Intelligent Data Analysis 19 (2015) 1171–1192
DOI 10.3233/IDA-150764
IOS Press

Mining multidimensional contextual outliers from categorical relational data

Guanting Tang[a,*], Jian Pei[a], James Bailey[b] and Guozhu Dong[c]
[a]*School of Computing Science, Simon Fraser University, Burnaby, BC, Canada*
[b]*Department of Computing and Information Systems, The University of Melbourne, Melbourne, Australia*
[c]*Department of Computer Science and Engineering, Wright State University, Dayton, OH, USA*

Abstract. A wide range of methods have been proposed for detecting different types of outliers in both the full attribute space and its subspaces. However, the interpretability of outliers, that is, explaining in what ways and to what extent an object is an outlier, remains a critical issue.

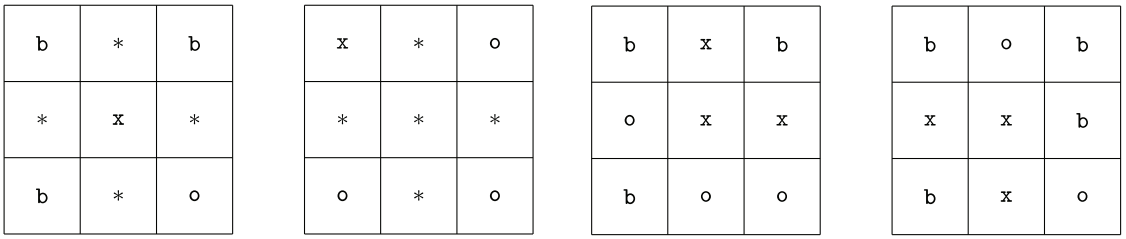
In this paper, we focus on improving the interpretability of outliers. Particularly, we develop a notion of *multidimensional contextual outliers* to model the context of an outlier, and propose a framework for contextual outlier detection. Intuitively, a contextual outlier is a small group of objects that share strong similarity with a significantly larger reference group of objects on some attributes, but deviate dramatically on some other attributes. In contextual outlier detection, we identify not only the outliers, but also their associated contextual information including (1) comparing to what reference group of objects the detected object(s) is/are an outlier; (2) the attributes defining the unusual behavior of the outlier(s) compared against the reference group; (3) the population of similar outliers sharing the same context; and (4) the outlier degree, which measures the population ratio between the reference group and the outlier group. We present an algorithm and conduct extensive experiments to evaluate our approach.

Keywords: Outlier detection, context, categorical data, relational data

1. Introduction

Outlier detection is an important data mining task with broad applications, such as business intelligence and security monitoring. As to be reviewed in Section 5, a wide range of methods have been proposed to detect different types of outliers on various kinds of data. Interpretability of outliers, however, remains a serious concern. More often than not, an analyst may want to see not only the outliers detected, but also insightful explanations about the outliers. Particularly, an analyst may want to know, for an outlier, a reference group of objects which the outlier deviates from in some aspects and shares similarity with in some other aspects, and a set of features manifesting the outlier's unusual/deviating behavior, the outlier degree, and the other similar outliers sharing the same context. Such contextual information can help an analyst to better understand and investigate individual outliers and propose action plans suitable for such outliers.

*Corresponding author: Guanting Tang, School of Computing Science – Simon Fraser University, 8888 University Drive, Burnaby, BC, V5A 1S6, Canada. Tel.: +1 778 782 6851; Fax: +1 778 782 3045; E-mail: gta9@cs.sfu.ca.



(a) An outlier group cover: 2 (b) A reference group cover: 35 (c) Outlier instance c_1 (d) Outlier instance c_2

Fig. 1. A contextual outlier in the tic-tac-toe data set.

Example 1 (Motivation 1). Figure 1 shows a contextual outlier found from a data set containing all the 958 possible board configurations at the end of tic-tac-toe games. (For details on this data set, see Section 6.) Here, "x" and "o" mark positions occupied by the x and o players respectively, "b" marks positions not occupied by any player, and "$*$" is a wildcard matching any value among "x", "o", and "b".

Figures 1(c) and (d) show two rare situations, where 3 of the 4 corners are not occupied by any players at the end of the games. These two rare situations can be summarized into an outlier group using wildcard symbol "$*$", as shown in Fig. 1(a). To manifest the outlier, Fig. 1(b) shows a reference group where "x" occupies the left-top corner and "o" the other three corners. The reference group and the outlier group share the common feature that the right-bottom corner is taken by "o". The reference group, which is matched by 35 configurations, is dramatically more popular than the outlier group, which is matched by only two configurations.

The patterns of the outlier and reference groups suggest that it may be a good strategy to occupy as many corners in the game, since 33 out of the 35 configurations matching Fig. 1(b) are won by "o". □

Example 2 (Motivation 2). In an insurance company, a fraud analyst may not only want to find out all fraud suspects, but also the insightful reasons about why the identified cases are suspicious. Particularly, for a small group of fraud suspects, the analyst may want to know in what aspects they share similarity with the normal patients and in what aspects they deviate dramatically from the normal patients.

For example, a scenario interesting to the analyst may look like "Among the patients who consume narcotic drugs in the region under investigation, a small group of 10 patients purchasing narcotic drugs from more than 60 different pharmacies is an outlier group, comparing to a reference group of 3000 patients buying narcotic drugs from fewer than 5 different pharmacies." Among the patients who take narcotic drugs in the Greater Vancouver area, the small group of 10 patients have a very different purchase pattern compared against the majority. The 10 patients in the outlier group have a high probability to submit fraud claims to the insurance company. □

The outlier group and the reference group cannot be found by the existing outlier detection methods. To the best of our knowledge, even though an existing outlier detection method can detect the outlier group, it cannot find the reference group that clearly manifests the outliers.

We argue that the contextual information about outliers should be an integral component in the outlier detection process. Unfortunately, most of the existing outlier detection methods do not provide rich and detailed contextual information for outlier analysis.

In this paper, we tackle the problem of contextual outlier detection on categorical data, and we do so by making three main contributions. First, we develop a notion of multidimensional contextual outliers to model the context of an outlier. Intuitively, a contextual outlier is a small group of objects that share similarity, on some attributes, with a significantly larger reference group of objects, but deviate

dramatically on some other attributes. An example is: "Among the computer science senior undergraduate students at University X, a small group of 3 students not enrolled in the data structure course is an outlier against the reference group of 128 students enrolled in the course."

In contextual outlier detection, we identify not only the outliers, but also their associated contextual information including (1) comparing to what reference group of objects the detected object(s) is/are an outlier; (2) the attributes defining the unusual behavior of the outlier(s) compared against the reference group; (3) the population of similar outliers sharing the same context; and (4) the outlier degree, which measures the population ratio between the reference normal group and the outlier group.

Second, there may exist many contextual outliers in a data set, and some of them can be very similar or even "equivalent" to each other. It is clearly important to identify and reduce redundancy among outliers in order to assist users to effectively analyze the outliers. We develop an approach to systematically identify redundant contextual outliers and propose a concise representation of contextual outliers.

Third, we design a simple, yet effective algorithm that leverages the state-of-the-art data cube computation techniques. The focus of our method is to find outliers together with their contextual information. We conduct extensive experiments to evaluate the feasibility and usefulness of our approach.

The rest of the paper is organized as follows. We propose the notion of contextual outliers in Section 2, and give a general analysis of the collection of contextual outliers in Section 3. We develop a contextual outlier detection algorithm in Section 4. In Section 5, we review related work and highlight the differences between our work and existing methods. We evaluate our approach in Section 6. We conclude the paper and discuss possible extensions in Section 7.

2. Contextual outliers

In this paper, we consider outlier detection on multidimensional categorical data. Specifically, we consider a *base table* $T(A_1, \ldots, A_n)$, where $A_1, \ldots, A_n$ are categorical attributes with finite domains. We assume that each object is represented by a tuple in the base table and is associated with an identifier tid, which is used as a reference to the object only, and does not carry any other meaning. For an object $t \in T$, let $t.A_i$ and $t.tid$ denote the value of t on attribute A_i $(1 \leqslant i \leqslant n)$ and the identifier of t respectively.

A *subspace* is a subset of attributes. In order to summarize a group of objects, we add a wildcard meta-symbol $*$ to the domain of every attribute A_i $(1 \leqslant i \leqslant n)$. Symbol $*$ matches any possible values in the domain. A *group-by tuple* (or *group* for short) is a tuple $g = (g.A_1, \ldots, g.A_n)$ such that $g.A_i$ takes either a value in the domain of A_i or meta-symbol $*$. The *cover* of g is the set of objects in T matching g, that is, $cov(g) = \{t \in T \mid t.A_i = g.A_i \text{ for all } i \text{ such that } (1 \leqslant i \leqslant n \& g.A_i \neq *)\}$. The set $space(g) = \{A_i \mid 1 \leqslant i \leqslant n, g.A_i \neq *\}$ is called the *subspace* of g. And the set $avs(g) = \{A_i = g.A_i \mid 1 \leqslant i \leqslant n, g.A_i \neq *\}$ is called the *non-$*$ attribute-value set* (*AVS* for short) of group g. Please note that in the definition of $avs(g)$, "$A_i =$" is text and $g.A_i$ is a value (of A_i). Example 2 contains an example. For an AVS V, we overload the operator $space(\cdot)$ by defining $space(V) = \{A_i \mid A_i \text{ occurs in } V\}$. Thus, $space(avs(g)) = space(g)$ always holds.

For two distinct groups g_1 and g_2, g_1 is an *ancestor* of g_2, and g_2 a *descendant* of g_1, denoted by $g_1 \succ g_2$, if $avs(g_1) \subset avs(g_2)$, that is, for every attribute A_i $(1 \leqslant i \leqslant n)$ such that $g_1.A_i \neq *$, we have $g_1.A_i = g_2.A_i$. We write $g_1 \succeq g_2$ if $g_1 \succ g_2$ or $g_1 = g_2$.

Property 1 (Monotonicity). For two groups g_1 and g_2 such that $g_1 \succ g_2$, $cov(g_1) \supseteq cov(g_2)$. □

Table 1
A table T of a set of customers

c-id	*City*	*Type*	*Branch*	*Package*
C1	L_1	T_1	B_1	Gold
C2	L_1	T_1	B_1	Silver
C3	L_1	T_1	B_1	Gold
C4	L_1	T_1	B_2	None
C5	L_2	T_2	B_1	Silver
C6	L_1	T_2	B_1	Gold

Example 3. Consider the base table T in Table 1, which contains the cities, customer-types, home-branches, and service packages of some investment service customers. For group $g = (L_1, T_1, *, *)$, $cov(g) = \{C1, C2, C3, C4\}$, $space(g) = \{\text{city}, \text{type}\}$, and $avs(g) = \{\text{city} = L_1, \text{type} = T_1\}$.

Let $g' = (L_1, T_1, B_1, *)$. Then, g is an ancestor of g' and g' a descendant of g, that is, $g \succ g'$. Moreover, $cov(g') = \{\text{C1, C2, C3}\} \subset cov(g)$. □

We are now ready to define contextual outliers. Intuitively, for a group of outlier objects, the contextual information consists of a group of reference objects that manifest the outlier group in a subspace. The comparison of the two groups in population size is also included.

Definition 1 (Contextual outlier). Let T be a base table, and r, o be two groups such that $space(r) = space(o) \neq \emptyset$. Given an *outlier degree threshold* $\Delta > 1$, the pair (r, o) is a *contextual outlier* if the *outlier degree* $deg(r, o) = \frac{|cov(r)|}{|cov(o)|} \geqslant \Delta$. We call r the *reference group*, o the *outlier group*, $out(r, o) = space(r) - space(cond(r, o))$ the *outlier subspace*, and $cond(r, o) = avs(r) \cap avs(o)$ the *shared AVS*. It is possible that $cond(r, o)$ is empty. □

The shared AVS $cond(r, o)$ provides a context subspace for the outlier analysis about o. The objects in groups o and r belong to the same context subspace, that is, they take the same values on those attributes that occur in $cond(r, o)$. If $cond(r, o) = \emptyset$, r and o do not share any common features. In such a special case, o is a global outlier that is small in population and different from a large reference group r in space $space(o) = space(r)$.

The reference group r indicates the normal or dominating objects to which o is compared. The outlier group o and the outlier subspace $out(r, o)$ indicate the outlier objects $cov(o)$ and the attributes that manifest the deviation of o from r. The outlier degree measures how exceptional the group o is when compared to r. The larger the outlier degree is, the more outlying o is.

Example 4. In T (Table 1), let the outlier degree threshold be $\Delta = 2$. Then, $(r, o) = ((L_1, T_1, B_1, *), (L_1, T_1, B_2, *))$ is a contextual outlier, where $(L_1, T_1, B_1, *)$ is the reference group, $\{\text{city} = L_1, \text{type} = T_1\}$ is the shared AVS, $(L_1, T_1, B_2, *)$ is the outlier group, $\{\text{branch}\}$ is the outlier subspace, and the outlier degree is $deg(r, o) = 3$.

One object may be an outlier in more than one context. For example, customer C4 is an outlier in the above context and $((*, T_1, *, \text{Gold}), (*, T_1, *, \text{None}))$. □

Using only an outlier degree threshold may lead to many contextual outliers, since many groups of very small cover size, such as 1, may be identified as outliers. We will address this issue in Section 3.3.

The definitions and frequently used notations are summarized in Table 2.

3. Contextual outlier analysis

Enumerating all possible contextual outliers in a base table is ineffective due to three reasons. First,

Table 2
Summary of definitions and frequently used notations

Notation	Description
$T(A_1, \ldots, A_n)$	A base table, where $A_1, \ldots, A_n$ are categorical attributes with finite domains.
$*$	A widecard meta-symbol, matches any possible values in a domain.
$g(=(g.A_1, \ldots, g.A_n))$	A group-by tuple (or group for short), $g.A_i$ takes either a value in the domain of A_i or meta-symbol $*$.
$cov(g)$	The cover of g, the set of objects in T matching g, that is, $cov(g) = \{t \in T \mid t.A_i = g.A_i$ for all i such that $(1 \leqslant i \leqslant n \& g.A_i \neq *)\}$.
$space(g)$	The subspace of g, $space(g) = \{A_i \mid 1 \leqslant i \leqslant n, g.A_i \neq *\}$.
$avs(g)$	The non-$*$ attribute-value set (AVS for short) of g, $avs(g) = \{A_i = g.A_i \mid 1 \leqslant i \leqslant n, g.A_i \neq *\}$.
$g_1 \succ g_2$	g_1 is an ancestor of g_2, and g_2 a descendant of g_1, if $avs(g_1) \subset avs(g_2)$.
(r, o)	A contextual outlier, where r is the reference group and o is the outlier group.
$deg(r, o)$	The outlier degree of contextual outlier (r, o), $deg(r, o) = \frac{\lvert cov(r)\rvert}{\lvert cov(o)\rvert}$.
Δ	The outlier degree threshold.
$cond(r, o)$	The shared AVS of contextual outlier (r, o), $cond(r, o) = avs(r) \cap avs(o)$.
$out(r, o)$	The outlier subspace of contextual outlier (r, o), $out(r, o) = space(r) - space(cond(r, o))$.
$\alpha(r, o)$	The significance of a contextual outlier (r, o) (Definition 6).
$\Phi(g_1, g_2)$	The assembly of an ordered pair (g_1, g_2) (Definition 7).

some contextual outliers are highly similar and even equivalent to each other. Outlier detection is often followed by business actions, which are expensive to perform. Including redundant outliers may lead to unnecessary extra cost in the "analysis and actions" process and may also overwhelm users. Second, it is informative and important to analyze the relationships among outliers. The contextual outliers in a data set may not be independent of each other. A systematic analysis of the relationships among outliers may, for example, support business decisions based on the relationships among a collection of outliers, instead of on individual outliers only. Third, as mentioned at the end of Section 2, using an outlier degree threshold alone may lead to many insignificant contextual outliers, which may overwhelm users in practice. In this section, we conduct contextual outlier analysis to address the above three aspects.

3.1. Redundancy removal using closures

We immediately observe the following:

Lemma 1 (Non-closure attributes). For two contextual outliers (r_1, o_1) and (r_2, o_2) in a base table T, if $r_1 \succ r_2$, $o_1 \succ o_2$, $cov(r_1) = cov(r_2)$, and $cov(o_1) = cov(o_2)$, then $deg(r_1, o_1) = deg(r_2, o_2)$. □

In the two contextual outliers (r_1, o_1) and (r_2, o_2) in Lemma 1, the two groups r_1 and r_2 capture the same set of objects. Hence (r_1, o_1) is redundant given (r_2, o_2) or vice versa. Since $r_1 \succ r_2$, r_2 contains some extra attributes in addition to those in r_1. Hence r_2 is more informative and descriptive than r_1 as a reference group. It is better to include (r_2, o_2) for outlier analysis.

Definition 2 (Closure group/outlier). Given a base table T, a group g is a *closure group* if for any descendant group $g' \prec g$, $cov(g') \subset cov(g)$. (r, o) is called a *closure outlier* if there does not exist another contextual outlier (r', o') such that $r' \prec r$, $o' \prec o$, $cov(r) = cov(r')$, and $cov(o) = cov(o')$. □

Example 5 (Closure group/outlier). In table T (Table 1), for contextual outliers $u_1 = ((*, T_1, B_1, *), (*, T_1, B_2, *))$ and $u_2 = ((L_1, T_1, B_1, *), (L_1, T_1, B_2, *))$, since $cov((*, T_1, B_1, *)) = cov((L_1, T_1, B_1, *))$ and $cov((*, T_1, B_2, *)) = cov((L_1, T_1, B_2, *))$, u_1 is redundant given u_2. The reference group in u_2 is a closure one. It can be verified that u_2 is a closure outlier. □

We now establish a relationship between closure groups and closure outliers.

Theorem 1 (Closure group/outlier). Contextual outlier (r, o) is a closure outlier if and only if either r or o is a closure group.

Proof. (If) There are two cases. In the first case, r is a closure group. Then, there does not exist (r', o') such that $r' \prec r$ and $cov(r) = cov(r')$. In the second case, o is a closure group. Then, there does not exist (r', o') such that $o' \prec o$ and $cov(o) = cov(o')$. In both cases, (r, o) satisfies the definition of closure outliers.

(Only-if) We prove by contradiction. Assume that (r, o) is a closure outlier, but neither r nor o is a closure group. Then, there exist r' and o' such that $r' \prec r$, $o' \prec o$, $cov(r) = cov(r')$ and $cov(o) = cov(o')$. This contradicts the assumption that (r, o) is a closure outlier. □

We can generalize the idea of closure outliers to reduce redundancy further.

Example 6. Consider T (Table 1) and the outlier degree threshold $\Delta = 3$. C4 is in two contextual outliers, $u_1 = ((L_1, *, B_1, *), (L_1, *, B_2, *))$ with $deg(u_1) = 4$, and $u_2 = ((L_1, T_1, B_1, *), (L_1, T_1, B_2, *))$ with $deg(u_2) = 3$. Comparing to u_1, the reference group in u_2 is more specific and thus is closer to the outlier group and more informative. □

We generalize the observation in Example 6 and introduce the notion of tight outlier, which is essentially an outlier and its most specific reference group.

Definition 3 (Tight outlier). Let T be a base table and (r, o) a contextual outlier with respect to outlier degree threshold Δ. We call (r, o) a *tight outlier* if there does not exist another outlier (r', o') with respect to threshold Δ such that $r \succ r'$ and $cov(o) = cov(o')$. □

Corollary 1. A tight outlier is a closure outlier. □

As shown in Example 6, not every closure outlier is tight.

3.2. Relationships among outliers

An outlier group may have more than one reference group in the same outlier space. Consider two contextual outliers (r_1, o) and (r_2, o) such that $space(r_1) = space(r_2)$, $cond(r_1, o) = cond(r_2, o)$. The two outliers have the same outlier subspace, since $out(r_1, o) = space(r_1) - space(cond(r_1, o)) = space(r_2) - space(cond(r_2, o)) = out(r_2, o)$. If $|cov(r_1)| > |cov(r_2)|$, r_1 is a stronger reference group than r_2 and o deviates more from r_1 than from r_2. Thus, (r_2, o) is redundant given (r_1, o). In words, we only need to report the reference group that an outlier deviates most. This is modeled in the following notion of strong outliers.

Definition 4 (Strong outlier). A contextual outlier (r, o) is a *strong outlier* if there does not exist another outlier (r', o') such that $space(r) = space(r')$, $cond(r, o) = cond(r', o')$, $cov(o) = cov(o')$, and $|cov(r')| > |cov(r)|$. □

Example 7 (Strong outlier). For T (Table 1) and the outlier degree threshold $\Delta = 2$, consider contextual outliers $u_1 = ((*, *, *, \text{Gold}), (*, *, *, \text{None}))$ with $deg(u_1) = 3$, and $u_2 = ((*, *, *, \text{Silver}), (*, *, *, \text{None}))$ with $deg(u_2) = 2$. u_1 is stronger than u_2 in outlier degree, and they have the same shared AVS. u_2 is redundant given u_1. In fact, u_1 is a strong outlier. □

A group of objects may appear in multiple strong and tight contextual outliers, such as $C4$ in T (Table 1). To comprehensively understand an outlier group, we can group all contextual outliers together with respect to the same set of outlier objects.

Definition 5 (Contextual outlier group). Given an outlier degree threshold $\Delta > 1$, a set of contextual outliers $\{(r_1, o_1), \ldots, (r_l, o_l)\}$ is called a *contextual outlier group* if (1) $(r_1, o_1), \ldots, (r_l, o_l)$ are all strong and tight contextual outliers; (2) $cov(o_1) = \ldots = cov(o_l)$; and (3) there does not exist a proper superset of contextual outliers satisfying the above two requirements. We denote by $Context_\Delta(o)$ the contextual outlier group $\{(r_1, o_1), \ldots, (r_l, o_l)\}$ such that $cov(o_1) = \ldots = cov(o_l) = o$.

The *(geometric) average degree* of a contextual outlier group $Context_\Delta(o) = \{(r_1, o_1), \ldots, (r_l, o_l)\}$ is $deg(o, \Delta) = \sqrt[l]{\prod_{i=1}^{l} deg(r_i, o_i)}$. In particular, when $l = 0$, we have $deg(o, \Delta) = 1$. □

We use the geometric average degree instead of the arithmetic average to aggregate the outlierness of an outlier group in multiple contexts because the contexts in general may not be completely independent. The geometric average prefers outlier groups whose outlier degrees in multiple contexts are more balanced against those that are extremely outlying in only a small number of contexts, but not the others. Of course other aggregate functions might also be adopted.

3.3. Finding significant outliers

As mentioned earlier, using only an outlier degree threshold may lead to many contextual outliers, since many groups of very small cover size and with a small difference from large reference groups, such as on one attribute, may be identified as outliers. To tackle the problem and avoid overwhelming users by many insignificant outliers, we need to test the statistical significance of contextual outliers. In other words, we use statistical test to avoid reporting groups that are caused by random noise.

Intuitively, we use the global distribution in the base table as the background distribution by ignoring the tuples taking the same values with the reference group, and measure the statistical significance of the outlier group. For global outliers, we assume the uniform distribution as the background.

Definition 6 (Outlier significance). Let (r, o) be a contextual outlier in a base table T. The *background distribution* of (r, o) in subspace $out(r, o)$ is the distribution of tuples in $T - \{t \in T \mid t.A = r.A, \forall A \in out(r, o)\}$, if $cond(r, o) \neq \emptyset$, and the uniform distribution, otherwise.

The *significance* of a contextual outlier (r, o), denoted by $\alpha(r, o)$, is the p-value of the null hypothesis H_0: o has been generated from the tuples in $cov(cond(r, o)) - cov(r)$ according to the background distribution in space $out(r, o)$. □

In the above definition, $cov(cond(r, o)) - cov(r)$ is the set of tuples matching reference group r in subspace $cond(r, o)$ but not in subspace $out(r, o)$. The smaller the p-value $\alpha(r, o)$ is, the more significant the outlier is. The following shows the details of significance calculation.

Consider a contextual outlier (r, o). Let p_o be the probability of AVS $avs(o) - avs(r)$ in the background distribution. If the background distribution is uniform, then

$$p_o = \frac{1}{\prod_{A \in space(out(r,o))}(|A| - 1)}.$$

Let $m = |T - \{t \in T \mid t.A = r.A, \forall A \in out(r, o)\}|$ be the number of tuples matching r in subspace $cond(r, o)$, but not in subspace $out(r, o)$. Then, we have

$$\alpha(r, o) = \sum_{i=|cov(o)|}^{m} p_o^i (1 - p_o)^{m-i}$$

Example 8 (Outlier significance). Given T (Table 1), an outlier degree threshold $\Delta = 3$, and a significance threshold $s = 0.01$. We perform the outlier significance test on a strong and tight contextual outlier, $u_1 = ((L_1, *, B_1, Gold), (L_1, *, B_2, None))$ with $deg(u_1) = 3$ and $out(r, o) = \{$*Branch, Package*$\}$.

According to the previous discussion, we have

$$p_o = \frac{1}{(|Branch| - 1)(|Package| - 1)} = \frac{1}{(2-1)(3-1)} = \frac{1}{2}, \;\; m = 6 - 3 = 3, \;\; i = 1,$$

and thus

$$\alpha(r, o) = \left(\frac{1}{2}\right)^1 \left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2 \left(\frac{1}{2}\right)^1 + \left(\frac{1}{2}\right)^3 \left(\frac{1}{2}\right)^0 = \frac{3}{8} = 0.375.$$

Since $\alpha(r, o) > s$, u_1 is not a significant contextual outlier here. □

Based on the above discussion, we can define the problem of contextual outlier detection as, given a base table T, an outlier degree threshold $\Delta > 1$, and a significance threshold $s > 0$, find all strong and tight context outliers (r, o) such that $\alpha(r, o) < s$.

4. Detection algorithms

In this section, we develop an algorithm for contextual outlier detection. We observe that group-bys are essential units in both data cube computation and contextual outlier analysis, so we can exploit state-of-the-art data cube techniques in detecting contextual outliers.

Our method is inspired by Theorem 1. Since every closure contextual outlier must have either the reference group or the outlier group as a closure group, we can find all closure groups in the base table first, and then use the closure groups to assemble contextual outliers.

Finding closure groups and closure patterns has been well studied in frequent pattern mining [28,33] and data cube computation [21]. Given a base table T, we can adopt a state-of-the-art algorithm, such as the DFS algorithm in [21], to find all closure groups. Therefore, hereafter, we focus on how to use closure groups to assemble contextual outliers.

We define the following assembly operation to extract the common attributes in two closure groups.

Definition 7 (Assembly). Given two closure groups g_1 and g_2 on a base table T, the *assembly* of the ordered pair (g_1, g_2), denoted by $\Phi(g_1, g_2) = (g_1', g_2')$, is the ordered pair of groups such that for every attribute $A \in space(g_1) \cap space(g_2)$, $g_1'.A = g_1.A$ and $g_2'.A = g_2.A$; for any other attribute $B \in (T - space(g_1) \cap space(g_2))$, $g_1'.B = g_2'.B = *$. We call g_1' and g_2' the *reference group* and the *outlier group* of the assembly, respectively. □

Example 9 (Assembly). In T in Table 1, for closure groups $g_1 = (L_1, T_1, B_1, *)$ and $g_2 = (*, T_2, B_1, *)$, the assembly of the ordered pair $\Phi(g_1, g_2) = ((*, T_1, B_1, *), (*, T_2, B_1, *))$. □

It is easy to verify the following, which shows that contextual outliers are fixpoints for Φ.

Corollary 2. For any contextual outlier (r, o), $\Phi(r, o) = (r, o)$. □

Since the assembly operator takes an ordered pair of closure groups as the input and produces an ordered pair as the output, in general, $\Phi(g_1, g_2) \neq \Phi(g_2, g_1)$. Instead, the reference (outlier) group of the assembly $\Phi(g_1, g_2)$ is the outlier (reference) group of $\Phi(g_2, g_1)$. The assembly operation has the following nice property.

Algorithm 1 COD: the contextual outlier detection algorithm.

Require: $\mathcal{G}$: the complete set of closure groups; Δ: the outlier degree threshold; and s: the significance threshold

Ensure: the set of tight and strong contextual outliers

1: let $l = \max_{g \in \mathcal{G}}\{|cov(g)|\}$;
2: let O be the set of contextual outliers; set $O = \emptyset$;
3: **for** each closure group o such that $|cov(o)| \leqslant \frac{l}{\Delta}$ **do**
4: create a set L of tight and strong contextual outliers, set $L = \emptyset$;
5: **for** each closure group r such that (1) $|cov(r)| \geqslant \Delta|cov(o)|$; (2) $space(r) \subseteq space(o)$ or $space(r) \supseteq space(o)$; and (3) $\alpha(r, o) \leqslant s$ **do**
6: let $(r', o') = \Phi(r, o)$;
7: **if** $cov(r) = cov(r')$ **and** $cov(o) = cov(o')$ **and** there is no outlier in L that is stronger or tighter than $\Phi(r, o)$ **and** $\alpha(\Phi(r, o)) \leqslant s$ **then**
8: insert $\Phi(r, o)$ into L;
9: remove from L any outliers that $\Phi(r, o)$ is stronger or tighter than;
10: **end if**
11: **end for**
12: $O = O \cup L$;
13: **end for**
14: **return** O;

Corollary 3. For every tight and strong contextual outlier (r, o), there exists a unique ordered pair of closure groups (r', o') such that $\Phi(r', o') = (r, o)$, $cov(r) = cov(r')$, and $cov(o) = cov(o')$.

Proof. According to Theorem 1, either r or o must be a closure group. Without loss of generality, let us assume r is a closure group, and thus let $r' = r$. If o is not a closure group, there exists a unique closure group o' such that $cov(o) = cov(o')$ and $space(o) \subseteq space(o')$. Therefore, $\Phi(r', o') = \Phi(r, o') = \Phi(r, o)$. Using Corollary 2, we have $\Phi(r', o') = \Phi(r, o) = (r, o)$. □

Corollary 3 enables us to assemble closure groups into contextual outliers. Algorithm 1 presents the pseudocode of our detection method, COD (for Contextual Outlier Detection), which is explained in detail as follows.

For each closure group o, we consider all the other closure groups r such that $\Phi(r, o)$ is a contextual outlier. Obviously, $|cov(o)|$ cannot be larger than $\frac{l}{\Delta}$, where l is the largest cover size among all closure groups (calculated in Line 1). For each of such closure groups o, we iterate over all the other closure groups r such that $|cov(r)| \geqslant \Delta|cov(o)|$ and $\alpha(r, o)$ passes the significance threshold (the inner loop, Lines 5–11). To facilitate the access of closure groups according to their cover size, we sort all the closure groups in cover size ascending order.

For a pair of closure groups (r, o), we compute the assembly $\Phi(r, o) = (r', o')$. We only consider the contextual outliers $\Phi(r, o) = (r', o')$ such that $cov(r) = cov(r')$ and $cov(o) = cov(o')$. Otherwise, the outlier will be considered by some other closure groups according to Corollary 3 (the first two conditions in Line 7). If $\Phi(r, o)$ is strong and tight given the other contextual outliers using o as the outlier group (the second last condition in Line 7), and is statistically significant (that is, the significance is less than or equal to the significance threshold) (the last condition in Line 7), then we keep $\Phi(r, o)$ (Line 8), and use it to remove those contextual outliers found before that are not as strong or tight as $\Phi(r, o)$ (Line 9).

After the inner loop, all contextual outliers using o as the outlier group are computed, and the tight and strong ones are kept in L. Those tight and strong outliers are moved to O for outputting later. The iteration continues until all closure groups that may be outlier groups are examined.

COD checks every pair of closure groups (r, o) such that $\frac{|cov(r)|}{|cov(o)|} \geqslant \Delta$. The correctness follows with Corollary 3. Moreover, the algorithm is cubic in time with respect to the number of closure groups in T, that is, $|\mathcal{G}|$ in the algorithm. The problem of computing closure groups in a base table is #P-complete by a polynomial reduction from the #P-complete problem of frequent maximal pattern mining [37]. Our method is overall pseudo-polynomial.

5. Related work

Outlier detection has been extensively studied in literature from a number of different angles, such as statistical methods, proximity based methods, clustering based methods, supervised methods, semi-supervised methods, unsupervised methods, and so on.

Some of these studies are dedicated to identifying outliers from categorical data sets. Frequent pattern mining based methods [4,35], proximity based methods [2] and probabilistic model based methods [10, 11] are commonly used methods in outlier detection tasks for categorical data. Other methods, such as statistical methods and density based methods can be effectively adopted in categorical data by using proper data transformation approaches.

A survey of the existing methods is far beyond the capacity of this paper. We refer readers to some recent excellent surveys on the topic [1,8,19,24,25,29].

Given a set of data objects, most of the existing outlier detection studies focus on finding outlier objects that are significantly different from the rest of the data set. The context of outliers, which provides insightful and critical information for outlier analysis, is often missed. We address this issue in this study. Our method not only detects outliers, but also automatically identifies the corresponding contextual information that manifests the outliers at the same time.

To the best of our knowledge, only very few existing studies consider context in outlier detection. Song et al. [31] proposed the notion of conditional outliers to model the outliers manifested by a set of behavioral attributes (e.g. temperature) conditionally depending on a set of contextual attributes (e.g. longitude and latitude). The behavioral attributes and the contextual attributes are pre-defined. Our contextual outlier model does not need pre-defined behavioral and contextual attributes. Instead, it automatically identifies shared AVSs. Moreover, reference groups are not modeled in [31]. A mixture model is used in [31] to mine conditional outliers, which is infeasible in our model since here the subspaces are not pre-defined and change from one outlier to another. Valko et al. [32] detected conditional anomalies using a training set of labeled examples with possible label noise, which is different from our work here that no labeled data is assumed.

Li et al. [35] introduced a hypergraph-based outlier detection test (HOT) to identify outliers and explanation. Given a dataset and a minimum support threshold, a hypergraph is built based on the frequent itemsets in the dataset. Particularly, a vertex is a data object, and a hyperedge denotes a group of objects containing a frequent itemset. A deviation score has been designed to calculate the outlyingness of a data object in a certain attribute with respect to a hyperedge. Given a deviation threshold θ, a data object o is an outlier in attribute A with respect to a hyperedge he, if the deviation score of o in A with respect to he is lower than θ. This study is different from ours. First, the detected ouliers are very different. Outliers detected by HOT [35] are single data objects (vertexes). Outliers detected by our methods are group-by

tuples (hyperedges). Second, the explanations concerning the outlyingness are different. HOT [35] uses just one attribute to characterize the outlyingness of an outlier, while our method uses a set of attributes. Thus, the problem and the framework proposed by our work are more general than the ones proposed by HOT.

Wang and Davidson [34] used random walks to find context and outliers. Their problem settings and solution are fundamentally different from ours. First, a similarity function is needed in [34] to transform data to a similarity matrix and thus a graph on which random walks can be conducted. Second, due to the use of a similarity measure, the context and the outliers cannot be summarized by attributes and subspaces. Overall, due to the different problem settings and objectives, the random walk method in [34] and ours in this paper are orthogonal.

Kriegel et al. [20] proposed a method that detects an outlier with reference to the axis parallel subspace spanned by its neighbors. Müler et al. [26] proposed a technique for ranking outliers based on their degree of deviation in different subspace projections. While these studies also focus on subspace context for outliers, there are two key differences from our work. First, these studies focus on continuous datasets, while our focus is on categorical relational data. Second, our work proposes techniques for concise descriptions of sets of outliers. We also provide contextual descriptions for the outliers that are detected.

Recently, Smet and Vreeken [30] developed an outlier detection method OC^3, which assumes that outliers are generated by a distribution different from that generates the normal objects, and uses minimum description length (MDL) to detect outliers. Again, the notions of context, reference groups and outlier groups are not modeled simultaneously in OC^3. Angiulli et al. [4] studied a related by orthogonal problem. Given a multidimensional database and a query object in the database, find the top-k subset of attributes (i.e., subspaces) that the query object receives the highest outlier score. Their method does not find outliers directly. Moreover, it finds subspaces but does not find reference groups in outlier explanation.

A contextual outlier (r, o) identified in our method can be written as a pair of rules: $cond(r, o) \Rightarrow avs(r) - cond(r, o)$ for the reference group, and $cond(r, o) \Rightarrow avs(o) - cond(r, o)$ for the outlier group. There are a number of methods using rules in outlier detection [7,14,23,27,36]. Our method differs from the existing methods in several important aspects. First, most of the existing rule-based methods focus on detecting individual outliers, and may not be able to identify outlier groups and measure the outlyingness accordingly. Second, many existing rule-based methods use rules to model only the normal objects or strong associations. Outliers are individual objects that do not follow those rules. Those methods do not model and analyze context explicitly. Lastly, many existing methods, such as [10,22,36], set strict constraints on the size of the rules or the aggregate groups to be considered, such as a very small number of items/attributes allowed in a rule or only the parents and their sibling groups.

Our study is also related to previous work on emerging pattern mining ([13] and various subsequent publications) and contrast mining (see [12] and the references there). Conceptually, mining contextual outliers can be regarded as mining the non-redundant set of emerging patterns in all possible subspaces and under all possible shared AVSs as the constraints. Chen and Dong [9] provided an emerging pattern length based outlier detection method, but it is limited to global outliers. To the best of our knowledge the contextual outlier detection problem has not been systematically explored in the emerging/contrast pattern mining area.

More broadly, our study uses related concepts and techniques in data cube computation [5,17] and formal concept analysis [16]. However, to the best of our knowledge, no previous study systematically integrates the techniques to tackle the contextual outlier detection problem.

6. Experimental results

In this section, we report our empirical evaluation of COD using both real data sets and synthetic data sets. All experiments were conducted on a PC computer with an Intel Core Duo E8400 3.0 GHz CPU and 4 GB main memory, running the Microsoft Windows 7 operating system. The algorithms were implemented in C++ using Microsoft Visual Studio 2010.

We cannot identify any existing method that solves the exact same problem. The focus of our method is to find outliers with contextual information. Consequently, this paper does not intend to compete with the existing methods with respect to outlier detection accuracy or recall. We do compare our method with LOF [6] in Section 6.2.

6.1. Results on real data sets

We use several categorical data sets from the UCI repository [15]. We report our results on six data sets: adult, mushroom-sc, solar-flare, tic-tac-toe, credit-approval, and hayes-roth. Some statistics of the data sets are summarized in Table 3. Note that the mushroom-sc data set is made from the mushroom data set, by selecting all attributes related to mushrooms' shape and color. For the credit-approval and adult data sets, we keep the categorical attributes, and remove the numerical attributes. We also remove the records that having missing values in the selected data sets.

COD takes two parameters, the outlier degree threshold Δ and the significance threshold s. We report the results with respect to different combinations of Δ and s values for each data set. We evaluate COD in five aspects: outlier case studies, outlier analysis effectiveness (redundancy reduction and significance filtering), efficiency, scalability on dimensionality, and scalability on number of tuples.

6.1.1. Case studies

We demonstrate the effectiveness of contextual outlier detection using case studies on the tic-tac-toe, hayes-roth and mushroom-sc data sets. These three data sets are well known, and there is very little if any difference between these data sets and their original counterparts. Results on these data sets are hence easy to understand to the general audience.

An example on tic-tac-toe: The tic-tac-toe data set is composed of all the 958 possible board configurations at the end of tic-tac-toe games. It is assumed that "x" plays first and then "o" plays. There are 9 attributes, each corresponding to one tic-tac-toe square. An attribute takes "x" if the corresponding square is occupied by x, "o" if occupied by o, and "b" if blank.

Figures 1(a) and (b) show a contextual outlier group and a reference group, respectively. The outlier degree of this contextual outlier is 17.5 and the significance is 2.76×10^{-12}. This outlier corresponds to an end-of-game board where most of the corners (3 out of 4) are not occupied by any players. This is a rare occurrence, since occupying corner squares is a common winning strategy in tic-tac-toe games. The only two such configurations are c_1 and c_2, shown in Figs 1(c) and (d), respectively.

An example on hayes-roth: The hayes-roth data set records the information about 160 people on four attributes [3,15]. The first attribute, hobby, takes values uniformly at random [15] and we thus ignore it in our analysis, that is, all groups take value $*$ on the attribute. The description for the other three attributes are adopted from [3]: the second attribute, age, takes values in $\{30, 40, 50, r > 0\}$ (the meaning of value "$r > 0$" is unspecified in [3]); attribute education takes values in {junior-high, high-school, trade-school,

Table 3
The statistics of the data sets

Data set	*# of objects*	*# of attributes*	*# of closure groups*	*QC time (s)*
Adult	30,162	8	73,282	28.678
Mushroom-sc	8,124	8	6,265	1.879
Solar-flare	1,389	10	7,770	2.136
Tic-tac-toe	958	9	42,711	12.903
Credit-approval	690	8	5,707	1.446
Hayes-roth	160	4	277	0.047

Note: QC time refers to the time used to find all closure groups, that is, $|\mathcal{G}|$ in Algorithm 1.

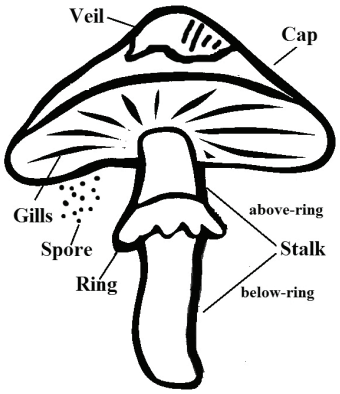


Fig. 2. Names for the parts of a mushroom.

college}; and the last attribute, marital-status, takes values in {single, married, divorced, widowed}. Table 4 shows some interesting contextual outliers with respect to $\Delta = 5$ and $s = 10^{-8}$.

In Table 4, outliers c_1 and c_2 share the same reference group. The reference group consists of 34 people whose marital-status is "single" and who have high-school degrees. Outlier group c_1 is a collection of 6 "divorced" college graduates and outlier group c_2 is a collection of 6 "single" trade school graduates. Outlier c_5 is interesting: among people who are divorced, those who are college graduates are outliers compared to those with high school degrees. c_9 shows that, among people who are 50 years old, the 2 with college degrees are outliers compared to the 16 with high school degrees. Another interesting outlier is c_{10}: in the age group of 30, the 4 people widowed are outliers compared to the 34 people married. Please note that, in the whole data set, there are 59 of high-school, 59 of junior-school, 29 of trade-school and 13 of college graduates. Given $\Delta = 5$, those of trade-school and college graduates are not outliers comparing to those of high-school and junior-school. The outliers can only be explained well using the contextual information.

An example on mushroom-sc: The mushroom-sc data set is made from data set mushroom [15]. We select all attributes that related to mushrooms' shape and color in our experiment. The mushroom-sc data set contains 8,124 individual mushroom records on 8 attributes. The attributes that we selected are (*cap-shape, stalk-shape, cap-color, gill-color, stalk-color-above-ring, stalk-color-below-ring, veil-color, spore-print-color*). Please refer to [15] for the detailed value description of each attribute. Table 5 shows some interesting contextual outliers with respect to $\Delta = 50$ and $s = 10^{-3}$.

Figure 2 shows the popular names used for different parts of a mushroom. In Table 5, outlier c_1 shows that, among the mushrooms that are white in the stalk above ring and the veil, the 64 mushrooms that are brown in the stalk below ring are outliers, compared to the 3,520 mushrooms that are white in the stalk below ring. Outlier c_2 tells us that, among the mushrooms that are white in the veil, brown in the spore print, and have convex caps, a small group of 32 mushrooms that are red in the stalk below ring are outliers, compared to a large group of 2,204 mushroom that are white in the stalk below ring. Outlier c_4 is interesting: among the mushrooms that are white in both the stalk below ring and the veil, and have tapering stalks, the 16 mushrooms that are brown in the gill and purple in the spore print are outliers, compared to the 864 mushrooms that are buff in the gill and white in the spore print.

6.1.2. Effectiveness of redundancy reduction and significance filtering

To evaluate the effectiveness of the outlier analysis techniques developed in Section 3, in addition to

Table 4
Some contextual outliers on data set hayes-roth ($\Delta = 5$, $s = 10^{-8}$). The underlined attributes indicate the shared AVSs

Outlier-id	*Reference group r*	*Outlier group o*	$deg(r,o) = \frac{\lvert cov(r)\rvert}{\lvert cov(o)\rvert}$	*Significance*
c_1	(∗, ∗, high-school, single)	(∗, ∗, high-school, divorced)	5.7 = 34/6	1.02×10^{-17}
c_2	(∗, ∗, high-school, single)	(∗, ∗, trade-school, single)	5.7 = 34/6	1.02×10^{-17}
c_3	(∗, ∗, high-school, single)	(∗, ∗, high-school, widowed)	8.5 = 34/4	5.05×10^{-10}
c_4	(∗, ∗, trade-school, married)	(∗, ∗, trade-school, widowed)	8.0 = 16/2	2.32×10^{-8}
c_5	(∗, ∗, junior-high, divorced)	(∗, ∗, college, divorced)	8.0 = 16/2	2.32×10^{-8}
c_6	(∗, 40, junior-high, ∗)	(∗, 40, college, ∗)	8.5 = 34/4	5.05×10^{-10}
c_7	(∗, 40, junior-high, ∗)	(∗, 40, trade-school, ∗)	5.7 = 34/6	1.02×10^{-17}
c_8	(∗, 40, junior-high, ∗)	(∗, 50, junior-high, ∗)	5.7 = 34/6	1.02×10^{-17}
c_9	(∗, 50, high-school, ∗)	(∗, 50, college, ∗)	8.0 = 16/2	2.32×10^{-8}
c_{10}	(∗, 30, ∗, married)	(∗, 30, ∗, widowed)	8.5 = 34/4	5.05×10^{-10}

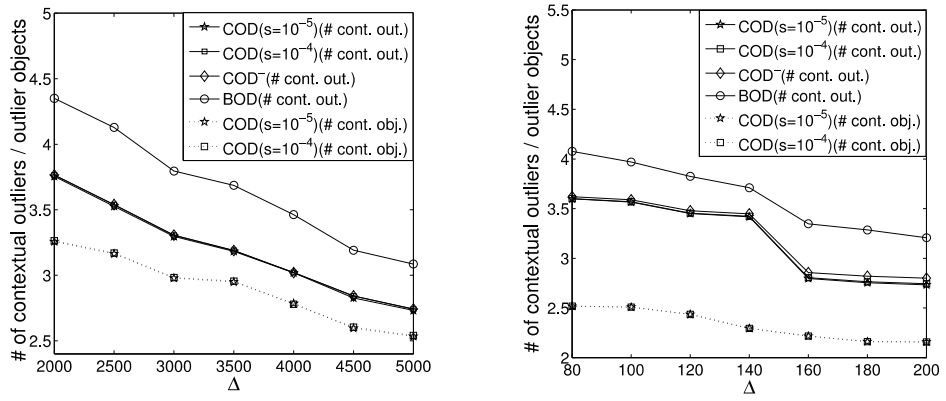
Table 5
Some contextual outliers on data set mushroom-sc ($\Delta = 50$, $s = 10^{-3}$). The underlined attributes indicate the shared AVSs

Outlier-id	Contextual outlier	$deg(r,o) = \frac{\lvert cov(r)\rvert}{\lvert cov(o)\rvert}$	*Significance*
c_1	*r*: (∗, ∗, ∗, ∗, *White*, *White*, *White*, ∗) *o*: (∗, *Enlarging*, ∗, *White*, *White*, *Brown*, *White*, *White*)	$55.0 = \frac{3520}{64}$	1.17×10^{-21}
c_2	*r*: (*Convex*, ∗, ∗, ∗, ∗, *White*, *White*, ∗) *o*: (*Convex*, *Enlarging*, ∗, ∗, ∗, *Red*, *White*, *White*)	$63.3 = \frac{2024}{32}$	1.29×10^{-18}
c_3	*r*: (*Convex*, ∗, ∗, ∗, ∗, ∗, *White*, *Brown*) *o*: (*Sunken*, *Enlarging*, ∗, ∗, *White*, *White*, *White*, *Brown*)	$62.5 = \frac{1000}{16}$	1.69×10^{-29}
c_4	*r*: (∗, *Tapering*, ∗, *Buff*, ∗, *White*, *White*, *White*) *o*: (∗, *Tapering*, ∗, *Brown*, *White*, *White*, *White*, *Purple*)	$54.0 = \frac{864}{16}$	1.39×10^{-12}
c_5	*r*: (*Convex*, *Tapering*, ∗, ∗, *White*, *White*, *White*, ∗) *o*: (*Convex*, *Enlarging*, ∗, ∗, *red*, *White*, *White*, *White*)	$54.0 = \frac{816}{16}$	5.31×10^{-31}

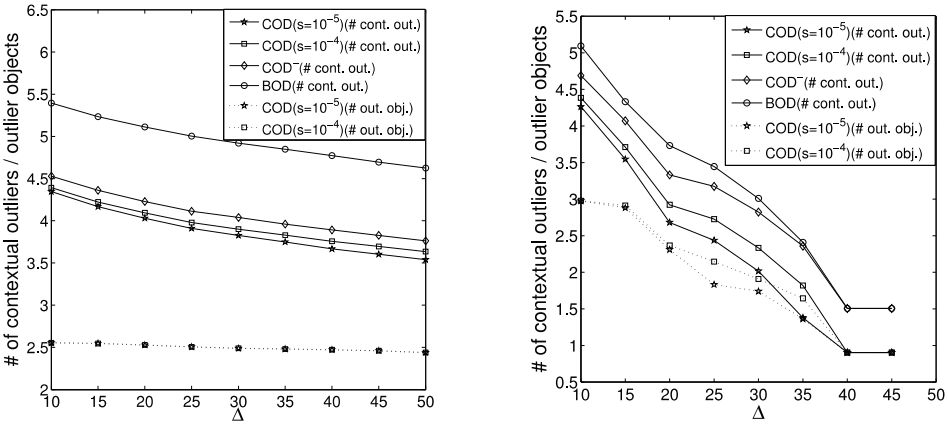
COD, we consider two simplified versions, COD$^-$ and BOD. Both of them work the same as COD except for the following changes. COD$^-$ does not apply the significance test for contextual outliers. BOD does not apply either the redundancy removal techniques or the significance test.

Figure 3 plots the number of contextual outliers and the number of outlier objects with respect to different Δ values. An object is called an outlier if it is contained in the outlier group of a context outlier. We set $s = 10^{-4}$ and $s = 10^{-5}$ respectively in COD. No contextual outliers exist when $\Delta \geqslant 50$ in the tic-tac-toe data set and when $\Delta \geqslant 20$ in the hayes-roth data set. COD outputs much less contextual outliers than COD$^-$ and BOD in all cases. The number of outlier objects is small, and decreases roughly linearly as Δ increases. One object may be contained in multiple contextual outliers. Multiple contextual outliers containing the same outlier object identify how the outlier object deviates from the majority in different subspaces.

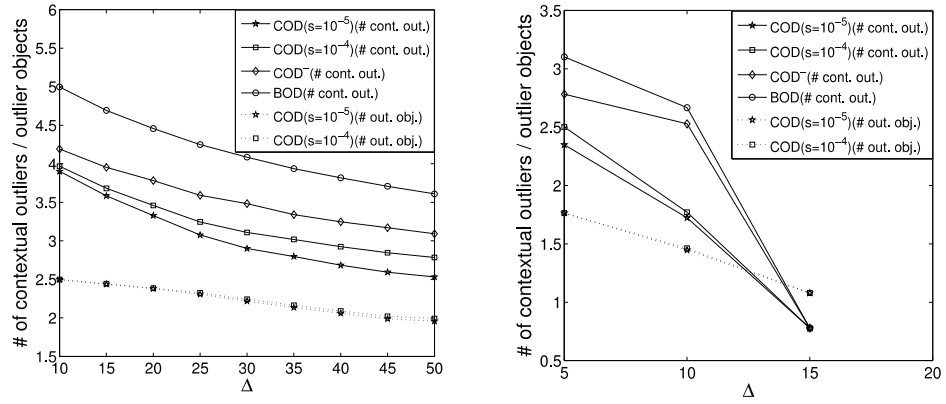
Table 6 shows the number of outlier objects, outlier groups and contextual outliers with respect to different significance threshold s in COD. As expected, the lower the significance threshold, the less outliers are reported. Moreover, the number of outlier groups is much smaller than that of contextual outliers. This shows that outlier groups are a concise summarization of contextual outliers.



(a) Adult (b) Mushroom-sc



(c) Solar-flare (d) Tic-tac-toe



(e) Credit-approval (f) Hayes-roth

Fig. 3. The number of contextual outliers/outlier objects w.r.t different Δ. The y-axis (# of contextual outliers/outlier objects) is in logarithmic scale.

Table 6
The number of contextual outliers w.r.t. significance threshold

$*$ s		Adult $\Delta = 2000$	Mushroom-sc $\Delta = 60$	Solar-flare $\Delta = 150$	Tic-tac-toe $\Delta = 30$	Credit-approval $\Delta = 70$	Hayes-roth $\Delta = 15$
10^{-3}	# of out. obj.	1,831	346	161	222	72	12
	# of out. grp.	2,429	676	212	222	98	6
	# of cont. out.	5,823	6,659	896	664	488	6
10^{-5}	# of out. obj.	1,807	346	124	55	34	12
	# of out. grp.	2,399	676	170	55	48	6
	# of cont. out.	5,686	6,658	487	104	167	6
10^{-7}	# of out. obj.	1,388	346	113	46	27	12
	# of out. grp.	1,745	676	134	46	27	6
	# of cont. out.	2,882	6,571	304	84	62	6
10^{-9}	# of out. obj.	1,314	346	107	42	22	12
	# of out. grp.	1,506	672	118	42	14	6
	# of cont. out.	2,213	5,872	255	76	24	6

6.1.3. Efficiency

Figure 4 compares the runtime of COD, COD$^-$ and BOD on the four real data sets with respect to various Δ thresholds. The closure group computation time is reported in Table 3, and is not included in Fig. 4.

The runtime of BOD is the least among the three methods. COD$^-$ takes a very small amount of extra time on top of BOD to identify tight and strong outliers. COD uses extra time on top of COD$^-$ to test the statistical significance. When Δ is small, the number of contextual outliers is large, and thus COD and COD$^-$ need more extra runtime. When Δ increases, the runtime difference between these three methods decreases quickly. In practice, Δ should not be set to a small value, since one often likes to find outliers that deviate from significantly larger trends. We notice that different setting of significance threshold s does not affect the runtime of COD in a noticeable way.

6.1.4. Scalability on dimensionality

We test the scalability of COD with respect to dimensionality on the real data sets. We keep the first k attributes and vary k from 2 to the dimensionality of the data sets. We report the results on two data sets, adult and solar-flare, which have the largest number of tuples and the highest dimensionality, respectively, among the six real data sets. Figure 5 shows the number of outlier objects with respect to dimensionality; and Fig. 6 shows the runtime with respect to dimensionality. Both the number of outlier objects and runtime increase when the dimensionality increases, since spaces of higher dimensionality can accommodate more outliers.

6.1.5. Scalability on number of tuples

We also test the scalability of COD with respect to the number of tuples on real data sets. Similar to Section 6.1.4, we report the results on two real data sets, adult and solar-flare. In the adult data set, we use the first k tuples and vary k from 7500 to the number of tuples of the data set. Similarly, in the solar-flare data set, we use the first k tuples as well and vary k from 350 to the number of tuples of the data set. Figure 7 shows the number of outlier objects with respect to number of tuples; and Fig. 8 shows the runtime with respect to number of tuples. Both the number of outlier objects and runtime increase when the number of tuples increases, since more tuples can accommodate more outliers. In all these cases, the runtime increases linearly with respect to the database sizes. In other words, our method is approximately linear with respect to the number of tuples empirically.

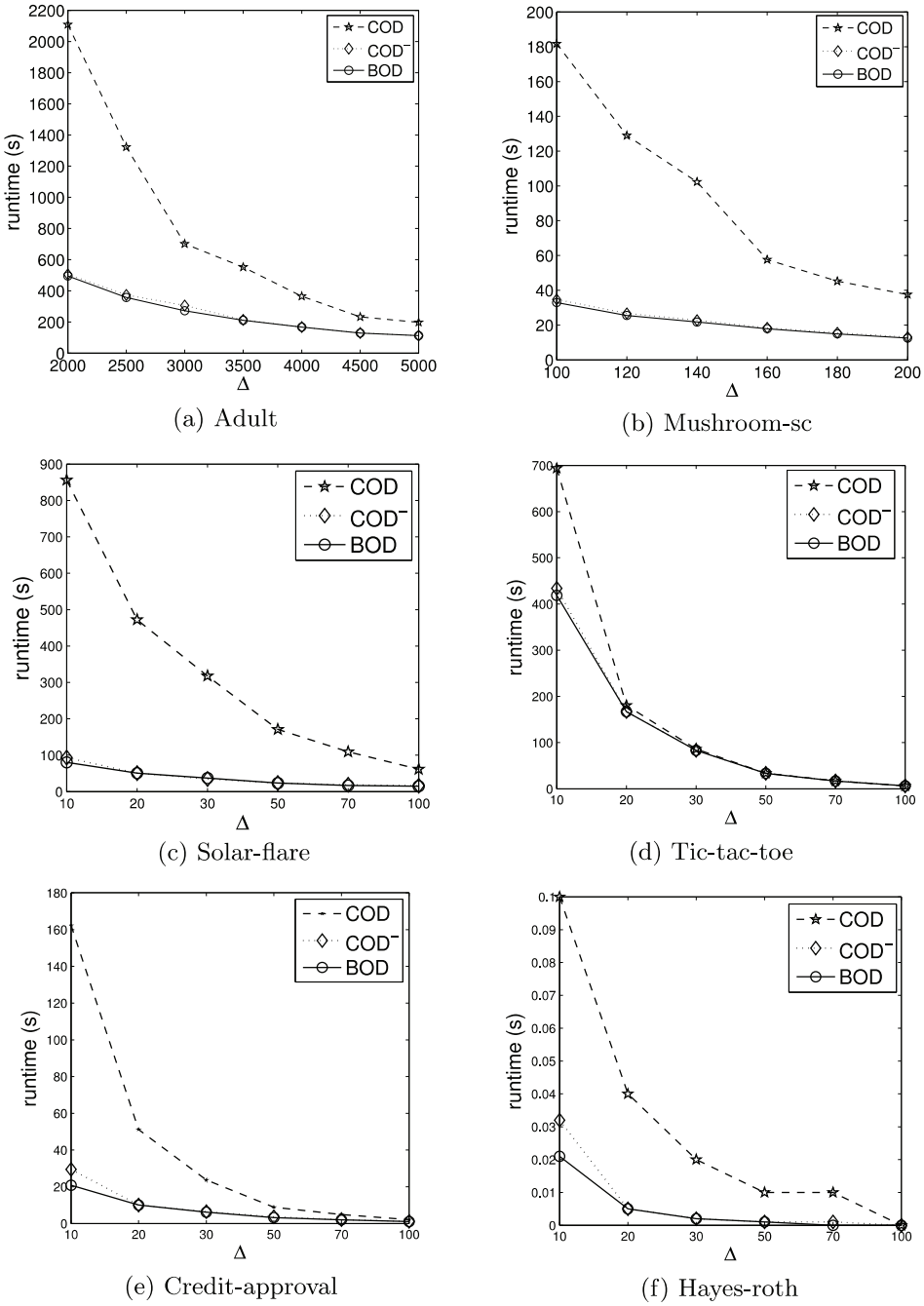
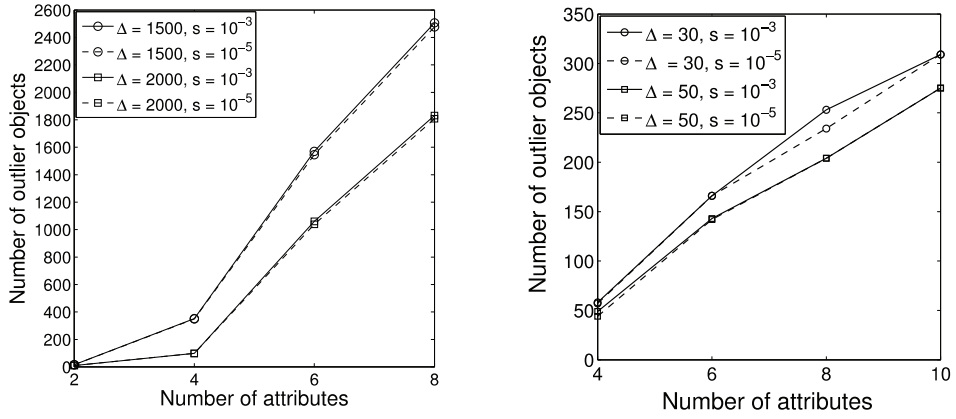


Fig. 4. The runtime of COD, COD$^-$ and BOD on the six real data sets ($s = 10^{-3}$ in COD).

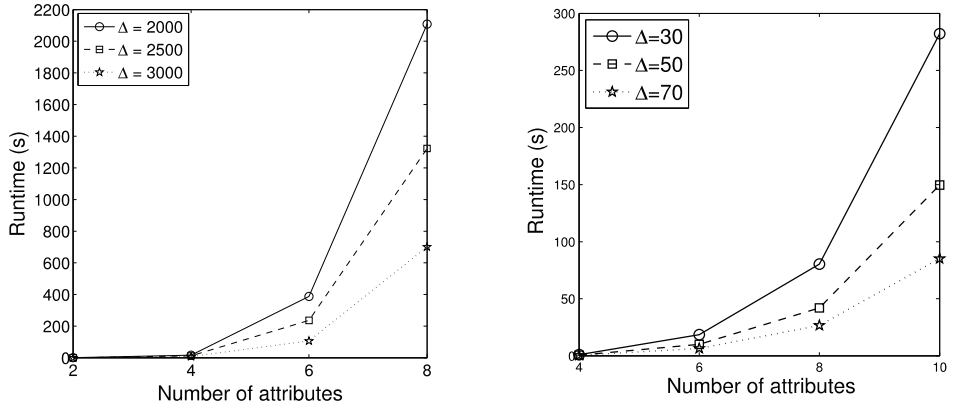
6.2. *Results on synthetic data sets*

In order to test the accuracy of COD, we use synthetic data sets, since most real data sets do not have the complete ground-truth information. We also use synthetic data sets to test the scalability of COD.



(a) Adult (b) Solar-flare

Fig. 5. The number of outliers of COD with respect to dimensionality.

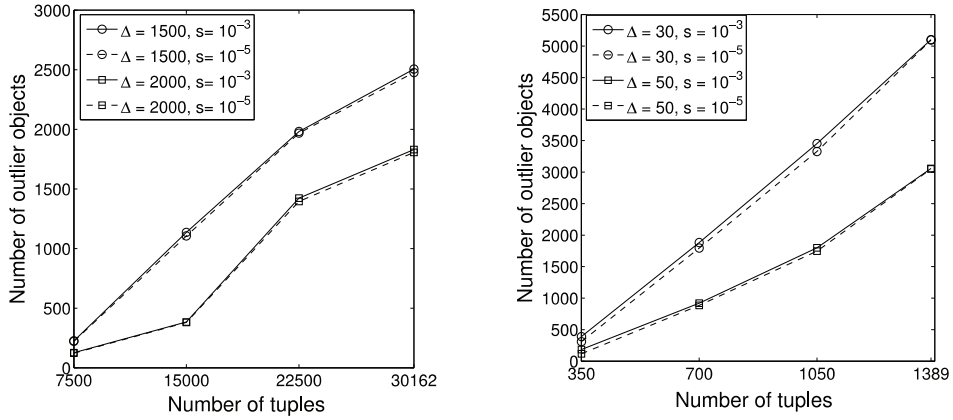


(a) Adult (b) Solar-flare

Fig. 6. The runtime of COD with respect to dimensionality ($s = 10^{-3}$).

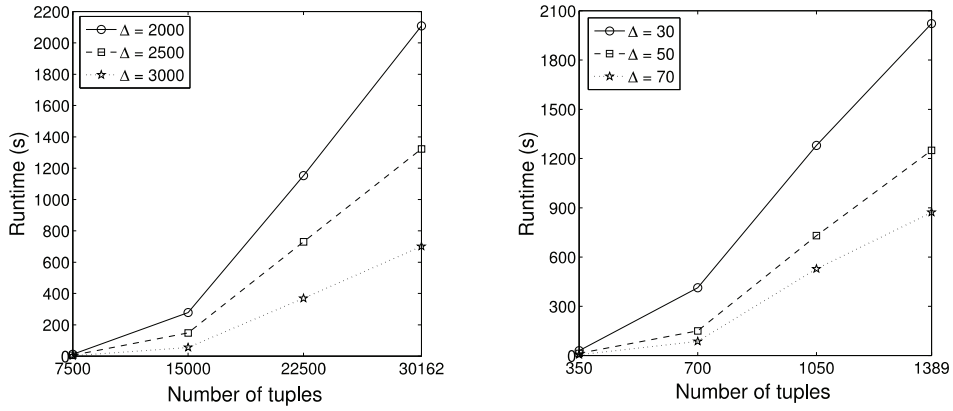
Given the outlier degree threshold $\Delta > 0$, the number of subspace outliers m_s, the number of global outlier m_g, the dimensionality d, the cardinality c, and the number of tuples in the data set, we generate a synthetic data set in three steps. First, we generate m_s subspace outliers (including the reference groups and the outlier groups) satisfying the outlier degree threshold requirement. For an outlier, the shared AVS and the outlier subspace are chosen randomly. Second, we generate m_g global outliers (including the reference groups and the outlier groups) in the same manner. Again, the outlier subspaces are chosen randomly. Last, we inject independent and uniformly distributed data to fulfill the requirement on number of tuples. A synthetic data set generated as such carries the ground truth on contextual outliers. In our experiments, we fix $\Delta = 50$, $m_s = 850$, $m_g = 150$, $d = 10$, $c = 100$. By default, the number of tuples is set to 1 million.

Table 7 shows the precision of COD. We repeat the experiments 10 times on 10 synthetic data sets generated independently using the same parameters, and report the average and the standard deviation of the precision and recall. The results show that our method always detects all outliers in the ground-truth (100% recall). At the same time, COD has a good precision. Please note that, although we implant



(a) Adult (b) Solar-flare

Fig. 7. The number of outliers of COD with respect to number of tuples.



(a) Adult (b) Solar-flare

Fig. 8. The runtime of COD with respect to number of tuples ($s = 10^{-3}$).

the seed outliers in the synthetic data set as the ground-truth, the noise injected in the synthetic data set may lead to some outliers that are not included in the ground-truth.

In Table 7, we also compare our method COD with LOF [6] using the implementation in Weka [18] and Hamming distance as the distance measure. The parameters *MinPtsLB* and *MinPtsUB* are set according to the suggestions in [6]. COD outperforms LOF on the synthetic data sets in both accuracy and recall. Please note that LOF cannot provide contextual information for outliers, and is not designed specifically for contextual outlier detection. In fact, we cannot identify any existing method that solves the exact same problem.

Figure 9 tests the scalability of COD with respect to the number of tuples in the database. We generate data sets of different sizes, from 100 thousand to 1 million tuples. We keep the other parameters the same. Again, for each configuration, we repeat the experiment 10 times, and report results in the figure. The results clearly shows that COD is scalable with respect to database size.

Table 7
The precision & recall of COD, and comparison with LOF

Methods & threshold setting	*Precision*		*Recall*	
	Avg.	*Std.*	*Avg.*	*Std.*
COD ($\Delta = 50$, $s = 10^{-7}$)	78.63%	10.77%	100%	0
LOF (*MinPtsLB* $= 10$, *MinPtsUB* $= 100$)	68.59%	11.59%	73%	13.04%

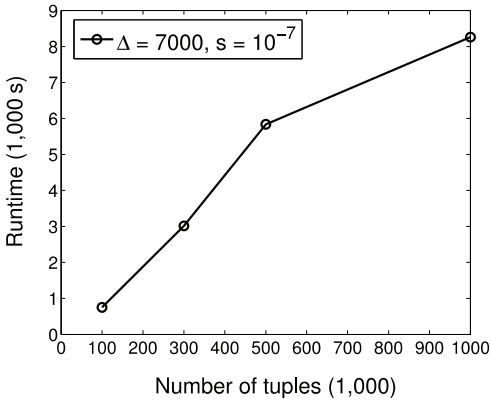


Fig. 9. The scalability of COD on synthetic data sets w.r.t. number of tuples.

7. Conclusions and future work

In this paper, we proposed a framework for contextual outlier detection. Our focus was to improve the interpretability of outliers. In particular, we argued that the context of an outlier should include a shared set of attribute-value pairs, a reference group, an outlier group, and an outlier degree measure. Moreover, we developed a concise representation for contextual outliers and presented a detection algorithm leveraging the state-of-the-art data cube computation techniques.

This paper only represents the first step in an ambitious journey towards contextual outlier detection and analysis. There are several important and interesting problems for future work. For example, we mentioned the concept of outlier groups (Definition 5). From the user's point of view, it would be interesting to develop efficient and pay-as-you-go methods to compute a ranked list of outlier groups. This would also help to eliminate the need for setting an outlier degree threshold. As another example, it would be interesting to explore the correlation among outlier groups, reference groups and contexts more generally. This may lead to valuable insights into the inherent characteristics of high dimensional data. Last but not least, developing more efficient and scalable algorithms for contextual outlier detection is another challenge for future study.

Acknowledgements

This work is supported in part by an NSERC Discovery grant and a BCIC NRAS Team Project. James Bailey is supported by ARC Grants DP140101969 and FT110100112. Part of the work by Guozhu Dong was performed when he was visiting Simon Fraser University under the support of an Ebco/Eppich visiting professorship. All opinions, findings, conclusions and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

References

[1] M. Agyemang, K. Barker and R. Alhajj, A comprehensive survey of numeric and symbolic outlier mining techniques, *Intell Data Anal* **10** (December 2006), 521–538.

[2] L. Akoglu, H. Tong, J. Vreeken and C. Faloutsos, Fast and reliable anomaly detection in categorical data, in: *CIKM* (2012), 415–424.
[3] J.R. Anderson and P.I. Kline, A learning system and its psychological implications, in: *Proceedings of the Sixth International Joint Conference on Artificial Intelligence*, IJCAI '79 **1** (1979), 16–21.
[4] F. Angiulli, F. Fassetti and L. Palopoli, Detecting outlying properties of exceptional objects, *ACM Trans Database Syst* **34**(1) (Apr 2009), 7:1–7:62.
[5] K. Beyer and R. Ramakrishnan, Bottom-up computation of sparse and iceberg cube, in: *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, SIGMOD '99, ACM, New York, NY, USA (1999), 359–370.
[6] M.M. Breunig, H.-P. Kriegel, R.T. Ng and J. Sander, Lof: identifying density-based local outliers, in: *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, SIGMOD '00, ACM, New York, NY, USA (2000), 93–104.
[7] P.K. Chan, M.V. Mahoney and M.H. Arshad, A machine learning approach to anomaly detection, Technical report, Florida Institute of Technology, Melbourne, Florida, USA, 2003.
[8] V. Chandola, A. Banerjee and V. Kumar, Anomaly detection: A survey, *ACM Comput Surv* **41** (July 2009), 15:1–15:58.
[9] L. Chen and G. Dong, Masquerader detection using OCLEP: One class classification using length statistics of emerging patterns, in: *Proceedings of WAIM Workshops: International Workshop on Information Processing Over Evolving Networks*, WINPEN '06, IEEE Computer Society, Washington, DC, USA (2006), 5.
[10] K. Das and J. Schneider, Detecting anomalous records in categorical datasets, in: *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, ACM, New York, NY, USA (2007), 220–229.
[11] K. Das, J.G. Schneider and D.B. Neill, Anomaly pattern detection in categorical datasets, in: *KDD* (2008), 169–176.
[12] G. Dong and J. Bailey, Overview of contrast data mining as a field and preview of an upcoming book, in: *Proceedings of ICDM Workshops: Workshop on Contrast Data Mining and Applications*, IEEE Computer Society, Washington, DC, USA (2011), 1141–1146.
[13] G. Dong and J. Li, Efficient mining of emerging patterns: discovering trends and differences, in: *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '99, ACM, New York, NY, USA (1999), 43–52.
[14] T. Fawcett and F. Provost, Adaptive fraud detection, *Data Min Knowl Discov* **1** (1997), 291–316.
[15] A. Frank and A. Asuncion, UCI machine learning repository, 2010,
[16] B. Ganter and R. Wille, *Formal Concept Analysis: Mathematical Foundations*, 1st edition, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1997.
[17] J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatrao, F. Pellow and H. Pirahesh, Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-totals, *Data Min Knowl Discov* **1** (1997), 29–53.
[18] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I.H. Witten, The weka data mining software: An update, *SIGKDD Explor Newsl* **11**(1) (2009), 10–18.
[19] V. Hodge and J. Austin, A survey of outlier detection methodologies, *Artif Intell Rev* **22** (October 2004), 85–126.
[20] H.P. Kriegel, E. Schubert, A. Zimek and P. Kroger, Outlier detection in axis-parallel subspaces of high dimensional data, in: *Proc of PAKDD* (2009), 831–838.
[21] L.V.S. Lakshmanan, J. Pei and J. Han, Quotient cube: how to summarize the semantics of a data cube, in: *Proceedings of the 28th International Conference on Very Large Data Bases*, VLDB '02, VLDB Endowment (2002), 778–789.
[22] S. Lin and D.E. Brown, An outlier-based data association method for linking criminal incidents, *Decis Support Syst* **41** (March 2006), 604–615.
[23] M.V. Mahoney and P.K. Chan, Learning rules for anomaly detection of hostile network traffic, in: *Proc of the 3rd IEEE International Conference on Data Mining*, ICDM '03 (2003), 601–604.
[24] M. Markou and S. Singh, Novelty detection: A review – part 1: Statistical approaches, *Signal Process* **83** (December 2003), 2481–2497.
[25] M. Markou and S. Singh, Novelty detection: A review – part 2: Neural network based approaches, *Signal Process* **83** (December 2003), 2499–2521.
[26] E. Muller, M. Schiffer and T. Seidl, Statistical selection of relevant subspace projections for outlier ranking, in: *Proceedings of the (2011), IEEE 27th International Conference on Data Engineering*, ICDE '11, IEEE Computer Society, Washington, DC, USA (2011), 434–445.
[27] K. Narita and H. Kitagawa, Detecting outliers in categorical record databases based on attribute associations, in: *Proceedings of the Tenth Asia-Pacific Web Conference on Progress in WWW Research and Development*, APWeb '08, Springer-Verlag, Berlin, Heidelberg (2008), 111–123.
[28] N. Pasquier, Y. Bastide, R. Taouil and L. Lakhal, Discovering frequent closed itemsets for association rules, in: *Proceedings of the Seventh International Conference on Database Theory*, ICDT '99, Springer-Verlag, London, UK

(1999), 398–416.
[29] A. Patcha and J.-M. Park, An overview of anomaly detection techniques: Existing solutions and latest technological trends, *Comput Netw* **51** (2007), 3448–3470.
[30] K. Smets and J. Vreeken, The odd one out: Identifying and characterising anomalies, in: *Proceedings of the SIAM International Conference on Data Mining*, SDM '11, SIAM (2011), 804–815.
[31] X. Song, M. Wu, C. Jermaine and S. Ranka, Conditional anomaly detection, *IEEE Trans on Knowl and Data Eng* **19** (2007), 631–645.
[32] M. Valko, B. Kveton, H. Valizadegan, G.F. Cooper and M. Hauskrecht, Conditional anomaly detection with soft harmonic functions, in: *Proceedings of the (2011), IEEE 11th International Conference on Data Mining*, ICDM '11, IEEE Computer Society, Washington, DC, USA (2011), 735–743.
[33] J. Wang, J. Han and J. Pei, Closet+: searching for the best strategies for mining frequent closed itemsets, in: *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, ACM, New York, NY, USA (2003), 236–245.
[34] X. Wang and I. Davidson, Discovering contexts and contextual outliers using random walks in graphs, in: *Proceedings of the (2009), Ninth IEEE International Conference on Data Mining*, ICDM '09 (2009), 1034–1039.
[35] L. Wei, W. Qian, A. Zhou, W. Jin and J.X. Yu, Hot: Hypergraph-based outlier test for categorical data, in: *Proceedings of the 7th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, PAKDD'03, Springer-Verlag, Berlin, Heidelberg (2003), 399–410.
[36] W.K. Wong, A. Moore, G. Cooper and M. Wagner, Rule-based anomaly pattern detection for detecting disease outbreaks, in: *Proceedings of the 18th National Conference on Artificial Intelligence*, ENAI '02, American Association for Artificial Intelligence, Menlo Park, CA, USA (2002), 217–223.
[37] G. Yang, The complexity of mining maximal frequent itemsets and maximal frequent patterns, in: *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, ACM, New York, NY, USA (2004), 344–353.