0.801	0.855	∗**0.980**
Mushroom	0.371	0.709	0.922	**0.987**	0.907	0.892	0.892	0.936
Dermatology	**0.884**	0.418	0.641	0.810	0.726	0.638	0.397	0.727
Reuters	0.793	0.684	○	•	0.861	0.865	0.802	∗**0.883**
Newsgroup	0.670	0.289	○	•	0.684	0.708	0.635	∗**0.735**
Advertise	0.597	0.547	○	•	0.689	**0.710**	0.673	**0.704**
Arrhythmia	0.729	0.798	○	•	0.803	0.755	0.771	∗**0.813**
Kddcup99	0.553	0.804	0.997	0.943	**0.998**	0.925	0.920	0.996
U2r	0.626	0.799	•	0.968	0.986	0.985	0.943	∗**0.990**
Census	•	•	•	•	0.623	0.627	**0.724**	0.676
Hypothyroid	0.603	0.541	0.631	**0.688**	0.562	0.509	0.616	0.667
Sick	0.596	0.490	0.564	0.621	0.523	0.471	**0.641**	0.602
Annthyroid	0.669	0.599	0.688	0.696	0.632	0.503	0.603	∗**0.697**
Lymph	0.994	0.953	0.979	0.996	0.995	0.847	0.833	∗**0.997**
Covertype	0.520	0.479	0.838	•	**0.977**	0.930	0.925	**0.974**
Linkage	•	•	0.910	0.972	**1.000**	0.998	0.898	0.993
Breastw	0.435	0.822	**0.991**	0.990	0.987	0.952	0.971	0.990
Spambase	0.653	0.649	○	**0.787**	0.785	0.658	0.717	0.770
Mnist	0.801	0.706	0.635	0.799	**0.835**	0.807	0.798	0.799
Har	0.318	0.998	○	•	0.988	0.922	0.996	∗**1.000**
Secom	0.533	0.553	○	•	0.537	0.534	0.548	∗**0.562**
Isolete	0.835	0.983	○	•	**1.000**	**1.000**	**1.000**	**1.000**
Mfeat	0.755	0.722	○	•	**0.947**	0.834	0.848	0.938
Avg. Rank	5.2	5.7	3.6	2.3	2.9	4.4	4.3	1.8
#Top3	3/23	3/23	7/14	11/15	17/25	10/25	7/25	24/25
w:l:d	2:21:0	0:23:0	2:11:1	6:6:3	5:16:4	2:21:2	2:21:2	–

Bold face: The best or equivalent to the best AUC.
∗: Significantly better detection performance of SPAD over all other contenders.
•: Did not complete in 24 h.
○: Did not complete due to insufficient memory with 8 GB.

Table 4. Total runtime (in seconds).

Data set	LOF-L2	LOF-OF	FPOF	CMPRX	*i*Forest	Sp-L2	Sp-OF	SPAD
Chess	5.14	14.69	0.49	148.61	0.59	0.12	0.24	0.13
Nursery	3.23	15.29	0.52	54.15	0.51	0.12	0.21	0.10
Solar	0.24	0.51	0.35	20.20	0.40	0.07	0.12	0.04
Mushroom	5.07	16.07	100.79	363.59	0.56	0.14	0.30	0.08
Dermatology	0.33	0.48	1848.42	128.62	0.66	0.05	0.06	0.03
Reuters	2198.30	1879.72	○	•	49.85	11.11	19.75	2.25
Newsgroup	5520.84	5126.25	○	•	76.33	31.68	43.98	5.26
Advertise	193.66	177.44	○	•	2.03	1.53	2.06	0.93
Arrhythmia	1.35	3.56	○	•	0.73	0.07	0.25	0.19
Kddcup99	3527.47	2419.10	79488.06	6272.76	4.47	1.10	1.91	0.86
U2r	1448.58	1987.84	•	4342.38	4.34	1.02	1.71	0.85
Census	•	•	•	•	25.84	5.60	13.29	3.40
Hypothyroid	9.57	15.93	1536.38	297.69	0.73	0.14	0.27	0.17
Sick	4.56	8.60	1076.37	149.14	0.66	0.14	0.26	0.15
Annthyroid	20.00	34.86	167.05	204.02	0.73	0.16	0.31	0.14
Lymph	0.08	0.24	1.79	28.03	0.21	0.02	0.03	0.02
Covertype	40586.59	85501.06	14.23	•	15.03	1.71	6.34	2.22
Linkage	•	•	137.87	43976.78	255.80	19.76	49.11	17.58
Breastw	0.54	0.72	0.37	28.45	0.39	0.03	0.07	0.09
Spambase	3.94	8.48	○	950.53	0.66	0.12	0.34	0.13
Mnist	366.48	2964.48	6072.35	13174.59	2.00	0.65	3.20	0.65
Har	398.22	1383.15	○	•	1.39	1.30	4.52	0.78
Secom	10.16	42.24	○	•	1.03	0.36	1.40	0.26
Isolete	4.61	22.90	○	•	1.25	0.18	0.90	0.14
Mfeat	1.75	7.33	○	•	1.33	0.11	0.55	0.12
Avg. Rank	5.0	5.8	5.9	7.0	4.1	1.6	2.8	1.4

Bold face: The best or equivalent to the best AUC.
∗: Significantly better detection performance of SPAD over all other contenders.
•: Did not complete in 24 h or 86400 s.
○: Did not complete due to insufficient memory with 8 GB.

in the experiments, LOF took the longest, close to one day, in Covertype which has less than 300000 instances; COMPREX took about half a day in Linkage; and FPOF took about 22 h in Kddcup99 which has 41 attributes and less than 65000 instances. Note also that COMPREX and FPOF would take more than a day to complete even for small data sets such as Isolete, Mfeat and Arrhythmia which have less than 1000 instances and 700 attributes.

4 Detecting Anomalies Which Rely on Multiple Attributes

Despite its strong assumption of attribute independence, SPAD produced superior performance than other state-of-the-art anomaly detectors in many data sets. But, it has a limitation in detecting anomalies which rely on multiple attributes as it does not capture the relationship between attributes. An example is shown in Fig. 1 where anomalies are not different from normal instances in any single attribute; but they exhibit outlierness only if both attributes are examined.

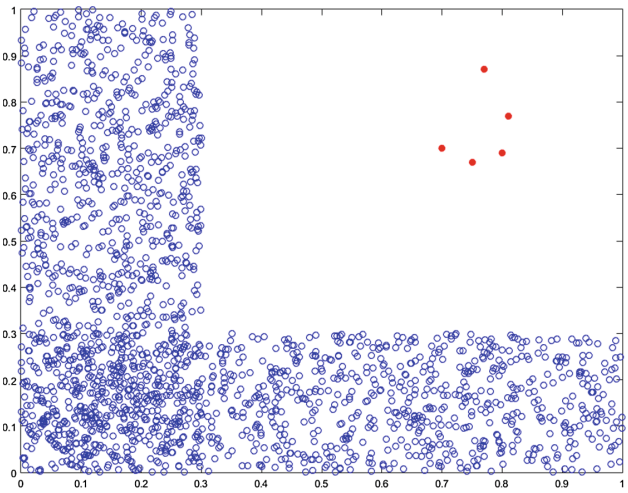

Fig. 1. An example where SPAD fails.

In order to handle such a situation, we propose a parametrized model called SPAD$_r$, in which data in the original m-dimensional space $\mathcal{X}$ are embedded into a new s-dimensional space $\mathcal{Z}$. Each dimension (or attribute) in $\mathcal{Z}$ is a product set of r dimensions (or attributes) in $\mathcal{X}$. The instances can be ranked using Eq. 7 in the new space $\mathcal{Z}$. In the new space, we are able to inspect data distribution along various combinations of the original attributes to identify those outliers which are otherwise difficult to identify.

To guarantee that all possible r attributes are represented in $\mathcal{Z}$, a set S of $\binom{m}{r}$ combinations is required, which is intractable even for moderate values of m and r. Instead, we randomly generate a subset of attribute combinations $\mathcal{S} \subset S$ ($|\mathcal{S}| = s \ll \binom{m}{r}$) as follows: A random order of m attributes (in $\mathcal{X}$) is generated. Then m sets of new attributes are formed from the ordered attributes through a moving window of size r (the attributes ordering is considered as a circular sequence). As an example, for $r = 2$ and a random ordering $[x_3, x_1, x_2]$ of three attributes in $\mathcal{X}$, three new attributes (z_1, z_2, z_3) are formed as $[(x_3 \wedge x_1), (x_1 \wedge x_2),$

$(x_2 \wedge x_3)]$. The above is repeated t times to produce $s = mt$ new attributes, where t is chosen such that $s \ll \binom{m}{r}$. Note that the simplest version with $r = 1$ and $t = 1$ is the SPAD used in Sect. 3.

SPAD$_r$ has the advantage that the size of $\mathcal{S}$ is linear in t and m; and each attribute is used exactly r times. The time complexity for SPAD$_r$ is $O(nmtr)$ and the space complexity is $O(mtw^r)$ since each attribute in $\mathcal{Z}$ has w^r labels.

If anomaly $\mathbf{x}$ relies on r attributes to be identified, and as long as one of the attributes in $\mathcal{Z}$ represents the r attributes in $\mathcal{X}$, then it can be detected by SPAD$_r$ because it has low probability in that particular attribute in $\mathcal{Z}$.

In a data set, the appropriate setting of r in SPAD$_r$ depends on the number of attributes required to detect anomalies. Figure 2 shows three examples where $r = 1$ is enough in Sick; $r = 10$ is required in Mushroom; and $r = 5$ is the best in Mfeat. Though we have used $t = 1$ in these examples, we found that in some cases, $t > 1$ can further improve the AUC of SPAD$_r$.

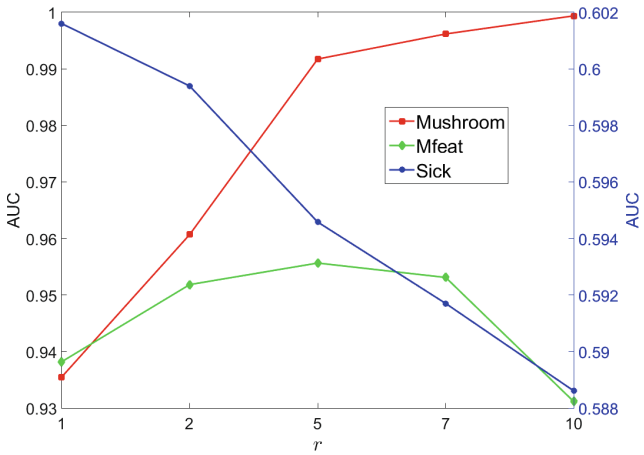

Fig. 2. AUC of SPAD$_r$ w.r.t r ($t = 1$) (The curve of Sick is using the right y-axis) (Color figure online).

One can see some similarity of SPAD$_r$ with subspace anomaly detection methods such as HICS [16]. The fundamental difference is that SPAD$_r$ avoids the expensive search in HICS to find subspaces by considering random subspaces; and SPAD$_r$ runs significantly faster than those complex subspace search based methods.

5 Conclusions

We show that the simple parameter-free anomaly detection method based on univariate probabilities is the fastest method and works as effective as, if not better

than, five state-of-the-art anomaly detection methods. Its anomaly detection performance is consistent across different domains – categorical only, numeric only and mixed domains. It is the method of choice in big data and data streams, as far as we know.

The parametrized version can be used to further improve the detection performance of the parameter-free version in data sets where the detection of anomalies relies on multiple attributes.

Acknowledgments. We would like to thank Prof. Takashi Washio for providing very useful comments and suggestions. We are thankful to the anonymous reviewers for their critical comments to improve the quality of the paper.

References

1. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: a survey. ACM Comput. Surv. **41**(3), 15: 1–15: 58 (2009)
2. Ramaswamy, S., Rastogi, R., Shim, K.: Efficient algorithms for mining outliers from large data sets. In: Proceedings of the 2000 ACM SIGMOD Conference on Management of Data, pp. 427–438 (2000)
3. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: LOF: identifying density-based local outliers. In: Proceedings of ACM SIGMOD Conference on Management of Data, pp. 93–104 (2000)
4. Liu, F., Ting, K.M., Zhou, Z.H.: Isolation forest. In: Proceedings of the Eighth IEEE International Conference on Data Mining, (ICDM), pp. 413–422 (2008)
5. Sugiyama, M., Borgwardt, K.M.: Rapid distance-based outlier detection via sampling. In: Proceedings of the 27th Annual Conference on Neural Information Processing Systems, Lake Tahoe, Nevada, United States, pp. 467–475 (2013)
6. He, Z., Xu, X., Huang, J.Z., Deng, S.: FP-outlier: frequent pattern based outlier detection. Comput. Sci. Inf. Syst. **2**(1), 103–118 (2005)
7. Akoglu, L., Tong, H., Vreeken, J., Faloutsos, C.: Fast and reliable anomaly detection in categorical data. In: Proceedings of the 21st ACM Conference on Information and Knowledge Management (CIKM), pp. 415–424 (2012)
8. Goldstein, M., Dengel, A.: Histogram-based outlier score (hbos): a fast unsupervised anomaly detection algorithm. In: Proceedings of the 35th German Conference on Artificial Intelligence (KI-2012), pp. 59–63 (2012)
9. Chandola, V., Boriah, S., Kumar, V.: Similarity measures for categorical data: a comparative study. Technical report TR 07–022, Department of Computer Science and Engineering, University of Minnesota, USA (2007)
10. Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. In: Proceedings of the 1993 ACM SIGMOD Conference on Management of Data, pp. 207–216 (1993)
11. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley-Interscience, New York (2000)
12. Bache, K., Lichman, M.: UCI machine learning repository, University of California, Irvine, School of Information and Computer Sciences (2013). http://archive.ics.uci.edu/ml
13. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. SIGKDD Explor. Newslett. **11**(1), 10–18 (2009)

14. Bay, S.D., Schwabacher, M.: Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In: Proceedings of the Ninth ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 29–38 (2003)
15. Beygelzimer, A., Kakade, S., Langford, J.: Cover trees for nearest neighbor. In: Proceedings of the 23rd International Conference on Machine Learning, pp. 97–104 (2006)
16. Keller, F., Mller, E., Bhm, K.: HiCS: high contrast subspaces for density-based outlier ranking. In: Proceedings of ICDE, pp. 1037–1048. IEEE Computer Society (2012)