An efficient sampling algorithm for uncertain abnormal data detection in biomedical image processing and disease prediction

Fei Liu\textsuperscript{a,}\textsuperscript{*}, Xi Zhang\textsuperscript{b} and Yan Jia\textsuperscript{a}

\textsuperscript{a}School of computer, National University of Defense Technology, 410073, Changsha, Hunan, China
\textsuperscript{b}School of Biomedical Engineering, Fourth Military Medical University, 710032, Xian, Shaanxi, China

Abstract. In this paper, we propose a computer information processing algorithm that can be used for biomedical image processing and disease prediction. A biomedical image is considered a data object in a multi-dimensional space. Each dimension is a feature that can be used for disease diagnosis. We introduce a new concept of the top \((k_1, k_2)\) outlier. It can be used to detect abnormal data objects in the multi-dimensional space. This technique focuses on uncertain space, where each data object has several possible instances with distinct probabilities. We design an efficient sampling algorithm for the top \((k_1, k_2)\) outlier in uncertain space. Some improvement techniques are used for acceleration. Experiments show our methods' high accuracy and high efficiency.

Keywords: Biomedical image, disease diagnosis, computer information processing, abnormal detection, outlier, uncertain

1. Introduction

Biomedical image processing is an important tool for disease diagnosis and predication. Data processing using computers is the basic technique in this field. For example, urinary bladder cancer has become the fourth most common cancer among males [1]. The early detection of bladder cancer is extremely important. With recent advances in imaging and visualization techniques, virtual cystoscopy (VCy), which is based on volumetric computed tomography (CT) or magnetic resonance (MR) imaging data, has revealed its potentials for detecting bladder abnormalities [2]. As reported in [3], bladder cancerous tissue invades gradually from the mucosa into the wall muscles, inducing morphological changes and texture changes in the bladder wall. Therefore, the bladder image (MR or CT) contains some important clinical information for bladder cancer prediction, such as bladder wall thickness (BWT) [4], textural grey-level intensity, and many other features. Figure 1 show an MR bladder image acquired from a patient. The red and yellow contours represent the inner and outer borders of a bladder wall, respectively. Naturally, the BWT of the abnormal region is larger than the normal wall tissue. After we collect these abstract

\textsuperscript{*}Address for correspondence: Fei Liu, Team 7, School of Computer, National University of Defense Technology, Changsha, Hunan, 410073, China. Tel: +86 15873192183; Fax: +86 0731 84574614; E-mail:1986figo@163.com.

0959-2989/15/$35.00 © 2015 – IOS Press and the authors.

This article is published with Open Access and distributed under the terms of the Creative Commons Attribution and Non-Commercial License.
features together, we can construct a multi-dimensional space. Each dimension is a feature, and each biomedical image is a data object in the space. In a set of many biomedical images, the data objects far away from their neighbor objects can be considered abnormal, because some of their features’ values are very different from the others’. The difference could be caused by pathological changes. For example, in the two-dimensional space in Figure 2, A is considered 'abnormal', because its feature values depart from the majority. B is considered 'normal' because it is similar to many of the surrounding data objects.

At the same time, uncertainty is inherent in real application due to various factors, like noise, device imprecision, incompleteness of data, delay in data transfer, or even manual misplay [5–7]. In this case, the feature values of a data object in some dimensions would be uncertain. For example, in Figure 3, data object A could be represented by three different instances: a, b, or c. It’s difficult to obtain A’s distance from its neighbors and compare A with other objects. Our research in this paper focuses on this problem and attempts to detect abnormal data objects, denoted as outliers, in uncertain space.

An outlier is an observation that deviates so much from the other observations that it arouses suspicion that it was generated by a different mechanism [8]. Outlier detection is a basic technique in data analysis and is widely used in many applications. Many kinds of outlier have been proposed, such as the Distribution-based outlier, the Distance-based outlier, the Density-based outlier, and so on. In this paper, we use the concept of a Distance-based outlier, where the distance between a data object and its neighbors are used as the measurement for outlier detection.

We use the classic x-tuple model [9] and possible world semantics [10] to describe uncertain data. An uncertain data object (abbreviated as object) is noted as an x-tuple, containing several tuples. Each tuple belonging to an x-tuple is a possible data instance (abbreviated as instance) of the corresponding object. Each instance has an appearance probability. Instances belonging to the same object are exclusive. The sum of these instances’ probabilities is no more than 1. For example, in Table 1, \( t_{11} \) and \( t_{12} \) are two tuples of x-tuple \( T_1 \), and \( t_{11} \) and \( t_{12} \) cannot both appear at the same time. \( t_{21} \) with probability \( p_{21} \) is the unique tuple of x-tuple \( T_2 \). If \( p_{11}+p_{12}<1 \), \( T_1 \) would not appear with a probability \( 1-p_{11}p_{12} \).

Based on the x-tuple model, a possible world is a subset of tuples from different x-tuples. There would be no more than one tuple of the same x-tuple appearing in a possible world. Table 2 shows six possible worlds produced from the x-tuples in Table 1. \( \{t_{11}\} \) is the second possible world with a probability \( p_{11}(1-p_{21}) \). It contains only one tuple, \( t_{11} \).

The remainder of this paper is organized as follows. In section 2, some preliminary definitions are introduced. A naive sampling method is described in section 3. Section 4 proposes an efficient sampling algorithm to accelerate the naive sampling method. We experimentally evaluate our algorithms in section 5. We introduce some existing research related to our work in section 6 and conclude our contribution in section 7.
2. Preliminary

An uncertain dataset is constituted by many x-tuples. A large number of possible worlds would be produced as instances of the uncertain dataset. Let $D$ be an uncertain dataset, $t$ be a tuple and $T(t)$ be the x-tuple containing $t$. $P(t)$ is $t$’s appearance probability and $P(T)=\sum_{t\in T} P(t)$. Let $W$ be the set of all possible worlds produced from $D$. For a possible world $w \in W$, $t \in w$ means tuple $t$ appears in $w$, and $T \in w$ means that a tuple of $T$ appears in $w$. Let $P(w)$ be the probability of $w$, and $P(w)=\prod_{t \in w} P(t) \prod_{T \notin w} (1-P(T))$. For each tuple $t$, we assign it an outlier score $s(w, t)$ in possible world $w$, which is the average distance between $t$ and its $n$ nearest neighbors in $w$. Based on the above assumptions, we propose following definitions.

**Definition 1** Top $k_1$ outliers in a possible world: In a possible world $w$, the top $k_1$ outliers are the $k_1$ tuples with the largest outlier scores.

**Definition 2** $P_{k_1}$ probability: For an x-tuple $T$, its $P_{k_1}$ probability, $P_{k_1}(T)$, is the probability sum of all possible worlds where one of $T$’s tuples is a top $k_1$ outlier. Formally, $P_{k_1}(T)=\sum_{w \in W(T)} P(w)$. $W(T)$ is the set of possible worlds, where there is a tuple $t \in T$ and $t$ is a top $k_1$ outlier.

**Definition 3** Top $(k_1, k_2)$ outliers in an uncertain dataset: In an uncertain dataset $D$, the top $(k_1, k_2)$ outliers are the $k_2$ x-tuples with the highest $P_{k_1}$ probabilities.

The straightforward calculation of an x-tuple’s $P_{k_1}$ probability according to definition 2 needs to traverse all possible worlds. It is a #P-complete [11,12] problem leading to a time cost that is too high. Therefore, we propose an approximate method based on sampling to detect the top $(k_1, k_2)$ outliers with high accuracy and efficiency.

3. Naive sampling method

To avoid enumerating all possible worlds, we sample x-tuples one by one to produce samples of possible worlds. In each sampled possible world, the top $k_1$ outliers can be detected in a deterministic situation. For each x-tuple $T$, let $F_{k_1}(T)$ be $T$’s frequency of being a top $k_1$ outlier in $M$ sampled possible worlds. Based on the law of large numbers [13,14], $F_{k_1}(T)$ converges to $P_{k_1}(T)$ in a large sampling frequency. We use $F_{k_1}(T)$ as the approximation of $P_{k_1}(T)$ and sort all x-tuples according to $F_{k_1}(T)$ in descending order. The first $k_2$ x-tuples are output as the top $(k_1, k_2)$ outliers. We call this process the naive sampling algorithm, whose details are shown in algorithm 1. The classic RBRP [15] algorithm is used to detect the top $k_1$ outliers in a possible world.
Algorithm 1 Naive sampling algorithm

**Input:** $M$, $D$, $k_1$, $k_2$, $F$, $F[i]=0$, $0<i \leq |D|$, $m=0$.

**Step 1:** sample each x-tuple in $D$ to construct a possible world $w$.

**Step 2:** calculate outlier score for each tuple $t$ using RBRP.

**Step 3:** detect top $k_1$ outliers in $w$.

**Step 4:** for each x-tuple $T_i$ and each tuple $t \in T_i$, if $t$ is a top $k_1$ outlier, $F[i] = F[i] + 1$.

**Step 5:** if $m < M$, go to step 1.

**Output:** $k_2$ x-tuples with largest values $F[i]/M$.

4. Efficient sampling algorithm

Although the naive sampling algorithm can detect the top $(k_1, k_2)$ outliers successfully, a high frequency of sampling is needed to get high accuracy. RBRP algorithm must be repeated in each sampled possible world. Clustering and detecting the $n$ nearest neighbors’ in the RBRP algorithm leads to a high time cost. In order to overcome this problem, we propose an efficient sampling algorithm.

We find that the $n$ nearest neighbors of a tuple always exist nearby. We maintain a local region for a tuple, where most of its possible neighbors are located; then we can detect its $n$ nearest neighbors just in the local region in each possible world. For tuple $t$, let $L_N(t)$ be $t$’s nearest neighbors list. For any tuple $t_i$ in $L$, $T(t_i) \neq T(t)$. All tuples in $L_N(t)$ are sorted in ascending order according to their distance from $t$. The RBRP algorithm can be used to construct $L_N(t)$. Let $\mu$ be the sum of all tuple’s probability in $L_N(t)$. When $\mu$ is large enough, the $n$ nearest neighbors of $t$ in a possible world will always appear in $L_N(t)$. In order to detect $t$’s $n$ nearest neighbors in a sampled possible world, we can just search $L_N(t)$ in order to get the first $n$ appearing tuples. The outlier score can then be calculated easily. This procedure is in linear time cost and much more efficient than the naive sampling algorithm. The steps of the efficient sampling algorithm are shown in algorithm 2. In this way, the clustering operation of the RBRP algorithm is executed once before sampling, and the detection of the $n$ nearest neighbors for each tuple is greatly accelerated.

4.1. Improvement

Although the efficient sampling algorithm avoids repeating clustering and improves the efficiency of detecting the $n$ nearest neighbors, it’s also redundant. Because $k_1$ is always much smaller than $|D|$ and $n$ is always much smaller than $|L_N(t)|$, it’s unnecessary to sample all x-tuples to produce a possible

Algorithm 2 Efficient sampling algorithm

**Input:** $M$, $D$, $k_1$, $k_2$, $F$, $F[i]=0$, $0<i \leq |D|$, $m=0$.

**Step 1:** detect $|L_N(t)|$ possible $n$ nearest neighbors for each tuple $t$ and construct $L_N(t)$.

**Step 2:** sample each x-tuple in $D$ to construct a possible world $w$.

**Step 3:** calculate outlier score for each tuple $t$ using $L_N(t)$.

**Step 4:** detect top $k_1$ outlier in the possible world.

**Step 5:** if $t \in T_i$ is a top $k_1$ outlier, $F[i] = F[i] + 1$.

**Step 6:** if $m < M$, go to step 2.

**Output:** $k_2$ x-tuples with largest values $F[i]/M$. 

then we search the tuples in \( L \) and \( T(t_j) \). We sample x-tuple \( T(t_i) \) first. If \( t_i \) is not selected, then we sample x-tuple \( T(t_j) \). If \( t_i \) is selected, then we search the tuples in \( L_N(t_i) \) to get \( t_i \)'s \( n \) nearest neighbors. Suppose the first tuple in \( L_N(t_i) \) is \( t_k \), and \( T(t_k) \) has been sampled, but \( t_k \) is not selected. Then \( t_k \) is jumped over, and the next tuple in \( L_N(t_i) \) is checked. If \( t_k \) is selected, \( t_k \) is added to \( t_i \)'s \( n \) nearest neighbors in this possible world. Then we check the next tuple in \( L_N(t_i) \) until the \( n \) nearest neighbors of \( t_i \) are detected. Then, \( t_i \)'s outlier score in the possible world is calculated. After that, \( t_j \) is processed similarly. If no less than \( k_1 \) tuples in \( L \) have been processed, the \( k_1 \) th largest outlier score can be a threshold \( h \). For any following tuple \( t_x \), if the upper bound of \( t_x \)'s outlier score is smaller than \( h \), \( t_x \) can be pruned. The upper bound of tuple \( t \) can be calculated using function \( \sum_{1 \leq i \leq n'} d(t, t'_i) + \sum_{|L_N(t)|-n'+1 \leq k \leq |L_N(t)|} d(t, L_N(t)[k]) \), where \( t'_i \) is one of \( t \)'s \( n \) nearest neighbors that have been detected and \( L_N(t)[k] \) is the \( k \)th tuple in \( L_N(t) \). For example in Figure 4, let \( L=\langle t_1, t_2, t_3, t_4 \ldots \rangle \), \( T(t_1)=T(t_2) \). \( t_1 \) is the first tuple and \( T(t_1) \) is sampled. Suppose \( L_N(t_2) = \langle t_3, t_4 \ldots \rangle \). Then \( t_3, t_4 \ldots \) are checked one by one until the \( n \) neighbors of \( t_2 \) are detected. Then \( t_2 \)'s outlier score can be calculated. Suppose both \( t_3 \) and \( t_4 \) are selected. Then the tuples in \( L_N(t_3) \) are checked to detect \( t_3 \)'s \( n \) nearest neighbors. Suppose \( L_N(t_3)=\langle t_2, t_4 \ldots \rangle \). Because \( T(t_2) \) and \( T(t_4) \) have been processed, they will not be redundantly sampled.

5. Experiments

In this section, we conduct an empirical study on a PC with a 2.5 GHz Intel Core i5 CPUs, 8.0 GB main memory, running on the Microsoft Windows 7 operating system. Because our research is based on the novel proposed definitions, we compare the three algorithms we proposed in this paper, the naive sampling algorithm, the efficient sampling algorithm, and the improved efficient sampling algorithm. We construct a synthetic dataset to simulate data objects with the features of biomedical images. In the synthetic dataset, every object contains no more than 10 instances. The number of instances of a data object is decided randomly. For each feature, the value is in region (0,1), satisfying normal distribution with both the expectation and variance randomly produced in (0,1).

We first test the accuracy of our algorithms. We compare the efficient sampling algorithm with the different parameter values of \( \mu \) and the naive sampling algorithm. We use the result of the naive sampling
algorithm with a sampling frequency $10^5$ as the real outliers. In Figure 5(a), $n=5$, $k_1=10$, and $k_2=20$. The x-axis shows the sampling frequency. The y-axis shows the accuracy of outlier detection. In Figure 5(b), $n=5$, $k_1=10$, and $k_2=30$. In Figure 5(c), $n=5$, $k_1=20$ and $k_2=20$. A larger $\mu$ means a larger $L_N(t)$ and higher memory cost, but it also means higher accuracy. As shown in these figures, when $\mu$ is larger than 5, the accuracy of the efficient sampling algorithm is no less than 90% of the naive sampling algorithm’s accuracy.

We then test the efficiency of our algorithms. Figure 6(a) shows the time cost of the three algorithms. The x-axis shows four groups of histograms. The first group corresponds to the parameter set \{\mu=5, n=5, k_1=10, k_2=20\}. The second one corresponds to the parameter set \{\mu=5, n=5, k_1=10, k_2=30\}. The third one corresponds to the parameter set \{\mu=5, n=10, k_1=10, k_2=20\}. The y-axis shows the time cost (millisecond). It is obvious that the efficient sampling algorithm greatly increases efficiency, and the improvement method leads to further acceleration.

Finally, we show the scalability of the algorithms. Figure 6(b) shows the time cost of the three algorithms with increasing sampling frequencies. The x-axis shows the changing sampling frequency, and the y-axis shows the time cost (millisecond). The results show that the naive sampling algorithm costs much more time than the other algorithms, and the improved efficient sampling algorithm saves the most time. Figure 6(c) shows the time cost of the three algorithms with a sampling frequency of 100 at different dataset sizes. The x-axis shows the changing dataset sizes. The y-axis is the logarithmic value of the time cost (millisecond). We can find that the efficient sampling algorithm, especially the improved one performs much better than the naive sampling algorithm.

6. Related work

In order to detect outliers in uncertain space, [16] proposes an outlier model and the detection method on uncertain data. However, they do not consider the diversity of a data object where a data object is only
represented by a single instance, so they cannot be used for uncertain biomedical images processing with many possible instances. In their research, grid-based pruning is used to improve efficiency. The grid is only discussed in two dimensions and is not suitable for high-dimension space.

7. Contribution

We propose a new definition of outliers in uncertain space and offer the naive sampling algorithm. We design an efficient sampling algorithm to avoid redundant clustering operations and accelerate the detection of $n$ nearest neighbors. Then we further improve its efficiency using sorting and pruning strategies. Experiments show the effectiveness of our methods.

Acknowledgements

This research is supported by the National High-Tech R&D Program of China (No.2012AA012600, 2012AA01A401, 2012AA01A402), National Natural Science Foundation of China (No. 61202362), State Key Development Program of Basic Research of China (No. 2013CB329601) and Project funded by China Postdoctoral Science Foundation (No. 2013M542560).

References