POST-PROCESSING OF CLASSIFIERS - KDD

Rafael de Souza Teixeira, Jair B Domeneghi Colmanetti, Deborah Ribeiro Carvalho
Pontifícia Universidade Católica do Paraná (PUC-PR)
E-mails: drdrcarvalho@gmail.com

Abstract: Data Mining potentializes the use of the large volumes of data stored by companies, discovering useful patterns to support decision making. However, it is common that the quantity of these discovered patterns makes the respective analysis unfeasible. This paper proposes and tests two strategies based on generalization to post-process discovered patterns in a decision tree format. Both strategies were tested on a decision tree discovered within a data range of 1,617 employees, having as class attributes the fact that they either were or weren't awarded a sick leave (absence) for orthopedic treatment. The results achieved from the adoption of these two strategies show a reduction of around 60% of the patterns to be analyzed.

Keywords: KDD, Post-processing, Decision tree, Generalization

1. INTRODUCTION

Since the increase in capacity and speed of data storage, information technology has been continuously looking for strategies which optimize the potential of this stored data in support of decision taking.

Among the available alternatives, there is the Knowledge Discovery in Database (KDD), which involves three big stages: pre-processing, data mining and post-processing (Fayyad, Piatetsky-Shapiro, & Smyth, 1996). At the data mining stage, patterns are discovered among the data which were not yet realized and will be evaluated by the administrators during the post-processing stage. However, it is quite frequent for the pattern volumes to exceed the evaluation capacity, demanding strategies which minimize the challenges involved.

Some effort has been made in order to facilitate post-processing. Chen et al. (2004) adopted the decision tree method to diagnose computer system failures. He also made use of several techniques to remove ramifications of these trees, by initially getting rid of all rules which end in success, as his aim was to evaluate failure. He then went on to remove ramifications with low coverage.

In one of his articles, Qiang et al. (2007) claims that decision trees only produce results for visualization purposes, ranked by interest. They do not present solutions which can change something unfavorable into favorable. In the article, the authors work with algorithms in order to extract these actions from the trees.

Milani and Carvalho (2013) suggested a tool which facilitated not only post-processing, but also the two other stages which come before that. It is available, specially at the post-processing stage, the transformation of decision trees into sets of rules which eliminate the redundancies present before the rule, as well as the attribution of interest measures from several generalizations.

Therefore, this paper aims at proposing and implementing two strategies to post-process patterns discovered in the format of classifiers. It is important to highlight that the implementation is available from the contact with the authors.

4. METHODOLOGY

The proposed strategy post-processes classifiers represented in the format of decision tree, obtained from the J48 available at the WEKA tool - Waikato Environment for Knowledge Analysis (HALL et al., 2009). The representation in the format of decision tree was chosen due to the fact that it facilitates the administrator's understanding when it comes to indentifying the attributes most strongly related to the problem investigated. The WEKA tool was selected because of its wide use in processes which involve KDD.

Two functionalities are expected:

- Option 1 – prunes ramifications from a specific level parameterized by the user;
- Option 2 – Eliminates rules of the tree which do not attend a specific limit of example coverage.

In option 1, each path between the root node and the respective leaf node is converted into a rule, but presenting only the conditions in the antecedent, according to the maximum level desired, parameterized by the user. For each pruned rule, it is also totaled the number of records covered by it, according to the domain values of the attribute, which labels the leaf node, selected as class at the discovery of the decision tree.

In option 2, each path between the root node and the respective leaf node either can or can't be totally eliminated from the resulting set of rules, in case it doesn't satisfy the criterion of minimum number of records covered, also parameterized by the user.

From figure1, it is possible to find the pseudocode of the proposed implementation.

```
BEGIN
    option ← INTEGER
    IF option = 1 THEN: //prunning ramifications per level
        file ← File J48
        level ← INTEGER

        FOR line IN file:
            IF line IS TREE:
                tree ← line
        FOR line IN tree:
            actual_level ← COUNT '|' IN line
            IF actual_level > level:
                dictionary, variable ← search_values()
                tree ← variable + dictionary[variable]
                REMOVE line
        file ← tree
    IF option = 2 THEN: //prunning ramifications by percentage
        file ← FILE J48
        percent ← INTEGER
        FOR line IN file:
            IF line IS TREE:
                tree ← line
        FOR line IN tree:
            IF line HAS '(':
                IF rule_data_total > (percent * tree_data_total):
                    EXCLUDE rule
        file ← tree
    SAVE file
END
```

Figure1: Pseudocode of the Post-processing algorithm

The totaling of all records covered by each rule is obtained from the ramification of the generated tree. The search_values() function is responsible for the selection of the variable present in each line, as well as the respective values of coverage (Figure2).

```
String search_values()
BEGIN
        variable ← SELECT IN line BETWEEN ':' AND '('
        positive_value ← SELECT IN line BETWEEN '(' E '\' OR '(' E ')'
        negativer_value ← SELECT IN linha BETWEEN '\' AND ')'
        IF negative_value <> NULL
                dictionary[variable] ← positive_value – negative_value
        ELSE
                dictionary[variable] ← positive_value
        return dictionary, variable
END
```

Figure2 Pseudocode of the lookup function

The algorithm was implemented using the Python v3.5 programming language.

The project to experiment the database was submitted and authorized by the Research Ethics Committee of PUC-PR, Nº 1.183.459.

5. RESULTS

To better exemplify the workings and the exit of the proposed algorithm, a database which contained data of employees who either were or weren't awarded a sick leave due to orthopedic issues was adopted. For these two groups, it was taken into account the total use of 82 orthopedic procedures, demanded between 2007 and 2013. These 82 procedures were indicated by five orthopedic doctors. In order to find the classifier, the statuses 'on sick leave' or 'not on sick leave' were adopted.

```
tempo_de_empresa <= 36
|       tempo_de_empresa <= 5
|       |       res7 <= 0
|       |       |       res1 <= 3: N (36.0/2.0)
|       |       |       res1 > 3: S (12.0/3.0)
|       |       res7 > 0: S (3.0)
|       tempo_de_empresa > 5
|       |       res1 <= 2
|       |       |       res52 <= 1
|       |       |       |       res82 <= 0
|       |       |       |       |       res2 <= 5
|       |       |       |       |       |       res67 <= 0: N (371.0/176.0)
|       |       |       |       |       |       res67 > 0: S (123.0/47.0)
|       |       |       |       |       res2 > 5
|       |       |       |       |       |       res77 <= 0: S (19.0)
|       |       |       |       |       |       res77 > 0
|       |       |       |       |       |       |       res64 <= 0: S (4.0)
|       |       |       |       |       |       |       res64 > 0: N (2.0)
|       |       |       |       res82 > 0: S (212.0/41.0)
|       |       |       res52 > 1: S (37.0/2.0)
|       |       res1 > 2: S (687.0/74.0)
tempo_de_empresa > 36
|       res77 <= 2 RM
|       |       res56 <= 1: N (99.0/24.0)
|       |       res56 > 1: S (6.0/1.0)
|       res77 > 2: S (6.0)
```

Figure3 Discovered tree, to be post-processed

The demand frequency by procedure was totaled by employee. The following attributes were also incorporated into the attributes: gender, age, number of years at the company, job, work place and the type of leave, being: “yes” or “no”, in case the employee has been awarded a sick leave due to orthopedic issues.

The decision tree was discovered (Figure3) from this generated file, which contemplates 1,617 employees, being 1,143 on sick leave and 474 not on sick leave due to orthopedic issues.

It is possible to realize that the discovered tree presents 22 paths, considered complete rules, between the root node and the respective leaf node labeled S (on leave) and N (not on leave). The attribute most strongly related to the fact that the employee was awarded sick leave or not is the number of years the employee (tempo_de_empresa) has worked at the company, being 36 years the cut value.

```
Numero de niveis maximo: 2
-----------------------------------------------------
tempo_de_empresa <= 36
|           tempo_de_empresa <= 5
|           |           res7 <= 0
N  = 37
S  = 11
pruned: 2
tempo_de_empresa <= 36
|           tempo_de_empresa <= 5
|           |           res7 > 0
S  = 3
pruned: 0
tempo_de_empresa <= 36
|           tempo_de_empresa > 5
|           |           res1 <= 2
N  = 287
S  = 481
pruned: 12
tempo_de_empresa <= 36
|           tempo_de_empresa > 5
|           |           res1 > 2
S  = 613
~S  = 74
pruned: 0
tempo_de_empresa > 36
|           res77 <= 2 RM
|           |           res56 <= 1
N  = 75
~N  = 24
pruned: 0
tempo_de_empresa > 36
|           res77 <= 2 RM
|           |           res56 > 1
S  = 5
~S  = 1
pruned: 0
tempo_de_empresa > 36
|           res77 > 2
S  = 6
pruned: 0
```

Figure4 Tree segment post-processed from option 1

It was possible to come up with a set of rules presented in Figure 4 by post-processing the discovered tree, considering Option 1 which facilitates the identification of attributes most strongly related to the fact that the employee has been awarded sick leave or not, due to orthopedic issues. The parameterization value for the referred processing was two for the maximum number of 2 maximum limits. This means that the user recognizes it as the limit of representation of the most strongly related attributes.

Analyzing the first rule of the set, it is possible to realize that the number of years at the company establishes an interval between 5 and 36 years of work and that the next attribute most strongly related is the number of procedures labeled as “res7”. In this context, two ramifications were pruned and the pre-named class is N (Not on leave), with the coverage of 48 (37+11), being 37 of the N class and 11 of the S class (On leave).

The tree segment demonstrated in figure5 was obtained by considering the post-processing of the Tree (Figure3) starting from the elimination of the ramifications with coverage inferior to the specified, having adopted 0.05 as limit value.

```
Porcentagem selecionada: 0.05
Valor correspondente absoluto: 80.85
-------------------------------------------------------
tempo_de_empresa <= 36
|       tempo_de_empresa > 5
|       |       res1 <= 2
|       |       |       res52 <= 1
|       |       |       |       res82 <= 0
|       |       |       |       |       res2 <= 5
|       |       |       |       |       |       res67 <= 0: N (371.0/176.0)
|       |       |       |       |       |       res67 > 0: S (123.0/47.0)
|       |       |       |       res82 > 0: S (212.0/41.0)
|       |       res1 > 2: S (687.0/74.0)
tempo_de_empresa > 36
|       res77 <= 2 RM
|       |       res56 <= 1: N (99.0/24.0)
```

Figure5 Tree segment post-processed from option 2

From the post-processed Tree (Figure3), all ramifications with coverage lower than 80 records were eliminated, generalizing the discovered tree. The absolute correspondent value of 80.85 corresponds to the percentage parameterized on the base total record from which the tree was discovered. From 22 constant rules in the originally discovered tree, there was a reduction to only 5 rules (Figure5).

6. CONCLUSIONS

This paper proposed and tested two strategies to post-process patterns discovered in the format of classifiers represented from decision trees.

Starting from the first option of a tree with 27 conditions, it was possible to reduce the analysis to 11 conditions. Considering the second option of post-processing, it was possible to reduce from 27 ramifications to 13 ramifications.

In both cases, it is clear that this generalization of the two strategies facilitates the administrator’s analysis of the discovered patterns.

It is important to highlight that the implementation is available from the contact with the authors.

REFERENCES

CHEN, M. et al. **Failure diagnosis using decision trees**. Autonomic Computing, 2004. Proceedings. International Conference On, 36 – 43, 2004. Disponível em: < http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=1301345> Acesso em: 27 dez. 2014

HOLMES, G. DONKIN, A. WITTEN, I. H. **WEKA:A Machine Learning Workbench**. Intelligent Information Systems,1994. Proceedings Of The 1994 Second Australian And New Zealand Conference On, 357 – 361, 1994. Disponível em: < http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=396988> Acesso em: 27 dez. 2014

JOHNSON-THOMPSON M. C.; GUTHRIE J. **Ongoing research to identify environmental risk factors in breast carcinoma**. *Cancer* 2000: 88 (S5):1224-1229.

MILANI, C. S.; CARVALHO, D. R.; **Prepos Environment: A Simple Tool For Discovering Interesting Knowledge**. Iberoamerican Journal Of Applied Computing, v. 3, n. 2, 41-52, 2013. Disponivel em: <http://www.revistas2.uepg.br/index.php/ijac/article/view/6141> Acesso em: 19 jan. 2015

QIANG, Y.; JIE, Y.; CHARLES, L.; RONG, P. **Extracting Actionable Knowledge from Decision Trees**. IEEE Transactions On Knowledge And Data Enginnering, v. 19, n. 1, 43-56, 2007. Disponível em: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4016514> Acesso em: 29 nov. 2014

WEKA. Hamilton, Nova Zelândia, MACHINE LEARNING GROUP, Univ. de Waikato. Disponível em: <http://www.cs.waikato.ac.nz/ml/weka> Acesso em: 8 jan. 2015