CHAPTER 5

MIXED ATTRIBUTE OUTLIER FACTOR MODEL

In this chapter, a new methodology Mixed Attribute Outlier Factor (MAOF) score algorithm has been introduced. This approach has been derived based on mixed attribute dataset. This dataset contain two parts. The first part is the categorical part and the other one is continuous part. Attribute Value Frequency (AVF) score is calculated as the average of frequencies of all values included in a record for categorical part and cosine dot product between mean record of the dataset and the remaining data records for numerical part. This method is experimented on Bank Marketing Data [1]. This method finds considerable true positives from each sample. The existing Otey [3] and ODMAD [2] algorithms are also presented in this chapter.

5.1 Otey Score Algorithm

In this approach outlier score is computed based on partitioning the entire mixed attribute dataset in to two parts. First part contains categorical subspace and the second part contains numerical subspace. Outlier factor of categorical part is denoted as Score_1 (x_i) and outlier score of numerical part is denoted by Score_2 (x_i). This approach is described based on links between attributes.
This approach is derived from frequent patterns like below:

Let $V$= set of all distinct categorical values included in Dataset

$C$= Set of all combinations (itemsets I) of distinct attribute values

i.e. $P = FS \cup IFS$ \hspace{1cm} (5.1.1)

Where

$FS$: Set of Frequent combinations of values such that the $sup(combination) \geq user$ defined threshold value.

$IFS$: Set of infrequent Itemsets

Sup (I) = support of itemset I

Now the outlier score of the categorical part is derived as given below:

$$Score_{C(x)}(I) = \sum_{I \in FS(x)} \frac{1}{|I|}$$ \hspace{1cm} (5.1.2)

Let $C_I$ is the covariance of the itemset $I$,

$C_{ij}$ is the covariance of $I$ in ‘i’ and ‘j’ attributes from numerical part

$C_{xij}$ is the covariance of $I$ in ‘i’ and ‘j’ attributes from numerical part for the object $x_i$

$$C_{ij}^{x} = (x_i - \mu_i)(x_j - \mu_j)$$ \hspace{1cm} (5.1.3)
The violation score of an object $x_i$ is defined by the equation (5.1.4) as follows:

$$V_i(x_i) = \sum_i \sum_j v_i(x_i)$$  \hspace{1cm} (5.1.4)$$

$$v_i(x_i) = \begin{cases} 1 & \text{if } c_{ij}^{\text{old}} - c_{ij}^{\text{old}} \leq \varepsilon \\ 0 & \text{otherwise} \end{cases}$$  \hspace{1cm} (5.1.5)$$

$$\sigma_{c_i}^2 = \frac{1}{\text{sup}(I) - 1} \sum_{i \in I \in x_i} (C_{ij}^{\text{old}} - C_{ij}^{\text{old}})^2$$  \hspace{1cm} (5.1.6)$$

Where $\frac{C_{ij}^{\text{old}} - C_{ij}^{\text{old}}}{\sigma_{c_i}}$ follows the normal distribution

Now the outlier score of $x_i$ in a dataset

$$\text{Score}2(x_i) = \sum_{l \in \omega} \left( \frac{1}{|l|} \right) / (C_1 \lor C_2)\text{isTrue}$$  \hspace{1cm} (5.1.7)$$

Where C1: sup (l) ≤ threshold value C2: sup (l) > δ, where δ is a maximum disturbed score. Based on these two scores we can find outlier factor of each object in mixed attribute dataset.

### 5.2 ODMAD Score

This algorithm is based on both categorical and numerical attributes. i.e mixed attributes. The ODMAD computes two scores (Score1, and Score2) defined by the equations 5.2.1 and 5.2.2.
\[ Score_1(xi) = \sum_{|IF(xi)| = 1}^{\text{MAXLEN}} \frac{1}{\text{sup}(IF(xi)) \times |IF(xi)|} \]  

(5.2.1)

\text{MAXLEN} = \text{user entered maximum length of infrequent itemset},

\text{Sup} (IF (xi)) = \text{support of infrequent itemset in object } xi,

\[ |IF(xi)| = \text{length of infrequent itemset in object } xi, \]

\[ Score_2((xi)) = \frac{1}{|a \in xi^c|} \sum_{\forall a \in xi^c} \text{COS}(xi^N, \mu_a) \]  

(5.2.2)

\text{Where,} \text{COS}(xi^N, \mu_a) = \sum_{j=1}^{m_a} \frac{1}{\|x_j^N\|} \frac{\mu_a}{\mu_a} \]  

(5.2.3)

Here ‘a’ is a categorical value included in the object \( xi \). Based on the above scores the outlier factor is found in ODMAD. In both approaches finding frequent itemsets is a big problem. So we proposed a new approach called MAOF.

### 5.3 Proposed Mixed Attribute Outlier Factor Method

In this approach outlier factor is found with forming any frequent patterns in an object. Instead of this the attribute value frequency has been proposed. From the above two approaches number of scans of a dataset is required. Proposed method needs only on scan of the dataset. This proposed method finds again two scores, one is for categorical part
of dataset and other is for numerical part of the dataset. Score1 is defined like below:

Let there are ‘m’ categorical attributes and ‘n’ numerical attributes in a dataset.

\[
Score_1(C(x)) = \sum_{j=1}^{m} \frac{\sup(x_i)}{|D|}
\]  

(5.3.1)

\[
Score_2(N(x)) = COS(N(x)) = \frac{\langle \mu_{in}, x_i \rangle}{\|\mu_{in}\| \cdot \|x_i\|}
\]  

(5.3.2)

Here \( \mu_{in} \) is a vector of means of all records in Numerical part of the Dataset.

\( x_i \) is the vector of all attribute values in the numerical part of the \( i^{th} \) object.

MAOF factor can be defined as

\[
MAOF(x) = \frac{Score_1(C(x)) + Score_2(N(x))}{2}
\]  

(5.3.3)

MAOF always gives the values between minus one to one.

### 5.4 Implementation

The Bank Marketing Data is used for the experimentation of MAOF method. The Bank marketing Dataset used in this work contains seven
categorical attributes and two continuous attributes. All the other attributes categorical and numerical parts are discarded for the experiments. This dataset contains 45212 records, but 2500 records are selected randomly using Clementine 11.1 for the MAOF experiments. Again from these 2500 records different sample techniques are used to select different samples. 1-in-2, 1-in-5, 1-in-8, 1-in-10 samples are applied on 2500 records to select different number of samples. These samples are selected like that one sample from each two records, one sample from each 5 records, one sample from each 8 records, and one sample from each 10 records. Now 45 records are created randomly with completely different values in each attributes and mixed up with all the samples selected. Then MAOF is applied on all the samples. Results are given in 4.7.

5.5 Experimental Results

All the experiments are conducted on a workstation with a Pentium ® D, 2.80 GHz processor and 1.24 GB of RAM. Coding is developed in java platform and ran our own implementation of MAOF algorithm for MAOF scores for each record. All preprocessing operations of the data are done by Clementine11.1. The Bank Marketing Dataset used for the experimentation is taken from the UCI ML data repository [5].
Table 5.1 Comparison of results for different k values in 1-in-2

<table>
<thead>
<tr>
<th>Sample</th>
<th>Number of true and false outliers are found for different K values given as input</th>
</tr>
</thead>
<tbody>
<tr>
<td>K=10</td>
<td>K=20</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>1-in-2</td>
<td>10</td>
</tr>
</tbody>
</table>

Figure 5.1 Comparison of true positives and false positives for different k
In this sample for \( k=10, k=20, k=30 \) MAOF found all positives and no false positives. For \( k=40 \) MAOF found 39 true positives and one false positive. For \( k=50 \) MAOF found the same 39 true positives but 11 false positives. By the above experiments when the input is increasing the precision is decreasing. For all the samples the MAOF gave the same results. Even the total records are decreased in each sample, MAOF i the efficiency of finding true positives are same in each sample. The results of all samples are given in the table 5.2 and the graph of true positive found by MAOF in each sample is given in Figure 5.2. Outliers found graph by MAOF is given in Figure 5.3.

**Table 5.2** Comparison of the number of outliers found for different input ‘\( K \)’

<table>
<thead>
<tr>
<th>Sample</th>
<th>Number of true and false outliers are found for different K values given as input</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>K=10</strong></td>
</tr>
<tr>
<td></td>
<td>True</td>
</tr>
<tr>
<td>1-in-2</td>
<td>10</td>
</tr>
<tr>
<td>1-in-5</td>
<td>10</td>
</tr>
<tr>
<td>1-in-8</td>
<td>10</td>
</tr>
<tr>
<td>1-in-10</td>
<td>10</td>
</tr>
</tbody>
</table>
Figure 5.2 Comparison of true positives found in different samples for different inputs
Figure 5.3 Outliers found by MAOF

From the Figure 5.3 red stars are outliers. It has shown that the red points are under the inliers because of low frequent records.

** Summary **

The proposed MAOF method is simple when compared with all the existing methods. Time complexity is also very less because it needs only one scan of the entire database and whereas existing systems needs
thousands of scans. The next chapter deals with the conclusion and future direction of the work.