A Method for Refining Knowledge Rules Using Exceptions

Ronaldo Cristiano Prati, Maria Carolina Monard, and André C. P. L. F. de Carvalho

Laboratory of Computational Intelligence - LABIC
Department of Computer Science and Statistics - SCE
Institute of Mathematics and Computer Science - ICMC
University of São Paulo - Campus of So Carlos
P. O. Box 668, 13560-970, So Carlos, SP, Brazil
Phone: +55-16-273-9692. FAX: +55-16-273-9751.
{prati, mcmonard, andre}@icmc.usp.br

Abstract. The search for patterns in data sets is a fundamental task in Data Mining, where Machine Learning algorithms are generally used. However, Machine Learning algorithms have biases that strengthen the classification task, not taking into consideration exceptions. Exceptions contradict common sense rules. They are generally unknown, unexpected and contradictory to the user believes. For this reason, exceptions may be interesting. In this work we propose a method to find exceptions out from common sense rules. Besides, we apply the proposed method in a real world data set, to discover rules and exceptions in the HIV virus protein cleavage process.

1 Introduction

Data collection is a daily activity of many organizations in business, science, education, medicine, and others. The volume of the data contained in those data sources often exceeds our ability to analyze it efficiently, resulting in a gap between the data collection and its understanding. Data mining is a relatively new research area, which aims to fill this gap, generally looking for rules that express patterns and/or building models from data.

A crucial aspect of data mining is that the discovered rules should be somehow interesting to the domain expert, where the term interesting arguably has to do with surprise (unexpectedness), usefulness and novelty [3]. Another important aspect is that, besides describing the data in a more concise way, the discovered knowledge should be comprehensible to the domain expert [9].

In order to extract that sort of knowledge several methods can be used, and symbolic Machine Learning (ML) algorithms are among these methods. These algorithms induce symbolic hypotheses that are intelligible to humans such as hypotheses represented as a set of rules. This sort of representation may allow us to "explain" the data. Besides, symbolic knowledge could provide some

useful insights that may help the domain expert to understand the process that produced the data.

However, rule learning algorithms are mainly designed to induce classification and prediction rules that can predict new cases with high accuracy. In order to accomplish this goal, ML algorithms generally use domain independent biases[1] and heuristics to induce a classifier consisting of a small set of rules. From the knowledge discovery point of view, this approach has two main disadvantages: first, in order to produce a small set of rules, ML algorithms favor the discovery of general rules, which have high predictive accuracy and correlation. However, this sort of rules generally express common sense knowledge, resulting in many interesting and useful rules not being discovered. Second, the domain independent biases, especially the ones related to the language used to express the knowledge, could induce rules difficult to understand.

Exceptions are defined as rules that contradict common believes. This kind of rules can play an important role in the process of understanding the underlying data, as well as in making critical decisions. By contradicting the user's common believes, exceptions are bound to be interesting. Another important aspect regarding exceptions is that they are always associated with a common sense rule. This implies that the expression of exceptions is performed in a localized manner [6]. This locality concept makes exceptions easy to understand by the domain expert, since the general concept is described first and, afterwards, the exceptions are added as a refinement to the general concept.

Several approaches have been proposed to generate exceptions. Nevertheless, these approaches are mainly designed to association rule mining. In this work we propose a method that generates exceptions out of general classification rules. Furthermore, we apply our proposed approach to a HIV protease cleavage data set in order to induce knowledge rules and exceptions from it.

This work is organized as follow: section 2 presents the motivation of this work. Section 3 presents the exception concept used in this work. Section 4 describes our approach to generate exceptions out of general classification rules. A case stuy is shown in Section 5 and finally, Section 6 concludes the paper.

2 Motivation

In this work we are interested in symbolic ML algorithms that induce rules[2] from a dataset D, which consists of a set of n instances described by m distinctive attributes $X_1, X_2, \ldots, X_m$. The rules induced in such a dataset are targeted at a specific attribute, often called class attribute, which can assume one of k possible distinct values, called labels or classes. The aim of the set of rules induced by a

[1] Bias can be defined as any basis for choosing one generalization over another, other than strict consistency with the data.

[2] Some symbolic ML algorithms can also induce Decision Trees. As we can always rewrite a decision tree as a set of rules, from now on the term **rule** represents either a rule directly induced by a ML algorithm or the one obtained by rewriting a branch of a decision tree as a rule.

ML algorithm is to classify an instance that has an unknown class value with one of the k possible classes. We focus on symbolic algorithms because the user normally wants to know both the unknown class of an instance and how the other attributes are related to the target attribute. The set of rules R_j, $j = 1, \ldots, p$, induced by these algorithms are generally in the format

$$\text{R: } \underbrace{\textit{if } < \text{condition} >}_{\textit{Body or } B} \textit{ then } \underbrace{< \text{class} = C_i >}_{\textit{Head or } H},$$

or, in brief, $B \rightarrow H$. For rule learning algorithms that have the same representational power as propositional logic, condition is a disjoint of restrictions among the attributes, such as X_i op value, where op can be anyone in the set $\{=, <, >, \leq, \geq, \in\}$ and C_i is one of the k possible classes.

Classical rule learning algorithms are mainly developed to induce sets of rules for classification or prediction tasks, whose aim is to predict or classify new instances with as high accuracy as possible. In other words, these algorithms try to induce rules with high accuracy and support, so that these rules are gathered in a final set of rules, called classifier, that has high accuracy. Although this approach produces consistent classifiers, some of the induced rules may be both trivial and difficult to understand by humans.

The most trivial way to discover novel knowledge is to individually evaluate the rules that constitute the classifier, filtering the whole set of rules in order to select the most interesting ones, according to some objective or subjective criteria [4]. Since those rules are mainly induced focusing on the classification accuracy bias, they generally express common sense knowledge (*i.e.*, they are common sense or general rules). Even though general rules are consistent with the experts' expectations, in some activities is interesting to find out others kinds of rules outside the general ones.

Another important issue is how we interpret and understand the induced rules. ML algorithms can induce both disjoint and overlaid rules. Furthermore, overlaid rules can be either ordered (decision lists) or unordered (independent rules).

From a knowledge discovery point of view, rules in a decision list are difficult to understand by the domain expert since they are meaningful only in the context of all the preceding rules. Alternatively, disjoint and unordered rules can be individually interpreted. Nevertheless, the rules present in those sets of rules are uncorrelated with each other. In many DM applications, establishing a type of relationship among those rules can play an important role in obtaining a good overall understanding of the underlying relationships in the domain.

In our view, the construction of classifiers where the main emphasis lies on the classifier's accuracy fails to reflect the way humans construct and express hypothesis. This is not to say that this goal is irrelevant; notwithstanding, at the knowledge discovery level, and in some practical applications, it is almost worthless the directly application of those algorithms, since they fail on searching novelty patterns and/or expressing the discovered ones closer humans do.

3 Exceptions

From a knowledge representation point of view, one of the main features of rules is that they tend to have exceptions [6]. If we could represent the induced rules in that manner, they would be more intuitive for human users, since humans generally talk about knowledge in terms of general patterns and special cases. For instance, in medical applications, physicians always say that people with certain characteristics tend to have a particular disease; however, in some special situations, they may not develop the disease. Thus, more realistic rules are of the form '**if** P **then** u **unless** Q'. To represent such a rule, we can refine common sense rules by adding exceptions.

Intuitively, exceptions contradict a general or common sense rule. A common sense rule represents a common phenomenon that comes with high support and confidence in a particular domain. Therefore, exceptions to the rules are weak in terms of support, but having confidence similar to the common sense rules [5]. Support is a measure related to the relative frequency of the instances covered by a rule and confidence is related to its accuracy.

In this work, we use the exception concept given in [5], which structurally defines exception as show in Table 1, where the term B' also represents a non-empty set of conjunction of restriction among the attributes. For instance, if we had the common sense rule "*if a person is unemployed then it is not granted the person credit*", we could have an exception such as "*if a person is unemployed but his/her consort is employed then it is granted the person credit*"

$B \rightarrow H$	general rule
	high support, high confidence
$B \wedge B' \rightarrow \neg H$	exception rule
	low support, high confidence
$B' \rightarrow \neg H$	reference rule
	low support, low confidence

Table 1. Rule structure for exceptions

Exceptions help to solve the *understandability* problem. A set of isolated rules is not intuitive to the domain expert since these rules fragment the knowledge. Sets of rules expressed in that manner generally are difficult to read and understand because the domain expert cannot see any relationships among the rules. The concept of locality, which is implicit in exceptions rules, is more intuitive to the domain expert, since it allows him/her to see an overall picture of the domain first and then the special cases.

A related problem is the discovery of interesting or useful rules. The quest for a simple set of rules of the existing classification systems[3] results in many

[3] In general, classification systems use the Ockam's razor advice *"prefer the simplest hypothesis consistent with the data"*, in order to choose from multiple consistent hypothesis.

interesting and useful rules not being discovered. By contradicting the common sense rules, exceptions are generally more interesting and useful to the users. For instance, an exception can represent an important niche in a determined market. If the user could recognize this niche as an exception, he or she could apply to these consumers a more specific marketing campaign.

To illustrate this concept we use an artificial dataset, which is graphically represented in Figure 1. Two attributes, $A1$ and $A2$, and two classes, ▲ and ○, describe the instances. The domain of $A1$ is $\{a, b, c, d\}$, and the domain of $A2$ is $\{x, y\}$.

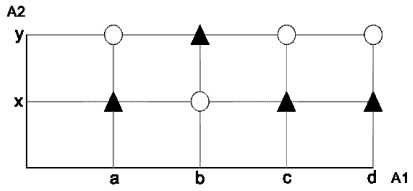

Fig. 1. The example data space

Rule learning algorithms can either treat the attribute $A1 = b$ as noise, generalizing the hypothesis or treat it as a true value, specializing the hypothesis. In the first case, the induced hypothesis consists of the rule set $\{A2 = x \rightarrow ▲$, $A2 = y \rightarrow ○\}$. In the second case, if we constrain the hypothesis language to a set of disjoints as several rule learning algorithms do, the induced hypothesis consists of a rule set having eight rules[4], one to each instance $\{A2 = x \wedge A1 = a \rightarrow ▲$, and so forth$\}$. However, we could represent the hypothesis in a more intuitive way, firstly generalizing and giving an overall idea about the concept, and afterwards treat the special cases locally specifying the exceptions. In this case, the hypothesis is represented as the set of rules $\{A2 = x \rightarrow ▲$ except if $A1 = b \rightarrow ○$ and $A2 = y \rightarrow ○$ except if $A1 = b \rightarrow ▲\}$.

4 Our proposed approach

Although several methods have been proposed in the literature in order to extract exceptions [10, 7, 5, 6], they are generally developed to treat association rule mining and cannot be used to extract exceptions out of classification rules. The main objective of this work is to propose a new method to find exceptions out from general classification rules. This method is mainly based in the following three key principles:

[4] If we enlarge the hypothesis language to contemplate negation or add the in ($\in$) operator (an attribute value is in a set of possible values), the hypothesis might consist of four rules. In both cases, the hypothesis constructed using exceptions is still more intuitive for humans.

1. A reasonable rule induction algorithm can summarize data and learn rules;
2. These algorithms have biases that favor the induction of rules with high support;
3. Exceptions should have low support in the whole dataset, otherwise they would be a common sense rule.

These three principles make difficult the direct induction of exceptions by traditional ML algorithms. The direct extraction of exceptions from a dataset is not a trivial task since ML algorithms biases favor the induction of general rules. Although we may relax these biases in order to induce rules with lower support, there is a high chance that the knowledge would be fragmented among the induced rules. All things considered, our proposal approach is divided in two steps, as follows:

Step 1 — induction of common sense rules. In this step, we use a traditional rule learning algorithm in order to induce general classification rules. As we are mainly interested in the induction of general rules, in this step the main objective is not allowing the induction of highly specialized rules. This can be done by properly configuring the parameters of the learning algorithm.

Normally, a user can stop here for his preliminary data mining probing. The user can also apply a filtering step in order to select the most interesting rules. In our approach, we also apply a filtering step, but with the intent of focusing on some rules to be further treated on the next step. This filtering step is based on the evaluation of a contingency matrix, as show on Table 2. In this table, B denotes the set of instances where the rule condition is true and its complement, $\overline{B}$, denotes the set of instances where the rule condition is false and analogously for H e $\overline{H}$. For each entry in the table, f_x denotes the relative frequency associated to the event x, i.e., f_{bh} denotes the relative frequency of the set of instances where both B and H are true, and so on. We use the value $f_{b\overline{h}}$ in order to filter the rules, as described on Step 2.

	H	$\overline{H}$	
B	f_{bh}	$f_{b\overline{h}}$	f_b
$\overline{B}$	$f_{\overline{b}h}$	$f_{\overline{b}\overline{h}}$	$f_{\overline{b}}$
	f_h	$f_{\overline{h}}$	1

Table 2. Contingency matrix for the rule $B \rightarrow H$

Step 2 — looking for exceptions. In this step we focus on rules that have the highest values of $f_{b\overline{h}}$ in order to search for possible exceptions on these rules. A high value of $f_{b\overline{h}}$ implies that there are instances on which the body of that rule is true but the class of the instances are not the same as the one foreseen by the head of the rule. In other words, there are instances misclassified by that rule.

After identifying the rules with high $f_{b\overline{h}}$ values, starts the search for reference rules that might be exceptions. For each of these rules, and only using the sub-

set of instances that are misclassified by that rule, we look for associations with attribute instances and the negative(s) class(es)[5] foreseen by the rule. If those associations have a minimum support and confidence values in the subset of instances, they represent a reference rule and the couple (general rule, confidence rule) represents one exception. It is worth to note that although reference rules have high support in the subset of instances used to look for exceptions, they probably have small support in the whole dataset.

A major concern about our proposed approach is that it cannot only extract exception out of general classification rules but also preserve the locality concept of exceptions.

5 Case Study

In order to illustrate our approach we choose a real world dataset, related to where a viral protease cleaves HIV viral polyprotein amino acid residues. This dataset is also used by [8, 2]. Table 3 summarizes this dataset.

# features	# instances	unknow values	classes
8 (nominal)	362	no	0 - non-cleavage (68,51%) 1 - cleavage (31,49%)

Table 3. Dataset description

Viral protease plays an important role on the viral cycle, since it is responsible by cleaving the precursor viral polyproteins (the substrate) at specific cleavage-recognition sites when they emerge from the ribosome of the host cell as one long sequence. In other words, proteases post-translate proteins chain into viral proteins and enzymes which will rise to new virus molecules that are then released for the infection of further cells. An understanding of viral protease specificity may help the development of future anti-viral drugs involving protease inhibitors by identifying specific features of protease activity for further experimental investigation.

Each instance of the HIV dataset consists of eight attributes that represent a recognition sequence followed by its class, related to its cleavage-ability. In its turn, each attribute on the recognition sequence represents one amino acid. The attributes on the recognition sequence are sequentially ordered, i.e., the first attribute represents the position one in the sequence, the second attribute the position two, and so forth. The size of the recognition sequence is the same as the size of viral protease. When one of the recognition sequences matches its counterpart in the viral protease, the cleavage occurs between position 4 and 5 of the recognition site. If it does not match, the cleavage does not occur.

[5] In the case of more than two classes, the set of classes not foreseen by H are the negative classes $\overline{H}$.

To avoid a possibility of the exceptions over-fitting the data, we repeated the experiment three times with different training and testing samples. The dataset was divided in three parts and in each experiment two parts were used for training and the other one for testing.

To apply the first step of the proposed methodology we used the See5 program, which induces symbolic decision trees. To generate smaller trees, the option of generating nodes having subsets of possible values for conditions rather than individual values was set up. As the transcription of the decision trees into rules produced a small number of rules, we decided to select all the rules in order to apply the second step.

On the second step, each rule obtained in the first step defines a subset of instances where we need to look for exceptions. As stated before, this subset contains all the instances that are covered by a rule but do not have the same class as the one foreseen by the rule. In order to look for possible reference rules, we applied the association rule-mining algorithm APRIORI [1]. We only select as possible reference rules the ones that present at least a support value 0.5. Further, if the possible reference rule appears at least twice in all the conducted experiments, we assume it is a reference rule and the couple (general rule, reference rule) as a truly exception.

Table 4 shows the tree induced by See5 on the first experiment, transcribed as rules. The numbers in parentheses at the end of each rule represent, respectively, the number of instances correctly and incorrectly covered by the rule.

R1.1	**if** pos4 ∈ {A,R,N,D,C,Q,E,G,H,I,K,P,S,T,W,V} **then** non-cleavage (149,16)
R2.1	**if** pos4 ∈ {L,M,F,Y} **and** pos5 ∈ {A,Q,G,K} **then** non-cleavage (12,3)
R3.1	**if** pos4 ∈ {L,M,F,Y} **and** pos5 ∈ {R,N,D,C,E,H,I,L,M,F,P,S,T,W,Y,V} **then** cleavage (81,15)

Table 4. Transcription of the tree generated by See5 into rules – Execution 1

For example, rule R1.1, which foreseen class non-cleavage, covers a total of 165 instances of the training set; 149 of these instances belong to class non-cleavage and the other 16 to class cleavage. In other words, rule R1.1 misclassified 16 instances of the training set. Using this subset, the association rule mining algorithm found the rule **if** pos6 = E **then** cleavage, which covers 9 of these 16 instances. Furthermore, in the test set, this exception rule correctly covers the only instance that was erroneously covered by rule R1.1, but 2 instances that were correctly covered by rule R1.1 are now erroneously covered.

The same steps were applied to rules R2.1 and R3.1. Rule R2.1, which also foreseen class non-cleavage, covers 15 instances of the training set, 3 of which misclassified with class coverage. Using this subset, the association rule mining algorithm found the rule **if** pos6 = E **then** cleavage, which correctly covers all

three instances. Furthermore, in the test set, this exception rule correctly covers 3 of 5 instances that were erroneously covered by rule R2.1, but 2 instances are now correctly covered by the exception rule. For rule R2.1, the proposed approach did not found any exception.

Table 5 shows the tree induced by See5 on the second experiment, transcribed as rules. Rule R1.2, which foreseen class non-cleavage, covers 173 instances of the training set; 158 of these instances belongs to class non-cleavage and the other 15 to class coverage. In this subset, the association rule mining also found the rule **if** pos6 = E **then** cleavage, which covers 8 instances from this subset. In the test set, this exception correctly covers all 5 instances that were mistakenly covered by the rule. One instance that was correctly covered by the rule R2.1 is now mistakenly covered though. For rule R2.2, our approach did not found any exceptions.

R1.2 **if** pos4 ∈ {A,R,N,D,C,Q,E,G,H,I,K,M,P,S,T,W,V}
then non-cleavage (158,15)
R2.2 **if** pos4 ∈ {L,F,Y}
then cleavage (84,22)

Table 5. Transcription of the tree generated by See5 into rules – Execution 2

Table 6 shows the tree induced by See5 on the third experiment, transcribed as rules. Rule R1.3, which foreseen class non-cleavage, covers 165 instances of the training set; 153 of these instances belongs to class non-cleavage and the other 12 to class coverage. In this subset, the association rule mining also found the rule **if** pos6 = E **then** cleavage, which covers 7 instances from this subset. In the test set, this exception correctly covers 3 of 5 instances that were erroneously covered by rule R2.1. For rule R2.3, our approach did not found any exceptions.

R1.3 **if** pos4 ∈ {A,R,N,D,C,Q,E,G,H,I,K,P,S,T,W,V}
then non-cleavage (153,12)
R2.3 **if** pos4 ∈ {L,M,F,Y}
then cleavage (81,15)

Table 6. Transcription of the tree generated by See5 into rules – Execution 3

As we can see in all three experiments, the proposed approach has found the same exception on rules that foreseen the same non-cleavage class. This enforces the evidences that this is a truly exception and not a causality on the data. To maximize the extracted knowledge, we also apply the proposed approach to the whole dataset. Table 7 shows the final hypothesis found. In this hypothesis, we have added the exception to the rule R1 due to its similarity to the rules induced on previous execution. This rule had covered 229 instances in the whole dataset,

and 17 of them were mistakenly covered. When we added the exception, 10 of these 17 examples were correctly covered.

R1	**if** pos4 ∈ {A,R,N,D,C,Q,E,G,H,I,K,P,S,T,W,V} **then** non-cleavage **exception: if** pos6 = E **then** cleavage
R2	**if** pos4 ∈ {L,M,F,Y} **and** pos5 ∈ {A,R,E,G,H,I,L,M,F,P,T,W,Y,V} **then** cleavage
R3	**if** pos4 ∈ {L,M,F,Y} **and** pos5 ∈ {N,D,C,Q,K,S} **then** non-cleavage

Table 7. Final hypothesis found using our proposal approach

The induced rules confirm the importance on the cleavage process of the amino acids in positions 4 and 5 of the substrate. This point is just where exists the linkage that the catalytic process occurs in order to cleave the substrate (scissile linkage). The generate exceptions also show the importance of the position 6 in this process. This point is also related in [8].

We also try to find out this exception using only a traditional ML algorithm (in this case, we used again the See5 algorithm, without setting up the option that generates subsets of values in the nodes). To this end, we stepwise relax the pruning confidence factor in 5% until the attribute position 6 appears in the induced tree (we stepwise relax this value from 25% (default) until 60%). The result is an induced decision tree with 55 leave nodes that can be translated into 55 rules. In two of these 55 rules the disjoint **if** pos6 = E appears. However, in this case, we cannot see the relationship between the generated rules.

6 Conclusion

This work presents a new methodology to find exceptions out of general classification rules. By contradicting common sense rules, exceptions are generally interesting and useful to users. Besides, exceptions are also more intuitive to humans, since they allow the user to see an overall picture of the domain first and then the special cases.

Furthermore, we also have applied our proposal approach to a HIV protease cleavage dataset, which formulates a hypothesis that consists of general rules and a exception. While the general induced rules are related to the biological process, the exceptions found can provide some insights in order to help the domain expert understands the underlying data. A natural extension of this work is the analysis and validation of the generated rules by domain experts.

References

1. R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In *ACM SIGMOD International Conference on Management of Data*, pages 207–216, Washington, D.C., 1993.
2. Y.-D. Cai and K.-C. Chou. Artificial neural network model for predicting hiv protease cleavage sites in protein. *Advances in Engineering Software*, 29(2):119–128, 1998.
3. U. M. Fayyad, G. Platestsky-Shapiro, and P. Smyth. *Advances in Knowledge Discovery and Data Mining*, chapter From Data Mining to Knowledge Discovery: An Overview, pages 1–30. AAAI Press, Menlo Park, CA, 1996.
4. A. A. Freitas. On rule interestingness measures. *Knowledge-Based Systems*, 12(5–6):309–315, October 1999.
5. F. Hussain, H. Liu, E. Suzuki, and H. Lu. Exception rule mining with a relative interesting measure. In *PAKDD-2000*, volume 1805 of *LNAI*, pages 86–97, Kyoto, Japan, April 2000. Springer-Verlag.
6. J. Kivinen, H. Mannila, and E. Ukknonen. Learning rules with local exceptions. In *Computational Learning Theory: EuroCOLT '93*, pages 35–46, Oxford, march 1994. Clarendon Press.
7. H. Liu, H. Lu, L. Feng, and F. Hussain. Efficient search of reliable exceptions. In *PAKDD'99*, volume 1574 of *LNAI*, pages 194–203. Springer-Verlag, 1999.
8. A. Narayanan, X. Wu, and Z. R. Yang. Mining viral protease data to extract cleavage knowledge. *Bioinformatics*, 18(Suppl. 1):S5–S13, 2002.
9. M. J. Pazzani, S. Mani, and W. R. Shankle. Beyond concise and colorful: Learning intelligible rules. In *KDD'97*, pages 235–238, Menlo Park, California, 1997. AAAI Press.
10. E. Suzuki. Autonomous discovery of reliable exception rules. In *KDD'97*, pages 159–176. AAAI Press, 1997.