Efficient Search of Reliable Exceptions

Huan Liu[1], Hongjun Lu[2], Ling Feng[3] and Farhad Hussain[1]

[1]School of Computing, National University of Singapore, Singapore, 117599
{liuh, farhad}@comp.nus.edu.sg
[2]Department of Computer Science, Hong Kong University of Science and Technology*
[3]Department of Computing, Hong Kong Polytechnic University
luhj@cs.ust.hk, cslfeng@comp.polyu.edu.hk

Abstract. Finding patterns from data sets is a fundamental task of data mining. If we categorize all patterns into *strong, weak,* and *random*, conventional data mining techniques are designed only to find strong patterns, which hold for numerous objects and are usually consistent with the expectations of experts. While such strong patterns are helpful in prediction, the unexpectedness and contradiction exhibited by weak patterns are also very useful although they represent a relatively small number of objects. In this paper, we address the problem of finding weak patterns (i.e., reliable exceptions) from databases. A simple and efficient approach is proposed which uses deviation analysis to identify interesting exceptions and explore reliable ones. Besides, it is flexible in handling both subjective and objective exceptions. We demonstrate the effectiveness of the proposed approach through a set of real-life data sets, and present interesting findings.

1 Introduction

Data mining has attracted much attention from practitioners and researchers in recent years. Combining techniques from the fields of machine learning, statistics and database, data mining works towards finding patterns from huge databases and using them for improved decision making. If we categorize patterns into

- strong patterns - regularities for numerous objects;
- weak patterns - reliable exceptions representing a relatively small number of objects; and
- random patterns - random and unreliable exceptions.

Conventional data mining techniques are only designed to find the strong patterns which have high predictive accuracy or correlation. This is because we normally want to find such kinds of patterns that can help the prediction task. However, in certain tasks of data mining, we may seek more than predicting: as strong patterns are usually consistent with the expectations of experts, we want to know what is in the data that we do not know yet. Therefore, in some cases, we are more interested in finding out those weak patterns outside the strong

* on leave from School of Computing, National University of Singapore.

ones. Usually, such patterns (reliable exceptions) are unknown, unexpected, or contradictory to what the user believes. Hence, they are novel and potentially more interesting than strong patterns to the user. For example, if we are told that "some kind of jobless applicants are granted credit", that will be more novel and interesting, as compared to "jobless applicants are not granted credit". Moreover, an exception rule is often beneficial since it differs from a common sense rule which is often a basis for people's daily activity [16].

1.1 Insufficient support from current data mining techniques

Most current data mining techniques cannot effectively support weak pattern mining. Taking association rule mining as an example, all association rules $X \Rightarrow Y$ with *support* s and *confidence* c in a database are found in two phases. In the first expensive phase, the database is searched for all frequent itemsets, i.e., sets of items that occur together in at least $s\%$ of instances (records) in the database. A fast level-wise algorithm Apriori [1, 2] was developed to address the nature of this problem. It derives the frequent k-itemsets from frequent $(k-1)$-itemsets. Apriori constructs a candidate set C_k from frequent itemset L_{k-1}, counts the number of occurrences of each candidate itemset, and then determines the set L_k of frequent itemsets based on pre-determined support s.

With the nice *downward* closure property (i.e., if a $(k+1)$-itemset has support s, all its k-subsets also have support s), Apriori can efficiently cross out some $(k+1)$-itemsets which cannot have s, and only check the remaining $(k+1)$-itemsets. For instance, suppose $Sup(\{jobless, not\_granted\}) = .70$ and $Sup(\{jobless, granted\}) = .30$. By setting support threshold as .50, the strong frequent itemset $\{jobless, not\_granted\}$ is kept while the latter is pruned right away, and no further itemset like $\{jobless, female, granted\}$ will be explored. The search space is thus reduced greatly.

After obtaining all frequent itemsets, the second phase will generate association rules from each frequent itemset so that more than $c\%$ of records that contain X also contain Y. Since relatively infrequent itemsets like {jobless, female, granted} are omitted, some reliable exception rules such as $jobless \wedge female \Rightarrow granted$ cannot be discovered although they show high confidence.

1.2 Intuitive extensions cannot help either

There are some intuitive ways we can think of to generate these weak patterns using traditional data mining techniques. To find exception association rules, for example, we can introduce two support thresholds $[s_1, s_2]$ as delimiters instead of only one $[s_1, 100\%]$, and search those itemsets whose support values fall into this range of $[s_1, s_2]$. However, in such case, the efficient pruning property - downward closure no longer exists, which may lead to an intolerable mining performance.

Another straightforward method is generate-and-remove: for a data set D, select the favorite induction algorithm, say C4.5 [12]; generate rules R on D; remove D' from D that are covered correctly by R; then generate rules R' on $D-D'$, repeat the process until no data is left. We can be sure that this procedure can find us many weak patterns. However, it is obvious that these weak patterns are induced independently of the strong patterns. An immediate problem is that there could be too many of weak rules. Although there are many, some interesting exceptions may be missed due to the bias of a chosen induction algorithm.

Generating all possible rules R_{all} is also a possible approach. It is certain that R_{all} will contain any interesting weak rules. However, there could be too many rules - it could be much more than the number of instances of data.

1.3 Related work

Generally, there are two schools in finding interesting and unexpected patterns: subjective vs. objective. The major objective factors used to measure the interestingness of discovered patterns include strength, confidence, coverage and simplicity [10, 3, 13].

Not all strong rules with high confidence and support are interesting as they may correspond to some prior knowledge or expectations. The importance of prior knowledge cannot be over-stressed [4]. [14, 9] propose two subjective measurements: unexpectedness and actionability, and define the unexpected measure of interestingness in terms of a belief system. [5] uses rule templates to find interesting rules from the set of all discovered association rules. [6] analyze the discovered classification rules against a set of general impressions that are specified using a representation language. Only those patterns conforming to these impressions are regarded as unexpected. The issue on interesting deviations is also discussed in [11].

As the interestingness of a pattern itself is subjective, a rule could be interesting to one user but not interesting to another. Thus, most of the previous work so far rely on users to distinguish those reliable exception patterns based on the existing concepts. One potential problem raised is that users' subjective judgments may be unreliable and uncertain in case the discovered rules are numerous. Recently, [16] proposes an autonomous probabilistic estimation approach which can discover all rule pairs (i.e., an exception rule associated with a common sense rule) with high confidence. Neither users' evaluation nor domain knowledge is required.

1.4 Our work

In this paper, we address the problem of finding weak patterns, i.e., reliable exceptions, from datasets. A simple yet flexible approach is proposed which works around the strong patterns and uses deviation analysis to find reliable exceptions. Different from previous work [16], we shun searching for strong common sense patterns; instead, we directly identify those exceptional instances and mine weak patterns from them. Besides the promised efficiency for weak pattern mining, the proposed method can also handle both subjective and objective exceptions. To demonstrate the effectiveness of this novel method, we apply it to a set of real-life data sets [8], and indeed find out some interesting exception patterns. From the mushroom data set, we can achieve more reliable exceptions, compared to the results from [16]. The remainder of the paper is organized as follows: section 2 gives a detailed description of the proposed approach. Experiments are conducted and some interesting exceptions are presented in section 3. Section 4 concludes this work.

2 A Focused Approach

Our approach is based on the following observations: (1) Any exception would have a low support [2] found in the data, otherwise it might be a strong pattern. (2) A reasonable induction algorithm can summarize data and learn rules. (3) Attributes in the rules are salient features. Observation (1) suggests that exceptions cannot be discovered from the data by applying standard machine learning techniques. Observations (2) and (3) allow us to focus on the important features so that we are more focused and an efficient method is possible for finding reliable exceptions. Our approach consists of the four phases below:

1. Rule induction and focusing.

This phase obtains the strong patterns. Normally, a user can stop here for his preliminary data mining probing. If the number of rules is too many, the user can choose to focus on the strongest rules or follow the suggestions in [6]. Let's assume that a few rules have caught our attention and we are curious to know any reliable exceptions with respect to these strong patterns. This is a filtering step that helps us focus on a few attributes quickly. If we are confident about what we want to investigate, i.e., we know the relevant attributes, then this step can be replaced by user specification of relevant attributes.

2. Contingency table and deviation.

Now we focus on some attribute in a rule. we can use these attributes and the class attribute to build a two-way contingency table that allows us to calculate deviations [15][1].

Class	Attribute				R-Total
	V_1	V_2	...	V_c	
C_1	$(n_{11})x_{11}$	$(n_{12})x_{12}$	...	$(n_{1c})x_{1c}$	$n_{1.}$
C_2	$(n_{21})x_{21}$	$(n_{22})x_{22}$	...	$(n_{2c})x_{2c}$	$n_{2.}$
...	...	...	...	...	
C_r	$(n_{r1})x_{r1}$	$(n_{r2})x_{r2}$	...	$(n_{rc})x_{rc}$	$n_{r.}$
C-Total	$n_{.1}$	$n_{.2}$	...	$n_{.c}$	n

In the table, x_{ij} are the frequencies of occurrence found in the data. $n_{ij} = n_{i.}n_{.j}/n$ which are the expected frequencies of occurrence, $n = \sum_{i=1}^{r}\sum_{j=1}^{c} x_{ij}$ - total, $n_{.j} = \sum_{i=1}^{r} x_{ij}$ is a column total (C-Total), and $n_{i.} = \sum_{j=1}^{c} x_{ij}$ is a row total (R-Total). Using the expected frequency as the norm, we can define the deviation as

$$\delta_{ij} = \frac{x_{ij} - nij}{n_{ij}}.$$

[1] For more than one attribute, we need to resort to multi-way contingency tables.

3. Positive, negative, and outstanding deviations.

Using the above definition to calculate deviations, we can expect to have three types: positive, zero, or negative. If the deviation is positive, it suggests that what is concerned is consistent with the strong patterns; if it is zero, it is the norm; and if it is negative, what's concerned is inconsistent with the strong patterns. The value of δ displays the magnitude of deviation. A large value means the deviation could be caused by chance. Since we are concerned about reliable exceptions and reliability is subject to the user's need, we specify a threshold δ_t to define how reliable is reliable. If $\delta_t > 0$, any deviation $\delta > \delta_t$ is positively outstanding; if $\delta_t < 0$, any deviation $\delta < \delta_t$ is negatively outstanding. Deviations are powerful and useful in this case as they provide a simple way of identifying interesting patterns in the data. This has been shown in the *KEFIR* [7] application.

4. Reliable exceptions.

After we have identified outstanding, negative deviations of attribute-values with respect to the class label, we can get all the instances that contain these attribute-value and class pairs, and perform further mining on the focused dataset - a window, using any data mining techniques we prefer. For example, we can continue searching for frequent itemsets to investigate the above exceptional combinations. As the number of instances is much smaller now than the original one, the mining performance should improve greatly. Reliable exceptions can be those longest frequent itemsets with high support in the window.

A strong association rule found in the window may itself be a strong rule that can be found in the whole data set. We need to make sure what we find are weak patterns - low support but high confidence. In other words, any sub-itemsets found in a reliable exception should be excluded if they hold high support in the whole data. A simple method is like this: assuming that X is a sub-itemset that does not include the negatively deviated attribute in a strong association rule found in the window, we can compare the support sup_{win} of X in the window and its counterpart sup_{who} in the whole data. Note that what we really want to check is $P(X, c)$ for the window and $P(X)$ for the whole data with respect to $X \rightarrow c$. If we consider their ratio, they are actually the confidence values. Therefore, if the difference $conf_{win} - conf_{who}$ is sufficiently large (as $conf_{win}$ is always 1, the large difference means a low $conf_{who}$), we should be satisfied that X's high confidence is unique to the window, otherwise, X does not have sufficient evidence to be included in the final reliable exception.

From exception rules to common sense and reference rules

Before we verify the proposed approach, we would like to say a few words about the exceptions found by our approach and those defined in [16]. Briefly, Suzuki suggested that for an exception rule, we should also be able to find its corresponding common sense rule and reference rule. A reference rule should hold low support and low confidence. Due to the nature of negative deviations, it is obvious that every exception rule found by our approach has a corresponding common sense rule (with positive deviation). A reference rule can also be found in our effort as we remove high confidence sub-itemsets mentioned in the previous paragraph. In summary, common sense and reference rules can be obtained based on exception rules.

3 Searching for Exceptions from Data Sets

In this section, we give detailed accounts of applying our approach to bench mark data sets [8].

3.1 Experiment 1 - the credit data set

The first data set used is the credit database which includes 10 attributes and 125 instances, recording situations where people are or aren't granted credit.

Step 1. Identifying relevant attributes

We ran C4.5rules on the data and obtained 5 rules in which the condition part of 3 rules contains attribute "Jobless". We thus consider it as relevant, and the class attribute is "Credit". As we can see, this step could be objective - the relevant attributes are obtained from an algorithm, or subjective - the relevant attributes are specified by an expert or user.

Step 2. Building contingency tables

Table 1 is a contingency table about "Jobless" and "Credit". For continuous attributes, the different range groups are drawn directly from the discretized results generated by the standard equal width method. The objective of the table is to determine whether the two directions of the pairings of "Jobless" and "Credit" are independent (as in the χ^2 test). Each cell of the table represents one of the $k = 2 \times 2 = 4$ categories of a two-directional classification of the $n = 125$ observations. The row-totals are $n_{1.} = 40$ for "Credit = No" and $n_{2.} = 85$ for "Credit = Yes", while the column-totals are $n_{.1} = 111$ for "Jobless = No" and $n_{.2} = 14$ for "Jobless = Yes". For example, x_{12} represents the (observed counts) number of applicants that are jobless and not granted a credit, while n_{12} represents the corresponding expected counts.

Table 1. (Expected) and observed cell counts for the "jobless" classification

Class	Attribute=Jobless		
grant	No	Yes	R-Total
No	(35.5) 28	(4.5) 12	40
Yes	(75.5) 83	(9.5) 02	85
C-Total	111	14	125

In contingency table analysis, if the two attributes are independent, the probability that an item is classified in any particular cell is given by the product of the corresponding marginal probabilities. This is because in a statistical sense, we know that independence of events A and B implies P($A \wedge B$) = P(A)P(B). It can be shown that the null hypothesis that the directions of classification are independent (in Table 1) is equivalent to the hypothesis that every cell probability is equal to the product of its respective row and column marginal probabilities.

Step 3. Identifying outstanding negative deviations

In Table 1, under the assumption of independence, we can compute the deviation δ_{ij} of the observed counts from the expected counts as $(\frac{x_{ij}-n_{ij}}{n_{ij}})$. Table 2 shows the deviations, sorted by their magnitudes.

Table 2. Deviation analysis of the jobless classification.

Jobless	Class	x_{ij}	n_{ij}	δ
Yes	No	12	4.5	* +1.68
Yes	Yes	02	9.5	* -.79
No	No	28	35.5	-.21
No	Yes	83	75.5	+.10

If the default threshold is $\delta_t = -.30$, we have two outstanding deviations denoted by "*" in Table 2. The large positive deviation is consistent with the trend suggested by the "norm". It is the large negative deviation that can lead to a reliable exception. The pattern "a jobless applicant is not given a credit grant" exhibits the most significant "positive deviation" at δ=+1.68. Such patterns are always statistically significant, intuitive and represented by strong patterns. For example, the association below has a very high confidence.

Jobless="Yes", ... $\rightarrow$ class="No" [confidence = .857]

The outstanding negative deviation ($\delta = -.79$) says that "Jobless" applicants are given "Credit". This is intuitively wrong, or unexpected, and naturally worthy of probing.

Step 4. Finding reliable exceptions

Following the findings suggested by the outstanding negative deviation, we gather the instances that satisfy Jobless = "Yes" and Credit = "Yes", or "the jobless being granted a credit". The satisfied instances form a window, the number of instances defines the window size. Applying Apriori, for this application, we obtain the weak pattern "A female jobless with little working experience is given credit" after removing those items with high overall confidence (> 0.70) such as Area = "Good" (0.72) and Married = "Yes" (0.71).

The number of exception candidates is determined by δ_t - the higher the threshold, the fewer the candidates. Although there may not be exceptions for every individual attribute, the more attributes, the more likely that we can find more exceptions.

3.2 Experiment 2 - the mushroom data set

This data set is obtained from [8]. It has 22 attributes, each has 2 to 12 possible values, and 8124 instances with binary class. The two types (classes) of mushrooms are *poisonous* or *edible*. For this data set, we want to verify if our approach can find the exceptions found in [16].

Step 1. Identifying relevant attributes

Since all the four rules in [16] contain *Stalk-root = ?*, we select *Stalk-root* and the class for the experiment.

Step 2. Building contingency tables

This step is routine and results are shown in the table below. We only include expected frequencies for "Stalk-root = ?".

Contingency table for the "Stalk-root" classification.

Type	Attribute=Stalk-root					R-Total
	b	c	e	r	?	
p	1856	44	256	0	(1195)1760	3916
e	1920	512	864	192	(1285) 720	4208
C-Total	3776	556	1120	192	2480	8124

Step 3. Identifying outstanding negative deviations

In the table below, we show the deviations for "Stalk-root = ?" and omit others. The outstanding negative deviation is -.44 for "Type = e".

Deviation analysis of the "Stalk-root = ?" classification.

Stalk-root	Type	x_{ij}	n_{ij}	δ
?	p	1760	1195	+.47
?	e	720	1285	-.44
...				

Step 4. Finding reliable exceptions

Using "Stalk-root = ?" and "Type = e", we find a window with 720 instances. If we simply run Apriori on the window to find frequent itemsets, we obtain two itemsets (recall that confidence for the window is always 1). For exception candidate number 1, we have the following itemset. "Chosen" below indicates whether an item is chosen to remain in the itemset.

Attribute-Value	Overall Conf	Chosen
stalk-root = ?	.29	Y
veil-type = p	.52	Y
stalk-shape = e	.46	Y
gill-size = b	.70	Y
bruises = f	.31	Y
ring-type = p	.79	N

After removing those with high overall confidence (> .70), we have rule 1 as found in [16] repeated here.

CS: bruises=f, g-size=b, stalk-shape=e $\rightarrow$ class=p

#1 RE: C, stalk-root=? $\rightarrow$ class=e

RR: stalk-root=? $\rightarrow$ class=e

where CS is a common sense rule, RE is a reliable exception, C in RE is the conditional part of CS, and RR is a reference rule.

For exception candidate number 2, after removing items with high overall confidence. We obtain a *super-itemset* for exception rules 2, 3, and 4 in [16] as follows. We repeat the three rules below.

Attribute-Value	Overall Conf	Chosen
stalk-root = ?	.29	Y
spore-print-color = w	.24	Y
veil-color = w	.51	Y
veil-type = p	.52	Y
stalk-shape = e	.46	Y
gill-size = b	.70	Y
gill-attachment = f	.51	Y
ring-number = t	.88	N
odor = n	.97	N

	CS: g-attachment=f, stalk-root=? → class=p
#2	RE: C, g-size=b, stalk-shape=3, veil-color=w → class=e
	RR: g-size=b, stalk-shape=3, veil-color=w → class=e
	CS: stalk-root=?, sp-color=w → class=p
#3	RE: C, g-attachment=f, g-size=b, stalk-shape=e → class=e
	RR: g-attachment=f, g-size=b, stalk-shape=e → class=e
	CS: stalk-root=?, sp-color=w → class=p
#4	RE: C, g-size=b, stalk-shape=3, veil-color=w → class=e
	RR: stalk-root=? → class=e

4 Conclusion

We propose here a simple approach that enables us to study reliable exceptions with respect to a rule of our interest or attributes specified by a user. The major techniques are deviation analysis, windowing, and conventional mining tools (e.g., Apriori for association rule mining). For the concerned attributes, we first find their negative deviations which determine the window, and then search reliable patterns from the window using any data mining tools we want. Reliable exceptions are those patterns that are only valid in the window.

This approach is efficient because (1) it works around focused attributes, thus avoiding search all attributes in the database; (2) we are only concerned about negative deviations; (3) it only scan the data once to create the window; and (4) the window size (i.e., number of instances) is usually much smaller than the number of instances in the data. Besides, such approach is also flexible to handle both subjective and objective prior knowledge. Weak pattern finding is an important area for effective, actionable, and focused data mining.

Acknowledgments

We would like thank Gurmit Singh, Danny Oh, and Soh Seng Guan for implementing some of the programs used for this project. Special thanks go to Kian Sing Ng for his work in the beginning of this project.

References

1. R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *Proc. of the ACM SIGMOD Conference on Management of Data*, pages 207–216, Washington D.C., USA, May 1993.
2. R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proc. of the 20th Conference on Very Large Data Bases*, pages 478–499, Santiago, Chile, September 1994.
3. W. Frawely, G. Piatesky-Shapiro, and C. Matheus. *Knowledge Discovery in database: an overview.* G. Piatesky-Shapiro and W. Frawley (eds.) Knowledge Discovery in Database, AAAI/MIT Press, 1991.
4. B.R. Gaines. An ounce of knowledge is worth a ton of data. In A.M. Segre, editor, *Proceedings of The Sixth International Workshop on Machine Learning*, pages 156 – 159. Morgan Kaufmann Publishers, Inc., 1989.
5. M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, and I. Verkamo. Finding interesting rules from large sets of discovered association rules. In *Proc. of the Third International Conference on Information and Knowledge Management*, pages 401–407, November 1994.
6. B. Liu, W. Hsu, and S. Chen. Using general impression to analyze discovered classification rules. In *Proc. of the Third International Conference on Knowledge Discovery and Data Mining*, pages 31–36, Newport Beach, California, USA, 1997.
7. C. Matheus, G. Piatetsky-Shapiro, and D. McNeil. An application of KEFIR to the analysis of healthcare information. In *Proc. of the Eleventh International Conference on Artificial Intelligence, Workshop on Knowledge Discovery in Databases*, pages 25–36, ., 1994.
8. C. Merz and P. Murphy. UCI repository of machine learning databases. Technical Report http://www.ics.uci.edu/ mlearn/MLRepository.html.Irvine, CA: University of California, Department of Information and Computer Science, 1996.
9. B. Padmanabhan and A. Tuzhilin. A belief-driven method for discovering unexpected patterns. In *Proc. of the Fourth International Conference on Knowledge Discovery and Data Mining*, pages 27–31, April 1998.
10. G. Piatetsky-Shapiro. *Discovery, analysis, and presentation of strong rules.* G. Piatesky-Shapiro and W. Frawley (eds.) Knowledge Discovery in Database, AAAI/MIT Press, 1991.
11. G. Piatetsky-Shapiro and C. Matheus. The interestingness of deviations. In *AAAI Workshop on Knowledge Discovery in Database*, pages 25–36, Seattle, Washington, July 1994.
12. J. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kauffmann Publishers, Inc., 1993.
13. A. Silberschatz and A. Tuzhilin. On subjective measures of interestingness in knowledge discovery. In *Proc. of the First International Conference on Knowledge Discovery and Data Mining*, pages 275–281, Montreal, 1995.
14. A. Silberschatz and A. Tuzhilin. What makes patterns interesting in knowledge discovery systems. *IEEE Trans. on Knoweldge and Data Engineering*, 8(6):970–974, 1996.
15. P. Smith. *Into Statistics*. Springer-Verlag, Singapore, 1998.
16. E. Suzuki. Autonomous discovery of reliable exception rules. In *Proc. of the 3rd International Conference on Knowledge Discovery and Data Mining*, pages 259–263, Newport Beach, CA, USA., August 1997.