A Practical Outlier Detection Approach for Mixed-Attribute Data

Mohamed Bouguessa

University of Quebec at Montreal
Department of Computer Science
Montreal, Qc, Canada
bouguessa.mohamed@uqam.ca

Abstract

Outlier detection in mixed-attribute space is a challenging problem for which only few approaches have been proposed. However, such existing methods suffer from the fact that there is a lack of an automatic mechanism to formally discriminate between outliers and inliers. In fact, a common approach to outlier identification is to estimate an outlier score for each object and then provide a ranked list of points, expecting outliers to come first. A major problem of such an approach is where to stop reading the ranked list? How many points should be chosen as outliers? Other methods, instead of outlier ranking, implement various strategies that depend on user-specified thresholds to discriminate outliers from inliers. *Ad hoc* threshold values are often used. With such an unprincipled approach it is impossible to be objective or consistent. To alleviate these problems, we propose a principled approach based on the bivariate beta mixture model to identify outliers in mixed-attribute data. The proposed approach is able to automatically discriminate outliers from inliers and it can be applied to both mixed-type attribute and single-type (numerical or categorical) attribute data without any feature transformation. Our experimental study demonstrates the suitability of the proposed approach in comparison to mainstream methods.

Keywords: Data Mining, Outlier detection, Mixed-attribute data, Mixture model, Bivariate beta.

---

1. Introduction

Outlier detection is the practice of identifying data points which are considerably different from the remaining data (Aggarwal, 2013; Cao, Si, Zhang, and Jia, 2010; Kriegel, Kroger, Schubert, and Zimek, 2011; Tan, Steinbach, and Kumar, 2006). Outlier detection is also known as exception mining or deviation detection because outlier points are exceptional in some sense or they have attribute values that deviate significantly from the expected or typical attribute values (Tan et al., 2006). Identifying outliers has practical applications in different domains such as intrusion and fraud detection, medical diagnosis, and many others (Fustes, Dafonte, Arcay, Manteiga, Smith, Vallenari, and Luri, 2013; Maervoet, Vens, Berghe, Blockeel, and Causmaecker, 2012; Alan and Catal, 2011). For example, in medical diagnosis, outliers may arise when the patient is afflicted with some disease, or suffers side-effects from a drug. Efficient detection of such outliers aids in identifying, preventing, and repairing the effects of malicious or faulty behavior (Penny and Jolliffe, 2011).

Approaches to outlier detection can be categorised as supervised, semi-supervised, and unsupervised (Angiulli and Fassetti, 2014). In principle, supervised, as well as semi-supervised learning methods, use labeled data to create a model which distinguishes outliers from inliers. On the other hand, unsupervised approaches do not require any labeled objects and detect outliers as points that are considerably dissimilar or inconsistent with respect to the remaining data using some quantified measures of outlierness (Aggarwal, 2013). To implement supervised and semi-supervised outlier detection methods, we should first label the training data (Wu and Wang, 2013). The problem here is that labeled data samples are more difficult, expensive and time consuming to obtain than unlabeled ones. This is why unsupervised approaches are more generally and widely used, since they do not require labeled information. In this paper we focus only on unsupervised outlier detection. For more surveys and details on outlier analysis, we refer the reader to Aggarwal (2013). In the following, we first describe some background information by providing a brief description of the key idea of some outlier detection approaches which are relevant to this work. Next, we discuss a number of elements that motivate this study and describe our contributions.

1.1. Background Information

Several unsupervised approaches have been proposed to identify outliers in numerical data. Such approaches can be broadly classified as statistical-based, distance-based, and density-based (Angiulli and Pizzuti, 2005). Statistical-based approaches attempt to fit the data set under investigation to a certain kind of distribution model (in general, the Gaussian model) (Yamanishi, Takeuchi, Williams, and Milne, 2000). Inliers occur in a high probability region of the model while outliers deviate strongly from the distribution. Distance-based approaches evaluate the outlierness of a point based on the distances to its k-nearest neighbors (kNN) (Angiulli and Pizzuti, 2005, 2002). Points with large kNN distance are defined as outliers. Finally, density-based approaches use the number of points within a specific local region of a data point in order to define local density (Breunig, Kriegel, Ng, and Sander, 2000). The local density values could be then used to measure how isolated a point is with respect to the surrounding objects (Wu and Wang, 2013).

The aforementioned approaches were specifically designed for numerical data. However, in several applications, attributes in real data sets are not numerical, but have categorical values. For categorical data sets, distance-based as well as density-based techniques must confront the problem of how to choose the measurement of distance or density (Wu and Wang, 2013). This poses a significant challenge in terms of generalizing algorithms for numerical data to the categorical domain (Aggarwal, 2013). To address this issue, a number of approaches have been proposed to deal with categorical data (Koufakou, Secretan, and Georgiopoulos, 2011; He, Xu, Huang, and Deng, 2005). Some of these approaches use the concept of frequent itemset mining to estimate an outlying score for each point. Inliers are those points which contain sets of items that co-occur frequently in the data sets, while outliers are likely to be the points

		A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8
Cluster 1	O_1	0.99	0.08	0.69	0.00	Q	A	A	T
	O_2	0.98	0.08	0.70	0.76	F	A	K	D
	O_3	0.96	0.09	0.71	1.00	G	F	W	W
	O_4	0.96	0.09	0.71	0.21	L	A	C	X
	O_5	1.00	0.10	0.68	0.08	L	A	D	Y
Cluster 2	O_6	0.16	0.08	0.26	0.55	M	N	X	S
	O_7	0.50	0.08	0.00	0.56	M	F	X	I
	O_8	0.70	0.10	0.94	0.58	M	M	X	M
	O_9	0.00	0.09	0.47	0.57	M	J	X	W
	O_{10}	0.32	0.08	0.34	0.56	M	G	H	P
Cluster 3	O_{11}	0.04	1.00	0.11	0.93	T	H	B	H
	O_{12}	0.04	0.32	1.00	0.95	N	T	C	H
	O_{13}	0.05	0.88	0.55	0.93	B	R	B	H
	O_{14}	0.05	0.23	0.17	0.94	O	P	B	H
	O_{15}	0.04	0.00	0.03	0.94	A	W	B	D
Outliers	O_{16}	0.60	0.67	0.09	0.11	C	X	F	Q
	O_{17}	0.73	0.91	0.95	0.96	D	Z	G	F
	O_{18}	0.40	0.74	0.31	0.29	R	U	K	C

Figure 1: Mixed-attribute data set with clustered objects and outliers.

that contain rare itemsets (Koufakou et al., 2011).

In many cases, categorical and numerical data are found in the same data set, as different attributes. This is referred to as mixed-attribute data (Aggarwal, 2013). Outliers are those objects containing attribute values that are dissimilar to or inconsistent with the remaining objects in both the numerical and the categorical space (Koufakou and Georgiopoulos, 2010; Otey, Ghoting, and Parthasarathy, 2006). To illustrate, Fig. 1 shows a small data set composed of 18 objects with four numerical attributes (A_1, A_2, A_3, and A_4) and four categorical attributes (A_5, A_6, A_7, and A_8). As can be seen from this figure, data objects $O_1, O_2, \ldots, O_{15}$ are grouped into three clusters, while the remaining points, that is, O_{16}, O_{17}, and O_{18}, are outliers which could not be located in any cluster. Note that in this figure each cluster is represented by a shade of gray and the unclustered background is white. Clusters thus exist in different subspaces spanned by different attributes. From Fig. 1, we can see that, in contrast to inliers (that is, the clustered objects), outliers contain dissimilar attribute values. In fact, compared to points that belong to clusters, outliers have non-correlated numerical attribute values along the numerical space and infrequent attribute values across the categorical space. On the other hand, from Fig. 1, we can see that objects grouped within clusters contain attribute values that are closely related along a specific subset of dimensions. For example, objects O_1, O_2, O_3, O_4, and O_5, which form cluster 1, contain correlated attribute values along the numerical attributes A_1, A_2, A_3, and a large number of common categorical attribute values along the categorical attribute A_6.

In practice, when faced with mixed-attribute data, it is common to discretize the numerical attributes and treat all the data as categorical so that categorical outlier detection algorithms can be applied to the entire data set. However, as suggested in Zhang and Jin (2010), discretizing numerical values into several bins could introduce noise or information losses. Improper discretizing thus would hamper the detection performance. To alleviate this problem, only few approaches (Koufakou and Georgiopoulos, 2010; Zhang and Jin, 2010; Otey, Ghoting, and Parthasarathy, 2006), have been proposed to handle outliers in the mixed-attribute space.

The approach proposed in Otey et al. (2006) is based on the concept of frequent itemsets to deal with categorical attributes, and the covariance for continuous attributes. Specifically, the authors in Otey et al. (2006) assign to each point an outlier score inversely proportionate to its infrequent itemsets. They also maintain a covariance matrix for each itemset to compute anomaly scores in the continuous attribute space. A point is likely to be an outlier if it contains infrequent categorical sets, or if its continuous values differ from the covariance violation threshold. It is worth noting that the work proposed by Otey et al. (2006) has the merit of being the first outlier detection algorithm for mixed-attribute data.

Koufakou and Georgiopoulos (2010) proposed an approach named ODMAD (Outlier Detection for Mixed Attribute Datasets). This algorithm calculates first, for each point in the categorical space, an outlier score which depends on the infrequent subsets contained in that point. Data points with score values less than a user-entered frequency threshold are isolated since they contain highly infrequent categorical values and may thus potentially correspond to outliers. This process results in a reduced data set based on which other outlier scores are calculated for the numerical space using the cosine similarity measure. As described in Koufakou and Georgiopoulos (2010), since minimum cosine similarity is 0 and maximum is 1, the data points with similarity close to 0 are more likely to be outliers. Experiments in Koufakou and Georgiopoulos (2010), show that ODMAD is fast and outperforms Otey's approach.

Zhang and Jin (2010) proposed a Pattern based Outlier Detection approach (POD). Patterns in Zhang and Jin (2010) are defined to describe the data objects as well as to capture interactions among different types of attributes. The more an object deviates from these patterns, the higher its outlier score. The authors in Zhang and Jin (2010) use logistic regression to learn patterns. These patterns are then used to estimate outlier scores for objects with mixed attribute. The top n points with the highest score values are declared as outliers. It is important to note that POD is not able to handle categorical values directly. To detect the target patterns, categorical attributes are first mapped into binary attributes. Then, these binary attributes are analyzed together with the original continuous attributes to detect outliers in the mixed-attribute space.

1.2. Motivations and Contributions

The area of outlier detection in mixed-attribute data offers several opportunities for improvement. There are just very few approaches around in the literature so far, yet there are a number of problems still to solve. For instance, the output of POD (Zhang and Jin, 2010) is a ranked list of points that represents the degree of outlierness of each point. The top n points in the list with the highest degree values are considered as outliers. This method encounters a major concern: at which level should this list be cut? Stated otherwise, starting from the first (ranked number one) object, how far should we go in that list? In general, no principled way is suggested on how many points should be selected from a ranked list. In some situations, the top n points are selected solely on the basis of specific knowledge of an application. Unfortunately, prior knowledge about the data under investigation is not always available.

Since a ranked list has a particular disadvantage because there is no clear cut-off point of where to stop consulting the results, thresholding has turned out to be important in detecting outliers. For instance, ODMAD (Koufakou and Georgiopoulos, 2010) and the approach proposed by Otey et al. (2006) implement various strategies that depend on user-specified thresholds to detect outliers. In real situations, however, it is rarely possible for users to supply the threshold values accurately. Outlier detection accuracy can thus be seriously reduced if an incorrect threshold value is used. The experiments conducted in Koufakou and Georgiopoulos (2010) on the impact of using various threshold values on the outlier detection accuracy corroborate our claim. Finally, it is worth noting that ODMAD and Otey's approach depend also on other input parameters such as the minimum support, the maximum length of itemset and the size of a window of categorical and numerical scores. Setting appropriate values of these parameters is not a straightforward task.

To alleviate the aforementioned drawbacks of existing approaches for detecting outliers in the mixed-attribute space, we propose in this paper a principled approach which is able to automatically identify outliers. In our approach, we first estimate an outlying score, for each object, in the numerical space and another score in the categorical space. Next, we associate to each data point a two dimensional vector containing the estimated scores: one dimension contains the score estimated in the numerical space while the second one contains the outlying score calculated in the categorical space. We assume that, in both spaces, outliers are characterised by high score values. Finally, we propose a statistical framework, based on the bivariate beta mixture, in order to model the estimated outlier score vectors. The goal is to cluster the estimated vectors into several components such that data points associated to the component with the highest score values correspond to outliers.

We have used the beta mixture mainly because it permits multiple modes and asymmetry and can thus approximate a wide variety of shapes (Dean and Nugent, 2013; Bouguila and Elguebaly, 2012; Bouguessa, 2012; Ji, Wu, Liu, Wang, and Coombes, 2005), while several other distributions are not able to do so. For example, the standard Gaussian distribution permits symmetric "bell" shape only. However, in many real life applications, the data under investigation is skewed with non-symmetric shapes. In this setting, as observed in Dean and Nugent (2013), and in Boutemedjet, Ziou, and Bouguila (2011), the standard Gaussian distribution may lead to inaccurate modeling (e.g. over estimation of the number of components in the mixture, increase of misclassification errors, etc.). In contrast to several distributions, the beta distribution is more flexible and powerful since it permits multiple symmetric and asymmetric modes, it may be skewed to the right, skewed to left or symmetric (Bouguila, Ziou, and Monga, 2006). This great shape flexibility of the beta distribution provides a better fitting of the outlier score vectors, which leads, in turn, to accurate detection of outliers. Our experimental results corroborate our claim.

We summarize the significance of our work as follows:

1. We view the task of identifying outliers from a mixture modeling perspective, on which we devise a principled approach which is able to formally discriminate between outliers and inliers, while previous works provide only a ranked list of objects expecting outliers to come first.
2. The proposed method automatically identifies outliers, while existing approaches require human intervention in order to set a detection threshold or to manually define the number of outliers to be identified. Furthermore, our method is general, in the sense that it is not limited to mixed-attribute data and it can be applied to single-type attribute (numerical or categorical) data without any feature transformation.
3. We conducted detailed experiments on several real data sets with mixed-attribute as well as with single-type attribute. The results suggest that the proposed approach achieves competitive results in comparison to mainstream outlier detection algorithms.

The rest of this paper is organized as follows. Section 2 describes our approach in detail. An empirical evaluation of the proposed method is given in Section 3. Finally, our conclusion is given in Section 4.

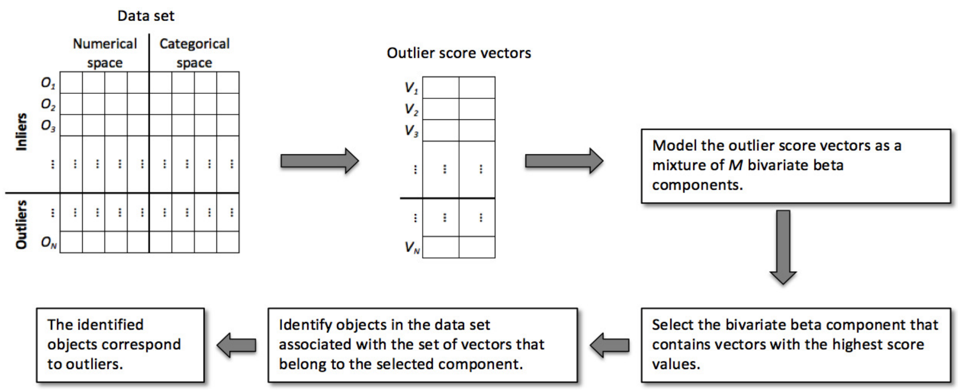


Figure 2: Workflow of the proposed approach.

2. Proposed Approach

We begin by fixing a proper notation to be used throughout the paper. Let $\mathcal{D} = \{O_1, \ldots, O_N\}$ be a set of N mixed-attribute data points. Each point contains A_n numerical attributes and A_c categorical attributes. The subspace of $\mathcal{D}$ that contains only numerical attributes is denoted S_{num}, while S_{cat} refers to the subspace of $\mathcal{D}$ which contains only categorical attributes. In this paper, we represent a data point O_i as $O_i = [O_i^n, O_i^c]$, such that $O_i^n = (o_{i1}^n, \ldots, o_{il}^n, \ldots, o_{iA_n}^n)$ and $O_i^c = (o_{i1}^c, \ldots, o_{it}^c, \ldots, o_{iA_c}^c)$, where o_{il}^n designates the l^{th} numerical attribute value and o_{it}^c corresponds to the t^{th} categorical attribute value. In what follows, we will call o_{il}^n a numerical 1D point and o_{it}^c a categorical 1D point.

In our approach, we propose first to estimate, for each object O_i, an outlier score in the numerical space and another score in the categorical space. Then, we associate to each data point a two-dimensional outlier score vector $\vec{V_i}$ containing the two estimated scores. Finally, based on $\{\vec{V_i}\}_{(i=1,\ldots,N)}$, we devise a probabilistic approach that uses the bivariate beta mixture model to automatically discriminate outliers from inliers in the full-dimensional space. Specifically, we first model $\{\vec{V_i}\}$ as a mixture of m bivariate beta components. We then select the component that corresponds to vectors with the highest score values. Data objects associated with the set of vectors that belong to the selected component correspond to outliers. Fig. 2 provides a simple visual illustration of the proposed approach. More details are given in the follows.

2.1. Estimating Outlier Score in the Numerical Space

It is widely accepted that outliers are data points that are considerably dissimilar from the remaining data (Aggarwal, 2013; Huang and Yang, 2011; Kriegel et al., 2011). In this setting, it is reasonable to assume that, in general, most of the attribute values of outlier objects projected along each of the dimensions in S_{num} tend to be far apart from the remaining attribute values (Tan et al., 2006). On the other hand, inliers have attribute values that tend to be closely related along several (or all) dimensions in S_{num}. Our assumption is based on the fact that inliers tend to form dense regions across several dimensions in the numerical space, while outliers are sparsely distributed. With this intuition in mind, we define the outlier score $\mathcal{ON}(O_i^n)$ for an object O_i in the numerical attribute space as

$$\mathcal{ON}(O_i^n) = \sum_{l=1}^{A_n} \log\big(W_N(o_{il}^n) + 1\big) \tag{1}$$

with

$$W_N(o_{il}^n) = \sum_{j=1}^{k} \Big[d_l\big(o_{il}^n, kNN_j(o_{il}^n)\big)\Big]^2 \tag{2}$$

where, for a specific dimension l in S_{num}, $kNN_j(o_{il}^n)$ denotes the j^{th} nearest (1D point) neighborhood of o_{il}^n and d_l denotes the distance between two numerical 1D points. In our case, this distance simply corresponds to the absolute value of the difference between two numerical attribute values of a specific dimension.

The outlier score defined in (1) is the sum, over all dimensions in the numerical space S_{num}, of the log of the weight $W_N(o_{il}^n)$. As described by (2), $W_N(o_{il}^n)$ computes the sum of the square of the distance between each 1D point o_{il}^n and its k nearest neighborhoods in dimension l. Intuitively, a large value of $W_N(o_{il}^n)$ means that o_{il}^n falls into a sparse region in which the k nearest neighborhood attribute values of o_{il}^n are loosely related, while a small value indicates that o_{il}^n

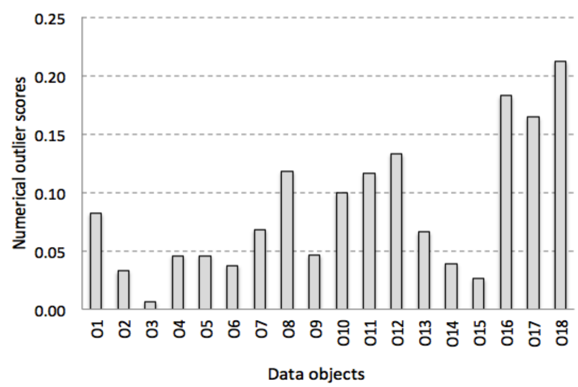


Figure 3: The estimated outlier scores in the numerical space for the data objects depicted by Fig. 1.

belongs to a dense region in which the k nearest neighborhood of o^n_{il} are closely related. Note that we have used the square power in (2) in order to favor the weight of the 1D points belonging to a sparse region.

The weight $W_N(o^n_{il})$ captures the degree of isolation of an attribute value with respect to its neighbors. The higher its weight, the more distant are its neighbors along dimension l of S_{num}. Accordingly, based on (2), we can surmise that objects that are sparsely distributed over S_{num} will receive high $\mathcal{ON}(O^n_i)$ values, while related points will receive low score values. This means that outliers will be characterized by high score values in contrast to inliers. As an illustration, Fig. 3 shows the estimated outlier scores in the numerical space for the data objects depicted by Fig. 1. As can be seen from Fig. 3, outlier objects O_{16}, O_{17}, and O_{18} have high score values in comparison to inliers, that is, $O_1, \ldots, O_{15}$.

It is important to note that we have used the logarithm function in (1) primarily to squeeze together the large values that characterize outliers and stretch out the smallest values, which correspond to inliers. This squeezing and stretching contributes to enhancing the contrast between largest and smallest values which helps in distinguishing outliers from the rest of the points. Finally, note that we have added 1 to $W_N(o^n_{il})$ in equation (1) to avoid the null value in the calculation of the logarithm, since it is possible to have $W_N(o^n_{il}) = 0$ in the likely case where an attribute has more than k duplicative values.

It is clear that the calculation of the k nearest neighbors is, in general, an expensive task, especially when the number of data points N is very large. However, since we are searching for the k nearest neighbors in the one-dimensional space, we can perform the task in an efficient way by sorting the values in each attribute and limiting the number of distance comparisons to a maximum of $2k$ values. The computation of the kNN distance is sensitive to the value of k, which is a limitation common to all kNN based approaches. However, we believe the problem this limitation creates for our approach does not have a major impact. This is because, since we estimate the kNN distances in the one-dimensional space only, the choice of the value of k is not as critical as in a multi-dimensional case. As suggested in Bouguessa and Wang (2009), to gain a clear idea of the sparseness of the neighborhood of a 1D point, we suggest using $k = \sqrt{N}$ as a default value.

2.2. Estimating Outlier Score in the Categorical Space

Virtually, as suggested in previous studies (Koufakou et al., 2011; Koufakou and Georgiopoulos, 2010; He et al., 2005), outliers in the categorical space are those points that have infrequent attribute categorical values for all dimensions compared to normal points. This means that every categorical 1D point of outlier objects is infrequent across all dimensions of S_{cat}, while inliers have several categorical 1D points which occur with higher frequency along several (or all) categorical attributes (Koufakou et al., 2011; Koufakou and Georgiopoulos, 2010). Based on such a definition, the outlier score $\mathcal{OC}(O_i^c)$ for an object O_i in the categorical attribute space is formulated as

$$\mathcal{OC}(O_i^c) = \sum_{t=1}^{A_c} \log\left(W_C(o_{it}^c)\right) \tag{3}$$

with

$$W_C(o_{it}^c) = f(o_{it}^c) \tag{4}$$

where $f(o_{it}^c)$ denotes the number of times o_{it}^c appears in a specific categorical dimension t of S_{cat}.

$\mathcal{OC}(O_i^c)$ is defined as the sum, across all dimensions in the categorical space S_{cat}, of the log of the weight $W_C(o_{it}^c)$, which, in turn, corresponds to the occurrence frequency of o_{it}^c in the categorical attribute t. Here, it is clear that rare categorical attribute values projected along dimension t will receive low weight values, while larger $W_C(o_{it}^c)$ values indicate that o_{it}^c is shared by several objects within dimension t. Accordingly, based on (3), points that share common categorical values across S_{cat} will get large $\mathcal{OC}(O_i^c)$ values, while data objects that have infrequent categorical values across S_{cat} will receive low $\mathcal{OC}(O_i^c)$ values. As a result, since outliers are those points whose attribute categorical values occur very rarely along each dimension in S_{cat} (Koufakou et al., 2011), it is easy to see that small values of $\mathcal{OC}(O_i^c)$ designate outliers and high scores correspond to inliers. Finally, note that, as with the numerical outlier score described by (1), we have used a logarithm function in (3) to enhance the contrast between larger and smaller weight values.

In this paper, as mentioned in Section 1, we assume that outliers are characterized by large score values in contrast to inliers. However, as just discussed, large $\mathcal{OC}(O_i^c)$ scores refer to inliers. To regularize such scores, we need to invert them. For this purpose, we simply take the difference between the observed score and the maximum possible estimated score $\mathcal{OC}_{max}$. The inverted score is estimated as

$$\mathcal{OC}_{inv}(O_i^c) = \mathcal{OC}_{max} - \mathcal{OC}(O_i^c) \tag{5}$$

It easy to show that this linear inversion doesn't affect the ranking-stability of the inverted scores:

$$\begin{aligned}
\mathcal{OC}(O_1^c) \leq \mathcal{OC}(O_2^c) \quad &\iff \quad -\mathcal{OC}(O_1^c) \geq -\mathcal{OC}(O_2^c) \\
&\iff \quad \mathcal{OC}_{max} - \mathcal{OC}(O_1^c) \geq \mathcal{OC}_{max} - \mathcal{OC}(O_2^c) \\
&\iff \quad \mathcal{OC}_{inv}(O_1^c) \geq \mathcal{OC}_{inv}(O_2^c).
\end{aligned}$$

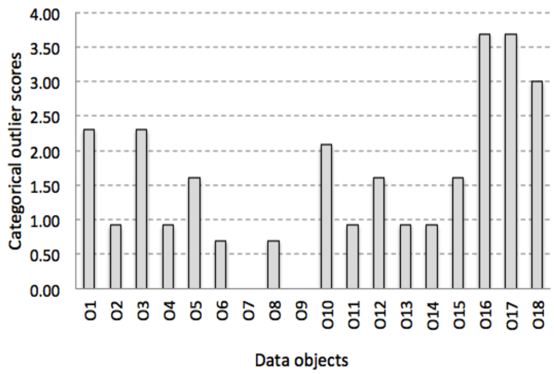


Figure 4: The estimated outlier scores in the categorical space for the data objects depicted by Fig. 1.

Accordingly, based on such a linear inversion, outliers will receive large score values while inliers will receive the lowest score values. In the remainder of this paper, unless otherwise specified, we use only the inverted categorical outlier score values. As an illustration, Fig. 4 shows the estimated outlier scores in the categorical space for the data objects depicted by Fig. 1. As can be seen from Fig. 4, outlier objects O_{16}, O_{17}, and O_{18} have high score values in comparison to inliers, that is, $O_1, \ldots, O_{15}$.

Finally, as the reader can notice, in our approach we treat numerical and categorical attributes independently in order to estimate outlier scores in the numerical and the categorical space. In other words, this means we assume the independence of both numerical and categorical attributes. Such an assumption is mainly based on the general definition of outliers, which relies on the fact that outlier objects contain attribute values that are dissimilar to or inconsistent with the remaining points. Stated otherwise, outliers may contain many atypical attribute values across most (or all) attributes of the data in comparison to inliers. Accordingly, investigating individual attributes in order to localize attribute values that deviate significantly from the expected or typical attribute values is appropriate to effectively detect outliers in the whole space.

2.3. Modeling Outlier Score Vectors

Once the outlier scores are estimated in both the numerical and the categorical spaces, we now focus on how to automatically identify outliers in the mixed-attribute space. To this end, we associate to each object O_i a two-dimensional vector $\vec{V_i}$ such that the first element of this vector corresponds to the outlier score of O_i in the numerical space, while the second element represent the outlier score of O_i in the categorical space. Then, based on the estimated vectors, we propose a probabilistic approach that uses the bivariate beta mixture model to automatically discriminate outliers from inliers in the full-dimensional space. The probabilistic model framework is described in the follows.

2.3.1. The Bivariate Beta Mixture Model

Since the beta distribution is defined on the interval [0,1], we should first, without loss of generality, normalize the estimated outlier score values between 0 and 1. Let $\vec{V_i} = (V_{i1}, V_{i2})^T$ where V_{i1} and V_{i2} represent, respectively, the normalized outlier scores $\mathcal{ON}(O_i^n)$ and $\mathcal{OC}_{inv}(O_i^c)$. Under a mixture of bivariate beta distribution,

$$\vec{V_i} \sim \sum_{m=1}^{M} \lambda_m \, \mathcal{B}_m(\vec{V_i}|\vec{x}_m, \vec{y}_m) \tag{6}$$

where $\mathcal{B}_m(\vec{V_i}|\vec{x}_m, \vec{y}_m)$ is the m^{th} bivariate beta distribution; M denotes the number of components in the mixture; $\vec{x} = \{\vec{x}_1, \ldots, \vec{x}_M\}$ and $\vec{y} = \{\vec{y}_1, \ldots, \vec{y}_M\}$. $\vec{x}_m$ and $\vec{y}_m$ are the parameters of the m^{th} component with $\vec{x}_m = (x_{m1}, x_{m2})^T$ and $\vec{y}_m = (y_{m1}, y_{m2})^T$. $\lambda = \{\lambda_1, \ldots, \lambda_M\}$ represents the mixing coefficients such that $\sum_{m=1}^{M} \lambda_m = 1$ and $\lambda_m > 0$.

The bivariate beta distribution can be obtained by cascading two beta variables together, that is, each element in the two-dimensional vector $\vec{V_i}$ is a scalar beta variable. In other words, the bivariate beta is the product of two univariate beta densities. Accordingly, the probability density function of the m^{th}

bivariate beta component is expressed as

$$\mathcal{B}_m(\vec{V}_i|\vec{x}_m,\vec{y}_m)=\prod_{d=1}^{2}\mathcal{B}(V_{id}|x_{md},y_{md}) \tag{7}$$

$\mathcal{B}(V_{id}|x_{md},y_{md})$ is the probability density function of the univariate beta distribution which is given by

$$\mathcal{B}(V_{id}|x_{md},y_{md})=\frac{\Gamma(x_{md}+y_{md})}{\Gamma(x_{md})\Gamma(y_{md})}V_{id}^{x_{md}-1}(1-V_{id})^{y_{md}-1} \tag{8}$$

where $\Gamma(.)$ is the gamma function given by $\Gamma(\alpha)=\int_0^\infty \beta^{\alpha-1}\exp(-\beta)d\beta;\beta>0$.

2.3.2. Maximum Likelihood Estimate

A common approach for estimating the unknown parameters x_{md} and y_{md}, $(m=1,\ldots,M;\ d=1,2)$ is the maximum likelihood estimation technique. The likelihood function corresponding to the m^{th} bivariate beta component $\mathcal{B}_m$ is defined as

$$\begin{aligned}\mathcal{L}\big(\mathcal{B}_m(\vec{V}_i|\vec{x}_m,\vec{y}_m)\big) &= \prod_{\vec{V}_i\in\mathcal{B}_m}\mathcal{B}_m(\vec{V}_i|\vec{x}_m,\vec{y}_m)\\ &= \prod_{\vec{V}_i\in\mathcal{B}_m}\prod_{d=1}^{2}\mathcal{B}(V_{id}|x_{md},y_{md})\end{aligned} \tag{9}$$

The logarithm of the likelihood function is given by

$$\log\big[\mathcal{L}\big(\mathcal{B}_m(\vec{V}_i|\vec{x}_m,\vec{y}_m)\big)\big]=\sum_{i=1}^{N_m}\sum_{d=1}^{2}\log\big[\mathcal{B}(V_{id}|x_{md},y_{md})\big] \tag{10}$$

where N_m is the size of the m^{th} component.

We note that the parameters pair $\{x_{md},y_{md}\}$ is independent from all other pairs. The problem of estimating the parameters of the model can thus be reduced to the estimation of the parameters pair $\{x_{md},y_{md}\}$ independently over each dimension of the outlier score vectors belonging to component m. In this setting, the value $\{\widehat{x}_{md}, \widehat{y}_{md}\}$ that maximizes the likelihood can be obtained by taking the derivative of the expectation of the log-likelihood function with respect to x_{md} and y_{md} and setting the gradient equal to zero as

$$\begin{bmatrix} \frac{\partial E\left(\log \left[\mathcal{L}(\mathcal{B}_m(\vec{V}_i|\vec{x}_m,\vec{y}_m)) \right] \right)}{\partial x_{md}} \\ \\ \frac{\partial E\left(\log \left[\mathcal{L}(\mathcal{B}_m(\vec{V}_i|\vec{x}_m,\vec{y}_m)) \right] \right)}{\partial y_{md}} \end{bmatrix} = 0 \tag{11}$$

where

$$\begin{aligned} \frac{\partial E\left(\log \left[\mathcal{L}(\mathcal{B}_m(\vec{V}_i|\vec{x}_m,\vec{y}_m)) \right] \right)}{\partial x_{md}} &= \sum_{i=1}^{N_m} \left[\frac{\partial}{\partial x_{md}} \log \left(\frac{\Gamma(x_{md}+y_{md})}{\Gamma(x_{md})\Gamma(y_{md})} V_{id}^{x_{md}-1} (1-V_{id})^{y_{md}-1} \right) \right] \\ &= \sum_{i=1}^{N_m} \left[\frac{\Gamma'(x_{md}+y_{md})}{\Gamma(x_{md}+y_{md})} - \frac{\Gamma'(x_{md})}{\Gamma(x_{md})} + \log(V_{id}) \right] \\ &= N_m \frac{\Gamma'(x_{md}+y_{md})}{\Gamma(x_{md}+y_{md})} - N_m \frac{\Gamma'(x_{md})}{\Gamma(x_{md})} + \sum_{i=1}^{N_m} \log(V_{id}). \end{aligned} \tag{12}$$

and

$$\begin{aligned} \frac{\partial E\left(\log \left[\mathcal{L}(\mathcal{B}_m(\vec{V}_i|\vec{x}_m,\vec{y}_m)) \right] \right)}{\partial y_{md}} &= \sum_{i=1}^{N_m} \left[\frac{\partial}{\partial y_{md}} \log \left(\frac{\Gamma(x_{md}+y_{md})}{\Gamma(x_{md})\Gamma(y_{md})} V_{id}^{x_{md}-1} (1-V_{id})^{y_{md}-1} \right) \right] \\ &= \sum_{i=1}^{N_m} \left[\frac{\Gamma'(x_{md}+y_{md})}{\Gamma(x_{md}+y_{md})} - \frac{\Gamma'(y_{md})}{\Gamma(y_{md})} + \log(1-V_{id}) \right] \\ &= N_m \frac{\Gamma'(x_{md}+y_{md})}{\Gamma(x_{md}+y_{md})} - N_m \frac{\Gamma'(y_{md})}{\Gamma(y_{md})} + \sum_{i=1}^{N_m} \log(1-V_{id}). \end{aligned} \tag{13}$$

Equations (11), (12) and (13) yield the following expression

$$\begin{bmatrix} N_m \left[\psi(x_{md}+y_{md}) - \psi(x_{md}) \right] + \sum_{i=1}^{N_m} \log(V_{id}) \\ \\ N_m \left[\psi(x_{md}+y_{md}) - \psi(y_{md}) \right] + \sum_{i=1}^{N_m} \log(1-V_{id}) \end{bmatrix} = 0 \tag{14}$$

where $\psi(.)$ is the digamma function given by $\psi(\alpha) = \frac{\Gamma'(\alpha)}{\Gamma(\alpha)}$.

Since the digamma function is defined though an integration, a closed-form solution to (14) does not exist. So the parameters pair $\{x_{md}, y_{md}\}$ can be estimated using the Newton-Raphson method (Ypma, 1995). Specifically, $\{x_{md}, y_{md}\}$ are estimated iteratively:

$$\begin{bmatrix} x_{md}^{(I+1)} \\ \\ y_{md}^{(I+1)} \end{bmatrix} = \begin{bmatrix} x_{md}^{(I)} \\ \\ y_{md}^{(I)} \end{bmatrix} - [\vec{h}_m]^T [\mathcal{H}_m]^{-1} \tag{15}$$

where I is the iteration index, h_m and $\mathcal{H}_m$ are respectively the vector of the first derivatives and the matrix of the second derivatives of the log likelihood function of the m^{th} component.

The vector $\vec{h}_m$ is defined as

$$\vec{h}_m = \begin{pmatrix} h_m^1 \\ h_m^2 \end{pmatrix} = \begin{pmatrix} \frac{\partial E\left(\log\left[\mathcal{L}(\mathcal{B}_m(\vec{V}_i|\vec{x_m},\vec{y_m}))\right]\right)}{\partial x_{md}} \\ \frac{\partial E\left(\log\left[\mathcal{L}(\mathcal{B}_m(\vec{V}_i|\vec{x_m},\vec{y_m}))\right]\right)}{\partial y_{md}} \end{pmatrix} \tag{16}$$

and the matrix $\mathcal{H}_m$ is expressed as

$$\mathcal{H}_m = \begin{pmatrix} \frac{\partial h_m^1}{\partial x_{md}} & \frac{\partial h_m^1}{\partial y_{md}} \\ \\ \frac{\partial h_m^2}{\partial x_{md}} & \frac{\partial h_m^2}{\partial y_{md}} \end{pmatrix},$$

where

$$\begin{aligned} \frac{\partial h_m^1}{\partial x_m} &= N_m\big[\psi'(x_{md}+y_{md}) - \psi'(x_{md})\big], \\ \frac{\partial h_m^1}{\partial y_m} &= \frac{\partial h_m^2}{\partial x_{md}} = N_m\big[\psi'(x_{md}+y_{md})\big], \\ \frac{\partial h_m^2}{\partial y_m} &= N_m\big[\psi'(x_{md}+y_{md}) - \psi'(\beta_{md})\big]. \end{aligned} \tag{17}$$

$\psi'(.)$ is the trigamma function given by $\psi'(\alpha) = \frac{\Gamma''(\alpha)}{\Gamma(\alpha)} - [\frac{\Gamma'(\alpha)}{\Gamma(\alpha)}]^2$.

The Newton-Raphson algorithm for the update of equation (15) converges, as our estimate of x_{md} and y_{md} change by less than a small positive value ϵ with each successive iteration, to $\hat{x}_{md}$ and $\hat{y}_{md}$. Note that, we have used in our implementation the method of moments estimators of the beta distribution (Bain and Engelhardt, 2000) to define starting values for $\{x_{md}^{(0)}, y_{md}^{(0)}\}$ in equation (15). In this technique, the expected mean of the distribution is equated to the sample mean and the expected variance to the sample variance. Specifically, the method of moments estimators are

$$\begin{aligned} \hat{x}_{md}^{(0)} &= \overline{\mu}_{md}\left[\frac{\overline{\mu}_{md}(1-\overline{\mu}_{md})}{\sigma_{md}^2} - 1\right], \\ \hat{y}_{md}^{(0)} &= (1-\overline{\mu}_{md})\left[\frac{\overline{\mu}_{md}(1-\overline{\mu}_{md})}{\sigma_{md}^2} - 1\right]. \end{aligned} \tag{18}$$

where $\overline{\mu}_{md}$ and σ_{md}^2 denote respectively the sample mean and variance of the normalized outlier score vectors belonging to the m^{th} component which are projected along dimension d.

2.3.3. EM Algorithm for the Bivariate Beta Mixture

Let $\mathcal{P} = \{\lambda_1, \ldots, \lambda_M, \vec{x}_1, \ldots, \vec{x}_M, \vec{y}_1, \ldots, \vec{y}_M\}$ denote the set of parameters of the mixture and $\mathcal{V} = \{\vec{V}_1, \ldots, \vec{V}_N\}$ the set of the normalized outlier score vectors. The usual choice for obtaining the maximum likelihood of the distribution parameters is the EM algorithm (Dempster, Laird, and Rubin, 1977). This algorithm is based on the interpretation of $\mathcal{V}$ as *incomplete* data. As mentioned in Figueiredo and Jain (2002), for finite mixture, the *missing* part is a set of N label vectors $\eta = \{\vec{\eta}_1, \ldots, \vec{\eta}_N\}$ associated with the N outlier score vectors, indicating to which component $\vec{V}_i$ belongs. Specifically, each $\vec{\eta}_i = (\eta_{i1}, \ldots, \eta_{im})^T$ is a binary vector, where $\eta_{im} = 1$ if $\vec{V}_i$ belongs to component m and $\eta_{im} = 0$ otherwise.

The complete data is thus defined by the sets η and $\mathcal{V}$. The likelihood of the complete data is then:

$$\mathcal{L}(\mathcal{V}, \eta|\mathcal{P}) = \prod_{i=1}^{N}\prod_{m=1}^{M}\left[\lambda_m\ \mathcal{B}_m(\vec{V}_i|\vec{x}_m, \vec{y}_m)\right]^{\eta_{im}} \tag{19}$$

and the complete log likelihood is:

$$\begin{aligned}
\log(\mathcal{L}(\mathcal{V},\eta|\mathcal{P})) &= \sum_{i=1}^{N}\sum_{m=1}^{M}\eta_{im}\,\log\big[\lambda_m\;\mathcal{B}_m(\vec{V}_i|\vec{x}_m,\vec{y}_m)\big] \\
&= \sum_{i=1}^{N}\sum_{m=1}^{M}\eta_{im}\,\log\big[\lambda_m\prod_{d=1}^{2}\mathcal{B}(V_{id}|x_{md},y_{md})\big] \\
&= \sum_{i=1}^{N}\sum_{m=1}^{M}\eta_{im}\big[\log(\lambda_m)+\sum_{d=1}^{2}\log(\mathcal{B}(V_{id}|x_{md},y_{md}))\big] \quad (20)
\end{aligned}$$

The EM algorithm can now be used to estimate $\mathcal{P}$. Specifically, the algorithm iterates between an Expectation step and an Maximization step in order to produce a sequence estimate $\{\hat{\mathcal{P}}\}^{(I)}, (I = 0, 1, 2, \dots)$, where I denotes the current iteration step, until the change in the value of the complete log-likelihood in (20) is negligible. Details of each step are given below.

In the Expectation step: each latent variable η_{im} is replaced by its expectation as follows

$$\hat{\eta}_{im}^{(I)} = E[\eta_{im}|\vec{V}_i,\mathcal{P}] = \frac{\hat{\lambda}_m^{(I)}\mathcal{B}_m(\vec{V}_i|\vec{\hat{x}}_m,\vec{\hat{y}}_m)}{\sum_{j=1}^{M}\hat{\lambda}_j^{(I)}\mathcal{B}_j(\vec{V}_i|\vec{\hat{x}}_j,\vec{\hat{y}}_j)} \quad (21)$$

In the Maximization step: the mixing coefficients $\{\lambda_m\}$ and the parameters $\{\vec{x}_1,\dots,\vec{x}_M,\vec{y}_1,\dots,\vec{y}_M\}$ are calculated using the values of $\widehat{\eta}_{im}$ estimated in the Expectation step. Specifically, the mixing coefficients are calculated as

$$\hat{\lambda}_m^{(I+1)} = \frac{\sum_{i=1}^{N}\hat{\eta}_{im}^{(I)}}{N},\; m = 1,\dots,M \quad (22)$$

The parameters $\{\vec{x}_m = (x_{m1}, x_{m2})^T\}_{(m=1,\dots,M)}$ and $\{\vec{y}_m = (y_{m1}, y_{m2})^T\}_{(m=1,\dots,M)}$ are estimated using the Newton-Raphson algorithm, based on (15), as described in the previous subsection.

Finally, note that, the EM algorithm requires the initial parameters of each component. Since EM is highly dependent on initialization, it will be helpful to perform initialization by means of clustering algorithms (Figueiredo and Jain, 2002). For this purpose we implement the k-means algorithm in order to

Algorithm 1: EM algorithm for the bivariate beta mixture

Input : $\{\vec{V}_i\}_{(i=1,\ldots,N)}$; M
Output: $\hat{\mathcal{P}} = \{\hat{\lambda}_1, \ldots, \hat{\lambda}_M, \vec{\hat{x}}_1, \ldots, \vec{\hat{x}}_M, \vec{\hat{y}}_1, \ldots, \vec{\hat{y}}_M\}$

begin
    `// Initialization`
    Apply the k-means algorithm to cluster the set $\{\vec{V}_i\}$ into M components;
    Estimate the initial set of parameters of each component using (18);
    repeat
        `// Expectation`
        Estimate $\{\hat{\eta}_{im}\}_{(i=1,\ldots,N;\ m=1,\ldots,M)}$ using (21);
        `// Maximization`
        Estimate $\{\hat{\lambda}_m\}_{(m=1,\ldots,M)}$ using (22);
        Estimate $\{\widehat{x}_{md}, \widehat{y}_{md}\}_{(m=1,\ldots,M;\ d=1,2)}$ using (15);
    until *the change in (20) is negligible*;
    Return $\hat{\mathcal{P}}$;
end

partition the set $\{\vec{V}_i\}_{(i=1,\ldots,N)}$, into M components. Then, based on such partition, we estimate the initial parameters of each component using the method of moment estimator of the beta distribution (Bain and Engelhardt, 2000) and set them as initial parameters to the EM algorithm. The detailed algorithm is summarized in Algorithm 1.

2.3.4. Estimating the Optimal Number of Components in the Mixture

The use of mixture of the bivariate beta distribution allows us to give a flexible model to describe the outlier score vectors. To form such a model, we need to estimate M, the number of components, and the parameters for each component. Several model selection approaches have been proposed to estimate M (Bouguessa, Wang, and Sun, 2006). In this paper, we implemented a deterministic approach that uses the EM algorithm described in Algorithm 1 in order to obtain a set of candidate models for the range value of M (from 1 to M_max, the maximum number of components in the mixture) which is assumed to contain the optimal M (Figueiredo and Jain, 2002). The number of components is then selected according to

$$\hat{M} = \underset{M}{\operatorname{argmin}} \left\{ \mathcal{C}\big(\hat{\mathcal{P}}, M\big) \right\}_{M=1,\ldots,M_{max}} \tag{23}$$

Algorithm 2: Estimating the number of components in the mixture

Input : $\{\vec{V}_i\}_{(i=1,\ldots,N)}$, M_max

Output: The optimal number of components $\hat{M}$

begin
    for $M = 1$ **to** M_max **do**
        if $M{=}{=}1$ **then**
            Estimate $\{\widehat{x}_d, \widehat{y}_d\}_{d=1,2}$ using (15);
            Estimate $\mathrm{ICL} - \mathrm{BIC}(\hat{\mathcal{P}}, M)$ using (24);
        else
            Estimate the parameters of the mixture using Algorithm 1;
            Estimate $\mathrm{ICL} - \mathrm{BIC}(\hat{\mathcal{P}}, M)$ using (24);
        end
    end
    Select $\hat{M}$, such that $\hat{M} = \underset{M}{\operatorname{argmin}}\ \mathrm{ICL} - \mathrm{BIC}(\hat{\mathcal{P}}, M)$;
end

where $\mathcal{C}\big(\hat{\mathcal{P}}, M\big)$ is some model selection criterion. Ji et al. (2005) found that the Integrated Classification Likelihood-Bayesian Information Criterion (ICL-BIC) performs well in selecting the number of components in the beta mixture. ICL-BIC has been also used in Dean and Nugent (2013) to select the number of beta mixture components. Accordingly, we use in our method ICL-BIC to identify the optimal number of components. The ICL-BIC criterion is given by

$$\mathrm{ICL} - \mathrm{BIC}(\hat{\mathcal{P}}, M) = -2\log(\mathcal{L}(\mathcal{V}, \hat{\eta}|\hat{\mathcal{P}})) + \mathcal{Q}_M \log(N) - 2\sum_{i=1}^{N}\sum_{m=1}^{M} \hat{\eta}_{im}\log(\hat{\eta}_{im}) \tag{24}$$

where $\mathcal{Q}_M$ denotes the number of parameters of the model with M components and $\log(\mathcal{L}(\mathcal{V}, \hat{\eta}|\hat{\mathcal{P}}))$ corresponds to logarithm of the likelihood at the maximum likelihood solution for the investigated mixture model. The number of components that minimize $\mathrm{ICL} - \mathrm{BIC}(\hat{\mathcal{P}}, M)$ is considered to be the optimal value for M. The procedure for estimating the number of components in the mixture is summarized in Algorithm 2.

2.3.5. Automatic Identification of Outlier

Once the optimal number of components has been identified, we focus now on detecting the bivariate beta component that corresponds to outliers. To this end, we used the results of the EM algorithm in order to derive a classification decision about which outlier score vector $\vec{V}_i$ belongs to which component in the mixture. In fact, the EM algorithm yields the final estimated posterior probability $\hat{\eta}_{im}$, the value of which represents the posterior probability that $\vec{V}_i$ belongs to component m. We assign $\vec{V}_i$ to the component that corresponds to the maximum value of $\hat{\eta}_{im}$. We thus divide the set of outlier score vectors into several components. As discussed earlier, in our approach we assume that outlier points are characterized by high score values. Therefore, we are interested by the bivariate beta component which contains vectors with the highest score values. To identify such a component, we first compute, for each component in the mixture, the average value of the numerical outlier scores and also the average value of the categorical outlier scores (that is, we compute the average of V_{i1} and V_{i2} per component). Then, we select the component with the largest average values as our target component. This simple strategy for determining which component to pick works well in practice since it fits our assumption, which is based on the fact that outlier points are characterized by large score values in both numerical and categorical space. Finally, once our target component is identified, we focus on the problem of detecting outlier objects. To this end, we identify the set of data objects that are associated with the outlier score vectors $\vec{V}_i$ that belong to the selected component. The identified objects are outliers. The steps described in Algorithm 3 can be implemented to automatically identify outliers.

Finally, it is worth noting that the proposed methodology could be also used to identify outlier objects in single-type (categorical or numerical) attribute data. In this particular case, we propose to associate to each object only one score ($\mathcal{ON}(O_i^n)$ or $\mathcal{OC}_{inv}(O_i^c)$, depending on the attribute type of the data under investigation). Then, to automatically discriminate between outliers and inliers, we can model the estimated scores as a finite mixture distribution using the univariate beta which is given by (8). Here, the problem is thus reduced from modeling a set of two-dimensional outlier score vectors $\{\vec{V}_i\}_{(i=1,\ldots,N)}$ (in the case of mixed-attribute data) to modeling a list of scalar

Algorithm 3: Automatic identification of outliers

Input : A data set $\mathcal{D}$
Output: A set of outliers $\mathcal{OUT}$

begin
  Estimate $\{\mathcal{ON}(O_i^n)\}_{(i=1,\ldots,N)}$ using (1);
  Estimate $\{\mathcal{OC}_{inv}(O_i^c)\}_{(i=1,\ldots,N)}$ using (3) and (5);
  Associate, to each object O_i in $\mathcal{D}$, a vector $\vec{V_i} = (V_{i1}, V_{i2})^T$ where V_{i1} and V_{i2} represent, respectively, the normalized values of $\mathcal{ON}(O_i^n)$ and $\mathcal{OC}_{inv}(O_i^c)$ in [0,1] ;
  Apply Algorithm 2 to cluster $\{\vec{V_i}\}_{(i=1,\ldots,N)}$ into M bivariate beta components;
  Use the results of the EM algorithm to decide about the membership of the outlier score vectore $\vec{V_i}$ in each component;
  Select the bivariate beta component that contains vectors with the highest score values;
  Identify objects in $\mathcal{D}$ associated with the set of $\vec{V_i}$ that belong to the selected component and store them in $\mathcal{OUT}$;
  Return $\mathcal{OUT}$;
end

outlier score values ($\{\mathcal{ON}(O_i^N)\}_{(i=1,\ldots,N)}$ or $\{\mathcal{OC}_{inv}(O_i^c)\}_{(i=1,\ldots,N)}$). In this setting, the parameters of the univariate beta mixture model to be estimated are $\{\lambda_m, x_m, y_m\}_{(m=1,\ldots,M)}$. These parameters and the optimal number of components in the mixture are estimated using the EM algorithm with the Newton-Raphson method and ICL-BIC as described in the above subsections. By doing so, we divide the outlier scores into several populations so that the larger scores can be identified and the associated objects are then declared as outliers.

3. Experimental Evaluation

In this section, we devise an empirical study to evaluate the suitability of the proposed approach. In the following, we first describe the technique that we have adopted to produce data for use in outlier detection and the performance metrics used in the evaluation. Next, we illustrate the effectiveness of our approach to identify outliers in mixed-attribute data. Finally, we devise further experiments to evaluate the performance of the proposed methodology in detecting outliers in single-type attribute data.

3.1. Data Preparation and Metrics

We draw the attention of the reader to the fact that, at the time of writing this paper, there is a shortage of standard benchmark data that can be used to evaluate outlier detection algorithms. Most of the publicly available labeled data are primarily designed for classification and machine learning applications. If the real data are unlabeled, then the evaluation of outlier detection accuracy must be done based on domain knowledge or with the help of a domain expert. However, this scenario is not practical for the purpose of evaluation since domain knowledge is not always available. All these factors make the evaluation of the proposed methodology a challenging task.

In this paper, we saliently illustrate the performance of our approach in handling outliers using real data from the UCI Machine Learning Repository [1]. Most of these data sets are labeled for classification purposes. Here, we have to be aware of the fact that these class labels are not the perfect ground truth in the sense that they do not correspond necessarily to potential outliers in the data. Keeping these issues in mind, we have adopted a principled way to produce real data for use in outlier detection.

In our experiments, similar to the work in (Das and Schneider, 2007), we create simulated outlier objects by randomly selecting attribute values. Specifically, in the numerical attribute space, we first normalize the attribute values of each numerical attribute onto the interval [0, 1] and the then inject outlier points whose attribute values are randomly selected from [0, 1]. As a result of this process, all the points in our data sets have coordinates in the range [0, 1] and are either normal points or outliers. Note that the outliers are distributed at random throughout the entire space. On the other hand, to obtain outliers in the categorical space, we inject novel objects in the data set in such a way that, for each dimension t, the attribute value of the newly generated object is randomly selected from the whole set of distinct categorical values that form

[1] http://archive.ics.uci.edu/ml/

dimension t in the original data. Outliers in the mixed-attribute space are a random combination of the newly generated objects in both the numerical and the categorical spaces.

For the purpose of evaluation, we used the following standard metrics: (1) Accuracy, which corresponds to the proportion of correctly partitioned objects, (2) True Positive Rate (TPR), measuring the proportion of outliers that are correctly identified as outliers, (3) False Positive Rate (FPR), corresponding to the proportion of inliers incorrectly classified as outliers, and (4) F-measure of the outliers class, corresponding to the harmonic mean between precision and recall of the outlier objects class.

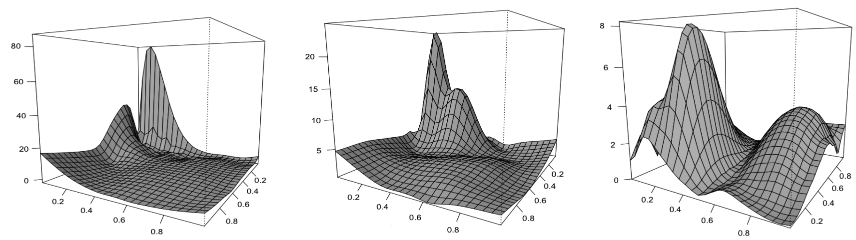
3.2. Experiments on Mixed-Attribute Data

The goal of the experiments conducted in this section are to evaluate the suitability of the proposed approach in handling outliers in mixed-attribute data. We compare the performance of our approach to that of ODMAD (Koufakou and Georgiopoulos, 2010), the most recent approach for detecting outliers in the mixed-attribute space. Note that ODMAD requires a number of parameters to be set by the user. For fairness in comparison, several values were tried for the parameters of ODMAD, following the suggestions in its original paper, and we report results for the parameter settings that produced the best results. Note that the selection of the best result here refers to the best F-measure value, since this metric represents a good trade-off between TPR and FPR.

We considered mixed-attribute data sets taken from the UCI Machine Learning Repository. As mentioned in the previous subsection, to obtain data sets for use in outlier detection, we generated outlier objects by randomly flipping attribute values. We fixed the number of outliers injected in each set to 10% of the original data set size under investigation. Fig. 5 summarizes the main characteristics of the data sets used in our experiments. Note that some data sets (such as Credit Approval, Automobile and Cylinder Bands) originally contain a number of objects with missing attribute values. In our experiments, we simply ignore such objects.

	#continuous attributes	#categorical attributes	#inliers	#outliers
Australian Credit Approval	6	8	690	69
German Credit	7	13	1,000	100
Credit Approval	6	9	653	65
Heart	5	8	270	27
Thoracic Surgery	3	14	470	47
Auto MPG	5	3	398	40
Automobile	15	11	159	16
Contraceptive Method Choice	2	7	1,473	147
AutoUniv (au_6)	5	7	1,000	100
Cylinder Bands	20	19	277	27

Figure 5: Mixed-attribute data sets characteristics.



(a) Credit Approval (b) Heart (c) AutoUniv (au_6)

Figure 6: Estimated density curve of the outlier score vectors that correspond to three mixed-attribute data sets.

We used our approach to identify outliers in each of the mixed-attribute data sets considered in these experiments. To this end, we set M_max to 5 and then, as discussed in Section 2, we selected the optimal number of components that minimize ICL-BIC. Here, the reader should be aware that the value of M_max is not limited to 5 and the user can set any other value. Interestingly, we found that the estimated outlier score vectors in each of the ten data sets are well fitted by three bivariate beta components. For the purpose of illustration and in order to not encumber the paper, we show in Fig. 6 the estimated probability density function of the outlier score vectors, that corresponds to Credit Approval, Heart and AutoUniv (au_6) only. Data points associated with the bivariate beta component that contains the score vectors with the highest values correspond to outliers. Recall that the identification of the component containing the highest score values follows the procedure described in Section 2.3.5.

	Accuracy		TPR		FPR		F-measure	
	Proposed	ODMAD	Proposed	ODMAD	Proposed	ODMAD	Proposed	ODMAD
Australian Credit Approval	98.77%	**98.94%**	**95.60%**	94.20%	**0.28%**	0.57%	**0.972**	0.942
German Credit	**98.72%**	96.54%	**100.00%**	81.00%	**1.40%**	1.90%	**0.934**	0.810
Credit Approval	**98.74%**	98.60%	**98.46%**	92.30%	1.22%	**0.76%**	**0.934**	0.923
Heart	**97.60%**	93.26%	**88.80%**	62.96%	**1.48%**	3.70%	**0.872**	0.630
Thoracic Surgery	**97.05%**	94.44%	**98.58%**	87.94%	**3.40%**	3.62%	**0.939**	0.879
Auto MPG	90.58%	**92.37%**	**87.50%**	57.50%	9.11%	**4.18%**	**0.625**	0.575
Automobile	**95.97%**	94.25%	**100.00%**	66.60%	4.40%	**3.14%**	**0.811**	0.666
Contraceptive Method Choice	**94.00%**	93.20%	**67.34%**	62.58%	**3.25%**	3.73%	**0.673**	0.523
AutoUniv	**94.54%**	87.27%	**88.18%**	30.00%	**4.82%**	7.00%	**0.746**	0.300
Cylinder Bands	**99.34%**	98.68%	**100.00%**	92.59%	**0.72%**	**0.72%**	**0.964**	0.926
Average	**96.53%**	94.75%	**92.45%**	72.77%	3.01%	**2.93%**	**0.847**	0.717

Figure 7: Performance results on mixed-attribute data sets.

Fig. 7 compares the proposed method with ODMAD. Shaded regions in this figure correspond to the best values of the four evaluation metrics considered in the experiment. As can be seen from Fig. 7, our approach achieves the highest true positive rates and F-measure values across all the data sets under investigation and reports low false positive rates with high accuracy values. In fact, the proposed method achieves, on average, an accuracy of 96.53%, TPR and FPR of 92.45% and 3.01% respectively and finally an F-measure of 0.847, all pointing to fairly accurate results. On the other hand, the results provided by ODMAD are, on average, reasonable but less competitive than those achieved by our approach. As depicted by Fig. 7, ODMAD reports, on average, an accuracy of 94.75%, TPR and FPR of 72.22% and 2.93% respectively and finally an F-measure of 0.717. Overall, in term of Accuracy, TP rate and F-measure, the proposed method performs better than ODMAD while the FPR achieved by both approaches are comparable.

From Fig 7, we observe that our proposed method reports an average 92.45% TP rate. This means that 7.55%, on average, of outliers were misclassified as inliers by our approach. This not necessarily an error, since data points have coordinates in the range [0, 1] and are either inliers or outliers. Outliers were randomly placed throughout the entire space. In this setting, it is probable that some of the outlier objects will have attribute values related to normal objects in the data set under investigation. Under these circumstances, it is possible that few outlier objects will have low outlier score values and consequently be considered as inliers.

To summarize, the results presented in Fig. 7, suggest that the proposed method performs well on different data sets. Furthermore, in contrast to ODMAD which suffers from its dependency on several input parameters (detection threshold, minimum support, the maximum length of itemset and the size of a window of categorical and numerical scores), our approach is able to accurately identify outliers in an automatic fashion. Such a notable feature of our approach illustrates its practical usability to effectively identify outliers in real-life applications. Another advantage of our approach is that it is able to handle outliers in single-type (numerical or categorical) attribute data without any feature transformation, while existing methods are not able to do so. The following two subsections investigate this point using real data sets characterized by only numerical or categorical attributes.

3.3. Experiments on Numerical Data

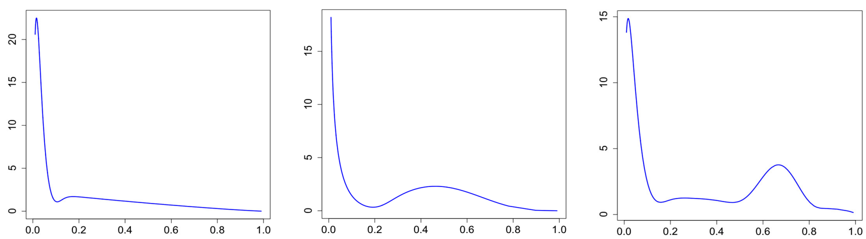
The experiments described in this section aim to illustrate the capability of the proposed methodology in detecting outlier objects in numerical data. As discussed at the end of Section 2, when the data contains only numerical attributes, we associate to each object the numerical score $\mathcal{ON}(O_i^n)$ given by (1). Then, we model these scores as a mixture of univariate beta mixture.[2] The parameters of the model $\{\lambda_m, x_m, y_m\}_{(m=1,\ldots,M)}$ and the optimal number of components in the mixture are estimated following the reasoning described in Section 2. This process results in grouping outlier scores into several beta components. Data objects associated with the beta component containing the highest score values are declared outliers.

Fig. 8 summarizes the main characteristics of the UCI numerical data sets used in the experiments. Note that, as with the experiments on mixed-attribute data, we have adopted the same technique to produce outliers, that is, normalizing the attribute values between 0 and 1 and then injecting outliers in the

[2] To fit the beta distribution, the estimated outlier scores should be first normalized in [0,1].

	#attributes	#inliers	#outliers
Cloud	10	2,048	205
Ecoli	7	336	34
Glass Identification	9	214	21
Image Segmentation	19	2,310	231
Istanbul Stock Exchange	8	538	54
Parkinson Speech	26	1,040	104
Wine Quality - Red	12	1,599	160
Wisconsin Diagnostic Breast Cancer	30	569	57
Yacht Hydrodynamics	7	308	31
Yeast	8	1,484	148

Figure 8: Numerical data sets characteristics.



(a) Ecoli (b) Wine Quality - Red (c) Breast Cancer

Figure 9: Estimated density curves of the numerical outlier scores that correspond to three numerical data sets.

data by generating objects whose attribute values are randomly selected from the interval [0,1]. The number of outliers injected in each data set corresponds to 10% of the original data set size. For each numerical data set, we estimated $\mathcal{ON}(O_i^n)$ for each object and then modelled these scores as a mixture of univariate beta distribution. To this end, we set M_max to 5 and selected the optimal number of components that minimize ICL-BIC. We found that the number of components varies from two to three beta components. For the purpose of illustration, Fig. 9 shows the density curve of the numerical outlier scores that correspond to three UCI data sets: Ecoli, Wine Quality - Red and Wisconsin Diagnostic Breast Cancer. The last component in each plot depicted by Fig. 9 represents the highest score values . Data points associated with the scores grouped in this component correspond to outliers.

To demonstrate the effectiveness of our approach, we compared its performance to that of kNN weighed outlier algorithm (kNNW) (Angiulli and Pizzuti, 2005, 2002). kNNW assigns a weight to each data point based on the sum of

	Accuracy		TPR		FPR		F-measure	
	Proposed	kNNW	Proposed	kNNW	Proposed	kNNW	Proposed	kNNW
Cloud	95.47%	**99.70%**	94.60%	**98.52%**	4.40%	**0.15%**	0.791	**0.985**
Ecoli	**99.18%**	98.90%	**93.90%**	**93.93%**	**0.29%**	0.59%	**0.954**	0.939
Glass Identification	**95.31%**	93.61%	**85.71%**	76.19%	**3.73%**	4.67%	**0.766**	0.681
Image Segmentation	99.43%	**99.74%**	**100.00%**	98.75%	0.61%	**0.14%**	0.970	**0.986**
Istanbul stock exchange	94.39%	**97.28%**	81.13%	**84.90%**	4.29%	**1.49%**	0.723	**0.849**
Parkinson Speech	98.34%	**99.39%**	99.16%	**96.66%**	1.74%	**0.33%**	0.915	**0.967**
Wine quality - red	**99.08%**	98.91%	**94.96%**	93.71%	**0.50%**	0.56%	**0.950**	0.940
Wisconsin Diagnostic Breast Cancer	97.69%	**99.18%**	**100.00%**	98.23%	2.98%	**0.50%**	0.952	**0.982**
Yacht Hydrodynamics	**97.63%**	90.53%	**73.33%**	46.66%	**0.00%**	5.19%	**0.846**	0.467
Yeast	**99.44%**	97.54%	**96.62%**	86.48%	**0.26%**	1.35%	**0.969**	0.864
Average	**97.60%**	97.48%	**91.94%**	87.40%	1.88%	**1.50%**	**0.884**	0.866

Figure 10: Performance results on numerical data sets.

the distances separating that point from its k nearest neighbors in such a way that outliers are characterized by high weights while inliers receive low weight values. After ranking data points based on the estimated weights, the top n points are identified as outliers. The implementation of this algorithm, and many other outlier detection approaches, is available in the ELKI Data Mining Framework [3] (Achtert, Kriegel, Schubert, and Zimek, 2013). Note that we have chosen kNNW for its effectiveness. In fact, in our empirical investigation, we have evaluated several other mainstream outlier detection algorithms, such as COP (Kriegel, Kroger, Schubert, and Zimek, 2012), LDOF (Zhang, Hutter, and Jin, 2009), LOCI (Papadimitriou, Kitagawa, Gibbons, and Faloutsos, 2003) and LOF (Breunig et al., 2000), already implemented in ELKI. We found that kNNW was the algorithm which performs well.

Fig. 10 illustrates the results of our approach and those of kNNW on the numerical data sets considered in the experiments. Shaded regions correspond to the best Accuracy, TPR, FPR and F-measure values. Recall that kNNW produces a ranked list of points expecting outliers to come first. Accordingly, to distinguish outliers from inliers, the user should specify the target number of outliers n. In this setting, and in order to compute the value of the four evaluation metrics used in the experiments (Accuracy, TPR, FPR and F-measure), we have simply set the value of n equal to the real number of outliers in the

[3] http://elki.dbs.ifi.lmu.de

	#attributes	#inliers	#outliers
Audiology (Standardized)	69	226	23
Congressional Voting Records	16	435	43
Lymphography	18	148	15
Mushroom	22	8214	821
Primary Tumor	17	339	34
Solar Flare	10	1389	139
Soybean (Large)	35	307	31

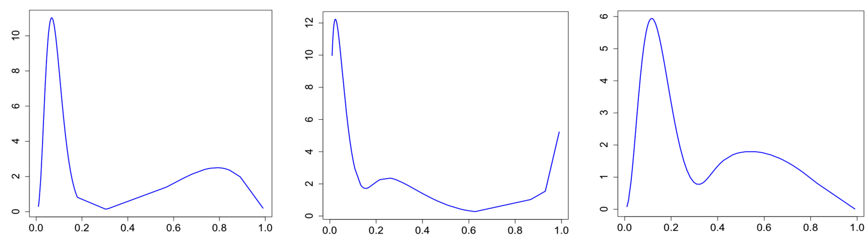
Figure 11: Categorical data sets characteristics.

data set under investigation. Finally note that, as with ODMAD, we have tried multiple values of k for kNNW, and we only report the best results, that is, those which correspond to the highest F-measure value.

As can be seen from Fig. 10, our approach achieves, on average, the highest Accuracy (97.60%), TPR (91.94%) and F-measure (0.884). On the other hand, kNNW reports the lowest average FPR (1.50%) while our approach achieves an average FPR of 1.88%. Overall, both competing algorithms show good performances. A significant advantage of our approach is that it is able to automatically discriminate outliers from inliers while with kNNW the user should specify how many points should be selected as outliers.

3.4. Experiments on Categorical Data

The aim of this section is to illustrate the suitability of the proposed approach for handling outliers in data sets with categorical attributes only. To this end, we selected a number of categorical data from the UCI Machine Learning Repository. Recall that these data sets are principally labeled for classification purposes. Accordingly, as discussed in Section 3.1, to produce data for use in outlier detection, we inject novel data points in such a way that each attribute value of each newly inserted object is randomly selected from the set of distinct categorical values that initially form the corresponding attribute in the original data. As with our previous experiments, the number of outliers injected in each data set corresponds to 10% of the original data set size. The main characteristics of the categorical data sets used in the experiments are summarized in Fig. 11.



(a) Audiology (b) Vote (c) Lymphography

Figure 12: Estimated density curves of the categorical outlier scores that correspond to three categorical data sets.

To identify outliers in each of the categorical data sets considered in these experiments, we estimated first $\mathcal{OC}_{inv}(O_i^c)$ for each object. These scores are then normalized in [0,1] and modelled as a mixture of univariate beta distribution. To identify the optimal number of components in the mixture, we set M_max to 5 and selected the number of components that minimize ICL-BIC. Interestingly, as with the experiment on numeric data, we found that the optimal number of components varies from two to three. Fig. 12 illustrates the density curve of the outlier scores corresponding to three data sets: Audiology, Congressional Voting Records (Vote) and Lymphography. The last component in each plot depicted by Fig. 12 represents the highest score values . Data points associated with the scores grouped in this component correspond to outliers. The knowledgeable reader can also observe in this rendering, and also from the pervious illustration of the estimated density curves depicted by Fig. 9 and Fig. 6, the great shape flexibility of the beta distribution which leads to accurate partitioning of the outlier scores.

Fig. 13 compares the effectiveness of our approach to that of a recent outlier detection approach for categorical data named Information-Theory Based Single-Pass (ITB-SP) (Wu and Wang, 2013). It has been empirically illustrated that ITB-SP is an effective approach which outperforms several existing categorical outlier detection algorithms. The implementation of this algorithm has been kindly provided by its authors. As the name implies, this approach harnesses information theory concepts to estimate an outlier score for each ob-

	Accuracy		TPR		FPR		F-measure	
	Proposed	ITB-SP	Proposed	ITB-SP	Proposed	ITB-SP	Proposed	ITB-SP
Audiology (Standardized)	99.09%	**100.00%**	**100.00%**	**100.00%**	0.01%	**0.00%**	0.952	**1.000**
Congressional Voting Records	92.17%	**92.21%**	**91.53%**	83.07%	7.64%	**5.05%**	**0.844**	0.830
Lymphography	**93.82%**	90.74%	**100.00%**	**100.00%**	**6.75%**	10.10%	**0.736**	0.651
Mushroom	95.19%	**98.16%**	**97.29%**	89.90%	5.00%	**1.00%**	0.786	**0.899**
Primary Tumor	**99.31%**	98.62%	**92.85%**	92.30%	**0.00%**	0.75%	**0.969**	0.923
Solar Flare	93.05%	**93.97%**	**87.68%**	66.66%	6.40%	**3.31%**	**0.695**	0.666
Soybean (Large)	**94.48%**	**94.48%**	**94.56%**	88.04%	5.53%	**3.58%**	**0.887**	0.880
Average	95.30%	**95.45%**	**94.84%**	88.57%	4.48%	**3.40%**	**0.838**	0.836

Figure 13: Performance results on categorical data sets.

ject. Specifically, the authors in Wu and Wang (2013) propose the concept of holoentropy as a new measure for outlier detection. As defined in Wu and Wang (2013), holoentropy is a combination between entropy and total correlation with attribute weighting, where the entropy measures the global disorder in the data and the total correlation measures the attributes relationship. Based on this concept, that is holoentropy, the authors formulate a function to estimate an outlier score for each object in such a way that outliers are characterized by high score values. The top n objects with the highest score values are declared as outliers. Note that, since ITB-SP requires the number of outliers in the data n to be specified by the user, and in order to compute Accuracy, TPR, FPR and F-measure, we have simply set the value of n equal to the real number of outliers in the data set under investigation.

As can be seen from Fig. 13, the average performance results for our approach and ITB-SP are quite similar except for the average TPR and FPR. Our method reports an average 94.84% of true positives while the average TPR of ITB-SP is 88.57%. This means that only 5.16%, on average, of outliers were misclassified as inliers by our approach while 11.43%, on average, of outliers were misclassified as inliers by ITB-SP. On the other hand, as illustrated by Fig. 13, we can see that ITB-SP achieves the lowest FPR, that is 3.40%, while the proposed method reports an average 4.48% of false positives. Overall, the results illustrated in Fig. 13 suggest that both approaches display good performance. Our approach has, however, the non-negligible advantage of automatically discriminating outliers from inliers while ITB-SP requires the number of outliers in the data to be specified by the user. As discussed earlier, in real applications for which no prior knowledge about the data is available, it is not always possible for the user to set accurately the value of this parameter.

4. Conclusion

In this paper, we have highlighted some limitations of existing outlier detection approaches for mixed-attribute data, including their dependency on user parameters, such as the detection threshold and the target number of outliers to be identified, which are difficult to tune and their incapability of formally discriminating between outliers and inliers. To alleviate these problems, we have proposed a principled approach that performs outlier detection in an automatic fashion.

In our approach, we first devised two functions in order to estimate, for each object, an outlier score in the numerical space and another score in the categorical space. Outliers in both spaces are characterized by high score values. Next, we associate to each data point in the data set under investigation a two-dimensional vector such that the first element of this vector corresponds to the estimated outlier score in the numerical space, while the second element corresponds to the outlier score estimated in the categorical space. Then, we model these vectors as a mixture of bivariate beta. The bivariate beta component that corresponds to the highest score values represents outliers. The beta distribution has been chosen due to its great shape flexibility which leads, in turn, to accurate fitting of the estimated outlier score vectors. We have described a statistical framework to illustrate how the bivariate beta mixture model can be used to identify outlier objects.

Finally, we have devised a detailed empirical study to illustrate the suitability of our approach in detecting outliers using several UCI data sets with mixed-attributes. We have compared the performance of the proposed method to that of ODMAD, the most recent approach for detecting outliers in the mixed-attribute space. The results show that our approach achieves results that are, in most cases, better than those of ODMAD. Moreover, we have performed further experiments to demonstrate the capability of our methodology in handling outliers in single-type attribute data without any feature transformation. Tests and comparison with previous ranking approaches on several numerical and categorical UCI data sets show that the proposed methodology exhibits competitive results.

Acknowledgements

The author gratefully thanks Dr. Shu Wu for providing the implementation of the ITB-SP algorithm. The author also would like to thank the reviewers for their valuable comments and important suggestions. This work is supported by research grants from the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

Achtert, E., Kriegel, H.-P., Schubert, E., & Zimek, A., 2013. Interactive data mining with 3D-parallel-coordinate-trees. In Proceedings of the ACM SIGMOD international conference on Management of Data, 1009–1012.

Aggarwal, C. C., 2013. Outlier Analysis. Springer.

Alan, O., & Catal, C., 2011. Thresholds based outlier detection approach for mining class outliers: An empirical case study on software measurement datasets. Expert Systems with Applications, 38 (4), 3440–3445.

Angiulli, F., & Fassetti, F., 2014. Exploiting domain knowledge to detect outliers. Data Mining and Knowledge Discovery, 28 (2), 519–568.

Angiulli, F., & Pizzuti, C., 2002. Fast outlier detection in high dimensional spaces. In Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery, 15–26.

Angiulli, F., & Pizzuti, C., 2005. Outlier mining in large high-dimensional data sets. IEEE Transactions on Knowledge and Data Engineering, 17 (2), 203–215.

Bain, L. J., & Engelhardt, M., 2000. Introduction to Probability and Mathematical Statistics, 2nd Edition. Duxbury Press.

Bezdek, J., 1981. Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum.

Bouguessa, M., 2012. Modeling outlier score distributions. In Proceedings of the 8th international conference on Advanced Data Mining and Applications, 713–725.

Bouguessa, M., & Wang, S., 2009. Mining projected clusters in high-dimensional spaces. IEEE Transactions on Knowledge and Data Engineering, 21 (4), 507–522.

Bouguessa, M., Wang, S., & Sun, H., 2006. An objective approach to cluster validation. Pattern Recognition Letters, 27 (13), 1419–1430.

Bouguila, N., & Elguebaly, T., 2012. A fully Bayesian model based on reversible jump MCMC and finite beta mixtures for clustering. Expert Systems with Applications, 39 (5), 5946–5959.

Bouguila, N., Ziou, D., & Monga, E., 2006. Practical bayesian estimation of a finite beta mixture through gibbs sampling and its applications. Statistics and Computing, 16 (2), 215–225.

Boutemedjet, S., Ziou, D., & Bouguila, N., 2011. Model-based subspace clustering of non-gaussian data. Neurocomputing, 73 (10-12), 1730–1739.

Breunig, M. M., Kriegel, H.-P., Ng, R., & Sander, J., 2000. LOF: Identifying density-based local outliers. In Proceedings of the ACM SIGMOD international conference on Management of Data, 93–104.

Cao, H., Si, G., Zhang, Y., & Jia, L., 2010. Enhancing effectiveness of density-based outlier mining scheme with density-similarity-neighbor-based outlier factor. Expert Systems with Applications, 37 (12), 8090–8101.

Das, K., & Schneider, J., 2007. Detecting anomalous records in categorical datasets. In Proceedings of the 13th ACM SIGKDD international conference on Knowledge Discovery and Data Mining, 220–229.

Dean, N., & Nugent, R., 2013. Clustering student skill set profiles in a unit hypercube using mixtures of multivariate betas. Advances in Data Analysis and Classification, 7 (3), 339–357.

Dempster, A., Laird, N., & Rubin, D., 1977. Maximum likelihood from incomplete data via the EM algorithm. Journal of Royal Statistical Society, (Series B), 39, 1–37.

Figueiredo, M. A. T., & Jain, A. K., 2002. Unsupervised learning of finite mixture models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24 (3), 381–396.

Fustes, D., Dafonte, C., Arcay, B., Manteiga, M., Smith, K., Vallenari, A., & Luri, X., 2013. SOM Ensemble for unsupervised outlier analysis. Application to outlier identification in the Gaia astronomical survey. Expert Systems with Applications, 40 (5), 1530–1541.

He, Z., Xu, X., Huang, J., & Deng, S., 2005. FP-Outlier: Frequent pattern based outlier detection. Computer Science and Information System, 2 (1), 103–118.

Huang, B., & Yang, P., 2011. Finding key knowledge attribute subspace of outliers in high-dimensional dataset. Expert Systems with Applications, 38 (8), 10147–10152.

Ji, Y., Wu, C., Liu, P., Wang, J., & Coombes, K., 2005. Applications of beta-mixture models in bioinformatics. Bioinformatics, 21 (9), 2118–2122.

Koufakou, A., & Georgiopoulos, M., 2010. A fast outlier detection strategy for distributed high-dimensional data sets with mixed attributes. Data Mining and Knowledge Discovery, 20 (2), 259–289.

Koufakou, A., Secretan, J., & Georgiopoulos, M., 2011. Fast outlier detection in large high-dimensional categorical data. Knowledge and Information Systems, 29 (3), 697–725.

Kriegel, H.-P., Kroger, P., Schubert, E., & Zimek, A., 2011. Interpreting and unifying outlier scores. In Proceedings of the 11th SIAM international conference on Data Mining, 13–24.

Kriegel, H. P., Kroger, P., Schubert, E., & Zimek, A., 2012. Outlier detection in arbitrarily oriented subspaces. In Proceedings of the 12th IEEE international conference on Data Mining, 379–388.

Maervoet, J., Vens, C., Berghe, G. V., Blockeel, H., & Causmaecker, P. D., 2012. Outlier detection in relational data: A case study in geographical information systems. Expert Systems with Applications, 39 (5), 4718–4728.

Otey, M. E., Ghoting, A., & Parthasarathy, S., 2006. Fast distributed outlier detection in mixed-attribute data sets. Data Mining and Knowledge Discovery, 12 (2–3), 203–228.

Papadimitriou, S., Kitagawa, H., Gibbons, P. B., & Faloutsos, C., 2003. LOCI: Fast outlier detection using the local correlation integral. In Proceedings of the 19th IEEE international conference on Data Engineering, 315–326.

Penny, K., & Jolliffe, I., 2011. A comparison of multivariate outlier detection methods for clinical laboratory safety data. Journal of the Royal Statistical Society. Series D (The Statistician), 50 (3), 295–308.

Tan, P.-N., Steinbach, M., & Kumar, V., 2006. Introduction to Data Mining. Addison Wesley.

Wu, S., & Wang, S., 2013. Information-theoretic outlier detection for large-scale categorical data. IEEE Transactions on Knowledge and Data Engineering, 25 (3), 589–602.

Yamanishi, K., Takeuchi, J.-I., Williams, G., & Milne, P., 2000. On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms. In Proceedings of the 6th ACM SIGKDD international conference on Knowledge Discovery and Data Mining, 320–324.

Ypma, T. J., 1995. Historical development of the Newton-Raphson method. SIAM Review, 37 (4), 531–551.

Zhang, K., Hutter, M., & Jin, H., 2009. A new local distance-based outlier detection approach for scattered real-world data. In Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, 813–822.

Zhang, K., & Jin, H., 2010. An effective pattern based outlier detection approach for mixed attribute data. In Proceedings of the 23rd Australasian Joint Conference on Advances in Artificial Intelligence, Lecture Notes in Artificial Intelligence, 6464, 122–131.