Modeling and Efficient Mining of Intentional Knowledge of Outliers

Zhixiang Chen*, Jian Tang+, Ada Wai-Chee Fu+,
Department of Computer Science, University of Texas-Pan American,
Edinburg TX 78539 USA.
email:chen@cs.panam.edu
+ Department of Computer Science, Chinese University of Hong Kong,
Shatin, N.T., Hong Kong.
email:tang@cse.cuhk.edu.hk, email:adafu@cse.cuhk.edu.hk

Abstract

In this paper, we study in a general setting the notion of outliered patterns as intentional knowledge of outliers and algorithms to mine those patterns. Our contributions consist of a model for defining outliered patterns with the help of categorical and behavioral similarities of outliers, and efficient algorithms for mining knowledge sets of distance-based outliers and outliered patterns. Our algorithms require only very limited domain knowledge, and no classified information. We also present an empirical study to show the feasibility of our algorithms.

keywords: *outlier detection, knowledge sets, categorical similarity, behavioral similarity, outliered patterns*

1 Introduction

Outliers are those data records that do not follow any pattern in an application. In some applications detecting outliers is more significant than detecting general patterns. For instance, calling card fraud in telecommunications [8], credit card fraud in banking and finance [14], computer intrusion in information systems [6], to name a few. In these applications, outliers may cause tremendous damages at the individual, business or even national level such as terrorist attacks, and therefore the methodologies to detect them are of crucial importance.

Extensive research has been done on the discovery of patterns [7, 11, 13, 16]. Although in principle these methods can also generate outliers as by-products, for efficiency and flexibility reasons, independent methods for outlier detections are in general more favorable. Most of the early schemes for outlier detection have been done in the field of statistics. These methods normally assume that the distribution of a data set is known in advance and try to detect outliers by examining the deviations of individual data objects based on such a distribution. In reality, however, a priori knowledge about the distribution of a data set is not always obtainable. Besides, these methods do not scale well for even modest number of dimensions as the size of data set increases.

Recent progresses in outlier detection include, among many others, the distance-based scheme (called $DB(d,n)$-outlier) proposed by Knorr and Ng in [9], the density-based scheme by Breuning, et al, in [4]; and the connectivity-based scheme by Tang et al, in [15]. All the above work considers outliers as being conceptually disjoint with patterns. In reality, however, some outliers themselves can be patterned. For example, consider a database that contains records for mobile phone calls. An outlier may result from an occasional action taken by a legal owner in an exceptional circumstance, or a malicious behavior of a thief. While in the former case the kind of behavior may be sporadic, in the latter case the thief may repeat the similar actions over time and as a result generate a pattern. In many cases, identifying the patterns of the outliers is of practical significance. For instance, for the above mobile phone database, discovering the unusual calling patterns may assist in revealing the illegal activities by thieves. The simplest method, of course, is to find all the outliers first and then apply any general clustering algorithm on the set of the outliers. The problems of this strategy are twofold. Firstly, any existing clustering algorithm essentially is based on the distributions of the data as being presented in the application. Thus its effectiveness depends on the accuracy that these distributions reflect the nature of the data. Since the number of outliers is usually small, their distributions as being presented to the user in general cannot be accurate for reflecting their true nature. Secondly, in the clustering algorithms, objects and their attributes are un-interpreted, and similarities between objects are evaluated based solely on their distances. On the other hand, outliers are semantically more meaningful. (They are outliers.) In order to model effectively the pattern in this context, we should make use of these semantics. Also, a user should be allowed to deal with patterns objectively, so that only those patterns that are to his/her interest will be discovered. In summary, we need different paradigm for the modeling of outliered patterns and algorithms to discover them in the applications.

To the best of our knowledge, no existing work has considered this problem with adequate generality. Although some application-specific mechanisms have been proposed [8], they deal only with isolated instances, and require a priori domain knowledge and some classified information in order to be trained in discovering the patterns of abnormal behaviors. In case the required knowledge is not available, these mechanisms do not work. Naturally, none of these approaches characterize the concept in a general setting, and hence do not offer insights into understanding the problem in the general case.

In this paper, we study in a general setting the notion of outliered patterns and methods to tackle them. Our contributions consist of a model for outliered pattern and several algorithms to identify them. Our algorithms require only very limited domain knowledge, and no classified information. We also present an empirical study to show the feasibility of our method.

2 A Model

2.1 Roles of Subspaces

The work [1] realized that in most cases, for the objects in a high dimensional space, it makes sense only to consider outliers for their projections on lower dimensional subspaces. On the other hand, the work in [10] noticed that an outlier is most meaningful in the minimal subspaces where it is an outlier. The above work suggests the important roles subspaces may play in outlier formulation. We observe that in the context of outliered patterns, all the subspaces do not play the same role, and it is necessary to categorize them based on the roles they play in outlier detection. We shall now substantiate the mobile phone database given in the previous section, and use it as an illustrating example in the subsequent discussions.

Example 1 *A mobile phone call is recorded by a vector of eight attributes* `<acct_num, home, from, to, time, date, duration, times>`. *Each phone must set up a unique account number before it can be used. The city where the account is set up is called the "*`home`*" of the phone. While the call can be made from/to anywhere, the rate is the lowest if it is from the home. The attribute* `times` *indicates the total number of calls made from* `acct_num` *in* `date`. *Intuitively, a legal call is likely to follow the following pattern: its* `from` *is its* `home` *city, its* `time` *is not very late at night, its* `duration` *is not very long if* `time` *is during office hours, and the number of* `times` *is not too high. Here the attributes* `from, to, time, duration` *and* `times` *can be viewed as indicators for normal/abnormal behaviors of calls. On the other hand, we usually do not use* `acct_num` *and* `home` *to distinguish normal calls from abnormal ones. Instead, we simply use them to identify the calls so that those that share the same account number and/or set up cities can be related. In addition, we note that the attribute* `date` *can neither indicate the status (i.e., normal or abnormal), nor provide identities of calls. Its role is for observation. For example, if a thief periodically makes long distance calls late at night with a long duration, this is observed at different* `dates`.

We will use the following notations. $\mathcal{D} = \{A_, \ldots, A_d\}$ is called a d-dimension set, where each A_i is the name of a dimension. Define $P_{+\mathcal{D}} = domain(A_1) \times domain(A_2) \times \ldots. \times domain(A_d)$. That is, $P_{+\mathcal{D}}$ is the d-dimensional space defined on the dimensions in $\mathcal{D}$. R denotes an arbitrary set such that $R \subseteq P_{+\mathcal{D}}$. For any dimension set A such that $A \subseteq \mathcal{D}$, and any object $o \in P_{+\mathcal{D}}$, $\prod_A(o)$ denotes the projection of o on P_{+A}. Define $\prod_A(R) = \{\prod_A(o) : o \in R\}$.

Definition 1 *The quadruple* (I, C, O, idc) *is a role arrangement for* $\mathcal{D}$ *if* I, C *and* O *constitute a partition of* $\mathcal{D}$. *The sets* P_{+I}, P_{+C} *and* P_{+O} *respectively are called identification, indication and observation subspaces of* $P_{+\mathcal{D}}$. idc *is a predicate defined on* $2^{P_{+I}}$. *We call* idc *an identification condition.*

Definition 2 R *is identified by* idc *if* $\forall R_1 \subseteq R, [idc(\prod_I(R_1))] = true]$. R *is fully identified by* idc *if* R *is identified by* idc *and for all* $S \supset R$, S *is not identified by* idc.

Thus for the mobile phone call database, a possible role arrangement is $I = \{acct_{num}, home\}$, $C = \{from, to, duration, time, times\}$, $O = \{date\}$, and $idc(X) = true$ iff $\forall x, y \in X$, $\prod_I(x) = \prod_I(y)$. The identification condition here identifies any set of phone calls made from the same phone device. In the following discussions, the symbols I, C, O, idc are used exclusively for role arrangement.

Definition 3 *An observation sequence for* R *is a sequence* $T =< T_1, \ldots, T_u >$ *where* $\cup_{i=1}^{u} T_i = \prod_O(R)$ *and for all* $1 \leq i < j \leq u$, $T_i \cap T_j = \emptyset$. *Each element set in* T *is called an observation spot. For any* R_1, R_1 *is observed at* T_i *if* $\prod_O(R_1) \subseteq T_i$. *For any* $o \in R_1$, o *is an outlier observed at* T_i *with respect to* R_1 *if* R_1 *is observed at* T_i *and* $\prod_C(o)$ *is an outlier with respect to* $\prod_C(R_1)$.

It is worth noting here that the three kinds of roles of the subspaces stated above in general belong to domain knowledge. Our model to be introduced in the next subsection will assume that users have this knowledge. We argue that this assumption is reasonable since for any application, if a user is interested in detecting outliers, he/she must have ideas of what his/her concern is. This means that he/she has some idea about the nature of attributes. In other words, he/she has some basic understanding of the meaning of the attributes, and the attributes that contribute to the abnormal behaviors.

2.2 Knowledge Sets of Outliers

We first review some concepts introduced in [10] that will be used in our model. Let $X \subseteq \mathcal{D}$ be any given nonempty set of attributes. Let U be a set of objects in P_{+X}, and $o \in U$ be an outlier. We say that o is a **non-trivial outlier** in the attribute set X if for all proper subset $Y \subset X$, $\prod_Y(o)$ is not an outlier in $\prod_Y(U)$, otherwise it is a trivial outlier in X. For a trivial outlier in X, the attributes in X do not give an accurate qualification for the outlier. For example, suppose a thief makes a call after midnight for one hour, while in general many legal calls are also made after mid-night, but last only a few minutes, and many calls last for more than one hour but earlier than mid-night. In this case the attribute set $\{time, duration\}$ accurately describe the nature of the illegal call, since, when viewed separately, neither "calling at midnight" nor "lasting for one hour" is special. On the other hand, if very few legal calls are made after midnight, then the same attribute set is not accurate in describing the nature of the unusual call. In this case the singleton set $\{time\}$ is more accurate. In this paper, if o is a non-trivial outlier in an attribute set X then we say that X is a *knowledge attribute set* of o (or *knowledge set* if no confusion is possible). Note that an outlier may have multiple knowledge sets. For example, if in addition to very few calls being made after midnight, we also have that very few calls last more than one hour, then both $\{times\}$ and $\{duration\}$ are knowledge sets for that unusual call.

2.3 Similarities

We are interested in the similarities based on which the repetitive nature of the outliered patterns can be established. In our model, the repetitive nature is observed at the observation spots. Thus the similarities are defined only on the objects that are observed at different observation spots. Depending on the applications, different kinds of similarities can be defined. We will introduce two kinds of similarities that fit different expectations of users. We first introduce some notations relating to knowledge sets of outliers.

Let o be an outlier. We define the **knowledge attribute class** for o as $KAClass(o) = \{X : X\ is\ a\ knowledge\ set\ for\ o\}$. Let U be a set of outliers. We define the **knowledge attribute cartel** on U as $KACartel(U) = \cup_{o \in U} KAClass(o)\}$. Namely, the knowledge attribute cartel on U is the set of all the knowledge attribute sets of outliers in U.

Let S be the set of database objects from which we want to find the outliered patterns, and $T =< T_1, \ldots, T_n >$ be the observation sequence for S. Let $S = \cup_{i=1}^{n} S_i$ where for all i, $1 \leq i \leq n$, S_i is observed at T_i, and Q_i be the set of all the outliers observed at T_i with respect to S_i. Assume $Q \subseteq Q_1 \cup \cdots \cup Q_n$ such that Q is fully identified by *idc* in $Q_1 \cup \cdots \cup Q_n$. For all i, $1 \leq i \leq n$, let $G_i = Q \cap Q_i$. Let $G = \{G_1, G_2, \ldots, G_n\}$. For any $X \in KACartel(G_i)$, let $G_i(X) = \{p : p \in G_i \ \&\ X \in KAClass(p)\}$. Namely, $G_i(X)$ is the set of all outliers in G_i with knowledge set X.

Definition 4 *Let $Z = KACartel(G_i) \cap KACartel(G_j)$ for any two outlier groups G_i and G_j in G. G_i and G_j are categorically similar with respect to a predefined positive threshold $\alpha > 0$, if*

$$\frac{|\bigcup_{X \in Z} G_i(X)|}{|G_i|} \geq \alpha \quad and \quad \frac{|\bigcup_{X \in Z} G_j(X)|}{|G_j|} \geq \alpha.$$

We call the fractions categorical rates.

In each categorical rate, the numerator is the number of outliers in group $G_i(G_j)$ that share some knowledge sets with some members in $G_j(G_i)$. Roughly speaking, two outlier groups are categorically similar with respect to a given threshold, if each group has sufficiently large number of members that share some knowledge sets with the members in the other group. Since a knowledge set essentially categorizes the outliers based on the minimal information that can describe them, in each of the categorically similar groups, most outliers can find some outliers in the same category in the other group.

Categorical similarity uses attribute names only. It addresses only the natures of the outliers, not their precise behaviors. To address the behaviors, it is necessary to use not only attribute names, but also the attribute values.

Let S, T and G be the same as defined above. Further let $Center(Y)$ be the center for any set of outliers Y.

Definition 5 *Let $Z = KACartel(G_i) \cap KACartel(G_j)$. For any two outlier groups G_i and G_j in G and any $\delta > 0$, let*

$$B_i = \bigcup_{X \in Z} \{G_i(X) : dist(Center(G_i(X)), Center(G_j(X))) \leq \delta\} \quad and$$

$$B_j = \bigcup_{X \in Z} \{G_j(X) : dist(Center(G_j(X)), Center(G_i(X))) \leq \delta\}$$

G_i and G_j are behaviorally similar with respect to three predefined positive thresholds α, β and δ, if G_i and G_j are categorically similar with respect to α, and

$$\frac{|\bigcup_{W \in B_i} W|}{|G_i|} \geq \beta \quad and \tag{1}$$

$$\frac{|\bigcup_{W \in B_j} W|}{|G_i|} \geq \beta, \tag{2}$$

We call the fractions the behavior rates.

The condition (1) indicates the following property: G_i has a high percentage of outliers forming a set of subgroups; each of those subgroups is characterized by a common knowledge attribute set of G_i and G_j; and the center of each subgroup in G_i is very close to the center of the corresponding subgroup in G_j. Condition (2) indicates the similar property for G_j. Intuitively, the behavioral similarity is meant to reveal the property that the two outlier groups have not only a very high percentage of common "categories" as characterized by their categorical similarity, but also a very high percentage of outliers in one of the two groups that lie very close to their corresponding outliers sharing the same knowledge set in the other group.

An issue here is how to determine the parameters. Like any other outlier detection schemes, these should be determined in an iterative try-and-refine manner. A detailed discussion on this issue is beyond the scope of this paper.

Example 2 *Consider the telephone call database described in* EXAMPLE 1. *Suppose the identification condition is defined as "all calls in the group have the same home", and the observation sequence contains two consecutive Saturdays, each observing the data records in the preceding seven days. Let us consider the fully identified group of unusual phone calls that have Hong Kong as their homes. Assume that during the first seven-day period,* 50% *of the calls are long lasting late night calls, with the center, (i.e., average), at (*2 *pm,* 1.5 *hours),* 20% *are made at day-time with high daily frequency, with the center at (*15 *times), and* 30% *are from overseas. Assume calls from overseas are always centered at (overseas)*[1].

During the second seven-day period, 10% *are day-time high frequency calls centering at (*6 *times),* 40% *are made late at night with long duration with the center of around (*3 *am,* 2 *hours),* 35% *are from overseas and* 15% *are to Tibet. Let* Y_1 *and* Y_2 *respectively be the outlier groups observed at the first and the second observation spots.*

Thus
$KACartel(Y_1) = \{\{time, duration\}, \{times\}, \{from\}\}$
and
$$KACartel(Y_2) = \{\{time, durations\}, \{times\}, \{from\}, \{to\}\}.$$
Suppose $\alpha = 80\%$,
$\beta = 70\%$,
and the value of δ *makes the following true:*
$dist((2pm, 1.5hours), (3am, 2hours)) < \delta$,
$dist((15times), (6times)) > \delta$.
We also have $dist((overseas), (overseas)) = 0$.

Thus the categorical rate for Y_1 *is* 100%*, and for* Y_2 *is* 85%. Y_1 *and* Y_2 *are categorically similar. The behavior rate for* Y_1 *is* 80%*, and for* Y_2 *is* 75%*. They are behaviorally similar, too. Observe that although the day-time high frequency calls in each group contribute to the categorical similarity, they do not contribute to the behavior similarity: they can be described by the same minimal set of attributes, but are not considered very close in the behavior. Another fact is that the calls to Tibet observed at the second observation spot do not contribute to either kind of similarities, since no call to Tibet has be observed at the first observation spot.*

[1]For simplicity, assume these groups do not overlap.

This example shows how the similarities reveal the natures/behaviors of the unusual phone calls at 'home' level. By varying the identification condition, one can refine/coarsen this level. For example, by defining the idc as "all calls with the same account number", one can detect the similarities among the unusual phone calls with the same account.

2.4 Category and Behavior Patterns of Outliers

Based on the categorical and behavior similarities introduced in the previous two subsections, we can now formulate the corresponding patterns. Let S, T and G be the same as defined in Section 2.3. We have

Definition 6 *A category pattern of outliers for* G *and* T *is a maximum sequence* $G_{i_1}, G_{i_2}, \ldots, G_{i_k}$ *such that* $1 \leq i_1 \leq i_2 \leq \cdots \leq i_k \leq n$*, and any two consecutive outlier groups in the sequence are categorically similar.*

Analogously, we define behavior patterns of outliers as follows.

Definition 7 *A behavior pattern of outliers for* G *and* T *is a maximum sequence* $G_{i_1}, G_{i_2}, \ldots, G_{i_k}$ *such that* $1 \leq i_1 \leq i_2 \leq \cdots \leq i_k \leq n$*, and any two consecutive outlier groups in the sequence are behaviorally similar.*

Algorithm Mine-Knowledge-Set(o, r, n)
$C_1 = K_1 = \emptyset$;
for $i = 1, 2, \ldots, d$ do
    find $L_i^{o,r}$; //all $L_i^{o,r}$ can be found via scanning database once
    if ($|N_r^{\{A_i\}}(o)| < n$) //$N_r^{\{A_i\}}(o)| \subseteq \prod_{\{A_i\}} L_i^{o,r}$
        $K_1 = K_1 \cup \{A_i\}$;
    else $C_1 = C_1 \cup \{L_i^{o,r}\}$;
for $k = 2, \ldots, d$ do
    $C_k =$ Knowledge-Set-Gen(C_{k-1})
    if ($C_k = \emptyset$) exit;
    for each $L_{i_1,\ldots,i_k}^{o,r} \in C_k$
        if ($|N_r^{\{A_{i_1},\ldots,A_{i_k}\}}(o)| < n$) //$N_r^{\{A_{i_1},\ldots,A_{i_k}\}}(o) \subseteq \prod_{\{A_i\}} L_{i_1,\ldots,i_k}^{o,r}$
            $K_k = K_k \cup \{A_{i_1}, \ldots, A_{i_k}\}$;
            $C_k = C_k - \{L_{i_1,\ldots,i_k}^{o,r}\}$;
return $\cup_k K_k$;

Fig. 1. Algorithm KSApriori

3 Finding Knowledge Sets of Outliers

The problem of finding knowledge sets of outliers is different from that of finding "strong outliers" studied in [10]. Strong outliers are defined with respect to *strong knowledge sets* (or *strong outlying spaces*). From [10], a knowledge set is strong if it contains at least one outlier but any of its subset contains no outliers. (Refer to Section 2.2.) In our context, we want to find all knowledge sets for every individual outlier, not just strong ones. A super set of a strong knowledge set (or strong outlying space) is not interesting in [10], but may be interesting in our context. E.g., given a strong knowledge set X and a group of strong outliers in it, X may have some super sets that are knowledge sets of some other outliers, and therefore those super sets contain valuable information that describes patterns of outliers in them. The approach in [10] ignores such information, and is therefore not applicable in our context.

It is worth noting here that our similarity model in the previous section is applicable to any existing outlier models. In the following, we will use distance-based outlier model proposed in [9] due to its simplicity, and develop algorithms for pattern discovery. We first design an innovative algorithm for mining knowledge sets of distance-based outliers in Fig. 1.

The distance-based outlier model specifies two parameters, a radius r and a cardinality n. An object o is an outlier with respect to r and n if $|\{p : dist(p, o) \leq r\}| < n$. In our algorithm, we assume that each attribute A_i has an interval domain V_i. Given any outlier $o = (o_1, o_2, \ldots, o_d)$ and a radius r, let $V_i^{o,r}$ be the subset interval of V_i with o_i as its center and r as radius. Furthermore, let $L_i^{o,r}$ be the set of all the objects in the subspace $V_1 \times \cdots \times V_{i-1} \times V_i^{o,r} \times V_{i+1} \times \cdots \times V_d$. In general, for any k attributes $A_{i_1}, A_{i_2}, \ldots, A_{i_k}$ with $i_1 < i_2 < \cdots < i_k$, let $L^{o,r}_{i_1,i_2,\ldots,i_k}$ be the set of objects in the subspace $V_1' \times \cdots \times V_d'$ such that for $j = 1, \ldots, d$,

$$V_j' = \begin{cases} V_j, & if\ j \notin \{i_1, i_2, \ldots, i_k\}, \\ V_j^{o,r}, & otherwise. \end{cases}$$

As can be seen easily, algorithm Mine-Knowledge-Set() resembles the well-known *apriori* algorithm in nature. The difference is that in our case, C_k can not be generated from K_{k-1} directly. It is obtained from projections in C_{k-1}. Procedure *Knowledge-Set-Gen()* also resembles the procedure *apriori-gen()* in the *apriori* algorithm. Like the procedure *apriori-gen()*, Knowledge-Set-Gen(C_{k-1}) first generates the set

$$\begin{aligned} \{L^{o,r}_{i_1,\ldots,i_k} : & L^{o,r}_{i_1,\ldots,i_{k-1}} \in C_{k-1} \& \\ & L^{o,r}_{i_2,\ldots,i_{k-1},i_k} \in C_{k-1} \& \\ & i_1 < i_2 < \cdots < i_k < i_{k+1}\}. \end{aligned}$$

Next, in the *prune* step, it deletes all $L^{o,r}_{i_1,\ldots,i_k}$ such that for some $(k-1)$-subset $\{i_{j_1}, \ldots, i_{j_{k-1}}\}$ of $\{i_1, \ldots, i_k\}$, $L^{o,r}_{i_{j_1},\ldots,i_{j_{k-1}}} \notin C_{k-1}$. The prune step requires testing that all $(k-1)$-subsets of a newly generated k-candidate-attribute set are contained in C_{k-1}. As in the *apriori* algorithm, we can use *hash-tree* to represent C_k to speed up such membership testing. At step 4, we need to find $N_r^{\{A_i\}}(o)$, the r-neighborhood of o in $\{A_i\}$. Since $N_r^{\{A_i\}}(o) \subseteq \prod_{\{A_i\}} L_i^{o,r}$, this can done with the help of $L_i^{o,r}$. Similarly, at step 11, we need to find $N_r^{\{A_{i_1},\ldots,A_{i_k}\}}(o)$, the r-neighborhood of o in $\{A_{i_1}, \ldots, A_{i_k}\}$. Because $N_r^{\{A_{i_1},\ldots,A_{i_k}\}}(o) \subseteq \prod_{\{A_i\}} L^{o,r}_{i_1,\ldots,i_k}$, this can done with the help of $L^{o,r}_{i_1,\ldots,i_k}$

Algorithm Mine-Knowledge-Set() finds all knowledge sets for any given outlier o with respect to r and n. It serves as the kernel of our algorithm FindKS (Find Knowledge Sets) as described in Fig. 2.

```
Algorithm FindKS(D, r, n)
1. S = ∅;
2. O = findOutlier(D, r, n);
3. for every o ∈ O do
4.     S = S ∪ {< Mine − Knowledge − Set(o, r, n), o >};
5. return S
```

Fig. 2. Algorithm FindKS

The procedure *findOutlier* finds outliers in the indication subspace of $\mathcal{D}$ with respect to r and n. We can use the efficient algorithms in [10] for this procedure. Mine-Knowledge-Set.

4 Discovering Patterns of Outliers

Since we have studied the problem of mining knowledge sets for outliers in the previous section we assume in this section that the knowledge set (and hence the knowledge attribute class) for each outlier is given. For each outlier o, let $m(o)$ denote a knowledge set of o. Recall that an outlier may have multiple knowledge sets. Because we assume that all the knowledge sets of the outlier groups are known, we can represent any outlier group G_i as a list of pairs

$$G_i \quad = \quad [< o_{i_1}, m(o_{i_1}) >, \ldots, < o_{i_{s_1}}, m(o_{i_{s_i}}) >].$$

Without loss of generality, we can further assume that attributes in each knowledge set are sorted according to some order.

The central part is to determine, for any given pair of outlier groups G_i and G_j, whether they are similar or not based on definitions 4 and 5. A straightforward method would be for each common knowledge set to the two groups, we scan each group to construct the maximal set of the outliers with that knowledge set. These maximal sets are then used to evaluate the two formulas. A more clever method is to use a counting tree where nodes are attributes and paths are knowledge sets. For each item $< o, m(o) >$ in the above list, we traverse the path $m(o)$. At the leaf node o is stored, together with the group id where o belongs. This structure allows us to store at the leaf node of each path a triple of the form $< G, O, c >$ where G is the group id, O is the set of all the outliers which share the path as one of their knowledge sets, and c is the center of O. (See [5] for more detail.)

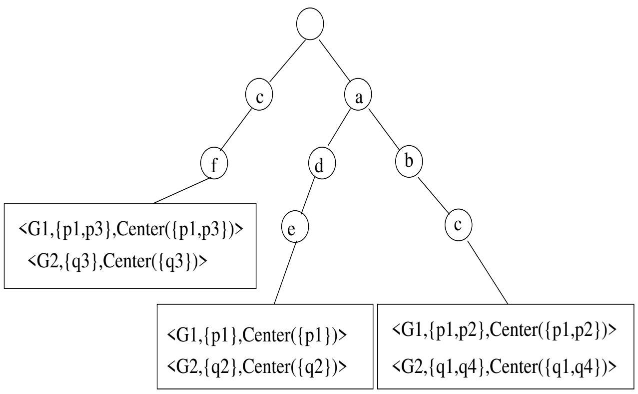

Figure 3. A Knowledge Set Counting Tree

Example 3 *Consider a set of attributes* $\{a, b, c, d, e, f\}$ *and two outlier groups* $G_1 = [< p1, \{a, b, c\} >, < p1, \{a, d, e\} >, < p1, \{c, f\} >, < p2, \{a, b, c\} >, < p3, \{c, f\} >]$ *and* $G_2 = [< q1, \{a, b, c\} >, < q2, \{a, d, e\} >, < q3, \{c, f\} >, < q4, \{a, b, c\} >]$. *A knowledge set counting tree for* G_1 *and* G_2 *is illustrated in Fig. 4.*

Note that in reality, in addition to the patterns themselves, the algorithm should also present to users the knowledge sets for the outliers in the patterns, and the center of each group of outliers that share the same knowledge set. This information serves as descriptions of the patterns that will assist users in analysing the patterns and revealing the causes for the repetitive behaviors of the outliers involved.

5 Empirical Analysis

Because our algorithms are the first in the study of mining outliered patterns, we have no other algorithms to compare with ours. As pointed out in section 3, algorithms in [10] for mining strong outliers and outlying space is related to our work, but they are not applicable to our context. Since the kernel of our algorithms is Mine-Knowledge-Set, we will focus on this algorithm to conduct our empirical analysis. The computing environment of our experiments is a Dell Precision 530 Workstation with dual Xeon 1.5 GHz processors. In order to have fair comparisons, all implementations are based on the same basic data structure.

As in [2], we use Poisson distributions to create synthetic data. First we consider a set of 20 attributes and generate 40 potential knowledge sets for an outlier. Note that those 20 attributes can form 1,048,575 non-empty knowledge sets. The size of a knowledge set is determined by a Poisson distribution with mean equal to 1, 2, 3, 4, 5, or 6, and attributes are randomly assigned to the set. To model that a knowledge set is *"maximal"* in the sense that it contains no other knowledge sets, some fraction of attributes in subsequent knowledge sets are chosen from attributes not contained in previous knowledge sets generated. Next, we generate 10k 20-dimensional data points to form a data sets. The data points are generated with a uniform distribution outside subspaces consisting of the potential knowledge sets. However, in order to smooth the over all distribution of data points, a small fraction of data points are generated within the subspaces consisting of the potential knowledge sets. In general, we use PxDySzk to denote a data set $\mathcal{D}$ with the following properties: it has 40 potential knowledge sets whose size follows a Poisson distribution with mean equal to x; its dimension is y, and it have $z \times 1,000$ many data points.

The performance results of algorithm Mine-Knowledge-Set for the six datasets are shown in Fig. 4 In our experiments, we choose $n = \alpha\%$ of N, where N is the total number of data points. In each of the first six illustrations of Fig. 4 the x-axis indicates decreasing values of α (hence it indicates decreasing values of n), the y-axis indicate execution times of algorithm Mine-Knowledge-Set with $r = 40, 60$, and 80. Three curves are plotted accordingly, which show trends of execution times for a fixed r value and variable decreasing n values.

The last two illustration in Fig. 4 show the scale-up performance of algorithm Mine-Knowledge-Set with respect to the size and the dimension of a dataset. We used the same Poisson distribution with mean equal to 4 to generate datasets and conducted experiments with three parameter settings $(40, 1)$, $(60, 0.75)$ and $(80, 0.5)$, where in each setting the first is the value of r and the second is the percentage α used to compute n. The execution times are normalized with respect to the times for dataset P4D20S10k in the seventh illustration, and for dataset P4D10S10k in the last illustration. As can be seen, the execution times scale linearly in both illustrations.

6 Discussion and Conclusion

Our model assumes that the user has an idea about the roles the attributes play so that he/she is able to subdivide the entire attribute set into the three required subsets. We have argued that this assumption in general is not too strong to be realistic. On the other hand, different users may divide the attribute set differently. In general, these will generate different results. To maximize the satisfaction toward the results, a user may want to use different division schemes and refine the results iteratively. Similarly, the decision on how to define an observation sequence, in particular, the size of an observation spot, is also an iterative process. If the observation spot is too small, the pattern may be hard to find since the number of outliers observed at each spot is not sufficiently large to contribute to a pattern. On the other hand, if it is too large, the pattern found may not reflect the reality.

The task of outlier detection has traditionally taken the view that the outliers are isolated events. This view ignores the possibilities that some outliers be related and form a pattern. In this paper, we propose a framework for modeling outliered patterns. We introduce two kinds of similarities that may exist among outliers, and propose algorithms that can be used to detect patterns based on these similarities.

In the future, we shall apply our model and algorithms on some real data and the results will be reported in the full version of the paper.

ACKNOWLEDGMENTS. Jian Tang is on leave from Memorial University of Newfoundland, Canada. Part of work was done while Zhixiang Chen was visiting Chinese University of Hong Kong. This research was supported by the Hong Kong RGC Research Grant Competitive Bids, Ref: 4179/01E, and the Hong Kong RGC Research Grant Direct Allocation, ID 2050279.

References

[1] C. Aggarwal and P.S. Yu: Outlier detection for high dimensional data, SIGMOD, 2001, pp 37-46.

[2] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A.I. Verkamo: Fast discovery of association rules, in U.M. Fayyad, et al., eds., *Advances in Knowledge Discovery and Data Mining*, pp 307-328, 1996.

[3] A. Arning, R. Agrawal, P. Raghavan: ”A Linear Method for Deviation detection in Large Databases”, KDD, 1996, pp 164 - 169.

[4] M. Breuning, Hans-Peter Kriegel, R. Ng, J. Sander: ”LOF: Identifying density based Local Outliers”, SIGMOD, 2000.

[5] Z. Chen, A. Fu, J. Tang, “Detection of Outliered Patterns”, Tech. Report, Dept. of CSE, Chinese University of Hong Kong, 2002.

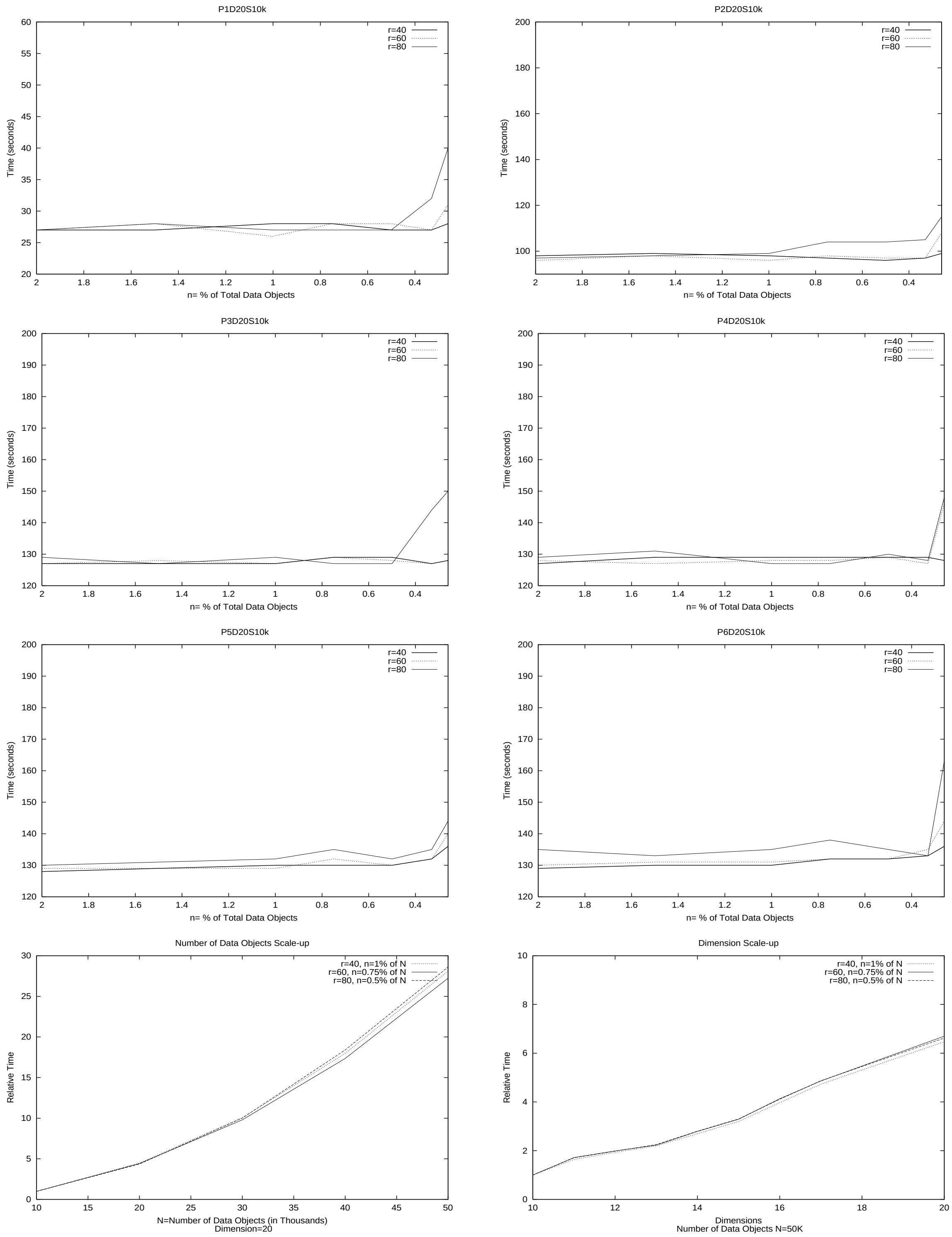

Figure 4. Execution Time and Scale-up Performance

[6] W. DuMouchel, M. Schonlau: ”A Fast Computer Intrusion Detection Algorithm based on Hypothesis Testing of Command transition Probabilities”, KDD, 1998, pp. 189 - 193.

[7] M. Ester, H. Kriegel, J. Sander, X. Xu: ”A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise”, KDD, 1996, pp 226 - 231.

[8] T. Fawcett, F. Provost: ”Adaptive Fraud Detection”, Data Mining and Knowledge Discovery Journal, Kluwer Academic Publishers, Vol. 1, No. 3, 1997, pp 291 - 316.

[9] E. Knorr, R. Ng: ”Algorithms for Mining Distance based Outliers in Large Datasets”, VLDB, 1998, pp 392 - 403.

[10] E. Knorr, R. Ng: ”Finding Intensional Knowledge of Distance-based Outliers”, VLDB, 1999, pp 211 - 222.

[11] R. Ng, J. Han: ”Efficient and Effective Clustering Methods for Spatial Data Mining”, VLDB, 1994, pp 144 - 155.

[12] S. Ramaswamy, R. Rastogi, S. Kyuseok: ”Efficient Algorithms for Mining Outliers from Large Data Sets”, SIGMOD, 2000, pp 427 - 438.

[13] G. Sheikholeslami, S. Chatterjee, A. Zhang: ”WaveCluster: A multi-Resolution Clustering Approach for Very Large Spatial Databases”, VLDB, 1998, pp 428 - 439.

[14] S. Stolfo, W. Fan, W. Lee, A. Prodromidis, P. Chan: “Cost-based Modeling for Fraud and Intrusion Detection: Results from the JAM Project”, In Proc. of DARPA Information Survivability Conference & Exposition - Vol. 2, 2000, pp 1130 - 1144.

[15] J. Tang, Z. Chen, A. Fu, D. Cheung, “A Robust Outlier Detection Scheme in Large Data Sets”, PAKDD, 2002.

[16] T. Zhang, R. Ramakrishnan, M. Linvy: ”BIRCH: An Efficient Data Clustering Method for Very Large Databases”, SIGMOD, 1996, pp 103 - 114.