Generalizing Knowledge in Decentralized Rule-Based Models

Pedro Strecht(✉), João Mendes-Moreira, and Carlos Soares

INESC TEC/Faculdade de Engenharia, Universidade do Porto,
Rua Dr. Roberto Frias, 4200-465 Porto, Portugal
{pstrecht,jmoreira,csoares}@fe.up.pt

Abstract. Knowledge generalization of ruled-based models, such as decision trees or decision rules, have emerged from different backgrounds. This particular kind of models, given their interpretability, offer several possibilities to be combined. Despite each distinct context, common patterns have emerged revealing the systemic nature of the problem. In this paper, we look at the problem of generalizing the knowledge contained in a set of models as a process formalizing the operations that can be addressed in alternative ways. We also include a set-up to evaluate generalized models based on their ability to replace the base ones from a predictive performance perspective, without loss of interpretability.

Keywords: Knowledge generalization · Rule-based models

1 Introduction

Rules are usually presented in the canonical form of IF *antecedent* THEN *consequent*. The antecedent is a conjunction of relational conditions implicating independent variables to predict the value of a target variable of interest, the consequent. Rule-based models [14] make use of a set of rules to describe how independent variables can explain the value of an objective variable. A popular example are decision trees [9] which offer a flowchart representation of rules promoting easier human interpretation. Another are decision lists [13] which present ordered rule-sets in the canonical form. Interpretability is an important property in domains where a decision support system is able to explain and justify its decisions [7]. Therefore, the number of organizations using rule-based models has been increasing.

Generating models to predict or describe a phenomenon in organizations with a decentralized activity presents challenges. An example is a company that does its sales through subsidiaries or even by authorized individual distributors. Each sale is carried out by a single subsidiary which is considered a business unit of the organization. Another is of a university offering numerous courses to its students. Each course is offered by a faculty, or a department, which are further examples of business units. These organizations have their problem domain broken down into what can be seen as several units, i.e., a decentralized

A. Monreale et al. (Eds.): ECML PKDD 2018 Workshops, CCIS 967, pp. 29–36, 2019.
https://doi.org/10.1007/978-3-030-14880-5_3

context. Such parallelism makes it increasingly common to generate not a single model but multiple models, each relating to a business unit. In the company example, each subsidiary can have a model to describe/predict its monthly sales level. In the university context, each course can have a model to describe/predict the performance of the students enrolled in it. Yet, the fact that these models are associated with only one unit makes it hard to find generalized knowledge representative of the whole organization. In the aforementioned examples, this could be the overall monthly sales level behavior of the organization or the overall performance behavior of the students of the university during an academic year.

In this paper, we look at the problem of how to gather and generalize the knowledge contained in a large number of rule-based models from organizations with distributed activity. Merging models has been presented in our previous work [12] as an approach to address the problem. However, it was explained deeply intertwined within the context of a case study. This entanglement also occurs in other works, together with distinct vocabulary to describe the same concepts. It is clear that there are patterns in the intermediate phases of each approach, even if named differently. We address this abstraction by presenting a process to generalize rule-based models, such as decision trees or decision rules.

The remainder of this paper is structured as follows. Section 2 presents related work on generalizing rule-based models. Section 3 describes the process to generalize rule-based models and Sect. 4 provides a conclusion.

2 Related Work

It is reasonable to differentiate generalizing rule-based models from ensemble learning, which, at an initial glimpse, may appear similar. Ensemble learning [8] consists of using the predictions made by a number of base models to make a single prediction. In contrast, generalizing models consists of using a set of base models to create a single model, which is the only one making a prediction. The goals of each technique are also quite different. While in ensemble learning it is focused on improving accuracy, in generalizing models it is concerned in obtaining aggregated models without significantly affecting accuracy. Moreover, model interpretability is a goal per se for generalizing models but not for ensemble learning.

Approaches to generalizing models fall into two major categories: analytical and mathematical. Analytical approaches were first introduced by Williams [15] and consist of breaking down a set of models into rules and then assemble them in order to create a generalized model. On the other hand, mathematical approaches consist in applying a mathematical function to a group of models which results in the generalized model. The process described in this paper fits in the context of analytical approaches.

Analytical approaches emerged essentially from two contexts. The first was to create models for systems based on distributed environments, i.e., where the data sources were scattered across different locations. The problem was presented as “mining data that is distributed on distant machines, connected by low transparency connections" [2]. The second was a consequence of the growth in the amount of data collected by information systems. It became necessary to create models that could manage large datasets [7]. At the time there was a lack of available resources to handle the task, being described as "a very slow learning process sometimes overwhelming the system memory" [4] or "the emergence of datasets exceeding available memory" [1].

In problems with naturally distributed data, every location has its own local dataset with identical format and structure. These are moved over a channel to a centralized location where they are joined into a monolithic dataset, i.e., a non-distributed dataset stored in a single location. A generalized model is then created using all available data. Still, such scenario presents a major problem: moving data may be unsafe, expensive or simply impossible due to its large volume. An alternative of moving data is to move the models instead. Models are created in each location, then moved through a channel to a centralized location, where they are combined in a generalized model [2]. In problems with the need to create models from large datasets, it is essential to artificially create distributed data. This is achieved by breaking down a large dataset into as many individual datasets as necessary until it becomes possible to create a model for each [1]. Under such circumstances, all base models are combined into a generalized one.

Contrarily to analytical approaches, mathematical approaches are quite different from each other and were designed to solve specific problems. Kargupta and Park [6], motivated by the need to analyse and monitor time-critical data streams using mobile devices, proposed an approach to combine decision trees using the Fourier Transform. As the decision tree is a function, it can be represented in a frequency domain, resulting in the model *spectrum*. Models are combined by the adding their spectra. Gorbunov and Lyubetsky [3] combine models by constructing a *supertree*, the "nearest" on average to a given set of trees. The method is tested on the domain of analysis of the evolution trees of different species. In this context, the problem is to map a set of gene trees into a species tree (the average tree). Shannon and Banks [10] describe *Maximum Likelihood Estimate* (MLE) to combine a set of classification trees into a single tree by finding a *central tree*. The approach was applied to a set of classification trees obtained from biomedical data.

3 Generalization of Rule-Based Models

Generalization of rule-based models is presented as a sequential process with abstract parts and a few that can be specialized. Given a set of datasets $(D_1, \ldots, D_n)$, the corresponding base models $(M_1, \ldots, M_n)$ are trained and evaluated (using metric η_i). Then, all the base models are organized into groups $(G_1, \ldots, G_k)$ with a generalized model being created for each group $(\Omega_1, \ldots, \Omega_k)$. Finally the generalized models are evaluated using previously unused parts of the base models datasets (with metric σ_i). Figure 1 depicts a high-level view of the experimental set-up of the process, while Algorithm 1 presents it in more depth.

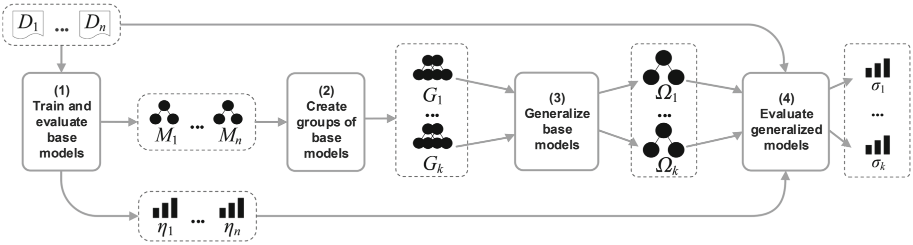
D_1 ... D_n
(1) Train and evaluate base models
M_1 ... M_n
(2) Create groups of base models
G_1
...
G_k
(3) Generalize base models
Ω_1
...
Ω_k
(4) Evaluate generalized models
σ_1
...
σ_k
η_1 ... η_n

Fig. 1. Experimental set-up

A preliminary task of the process is the creation of 10 folds for each dataset ($\{f_i^1, ..., f_i^{10}\} \in D_i$). Each generalized model has to be evaluated using unseen data, i.e., data not used in the creation of base models. As each base model is to provide rules to a generalized model, one fold of its associated dataset is put aside destined to incorporate a test dataset to evaluate that same generalized model. As a consequence, to ensure it remains new data, this fold (denoted as λ) is never included in the data for creating or evaluating base models. Instead of choosing a specific fold, the process of generalizing models and subsequent evaluation is performed 10 times, each using a different fold. Each fold maps in an iteration (λ) of the evaluation cycle. Base models (M_i^λ) are created and then evaluated using the data in all folds except the λ fold ($D_i \setminus f_i^\lambda$).

Algorithm 1. Process to Generalize Rule-based Models

```
Input: Datasets = $\{D_1, \ldots, D_n\}$
Output: Improvement scores = $\{\sigma_1, \ldots, \sigma_k\}$
for $\lambda$ such that $1 \leq \lambda \leq 10$ do
    for $i$ such that $1 \leq i \leq n$ do
        $\{M_i^\lambda, \eta_i^\lambda\} \leftarrow \text{TrainEvaluateBaseModel}(D_i \setminus f_i^\lambda)$
    end for
    $\{G_1^\lambda, \ldots, G_k^\lambda\} \leftarrow \text{CreateGroupsBaseModels}(\{M_1^\lambda, \ldots, M_n^\lambda\})$
    for $j$ such that $1 \leq j \leq k$ do
        $\Omega_j^\lambda \leftarrow \text{GeneralizeBaseModels}(G_j^\lambda)$
        $\sigma_j^\lambda \leftarrow \text{EvaluateGeneralizedModel}(\Omega_j^\lambda, \{f_1^\lambda, \ldots, f_p^\lambda\}, \{\eta_1^\lambda, \ldots, \eta_p^\lambda\})$
    end for
end for
for $j$ such that $1 \leq j \leq k$ do
    $\sigma_j \leftarrow \frac{1}{10} \sum_{\lambda=1}^{10} \sigma_j^\lambda$
end for
```

All base models are then organized into groups (G_j^λ), each to yield a generalized model (Ω_j^λ). Next, the evaluation test folds are assembled as test dataset for the generalized model ($\{f_1^\lambda, \ldots, f_p^\lambda\}$). The evaluation procedure takes the generalized model, the test dataset and the base models performances ($\{\eta_1^\lambda, \ldots, \eta_p^\lambda\}$

(p denoting the number of models in the group) resulting in an improvement score of the generalized model (σ_j^λ). This aim of this metric is to estimate whether there is gain (if positive) or loss (if negative) in predictive quality relative to the base models. Finally, as the evaluation cycle is replicated 10 times, the improvement scores of each generalized model are averaged across all iterations yielding the overall improvement score (denoted as σ_j).

3.1 Train and Evaluate Base Models

Models are rule-based classifiers, i.e., a set of IF-THEN rules. Due to one of the folds being reserved for evaluating the generalized model, base models are evaluated using 9-fold cross-validation set-up [11]. The evaluation score is conceptually denoted as η_i (which may be embodied, for example, with the F1-score [5]).

3.2 Create Groups of Base Models

In this procedure, the base models are gathered into groups. Models can be grouped reflecting a business driven criterion. For example, if a company is interested in knowing the performance of sales of its subsidiaries, it may want to group the models by geographic zone. Alternatively, there are applications where the creation of groups may be completely automated. An example is by criteria related to the complexity of the model, as the number of rules. In such cases, clustering techniques can be used to assist the creation of the groups. There may be applications where there is no need to create groups. Nevertheless, in order to maintain the process generic, it is considered that there is a single group with all the models.

3.3 Generalize Base Models

In this procedure, the base models in each group are generalized resulting in a new model, as described in Algorithm 2. Keeping up with the generality, it is required for the process to be independent of the language of the base models. In other words, it should be applied whether rules are extracted from decision trees or laid out in any other format. A possibility is for rules to be be represented as rows in a decision table with columns specifying the independent (x_i) and target ($\hat{y}$) variables. Therefore, before pursuing the combination of base models (M), these are converted to decision tables (denoted as T), and then generalized sequentially. Depending on the approach chosen to combine decision tables, there may be circumstances that generate an empty decision table. If so, the procedure skips that attempt and carries on selecting the next decision table to combine with the last one that succeeded (T_ω). After all the decision tables in the group are scanned, the final generalized decision table is converted back into the same language as the base models, yielding the generalized model (Ω). The next subsections detail these operations.

Algorithm 2. Generalize base models

```
**Input:** Group of base models $\{M_1, \ldots, M_p\}$
**Output:** Generalized model $\Omega$
$T_\omega \leftarrow \text{ExtractRules}(M_1)$
**for** $i$ such that $2 \leq i \leq p$ **do**
    $T_\theta \leftarrow \text{CombineRules}(T_\omega, \text{ExtractRules}(M_i))$
    **if** $T_\theta \neq \emptyset$ **then**
        $T_\omega \leftarrow T_\theta$
    **end if**
**end for**
$\Omega \leftarrow \text{BuildModel}(T_\omega)$
```

Extract Rules. This operation extracts the underlying rules of a model as rows in a decision table (T), using an approach in accordance with its language.

Combine Rules. This operation attempts to combine the rules if a pair of decision tables (T_1 and T_2) into one (T_θ), with the steps presented in Algorithm 3.

Algorithm 3. Combine rules

```
**Input:** Decision tables $T_1$ and $T_2$
**Output:** Combined decision table $T_\theta$
$T_\theta \leftarrow \text{CreateRules}(T_1, T_2)$
**if** $T_\theta \neq \emptyset$ **then**
    $T_\theta \leftarrow \text{JoinRules}(\text{ResolveConflicts}(T_\theta))$
**end if**
```

The operation *Create rules* implies a specific approach to derive the rules of the combined table. A common example is the intersection of the inner product of the rules of both tables [1,2,12]. The operation is replicated until all rules from both tables are combined. A possible consequence is that none of the rules of both tables overlap, resulting in the intersection to be an empty set. If this occurs the process stops. Although the operation is illustrated with intersection, it is important to highlight that it is generic, i.e., it can be carried out with any another function. A conflict exists if a pair of overlapping rules of T_1 and T_2 do not agree on the target variable value. The operation *Resolve conflicts* selects, for each conflict found, which value should be set to the target variable of the new rule. For example, an approach is to assign the target value of the rule covering the larger volume in the multidimensional decision space [1]. Another is to select the one created with more examples [12]. After this operation, the resulting decision table has no conflicts. The operation *Join rules* attempts to decrease the number of rules by identifying adjacent rules in the multidimensional decision space sharing the same class in the target variable. These can be joined together, thus reducing the number of rules.

Build Model. This operation converts a decision table back to the base model representation. For example, if the base models are decision trees, then the generalized model should also be a decision tree. This task presents unexpected challenges. An inevitable consequence of repeatedly changing and removing decision rules along the combination process is a final decision table frequently failing to cover the entire multidimensional space. An approach consists in artificially generating examples falling into each decision region of the final generalized decision table T_ω [12]. The examples of all rules are gathered in a dataset D^{T_ω} from which a model is trained (Ω).

3.4 Evaluate Generalized Models

In this procedure, a generalized model is evaluated following the steps in Algorithm 4. The predictive quality of the generalized models is measured by an *improvement score* (denoted as σ).

Algorithm 4. Evaluate generalized model

```
Input: Gener. model = $\Omega$, Test folds = $\{f_i, \ldots, f_p\}$, Perf. base models = $\{\eta_i, \ldots, \eta_p\}$
Output: Improvement score of generalized model = $\sigma$
for $i$ such that $1 \leq i \leq p$ do
    $\Delta_i \leftarrow \text{EvaluateModel}(\Omega, f_i) - \eta_i$
end for
$\sigma \leftarrow \frac{1}{p} \sum_{i=1}^{p} \Delta_i$
```

The fold that was put aside in each base model is used as test data to evaluate the generalized model. Evaluation consists in using the generalized model to make predictions on the test data and then comparing them with the true values of the target variable. The evaluation metric has to be the same as the one used to evaluate base models (e.g. if the F1-score was chosen to evaluate base models, then it should also be used to evaluate the generalized ones). As the aim is to estimate the variation in predictive quality of replacing the base models with a generalized one, the difference of performances (Δ_i) is assessed. If positive, then the generalized model performs better than the base model, otherwise, it performs worse. The cycle is replicated for all folds coming from each dataset of the base models associated with the generalized model. The overall performance of the generalized model results from the average of the differences of performances relative to all base models (σ) in the original group.

4 Conclusions

Generalizing rule-based models has emerged from approaches to solve different problems in particular contexts. Analytical approaches, which separate the rules of a set of models and then recombine them, although presented in a variety of forms, can be abstracted to a generic method. The main contribution of this paper is to describe a process that sequences the main procedures and then identifies the operations that can be deployed in different ways.

The representation of models as a set of decision rules facilitates the process of generalizing them. Then, the sub-problems of how to combine decision rules, resolve class conflicts of the target variable in overlapping rules, and build the generalized rule-based model remains open to different approaches, without loss of generality. Generalized models are evaluated by assessing their ability to replace the base models. Although the set-up to evaluate generalized models is part of the process, the evaluation metric itself is generic.

References

1. Andrzejak, A., Langner, F., Zabala, S.: Interpretable models from distributed data via merging of decision trees. In: Proceedings of the 2013 IEEE Symposium on Computational Intelligence and Data Mining. IEEE (2013)
2. Bursteinas, B., Long, J.: Merging distributed classifiers. In: Proceedings of the 5th World Multiconference on Systemics, Cybernetics and Informatics (2001)
3. Gorbunov, K., Lyubetsky, V.: The tree nearest on average to a given set of trees. Probl. Inf. Transm. **47**(3), 274–288 (2011)
4. Hall, L., Chawla, N., Bowyer, K.: Combining decision trees learned in parallel. In: Working Notes of the KDD-97 Workshop on Distributed Data Mining, pp. 10–15 (1998)
5. Han, J., Kamber, M., Pei, J.: Data Mining: Concepts and Techniques. Morgan Kaufmann, San Francisco (2011)
6. Kargupta, H., Park, B.: A fourier spectrum-based approach to represent decision trees for mining data streams in mobile environments. IEEE Trans. Knowl. Data Eng. **16**, 216–229 (2004)
7. Maimon, O., Rokach, L.: Data Mining and Knowledge Discovery Handbook, 2nd edn. Springer, Boston (2010). https://doi.org/10.1007/978-0-387-09823-4
8. Opitz, D., Maclin, R.: Popular ensemble methods: an empirical study. J. Artif. Intell. Res. **11**, 169–198 (1999)
9. Quinlan, J.: Induction of decision trees. Mach. Learn. **1**(1), 81–106 (1986)
10. Shannon, W.D., Banks, D.: Combining classification trees using MLE. Stat. Med. **18**(6), 727–740 (1999)
11. Stone, M.: Cross-validatory choice and assessment of statistical predictions. J. Roy. Stat. Soc.: Ser. B **36**(2), 111–147 (1974)
12. Strecht, P., Mendes-Moreira, J., Soares, C.: Merging decision trees: a case study in predicting student performance. In: Luo, X., Yu, J.X., Li, Z. (eds.) ADMA 2014. LNCS (LNAI), vol. 8933, pp. 535–548. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-14717-8_42
13. Weiss, S., Indurkhya, N.: Optimized rule induction. IEEE Expert **8**(6), 61 (1993)
14. Weiss, S.M., Indurkhya, N.: Rule-based machine learning methods for functional prediction. J. Artif. Intell. Res. **3**, 383–403 (1995)
15. Williams, G.: Inducing and combining multiple decision trees. Ph.D. thesis, Australian National University (1990)