# Mining Exceptions in Databases

Eduardo Corrêa Gonçalves, Ilza Maria B. Mendes, and Alexandre Plastino*

Universidade Federal Fluminense, Department of Computer Science,
Rua Passo da Pátria, 156 - Bloco E - 3º andar - Boa Viagem,
24210-240, Niterói, RJ, Brazil
{egoncalves, imendes, plastino}@ic.uff.br
http://www.ic.uff.br

**Abstract.** This paper addresses the problem of mining exceptions from multidimensional databases. The goal of our proposed model is to find association rules that become weaker in some specific subsets of the database. The candidates for exceptions are generated combining previously discovered multidimensional association rules with a set of significant attributes specified by the user. The exceptions are mined only if the candidates do not achieve an expected support. We describe a method to estimate these expectations and propose an algorithm that finds exceptions. Experimental results are also presented.

## 1 Introduction

Multidimensional association rules [4] represent combinations of attribute values that often occur together in multidimensional repositories, such as data warehouses or relational databases. An example is given by: ($Age$ = "30–35") ⇒ ($Payment$ = "*credit card*"). This rule indicates that consumers who are between 30 and 35 years old, are more likely to pay for their purchases using credit card. A multidimensional association rule can be formally defined as follows:

$$A_1 = a_1, \ldots, A_n = a_n \Rightarrow B_1 = b_1, \ldots, B_m = b_m \ ,$$

where $A_i$ $(1 \leq i \leq n)$ and $B_j$ $(1 \leq j \leq m)$ represent distinct attributes (dimensions) from a database relation, and $a_i$ and $b_j$ are values from the domains of $A_i$ and $B_j$, respectively. To simplify the notation, we will represent a generic rule as an expression of the form $A \Rightarrow B$, where $A$ and $B$ are sets of conditions over different attributes. The support of a set of conditions $Z$, $Sup(Z)$, in a relation $D$ is the percentage of tuples in $D$ that match all conditions in $Z$. The support of a rule $A \Rightarrow B$, $Sup(A \Rightarrow B)$, is given by $Sup(A \cup B)$. The confidence of $A \Rightarrow B$, $Conf(A \Rightarrow B)$, is the probability that a tuple matches $B$, given that it matches $A$. Typically, the problem of mining association rules consists in finding all rules that match user-provided minimum support and minimum confidence.

* Work sponsored by CNPq research grant 300879/00-8.

G.I. Webb and Xinghuo Yu (Eds.): AI 2004, LNAI 3339, pp. 1076–1081, 2004.
© Springer-Verlag Berlin Heidelberg 2004

In this work we propose a human-centered approach to mine *exceptions* from multidimensional databases. An example of this kind of pattern is given by: $(Age = \text{“30–35”}) \overset{\neg s}{\Longrightarrow} (Payment = \text{“}credit\ card\text{”})\ [Income = \text{“} < 1K\text{”}]$. This exception indicates that among the consumers who earn less than 1K, the support value of the rule $(Age = \text{“30–35”}) \Rightarrow (Payment = \text{“}credit\ card\text{”})$ is significantly *smaller* than what is expected.

Proposals for mining exception rules that contradict associations with high support and confidence can be found in [5, 7]. However, in our work, exceptions characterize rules that become much weaker in specific subsets of the database. Our approach was motivated by the concept of *negative association rules*, proposed in [6, 8], where a negative pattern represents a large deviation between the actual and the expected support of a rule.

This paper is organized as follows. In Sect. 2 we present the model for mining exceptions. We propose an algorithm in Sect. 3 and show experimental results in Sect. 4. Some concluding remarks are made in Sect. 5.

## 2 Exceptions

In order to explain our approach for mining exceptions, consider the *consumers data set* (Table 1). The data objects represent consumers of a hypothetical store. An association rule mining algorithm can obtain the following pattern from this database: “Female consumers have children” ($Sup = 40\%$ and $Conf = 66.67\%$). However, note that none of the women who earns more than 3K have children. Then, it would be interesting to infer the following *negative* pattern: “Female consumers who earn more than 3K do not have children”. This negative pattern came from the positive rule “Female consumers have children” and it was obtained because the support value of “Female consumers who earn more than 3K have children” is significantly lower than what was expected. This example illustrates an *exception* associated with a positive association rule. It can be represented as:

$$(Gender = \text{“}F\text{”}) \overset{\neg s}{\Longrightarrow} (Children = \text{“}Yes\text{”})\ [(Income = \text{“} > 3K\text{”})]\ .$$

**Definition 1.** *(Exception). Let $\mathcal{D}$ be a relation. Let $R : A \Rightarrow B$ be a multidimensional association rule defined from $\mathcal{D}$. Let $Z = \{Z_1 = z_1, \ldots, Z_k = z_k\}$ be a set of conditions defined over attributes from $\mathcal{D}$, where $Z \cap A \cap B = \emptyset$. $Z$ is named probe set. An exception related to the positive rule $R$ is an expression of the form $A \overset{\neg s}{\Longrightarrow} B\ [Z]$.*

Exceptions are extracted only if they do not achieve an expected support. This expectation is evaluated based on the support of the original rule $A \Rightarrow B$ and the support of the conditions that compose the probe set $Z$. The expected support for the *candidate exception* $A \Rightarrow B\ [Z]$ can be computed as:

$$ExpSup(A \Rightarrow B\ [Z]) = Sup(A \Rightarrow B) \times Sup(Z)\ . \tag{1}$$

An exception $E : A \overset{\neg s}{\Longrightarrow} B\ [Z]$ can be regarded as *potentially interesting* if the actual support value of the candidate exception $A \Rightarrow B\ [Z]$, given by

**Table 1.** Consumers data set

| Gender(G) | Age(A) | Income(I) | Children(C) | Car Owner(CO) |
|---|---|---|---|---|
| F | <18 | <1K | Yes | No |
| M | 26-30 | >3K | Yes | Yes |
| F | 26-30 | >3K | No | Yes |
| F | 18-25 | 1K - 3K | Yes | Yes |
| F | 31-40 | <1K | Yes | Yes |
| M | 18-25 | >3K | Yes | No |
| F | 26-30 | 1K - 3K | Yes | Yes |
| F | <18 | >3K | No | No |
| M | 18-25 | >3K | Yes | Yes |
| M | <18 | <1K | No | No |

$Sup(A \cup B \cup Z)$, is much lower than its expected support. The $IM$ index (*Interest Measure*) is used to calculate this deviation. This measure captures the type of dependence between $Z$ and $A \Rightarrow B$.

$$IM(E) = 1 - \left( \frac{Sup(A \Rightarrow B\ [Z])}{ExpSup(A \Rightarrow B\ [Z])} \right) . \tag{2}$$

The $IM$ index value grows when the actual support value is lower and far from the expected support value, indicating a negative dependence. The closer the value is from 1 (which is the highest value for this measure), the more the negative dependence is. If $IM(E) \approx 0$, then $Z$ and $A \Rightarrow B$ are independent. If $IM(E) < 0$, the actual support value is higher than the expected support value, indicating a positive dependence.

Consider the rule $R:\ (G = \text{“}F\text{”}) \Rightarrow (C = \text{“}Yes\text{”})$, presented at the beginning of this section. Two different values of the attribute *Income* will be used as probe sets and will be combined with this rule in an attempt to identify exceptions.

The actual support of the candidate $C_1 : (G = \text{“}F\text{”}) \Rightarrow (C = \text{“}Yes\text{”})\ [(I = \text{“} < 1K\text{”})]$ is 20%. The support of $R$ is 40% and the support of the probe set $Z_1 = \{(I = \text{“} < 1K\text{”})\}$ is 30%. According to (1), $ExpSup(C_1) = Sup(R) \times Sup(Z_1) = 40\% \times 30\% = 12\%$. The exception $E_1 :\ (G = \text{“}F\text{”}) \stackrel{-s}{\Longrightarrow} (C = \text{“}Yes\text{”})[(I = \text{“} < 1K\text{”})]$ is uninteresting because $IM(E_1) = 1 - (0.20 \div 0.12) = -0.67$.

The actual support of the candidate $C_2 : (G = \text{“}F\text{”}) \Rightarrow (C = \text{“}Yes\text{”})\ [(I = \text{“} > 3K\text{”})]$ is 0%. The support of the probe set $Z_2 = \{(I = \text{“} > 3K\text{”})\}$ is 50%. The expected support for $C_2$ is calculated as $40\% \times 50\% = 20\%$. The exception $E_2 :\ (G = \text{“}F\text{”}) \stackrel{-s}{\Longrightarrow} (C = \text{“}Yes\text{”})[(I = \text{“} > 3K\text{”})]$ is potentially interesting, because $IM(E_2) = 1 - (0 \div 0.20) = 1$.

In the next example, we will show that a high value for the $IM$ index is not a guarantee of interesting information. Consider the rule “Female consumers have a car” ($Sup = 40\%$ and $Conf = 66.67\%$), obtained from the *consumers data set*. Observing Table 1, we can also notice that none of the women who are under 18 years old have a car. These information could lead us to conclude that $(G = \text{“}F\text{”}) \stackrel{-s}{\Longrightarrow} (CO = \text{“}Yes\text{”})\ [(A = \text{“} < 18\text{”})]$ is an interesting negative

pattern, since the $IM$ value for this exception is 1. However, in reality, none of the consumers who are under 18 years old have a car, *independently if they are men or women*. Suppose these consumers live in a country where only the ones who are 18 years old or above are allowed to drive. Then, the exception $(G = \text{“}F\text{”}) \stackrel{-s}{\Longrightarrow} (CO = \text{“}Yes\text{”})\ [(A = \text{“} < 18\text{”})]$ represents an information that is certainly obvious and useless. Therefore it should not be mined. The $IM$ index was not able to detect the strong *negative dependence* between being under 18 years old and having a car.

**Definition 2.** *(Negative Dependence). Let $X = \{X_1 = x_1, \ldots, X_n = x_n\}$ and $Y = \{Y_1 = y_1, \ldots, Y_m = y_m\}$ be two sets of conditions where $X \cap Y = \emptyset$. The negative dependence between $X$ and $Y$, denoted as $ND(X, Y)$, is given by:*

$$ND(X,Y) = 1 - \left(\frac{Sup(X \cup Y)}{ExpSup(X \cup Y)}\right) = 1 - \left(\frac{Sup(X \cup Y)}{Sup(X) \times Sup(Y)}\right) . \quad (3)$$

The $DU$ index (*Degree of Unexpectedness*) is used to capture how much the negative dependence between a probe set $Z$ and a rule $A \Rightarrow B$ is *higher* than the negative dependence between $Z$ and either $A$ or $B$.

$$DU(E) = IM(E) - \max(ND(A,Z), ND(B,Z)) . \quad (4)$$

The greater the $DU$ value is from 0, the more interesting the exception will be. If $DU(E) \leq 0$ the exception is uninteresting. Consider, again, the rule $R: (G = \text{“}F\text{”}) \Rightarrow (C = \text{“}Yes\text{”})$ and the probe set $Z_2 = \{(I = \text{“} > 3K\text{”})\}$. First we should compute $ND(A,Z) = ND((G = \text{“}F\text{”}), (I = \text{“} > 3K\text{”})) = (1 - (0.20 \div 0.30)) = 0.33$; and $ND(B,Z) = ND((C = \text{“}Yes\text{”}), (I = \text{“} > 3K\text{”})) = (1 - (0.30 \div 35)) = 0.14$; The exception $E_2: (G = \text{“}F\text{”}) \stackrel{-s}{\Longrightarrow} (C = \text{“}Yes\text{”})\ [(I = \text{“} > 3K\text{”})]$ is, in fact, interesting because $DU(E_2) = 1 - max(0.33, 0.14) = 0.67$. Next, we give a formal definition for the problem of mining exceptions.

**Definition 3.** *(Problem Formulation). Let $MinSup \geq 0$, $I_{min} \geq 0$, and $D_{min} \geq 0$ denote minimum user-specified thresholds for Sup, IM, and DU. The problem of mining exceptions in multidimensional databases consists in finding each exception $E$ in the form $A \stackrel{-s}{\Longrightarrow} B\ [Z]$, which satisfies the following conditions:*

1. *(a) $Sup(A \cup Z) \geq MinSup$ and (b) $Sup(B \cup Z) \geq MinSup$;*
2. *$IM(E) \geq I_{min}$;*
3. *$DU(E) \geq D_{min}$.*

## 3 Algorithm

An algorithm for mining exceptions is given in Fig. 1. Phase 1 (line 1) identifies all probe sets. Phase 2 (lines 2-9) generates all candidate exceptions, combining each probe set in *ProbeSets* with each positive association rule in $PR$ (line 5). In order to compute the $IM$ and $DU$ indices, we need to count the actual

**Input:** *MinSup, $I_{min}$, $D_{min}$ - threshold values*; *PR - a set of multidimensional rules*; *Atrib - a set of attributes*; **Output:** *ME - a set of mined exceptions;*

**procedure** `FindExceptions`

1. *ProbeSets* = generate all possible probe sets from *Atrib*;
2. $CandidateExceptions = \emptyset$; $ConditionsSet = \emptyset$; $ME = \emptyset$
3. **for** each rule $R: A \Rightarrow B$ in $PR$ **do**
4. **for** each probe set $Z$ in *ProbeSets* **do**
5. $CandidateExceptions = CandidateExceptions \cup (A \Rightarrow B\ [Z])$;
6. $X' = \{\{A\}, \{B\}, \{Z\}, \{A, B\}, \{A, Z\}, \{B, Z\}, \{A, B, Z\}\}$ ;
7. $ConditionsSet = ConditionsSet \cup X'$ ;
8. **end for**;
9. **end for**;
10. perform a database scan to count the support of all sets in *ConditionsSet*;
11. **for** each candidate exception $E': A \Rightarrow B\ [Z]$ in *CandidateExceptions* **do**
12. **if** $Sup(A \cup Z) \geq MinSup$ and $Sup(B \cup Z) \geq MinSup$ and $IM(E') \geq I_{min}$ and $DU(E') \geq D_{min}$ **then** $ME = ME \cup (A \overset{-s}{\Longrightarrow} B\ [Z])$;
13. **end for**;

**Fig. 1.** Algorithm for mining exceptions in multidimensional databases

support values for the following sets: $\{A\}$, $\{B\}$, $\{Z\}$, $\{A, B\}$, $\{A, Z\}$, $\{B, Z\}$, and $\{A, B, Z\}$. The data structure *ConditionsSet* is used to keep counters for all these sets (lines 6-7). It can be implemented as a *hash tree*, for example. In phase 3 (line 10) an algorithm such as Apriori [1] counts the support of the sets stored in *ConditionsSet*. Finally, phase 4 (lines 11-13) generates the exceptions.

## 4 Experimental Results

The proposed algorithm was implemented and a test was carried out on the *Mushrooms data set* [2]. This database contains 8124 tuples and 22 attributes used to describe mushrooms. A target attribute classifies each mushroom as either *edible* or *poisonous*. We use the following threshold settings on the experiment: $MinSup = 0.20\%$, $I_{min} = 0.40$, and $D_{min} = 0.10$. The evaluated rule was $(Habitat = \text{“}Grasses\text{”}) \Rightarrow (Class \text{“}Edible\text{”})$, with Sup = 17.33% and Conf = 65.55%. It indicates that great part of the mushrooms specimens that grow on grasses are edible. We use the remaining 20 attributes to form the probe sets. The maximum size of $Z$ was restricted to 3.

Table 2 shows some of the mined exceptions, ranked by the *DU* index. The highest values for the *DU* measure (exceptions 1 and 3) were able to represent the best exceptions. The exception 1 shows a very interesting situation: $Z$ is independent of both $A$ and $B$. However, $Z$ and the original positive rule are highly negative dependent ($IM = 1$). The exceptions 26 and 43 show another interesting aspect: $Z$ and $B$ are positively dependent. However, once again, the $IM$ values are high. The exception 100 is less interesting due to the high negative

**Table 2.** Experimental results

| Rank | $Z$ (Probe Set) | $IM$ | $DU$ | $ND_{A,Z}$ | $ND_{B,Z}$ |
|---|---|---|---|---|---|
| 1 | ($CapShape$ = "$Flat$"), ($StalkShape$ = "$Enlarging$"), ($StalkSurfBelowRing$ = "$Smooth$") | 1.0000 | 0.9286 | 0.0714 | 0.0289 |
| 3 | ($GillColor$ = "$White$"), ($StalkSurfBelowRing$ = "$Ibrous$") | 1.0000 | 0.8274 | -0.0801 | 0.1726 |
| 26 | ($Bruises$ = "$True$"), ($Ring\ Number$ = "$Two$") | 1.0000 | 0.4600 | 0.5400 | -0.4610 |
| 43 | ($Population$ = "$Solitary$") | 0.8382 | 0.4214 | 0.4168 | -0.1986 |
| 100 | ($StalkColorBelowRing$ = "$Pink$") | 1.0000 | 0.2909 | 0.7091 | 0.4060 |

dependence between $Z$ and $A$. The adopted approach for mining exceptions was also applied to a real medical data set (the results can be found in [3]).

## 5 Conclusions

In this paper we addressed the problem of mining exceptions from multidimensional databases. The goal is to find rules that become much weaker in some specific subsets of the database. The exceptions are mined only if the candidates do not achieve an expected support. As a future work we intend to evaluate the interestingness of rules with large deviation between the actual and the expected confidence value. Moreover, the scalability of our algorithm should also be investigated, varying the parameters $MinSup$, $I_{min}$ and $D_{min}$.

## References


1. Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules. In 20th VLDB Intl. Conf. (1994).
2. Blake, C. L., Merz, C. J.: UCI Repository of Machine Learning Databases, *http://www.ics.uci.edu/˜mlearn/MLRepository.html*, Dept. of Inform. and Computer Science, University of California, Irvine (1998).
3. Goncalves, E. C., Plastino, A.: Mining Strong Associations and Exceptions in the STULONG Data Set. In 6th ECML/PKDD Discovery Challenge (2004).
4. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. 2nd edn. Morgan Kaufmann (2001).
5. Hussain, F., Liu, H., Suzuki, E., Lu, H.: Exception Rule Mining with a Relative Interestingness Measure. In 4th PAKDD Intl. Conf. (2000).
6. Savasere, A., Omiecinski, E., Navathe, S.: Mining for Strong Negative Associations in a Large Database of Customer Transactions. In 14th ICDE Intl. Conf. (1998).
7. Suzuki, E., Zytkow, J. M.: Unified Algorithm for Undirected Discovery of Exception Rules. In 4th PKDD Intl. Conf. (2000).
8. Wu, X., Zhang, C., Zhang, S.: Mining both Positive and Negative Association Rules. In 19th ICML Intl. Conf. (2002).