Detecting Outliers in High-Dimensional Datasets with Mixed Attributes

A. Koufakou[1], **M. Georgiopoulos**[1], **and G.C. Anagnostopoulos**[2]
[1] School of EECS, University of Central Florida, Orlando, FL, USA
[2] Dept. of ECE, Florida Institute of Technology, Melbourne, FL, USA

Abstract - *Outlier Detection has attracted substantial attention in many applications and research areas. Examples include detection of network intrusions or credit card fraud. Many of the existing approaches are based on pair-wise distances among all points in the dataset. These approaches cannot easily extend to current datasets that usually contain a mix of categorical and continuous attributes, and may be scattered over large geographical areas. In addition, current datasets usually have a large number of dimensions. These datasets tend to be sparse, and traditional concepts such as Euclidean distance or nearest neighbor become unsuitable. We propose ODMAD, a fast outlier detection strategy intended for datasets containing mixed attributes. ODMAD takes into consideration the sparseness of the dataset, and is experimentally shown to be highly scalable with the number of points and number of attributes in the dataset.*

Keywords: Outlier Detection, Mixed Attribute Datasets, High Dimensional Data, Large Datasets.

1 Introduction

Detecting outliers in data is a research field with many applications, such as credit card fraud detection [1], or discovering criminal activities in electronic commerce. Outlier detection approaches focus on detecting patterns that occur infrequently in the dataset, versus traditional data mining strategies that attempt to find regular or frequent patterns. One of the most widely accepted definitions of an outlier pattern is provided by Hawkins [2]: "An outlier is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism".

Most of the existing research efforts in outlier detection have focused on datasets with a specific attribute type, and assume that attributes are only numerical and/or ordinal, or only categorical. In the case of data with categorical attributes, techniques which assume numerical data need to first map the categorical values to numerical values, a task which is not a straightforward process (e.g., the mapping of a marital status attribute (married or single) to a numerical attribute). In the case of continuous attributes, algorithms designed for categorical data might use discretization techniques to map intervals of continuous space into discrete values, which can lead to loss of information.

A second issue is that many applications for mining outliers require the mining of very large datasets (e.g. terabyte-scale data). This leads to the need for outlier detection algorithms which must scale well with the size and dimensionality of the dataset. Data may also be scattered across various geographical areas, which implies that transferring data to a central location and then detecting outliers is impractical, due to the size of the data, as well as data ownership and control issues. Thus, the algorithms designed to detect outliers must minimize the number of data scans, as well as the need for excessive communication and required synchronization.

A third issue is the high dimensionality of currently available data. Due to the large number of dimensions, the dataset becomes sparse, and in this type of setting, traditional concepts such as Euclidean distance between points, and nearest neighbor, become irrelevant [3]. Employing similarity measures that can handle sparse data becomes imperative. Also, inspecting several, smaller "views" of the data can help uncover outliers, which would otherwise be "masked" by other outliers if one were to look at the entire dataset at once.

In this paper, we extend our work in [4] and we propose an outlier detection approach for datasets that contain both categorical and continuous attributes. Our method, Outlier Detection for Mixed Attribute Datasets (*ODMAD*), uses an anomaly score based on the categorical values of each data point. ODMAD then uses this score to find similarities among the points in the sparse continuous space. ODMAD is fast, efficiently handles sparse data, relies on minimal data scans, and lends itself to large and geographically distributed data.

The organization of this paper is as follows: Section 2 contains an overview of previous research in outlier detection. In Section 3, we present our outlier detection approach, ODMAD. Section 4 includes our experimental results, followed by our conclusions in Section 5.

2 Previous Work

The existing outlier detection work can be categorized as follows. *Statistical-model based* methods assume that a specific model describes the distribution of the data [5], which has the problem of obtaining the suitable model for each particular dataset and application [6]. *Distance-based* approaches (e.g. [7]) essentially compute distances among data points, thus become quickly impractical for large datasets (e.g., a nearest neighbor method has quadratic complexity with respect to the number of dataset points). Bay and Schwabacher [8] propose a distance-based method based on randomization and pruning and claim its complexity is close to linear in practice. Distance-based methods require data to be in the same location or large amounts of data to be transferred from different locations, which makes them impractical for distributed data. *Clustering* techniques can also be employed to first cluster the data, so that points that do not belong in the formed clusters are designated as outliers. However, these methods are focused on optimizing clustering rather than finding outliers [7]. *Density-based* methods estimate the density distribution of the data and identify outliers as those lying in relatively low-density regions (e.g. [9]). Although these methods are able to detect outliers not discovered by the distance-based methods, they become challenging for sparse high-dimensional data [10]. Other outlier detection efforts rely on Support Vector methods [11], Replicator Neural Networks [12], or using a relative degree of density with respect only to a few fixed reference points [13].

Most of the aforementioned techniques are geared towards numerical data and thus are more appropriate for numerical datasets or ordinal data that can be easily mapped to numerical values [14]. Another limitation of previous methods is the lack of scalability with respect to number of points and/or dimensionality of the dataset. Outlier detection techniques for categorical datasets have recently appeared in the literature (e.g. [15]). In [4], we experimented with a number of representative outlier detection approaches for categorical data, and proposed AVF (Attribute Value Frequency), a simple, fast, and scalable method for categorical sets. Otey et al. [6] presented a distributed and dynamic outlier detection method for mixed attribute datasets that has linear runtime with respect to the number of data points; however their runtime is exponential in the number of categorical attributes and quadratic in the number of numerical attributes. Regarding sparseness in high-dimensional data, Ertoz et al. [3] use the cosine function for document clustering. They construct a shared nearest neighbor (SNN) graph, and then cluster together high-dimensional points based on their shared nearest neighbors.

In this paper, we extend our previous work in [4] and propose *Outlier Detection for Mixed Attribute Datasets* (*ODMAD*), an outlier detection approach for sparse data with both categorical and continuous attributes. ODMAD exhibits very good accuracy and performance, it is highly scalable with the number of points and dimensionality of the dataset, and can be easily applied to distributed data. We compare ODMAD with the technique in [6] which is the existing outlier detection approach for distributed, mixed attribute datasets.

3 ODMAD Algorithm

The outlier detection proposed in this paper, ODMAD, detects outliers based on the assumption that outliers are points with highly irregular or infrequent values. In [4], we showed how this idea could be used to effectively detect outliers in categorical data. ODMAD extends the work in [4] to explore outliers in the categorical and in the continuous space of attributes. In this mixed attribute space, an outlier can have irregular categorical values only (*type a*), or irregular continuous values only (*type b*), or both (*type c*). The algorithmic steps of *ODMAD* are below:

In the *first* step, we inspect the categorical space in order to detect data points with irregular categorical values. This enables us to detect outliers of *type a* and *type c*.

In the *second* step, we 'set aside' the points found as irregular from the first step, and focus on the remaining points, in an attempt to detect the rest of the outliers (type *b*). Based on the categorical values of the remaining points, we concentrate on subsets extracted from the data, and work only on these subsets, one at a time. These subsets are considered so that we can identify outliers that would have otherwise been missed (*masked*) by more irregular outliers. To illustrate our point, consider the scenario in Figure 1. Outlier point O_2 is irregular with respect to the rest of the data points, while the second outlier, O_1, is closer to the normal points. In this case, outlier point O_2 masks the other outlier point, O_1. One solution to this problem could be to sequentially remove outliers. This implies several data scans, which is impractical for large or distributed data. In Section 3.3, we explain in more detail how we address this issue by considering subsets of the data.

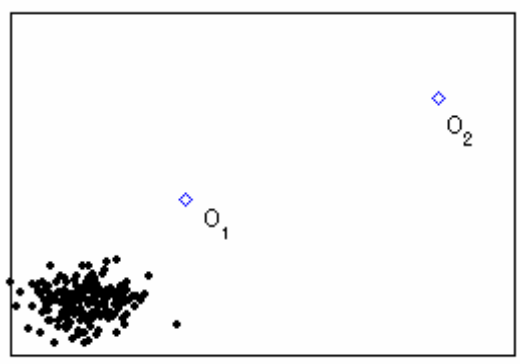

Figure 1: *Masking Effect* - Outlier O_2 is more irregular than normal points and outlier O_1, therefore O_2 will likely mask O_1.

3.1 Categorical Score

As shown in [4], the 'ideal' outlier in a categorical dataset is one for which each and every value of its values is extremely irregular (or infrequent). The *infrequent-ness* of an attribute value can be measured by computing the number of times this value is assumed by the corresponding attribute in the dataset. In [4] we assigned a score to each data point in the dataset that reflects the frequency with which each attribute value of the point occurs. In this paper, we extend this notion of '*outlierness'* to cover the likely scenario where none of the single values in an outlier point are infrequent, but the co-occurrence of two or more of its attribute values is infrequent.

We consider a dataset D with n data points, $\mathbf{x}_i$, $i = 1..n$. If each point $\mathbf{x}_i$ has m_c categorical attributes, we write $\mathbf{x}_i = [\ \mathbf{x}_{i1}\ ,\ \ldots,\ \mathbf{x}_{il}\ ,\ldots,\ \mathbf{x}_{imc}\ ]$, where $\mathbf{x}_{il}$ is the value of the l-th attribute of $\mathbf{x}_i$. Our anomaly score for each point makes use of the idea of an itemset (or set) from the frequent itemset mining literature [16]. Let I be the set of all possible combinations of attributes and their values in dataset D. Let S be a set of all sets d such that an attribute occurs only once in each set d:

$$S = \{d :\ d \in \text{power set } (I)\ \wedge\ \forall\ \ l, k \in d,\ \ l \neq k\}$$

where l and k represent attributes whose values appear in set d. We also define the length of d, $|d|$, as the number of attribute values in d, and the *frequency* or *support* of set d as $f(d)$, which is the number of points $\mathbf{x}_i$ in dataset D which contain set d. Following the reasoning stated earlier, a point is likely to be an outlier if it contains single values or sets of values that are infrequent. We say that a value or a set of values is infrequent if it appears less than *minsup* times in our data, where *minsup* is a user threshold. Therefore, a good indicator to decide if $\mathbf{x}_i$ is an outlier in the categorical attribute space is the score value, $Score_1$, defined below:

$$Score_1(\mathbf{x}_i) = \sum_{\substack{|d| = 1 \\ d \subseteq \mathbf{x}_i \\ f(d) \leq \min sup}}^{MAXLEN} \frac{1}{f(d) \times |d|} \qquad (1)$$

Essentially, we assign an anomaly score to each data point that depends on the infrequent subsets contained in this point. As shown in [6], we obtain a good outlier detection accuracy by only considering sets of length up to a user-entered *MAXLEN*. For example, let point $\mathbf{x}_i = [a\ b\ c]$, and $MAXLEN = 3$, the possible subsets of $\mathbf{x}_i$ are: a, b, c, ab, ac, bc, and abc. If subset d of $\mathbf{x}_i$ is infrequent, i.e. $f(d) \leq minsup$, we increase the score of $\mathbf{x}_i$ by the inverse of $f(d)$ times the length of d. In our example, if $f(ab) = 3$ and $minsup = 5$, ab is an infrequent subset of $\mathbf{x}_i$, and $Score_1$ will increase by $1/(3\times2) = 1/6$.

A higher score implies that it is more likely that the point is an outlier. If a point does not contain any infrequent subsets its score will be zero. $Score_1$ is inversely proportional to the frequency, as well as to the length of each set d that belongs to $\mathbf{x}_i$. Therefore, a point that has very infrequent single values will get a very high score; a point with moderate infrequent single values will get a moderately high score; and a point whose single values are all frequent and has a few infrequent subsets will get a moderately low score.

We note that $Score_1$ is similar to the one in [6]; however the latter does not make any distinction between sets of different frequency. We use the frequency of the sets to further distinguish between points that contain the same number of infrequent values. The benefit of our score becomes pronounced with larger datasets: for example, consider a dataset with a million data points and *minsup* of 10%. Also assume two categorical values: a, that appears only once in the dataset, and b, that appears in the dataset slightly less than a hundred thousand times. Using our score, a data point containing value a (very infrequent) will have a much higher score than a point with value b. Using the score by [6] the two values would add the same to the score. Therefore, our score better reflects the amount of irregularity in the data.

3.2 Continuous Score

Many existing outlier detection methods are based on distances between points in the entire dataset. In addition to the fact that this can be inefficient, especially for large or distributed data, it is very likely that in doing so, the algorithm might miss points which are not globally obvious outliers, but easier to spot if we focus on a subset of our dataset. Furthermore, the notion of a nearest neighbor does not hold as well in high dimensional spaces because the distance between any two data points becomes almost the same [17].

In our case of mixed attribute data, it is reasonable to believe that data points that share a categorical value should also share similar continuous values. Therefore, we can restrict our search space by focusing on points that share a categorical value, and then rank these points based on similarity to each other.

One issue that arises is how to identify similarities between points in high-dimensional data. The most prevalent similarity or distance metric is the Euclidean distance, or the L_2-norm. Even though the Euclidean distance is valuable in relatively small dimensionalities, its usefulness decreases as the dimensionality grows. Let us consider the four points below, taken from the KDDCup 1999 dataset (described in more detail in 4.1): the first two points are normal and the second two points are outliers (we removed the columns that had identical values for all four points). Using Euclidean distance we find correctly that point 1 is closest to point 2 and vice versa, but for points 3 and 4 we find that each is closest in Euclidean distance to point 1, i.e. the two outliers are more similar to a normal point than to each other.

1	0	0	0.002	0.002	0	0	0	0	1	0	0	0.004	0.004	1	0	1	0	0	0	0
2	08.2E-6	8.5E-6	0.02	0.02	0	0	0	0	0.9	0.2	0.2	1	0.9	0.9	0.01	0.04	0	0	0	0
3	0	0	0.002	0.002	0	0	1	1	1	0	0	1	0.004	0	0.5	1	0	0	1	1
4	0	0	0.002	0.002	1	1	0	0	1	0	0	1	0.004	0	0.99	1	1	1	0	0

This is mainly because the Euclidean distance assigns equal importance to attributes with zero values as to attributes with non-zero values. In higher dimensionalities, "the presence of an attribute is typically a lot more important than the absence of an attribute" [3], as the data points in high dimensionalities are often sparse vectors. Cosine similarity is a commonly used similarity metric for clustering in very high-dimensional datasets, e.g. used for document clustering in [3]. The cosine similarity between two vectors is equal to the dot product of the two vectors divided by the individual vector norms. Assuming non-negative values, minimum cosine similarity is 0 (non-similar vectors) and maximum is 1 (identical vectors). In our example with the four points above, the cosine function assigns highest similarity between points 1 and 2, and between points 3 and 4, so it correctly identifies similarity between normal points (points 1 and 2) and between outlier points (points 3 and 4).

In this paper, we used the cosine function to define similarities in the continuous space. Consider a data point $\mathbf{x}_i$ containing m_c categorical values and m_q continuous values. The categorical and continuous parts of $\mathbf{x}_i$ are denoted by $\mathbf{x}_i^c$ and $\mathbf{x}_i^q$ respectively. Let a be one of the categorical values of $\mathbf{x}_i$ which occurs with frequency $f(a)$. We identify a subset of the data that includes the continuous vectors corresponding to all points that share value a: $\{\mathbf{x}_i^q : a \in \mathbf{x}_i^c, i = 1..n\}$, which contains $f(a)$ vectors. The cosine similarity between the mean vector of this set, $\boldsymbol{\mu}_a$, and $\mathbf{x}_i^q$ is below:

$$\cos(\mathbf{x}_i^q, \boldsymbol{\mu}_a) = \sum_{j=1}^{m_q} \frac{\mathbf{x}_{ij}^q}{\left\|\mathbf{x}_{ij}^q\right\|} \times \frac{\mu_{aj}}{\left\|\mu_a\right\|} \qquad (2)$$

where $\|\mathbf{x}\|$ is the L_2-norm of vector $\mathbf{x}$. Finally, we assign the following score to each $\mathbf{x}_i$, for all categorical values a in $\mathbf{x}_i^c$:

$$Score_2(\mathbf{x}_i) = \frac{1}{\left|a \in x_i^c\right|} \sum_{\forall a \in x_i^c} \cos(\mathbf{x}_i^q, \mu_a) \qquad (3)$$

which is the summation of all cosine similarities for all categorical values a divided by the total number of values in the categorical part $\mathbf{x}_i^c$. As minimum cosine similarity is 0 and maximum is 1, the points with similarity close to 0 are more likely to be outliers.

Even though using the cosine similarity helps us better assess distances in a high-dimensional space, its use will not vastly improve our outlier detection accuracy in a large dataset with many different types of outliers. As we noted earlier in this section, we focus on specific subsets of the continuous space so as to identify outliers in smaller settings. In the next sections, we address the issue of having more than one outlier in a subset, and we outline which categorical values we use for $Score_2$ in Eq. (3).

3.3 Improving Accuracy

Many methods (e.g. [7]) assume that outliers are the data points that are irregular in comparison to the rest of the dataset, and that they can be globally detected. However, in many real datasets there are multiple outliers with different characteristics and their irregularity and detection depends on the rest of the outliers against which they are compared. This way, there could be outliers in our dataset that are masked by other, more irregular outliers (see Figure 1). The solution that we propose is to further use the knowledge that we obtain from the categorical scores to help alleviate this issue. Based on Eq. (1), data points with highly infrequent categorical values will have a very high $Score_1$. We can exclude these points with high $Score_1$ from the computation of our continuous score in Equations (2)-(3). The exclusion of these outlier points can be done in the following manner: as we compute the frequencies and means for each categorical value in our dataset, we identify highly infrequent categorical values. Based on this information, we can update the means for the rest of the categorical values that co-occur with the highly infrequent values. The details on how we select the values to exclude from the continuous subsets are given in the following sections.

3.4 Algorithm

ODMAD consists of two phases: the first phase calculates the necessary quantities for the algorithm (categorical values, frequencies, sets, and means); the second phase goes over each point in the dataset and decides if each point is an outlier or not, based on the scores described in Section 3.1 and 3.2. The pseudocode for the two phases is given in Figures 2 and 3, respectively.

As shown in Figure 2, for the score calculation from Eq. (1), we only gather the frequencies of certain sets: the pruned candidates. Pruned candidates are those infrequent sets such that all their subsets are frequent. These are the sets that are pruned at each phase of a Frequent Itemset Mining algorithm such as Apriori [16]. The reason behind this is that as mentioned in section 3.1, we are interested in either single categorical values that are infrequent, or infrequent sets containing single values that are frequent on their own. This makes ODMAD faster as shown in the following example.

Input: ***D*** – dataset (*n* points, m_q and m_c attributes)
minsup, *MAXLEN*
Output: *G* - Pruned Candidates & their frequencies;
A - Categorical values, means & frequencies
foreach point $\mathbf{x}_i$ ($i = 1..n$) **begin**
Add the categorical values of $\mathbf{x}_i$, their frequencies, & their means to *A*;
foreach *len* = 2..MAXLEN **begin**
Create candidate sets and get frequent itemsets;
Add pruned sets & their frequencies to *G* ;
end
end

Figure 2: First Phase of our Outlier Detection Approach *ODMAD*

Input: ***D*** - dataset (*n* points, m_q and m_c attributes)
G, A , *minsup*, *MAXLEN* ,
window, $\Delta score_{c,q}$, *low_sup*, *upper_sup*
Output: *outliers*
foreach point $\mathbf{x}_i$ ($i = 1..n$) **begin**
foreach categorical value *a* in $\mathbf{x}_i^c$ **begin**
If $f(a) < minsup$
$Score_1(\mathbf{x}_i)$ += $1/f(a)$;
end
If $low\_sup < f(a) \leq upper\_sup$
$Score_2(\mathbf{x}_i)$ += $\cos(\mathbf{x}_i^q, \boldsymbol{\mu}_a)$;
end
end
foreach pruned set *d* in *G* found in $\mathbf{x}_i^c$ **begin**
$Score_1(\mathbf{x}_i)$ += $1/(f(d) \times |d|)$;
end
If $Score_1 > \Delta score_c \times$ average $Score_1$ in *window* **or**
$Score_2 < \Delta score_q \times$ average $Score_2$ in *window*
flag($\mathbf{x}_i$) = outlier;
else
normal, add $Score_{1,2}$ to *window* scores;
end
end

Figure 3: Second Phase our Outlier Detection Approach *ODMAD*

Example. Assume we have two points, each with three categorical attributes: $\mathbf{x}_1 = [a\ b\ c]$ and $\mathbf{x}_2 = [a\ b\ d]$. If only single values *a*, *c* are infrequent with frequency equal to 5, the score is as follows:

$$Score_1(\mathbf{x}_1) = 1/f(a) + 1/f(c) = 2/5 = 0.4,$$
$$Score_1(\mathbf{x}_2) = 1/f(a) = 1/5 = 0.2$$

Since *a* and *c* are infrequent, we do not check any of their combinations with other values because they will also be infrequent. The sets we will not check are: *ab*, *ac*, *ad*, *bc*, *cd*. However, *bd* consists of frequent values, *b* and *d*, so we check its frequency. Assuming *bd* is infrequent, and $f(bd) = 4$, we increase the score of $\mathbf{x}_2$:

$$Score_1(\mathbf{x}_2) = 0.2 + 1/(f(bd) \times |bd|) = 0.325.$$

Note that at this point we stop increasing the score of both $\mathbf{x}_1$ and $\mathbf{x}_2$, because there are no more frequent sets. Therefore, in this scenario, we only need to check sets *a*, *c*, and *bd*, instead of all possible sets of length 1 to 3 contained in $\mathbf{x}_1$, $\mathbf{x}_2$. ■

As we identify categorical values and sets, we also update the corresponding mean vectors as discussed in Section 3.3. We use a user-entered frequency threshold, called *low_sup*, to indicate what values we consider 'highly infrequent'; then categorical values with frequency ≤ *low_sup* are 'marked' as 'highly infrequent'. As we described in 3.3, we exclude points that contain these 'highly infrequent' values from the mean in Eq. (2) of all other categorical values they co-occur with. For example: assume point **x** contains categorical values *a*, $f(a) \leq low\_sup$, and value *b*, $f(b) > low\_sup$. We exclude point **x** as follows:

$$\boldsymbol{\mu}_b = \frac{1}{f(b)-1} \times \left(\sum_{i=1, b \in \mathbf{x}_i}^{n} \mathbf{x}_i^q - \mathbf{x}^q \right)$$

In the second phase in Figure 3, we first find all categorical values in each point and update $Score_1$ in Eq. (1) accordingly. We do the same for all the pruned sets contained in the current point. Also, for each categorical value, we compute $Score_2$ using the updated mean we computed in the first phase. The continuous vectors we use are those that correspond to categorical values with frequency in (*low_sup*, *upper_sup*]. If a point has a value with frequency less than *low_sup*, its $Score_2$ will be 1, as it contains a highly infrequent categorical value. If a point has no values with frequency in (*low_sup*, *upper_sup*] it will have a $Score_2$ of 0. By applying a lower bound to the frequency range we exclude values with very infrequent categorical values, and by applying an upper bound we limit the amount of data points to which we assign a score in the continuous domain.

Finally, as we scan and score the data points, we maintain a window of categorical and continuous scores. We also employ a delta value for the detection of abnormal scores: $\Delta score_c$ for the categorical scores and $\Delta score_q$ for the continuous scores. As we go over the points in the second phase, if a point has a score larger (smaller in the case of $Score_2$) than the average score of the previous window of points by the corresponding Δ value, it is flagged as an outlier. Otherwise, the point is normal, and its non-zero scores are added to the window we maintain.

If each of the m_c categorical attributes has an average of *v* distinct values, the complexity upper bound is below:

$$T \leq n \times \left(m_q \times v \times m_c + \sum_{j=1}^{MAXLEN} v^j \binom{m_c}{j} \right)$$
$$= O(n \times m_q \times v + n \times (v m_c)^{m_c})$$

TABLE 1. DETECTION RATE ON THE KDDCUP 1999 TRAINING DATASETS (10% Training Set and Entire Training Set)

	10% training set		Entire Training set	
Attack Type	**ODMAD**	**Otey's**	**ODMAD**	**Otey's**
Back	**50**	**50**	75	**100**
Buffer overflow	**91**	36	**91**	**91**
FTP Write	**75**	**75**	**100**	**100**
Guess password	**100**	**100**	**100**	**100**
Imap	**100**	**100**	**50**	**50**
IP Sweep	**60**	30	**92**	76
Land	83	**100**	**100**	62
Load Module	**100**	40	**100**	80
Multihop	**100**	75	**75**	**75**
Neptune	**100**	67	**100**	90
Nmap	**100**	**100**	**88**	63
Perl	**100**	67	**100**	67
Phf	0	**75**	**0**	**0**
Pod	**75**	50	**87**	53
Port Sweep	**100**	67	**100**	64
Root Kit	**60**	40	**80**	40
Satan	**100**	67	**100**	50
Smurf	**57**	43	**88**	63
Spy	**100**	**100**	**100**	**100**
Teardrop	**100**	44	**100**	11
Warez client	**21**	4	9	**36**
Warez master	**100**	33	67	**100**

TABLE 2. EXECUTION TIME (SECONDS) FOR ODMAD VERSUS OTEY'S APPROACH ON THE KDDCUP 1999 TRAINING DATASETS

	ODMAD	Otey's Approach
10% Training Set	3.7	617.4
Entire Training Set	38	6014.9

Therefore ODMAD scales linearly with the number of data points, n, and with the number of continuous attributes, m_q, but seems to be scaling exponentially with the number of categorical attributes m_c. In practice our algorithm runs faster because we are using only the pruned candidates for the categorical value-based score. Otey's method in [6] has exponential time with respect to categorical attributes, and quadratic with the number of continuous. Moreover, the method in [6] requires a covariance matrix for each possible itemset in the dataset, while our method only requires a vector of length m_q (the mean vector) for each categorical value.

4 Experiments

4.1 Experimental Setup

We implemented our approach and Otey's approach [6] using C++. We ran our experiments on a workstation with a Pentium 4 1.99 GHz processor and 1 GB of RAM. We used the KDDCup 1999 intrusion detection dataset [18] from the UCI repository [19]. This dataset contains records that represent connections to a military computer network and multiple intrusions and attacks by unauthorized users. The raw binary TCP data were processed into features such as *connection duration, protocol type, number of failed logins*, etc. The KDD dataset contains a training set with 4,898,430 data points and a dataset with 10% training data points. There are 33 continuous attributes and 8 categorical attributes. Due to the large number of attacks in these datasets, we preprocess them such that attack points are around 2% of the dataset, and we preserve the proportions of the various attacks. We follow the same concept as in [6]: since network traffic packets tend to occur in bursts for some intrusions, we look at bursts of packets in the data set. Our processed dataset based on the entire training set contains 983,550 instances with 10,769 attack instances, and similarly for the 10% training dataset.

We compare our method with the one proposed in [6] as it is the only existing distributed outlier detection approach for mixed attribute datasets that scales well with the number of data points. We evaluate both algorithms based on two measures: outlier detection accuracy, or the outliers correctly identified by the approach as outliers, and the false positive rate, reflecting the number of normal points erroneously identified as outliers. We also compare the execution time of the two algorithms using the same data.

4.2 Results

The outlier detection accuracy or detection rate reflects how many points we detect correctly as outliers. In the KDDCup set, if we detect one point in a burst of packets as an outlier we mark all points in a burst as outliers, as in [6]. The false positive rate is how many normal points we incorrectly detect as outliers versus total number of normal points.

In Table 1, we depict the detection rate achieved from *ODMAD* versus the approach in [6] (better rates are in bold). In Table 2 we show the execution time in seconds for the two approaches. We used window = 40 for all experiments. We experimented with several values for the Otey's approach parameters, and in Table 1 we present the best results (we used: $\delta = 35$; *minsup* = 50% for the 10% set, and 10% for the entire training set; $\Delta score = 2$). For our approach we used: *upper_sup* = *minsup* = 10%; *low_sup* = 2%; $\Delta score_c = 10$, $\Delta score_q = 1.27$ (10% set); and $\Delta score_c = 10$, $\Delta score_q = 1.18$ (entire training set). As can be seen in Table 1, *ODMAD* has equal or better detection rate than Otey's approach for all but two of the attacks on the 10% training set, and all but three of the attacks for the entire training set. Moreover, the detection rates in Table 1 for the 10% dataset were achieved with a false positive rate of *4.32%* for ODMAD and *6.99*% for Otey's, while the detection rates for the entire training set were achieved with a false positive rate of *7.09*% for ODMAD, and *13.32*% for Otey's. Execution time for our approach is significantly faster as well, e.g. ODMAD processed the KDDCup 10% dataset in *38 seconds* while it took Otey's approach *100 minutes* for the same task. We attribute this mainly to the fact that the method in [6] creates

and checks a covariance for each and every possible set of categorical values, while ODMAD looks at single categorical values and the mean of their continuous counterparts.

We observed similar accuracy and performance for the KDD Test set, and we also conducted experiments to explore how ODMAD's performance varies with respect to the parameters (results not shown here due to space). Detection and false positive rates decrease as $\Delta score$ increases, as it reflects the magnitude of difference between scores in the data. The larger $\Delta score$ is, the higher the score difference needs to be for a point to be an outlier, and ODMAD will return less and less outliers. Also, the overall results indicate that good values for *upper_sup* are close to the value for *minsup*, and for *low_sup* close to 1-3% depending on the dataset size.

5 Conclusions

We proposed Outlier Detection for Mixed Attribute Datasets (*ODMAD*), a fast outlier detection algorithm for mixed attribute data that responds well to sparse high-dimensional data. ODMAD identifies outliers based on categorical attributes first, and then focuses on subsets of data in the continuous space by utilizing information from the categorical attribute space. We experimented with the KDDCup 1999 dataset, a benchmark outlier detection dataset, in order to demonstrate the performance of our approach. We found that ODMAD in most instances exhibits higher outlier detection rates (accuracy) and lower false positive rates, compared to the existing work in the literature [6]. Furthermore, ODMAD relies on two data scans and is considerably faster than the competing work in [6]. Extending our work for distributed data is the focus of our future work.

Acknowledgements: This work was supported in part by NSF grants 0341601, 0647018, 0717674, 0717680, 0647120, 05254209, 0203446.

6 References

[1] Bolton, R.J., Hand, D.J., "Statistical fraud detection: A review", *Statistical Science*, 17, pp. 235–255, 2002.

[2] Hawkins, D. *Identification of Outliers*. Chapman and Hall. 1980.

[3] Ertoz, L., Steinbach, M., Kumar, V. "Finding Clusters of Different Sizes, Shapes, and Densities in Noisy, High Dimensional Data", *Proc. of ACM Int'l Conf. on Knowledge Discovery and Data Mining (KDD)*, 2002.

[4] Koufakou, A., Ortiz, E., Georgiopoulos, M., Anagnostopoulos, G., Reynolds, K., "A Scalable and Efficient Outlier Detection Strategy for Categorical Data", *Int'l Conf. on Tools with Artificial Intelligence ICTAI*, October, 2007.

[5] Barnett, V., Lewis, T. *Outliers in Statistical Data*. John Wiley, 1994.

[6] Otey, M.E., Ghoting, A., Parthasarathy, A., "Fast Distributed Outlier Detection in Mixed-Attribute Data Sets", *Data Mining and Knowledge Discovery*, 2006.

[7] Knorr, E., Ng, R., and Tucakov, V., "Distance-based outliers: Algorithms and applications", *Very Large Databases Journal*, 2000.

[8] Bay, S.D. Schwabacher, M., "Mining distance-based outliers in near linear time with randomization and a simple pruning rule", *Proc. of ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, 2003.

[9] Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J., "LOF: Identifying density-based local outliers", *Proc. of the ACM SIGMOD Int'l Conf. on Management of Data*, 2000.

[10] Wei, L., Qian, W., Zhou, A., Jin, W., "HOT: Hypergraph-based Outlier Test for Categorical Data", *Proc. of 7th Pacific-Asia Conference on Knowledge Discovery and Data Mining PAKDD*, pp. 399-410, 2003.

[11] Tax, D., Duin, R., "Support Vector Data Description", *Machine Learning*, pp. 45–66, 2004.

[12] Harkins, S., He, H., Williams, G., Baster, R., "Outlier Detection Using Replicator Neural Networks", *Data Warehousing and Knowledge Discovery,* pp. 170-180, 2002.

[13] Pei, Y., Zaiane, O., Gao, Y., "An Efficient Reference-based Approach to Outlier Detection in Large Dataset", *IEEE Int'l Conference on Data Mining*, 2006.

[14] Hodge, V., Austin, J., "A Survey of Outlier Detection Methodologies", *Artificial Intelligence Review*, pp. 85, 2004.

[15] He, Z., Deng, S., Xu, X., "A Fast Greedy algorithm for outlier mining", *Proceedings of PAKDD*, 2006.

[16] Agrawal, R., Srikant, R., "Fast algorithms for mining association rules", *Int'l Conf. on Very Large Data Bases*, pp. 487–499, 1994.

[17] Beyer, K., Goldstein, J., Ramakrishnan, R., Shaft, U., "When is 'nearest neighbor' meaningful?", *Int'l Conf. Database Theory*, pp. 217–235, 1999.

[18] Hettich, S., Bay, S. *KDDCUP 1999 dataset, UCI KDD archive*. 1999.

[19] Blake, C., Merz, C. *UCI machine learning repository*. 1998.