Comprehensibility of Classification Trees–Survey Design

Rok Piltaver[1,2], Mitja Luštrek[2], Matjaz Gams[1,2], Sanda Martincic – Ipšic[3]
[1]Jozef Stefan Institute - Department of Intelligent Systems, Ljubljana, Slovenia
[2]Jozef Stefan International Postgraduate School, Ljubljana, Slovenia
[3]University of Rijeka - Department of Informatics, Rijeka, Croatia
rok.piltaver@ijs.si, mitja.lustrek@ijs.si, matjaz.gams@ijs.si, smarti@inf.uniri.hr

ABSTRACT: *Comprehensibility is the decisive factor for application of classifiers in practice. However, most algorithms that learn comprehensible classifiers use classification model size as a metric that guides the search in the space of all possible classifiers instead of comprehensibility - which is ill-defined. Several surveys have shown that such simple complexity metrics do not correspond well to the comprehensibility of classification trees. This paper therefore suggests a classification tree comprehensibility survey in order to derive an exhaustive comprehensibility metrics better reflecting the human sense of classifier comprehensibility and obtain new insights about comprehensibility of classification trees.*

Keywords: Classification Tree, Survey Design

Received: 23 October 2018, Revised 9 January 2018, Accepted 27 January 2019

DOI: 10.6025/isej/2019/6/1/15-20

1. Intorduction

Comprehensibility of data mining models, also termed interpretability [15] or understandability [1], is the ability to understand the output of induction algorithm [14]. Its importance has been stressed since the early days of machine learning research [17, 19]. Kodratoff even reports that it is the decisive factor when machine learning approaches are applied in industry [13]. Application domains in which comprehensibility is emphasized are for instance medicine, credit scoring, churn prediction, bioinformatics, and others [8].

A metric of comprehensibility is therefore needed in order to compare learning systems performance and as a (part of) heuristic function used by a learning algorithm [9, 21]. Majority of algorithms for learning comprehensible models use simple measures based on model size which may oversimplify the learned models. Humans by nature are mentally opposed to too simplistic representations of complex relations [7], therefore it is no surprise that empirical studies have shown comprehensibility to be negatively correlated with the complexity (size) of a classifier in at least some cases [1]. Such simple measures based on model complexity are therefore regarded as an over-simplistic notion of comprehensibility [8].

Those facts motivated us to propose a survey design, with the goal to derive an exhaustive comprehensibility metrics better reflecting the human sense of classifier comprehensibility. Obtained insights into evaluator's judgments about classifier comprehensibility will provide means for inducing definition of comprehensibility metrics that capture fine-grained differences in classifier comprehensibility and for evaluating the induced metrics. User survey based approach, which follows the observation that comprehensibility is in the eye of the beholder [16], is advocated; defining comprehensibility metric directly is not possible because it is comprehensibility is ill-defined [13].

2. Review of Related Work

According to [16] comprehensibility measures the "mental fit" [15] of the classification model, which has two main drivers: the type of classification model and its size or complexity. It is generally accepted that tree and rule based models are the most comprehensible while SVM, ANN and ensembles are in general black box models that can be hardly interpreted by users [8, 16, 20]; however there are domain and user specific exceptions from this rule of thumb. For a given classification model, the comprehensibility generally decreases with the size [2]. This principle is motivated by Occam's razor, which prefers simpler models over more complex ones [6]. Furthermore, a rule based model with few long clauses is harder to understand than one with shorter clauses, even if the models are of the same absolute size [20]. Comprehensibility also decreases with increasing number of variables and constants in a rule [20] and amount of inconsistency with existing domain knowledge [1, 18].

User-oriented assessment of classifier comprehensibility [1] compared outputs of several tree and rule learning algorithms and concluded that trees are more comprehensible then rules, and that in some cases tree size is negatively correlated with comprehensibility. Note that the trees included in the study were simple and were probably perceived as less comprehensible because they did not agree with the users' knowledge. Another study [12] (based on inexperienced users) compared comprehensibility of decision tables, trees and rules. The results showed that the respondents were able to answer the questions faster, more accurately and more confidently using decision tables than using rules or trees and were clearly able to assess the difficulty of the questions. Larger classifiers resulted in a decrease in answer accuracy, an increase in answer time, and a decrease in confidence in answers. Evidence that answering logical questions (e.g. validate a classifier) is considerably more difficult than classifying a new instance was found. However, proposition that cognitive fit of classifier with the given task type influences users' performance received limited support. A paper on comprehensibility of classification trees, rules, and tables, nearest neighbor and Bayesian network classifiers [8] stressed that graphical representation, hierarchical structure, including only subset of attributes in a tree, and independence of tree branches are advantages of classification trees. On the other hand, possible irrelevant attributes and replicated subtrees enforced by the tree structure decrease comprehensibility and may lead to overfitting. This can be mitigated by converting a tree into a rule set, which enables more flexible pruning resulting in a more comprehensible representation. Another recognized downside of classification trees is their Boolean logic-based nature as opposed to the probabilistic interpretation of naïve Bayes, which might be preferred in some applications [8].

This paper focuses on the comprehensibility of classification trees; however most of the suggested ideas could be analogically implemented on classification rules and tables as well. The survey design enables analysis of the influence of tree complexity and visualization on its comprehensibility. The complexity of classification tree is usually measured with the number of leaves or nodes in a tree or the number of nodes per branch [16, 20] while the suggested survey considers some additional complexity measures as well. The influence of visualization on comprehensibility has been stressed [16] but empirical studies are missing, therefore the suggested survey also considers visualization factors. The past empirical studies of classifier comprehensibility [1, 12] were performed only on homogenous groups of students, therefore we suggest adding data mining experts with different cultural background to the group of participants in future studies.

3. Survey Design

One possible way to estimate comprehensibility of a classifier is to present it to a survey respondent, who will analyze it, and then conduct an interview about comprehensibility. This approach is very time consuming and may be unintentionally biased by both involved persons, e.g. asking a question about comprehensibility of a model may help the respondent in comprehending the classifier. Therefore the indirect and more objective approach that was also used in previous studies [1, 12] is preferred. It measures the performance of respondents asked to solve tasks that involve interpretation and understanding of classifiers. The following subsections of the paper define the selected survey tasks, performance metrics, observed properties of classifiers, and strategies that prevent bias.

3.1 Survey Tasks (Question Types)

The comprehensibility survey consists of six tasks. The first task - **classify** asks respondent to classify an instance according to a given classifier (same as in [1, 12]). Tasks 2- 4 are based on [4], which reports that comprehensibility is required to explain individual instance classifications, validate the classifier, and discover new knowledge. Thus the second task - **explain** ask the respondent to answer which attributes values must be changed or retained in order to classify a given instance into another class. For example, which habits (values of attributes) would a patient with high probability of getting cancer (class) have to change in order to stay healthy? The third task - **validate** requires the respondent to check whether a statement about the domain is confirmed or rejected according to the presented classifier. For example: does the tree say that persons smoking more than 15 cigarettes per day are likely to get cancer. Similar questions were also asked in [12]. The fourth task - **discover** asks the respondent to find a property (attribute-value pair) that is unusual for instances from one class; this corresponds to finding a property of outliers. For example, people that lead healthy life are not likely to get cancer, except if they have already suffered from it in the past.

The fifth task - **rate** requests the user to give the subjective opinion about the classification trees on a scale with five levels: very easy to comprehend, easy to comprehend, comprehensible, difficult to comprehend, and very difficult to comprehend. Each label of the scale is accompanied with an explanation that relates to the time needed to comprehend the tree and difficulty of remembering it and explaining it to another person. The purpose of explanations is to prevent variation in subjective interpretations of the scale. The task intentionally follows the first four tasks in which the respondents use the classifiers and obtain hands on experience, which enables them to rank the comprehensibility. The classifiers are learned on a single dataset and visualized using Orange tool [5] in order to be consistent across all the tasks and enable reliable and prompt responding. For the same reason meaningful attribute and class names are used. The first five tasks measure the influence of classifier complexity (i.e. the number of leaves, depth, branching) while the final task measures the influence of different representations of the same tree on the comprehensibility.

Task six - **compare** asks the respondents to rate which of the two classification trees shown side by side is more comprehensible on the scale with three levels: the tree is much more comprehensible, the tree is more comprehensible, and the trees are equally comprehensible. One of the trees in this task is already used in the previous five tasks - serving as a known frame of reference - while the other one is a previously unseen tree with the same content but represented in different style. The position of a tree (left or right) is randomized in order to prevent bias, e.g. assuming that the left tree is always more comprehensible.

3.2 Performance Metrics

The tasks rate and compare are directed toward obtaining subjective opinions rated on the given scales. The tasks classify, explain, validate, and discover are directed toward objectively quantifying respondents' performance (e.g. time and correctness of answers). Corresponding performance metrics are derived from the six metrics proposed in the experiments on conceptual model understandability [11]. The first three are explicitly measured by the survey: the time needed to understand a model translates to time to answer a question (longer time - less comprehensible classifier); correctly answering questions about the content translates to the probability of correct answer (higher probability - more comprehensible classifier); the perceived ease of understanding is expressed with subjective judgment of a questions difficulty (rated on scale very easy, easy, medium, difficult and very difficult). The other measures are implicitly embedded in the survey design: difficulty of recalling a model is captured through descriptions of the five levels of comprehensibility scale in the rate task; problemsolving based on the model content is embedded in tasks 1- 4; and verification of model content is in the validate task.

3.3 Observed Classifier Properties

Motivated by the related work [1, 8, 12, 20] and authors' experience the following **tree complexity properties** are proposed: number of leaves or nodes, branching factor, number of nodes in a branch, and number of instances belonging to a leaf. Proposed tree complexity properties are systematically varied in the first five tasks of the survey. Also, the proposed **tree visualization properties** are varied in the compare task: using color to enhance readability (e.g. pie-charts corresponding to class distributions in nodes), layout of the tree based on the depth of subtrees, and general layout and readability of the visualized tree (e.g. plain text output vs. default Weka [10] and Orange [5] visualization). Additionally, the survey enables **contrasting**: meaningful names of attributes, attribute values and classes to meaningless ones; attributes with high information gain to the ones with low gain; and meaningful aggregated attributes contrasted to conjunctions of isolated attributevalue pairs (i.e. deep structure [20]). Finally, the survey design also enables various statistical analysis for the each single leaf (branch of the tree) or for the entire tree.

3.4 Avoiding Implicit Survey Bias

In order to prevent bias the following issues must be considered: choice of the classification domain, classifiers, and respondents group, and the ordering of questions. The **classification domain** has to be familiar to respondents - all of them are aware of relations among attribute values and classes and none of them have significant advantage of more in-depth knowledge about the domain. At the same time, the domain must be broad and rich enough to enable learning a range of classifiers with various properties listed in 3.3. Furthermore, choosing an interesting domain motivates the respondents to participate in the survey. The Zoo domain from the UCI Machine Learning Repository [3] meets all the requirements and is highly appropriate for general and heterogeneous population. It requires only elementary knowledge about animals expressed with 17 (mostly binary) attributes: are they aquatic or airborne, do they breathe, how many legs they have, do they have teeth, fins or feathers, etc. The Zoo domain induces 7 classes: mammals, fish, birds, amphibian, reptile, mollusk, and insect.

The selected **classifiers** must vary in complexity but not in other parameters that may influence comprehensibility and hence bias the results. In addition, classifiers are learned using well-known machine learning algorithm rather than manually constructed. Using different pruning parameters produces trees with different sizes. Higher branching factor can be achieved by replacing original binary attributes with constructed attributes, which can be interpreted as building deep models [20]. If possible, order of the leaves or at least their grouping in subtrees should remain the same as in the binary tree. Choosing a question for a given tree determines the number of nodes in a branch that the user will have to analyze in order to answer. In each group of questions a single parameter changes while the others remain constant. Finally, a well-known and comprehensible classifier visualization style must be used, e.g. Orange [5].

Order of the question may also induce bias. For example, the learning effect can occur: the respondents need more time to answer the first few questions, after that they answer quicker. Next, the performance of respondents drops if they get tired or loose motivation, therefore the number of questions must be limited. To prevent those effects, Latin square ordering is used, where each question occurs exactly once at each place in the ordering and subsequently each respondent gets a different ordering of the questions. Finally, starting each task with a test question (from the different domain) reduces the learning effect as well.

The survey design **assumes the following order of tasks**: starts with the simpler and progresses to more difficult ones. The compare and rate tasks, related to subjective opinions, are placed toward the end – after the respondents acquire experience with the classifiers.

Demographic data (DM knowledge, age, sex, language) reflects the heterogeneity of the respondents group and enables detailed analysis of classifier comprehensibility per different subgroups like students or experts. Hence, the **test group** consists of data mining experts on one hand and nonexperts with basic knowledge about classification on the other. Comparing the results of the two groups as well as considering the cultural background (e.g. different mother tongues), can provide new insights into classifier comprehensibility. Finally, obtaining statistically significant results requires high enough number of respondents.

4. Survey Implementation

This work proposes **online survey** in order to facilitate accurate measurements of time, automatic checking the correctness of answers, saving the answers in a database and allowing remote participation. Several tools for designing and performing online surveys exist but do not meet all of the design requirements (see section 3): Latin square design, measuring the time of answering each question, automatic translation to several languages, using templates to quickly define questions for a given task and automatically checking the correctness of answers. Therefore, custom online survey is implemented using MySQL database, PHP and JavaScript programming languages, and CSS for webpage formatting.

The **database** includes one table for demographic data with auto-increment user id as the primary key and one table per task with user id and question id as the primary key. Each task table includes a field representing question order number, a date-time field, and field(s) representing the respondents' answer. Tables for tasks 1-4 additionally include fields with the measured answering time, list of all respondent clicks and associated times, and the indicator of correct answer. PHP is used to dynamically **generate survey webpages** with correct ordering of questions for each respondent and storing the answers into the database. Question webpages are generated by a separate PHP script for each task based on a template and a simple data structure defining the questions. An additional PHP script is used as a library of shared functions and data structures: one represents instances used in the survey and the other terms (instructions, attribute names and value, classes, etc.) translated into English, Slovenian and Croatian languages. Additionally, PHP scripts are used for backing-up and checking correctness of answers, login and help pages, and a respondent home-page providing feedback on personal progress and performance compared to the group. **SVG images** representing the classification trees exported from Orange [5] were automatically translated into the three languages using a Java program – the translation table is the same as in the PHP library script.

JavaScript is used to **measure the time of answering** each question. When a webpage is opened, only the instructions and footer of the page are visible. Clicking on the button "*Start solving"* calls a JavaScript function that displays the question (e.g. table with attribute-value pairs and image of a tree) and the answer form (drop-down lists, radio buttons) and starts the timer. Changing a value of the answer form field records the relative time and action type. When the respondent clicks the "*Finish button"*, the answer fields are disabled, time is calculated, and question difficulty rating options are displayed. When the "*Next button"* is clicked, the collected values are assigned to hidden form fields in order to pass them to the PHP script that stores the data in the database and displays the next question.

A psychologist and two DM experts analyzed the initial survey and improved version was implemented based on their comments. It passed a **validation** test with 15 students answering the first task at the same time. Preliminary analysis of the results for 10 respondents is in line with the expectations, thus the survey is ready to be used in order to collect data about tree comprehensibility.

References

[1] Allahyari, H., Lavesson, N. (2011). User-oriented Assessment of Classification Model Understandability, 11*th Scandinavian Conference on AI*, 11-19.

[2] Askira-Gelman, I. (1998). Knowledge discovery: comprehensibility of the results. *In*: *Proceedings of the* 31*st Annual Hawaii International Conference on System Sciences,* 5, 247.

[3] Bache, K., Lichman, M. (2014). UCI Machine Learning Repository, http://archive.ics.uci.edu/ml. University of California, School of Information and Computing Sciences.

[4] Craven, M. W., Shavlik, J. W. (1995). Extracting Comprehensible Concept Representations from Trained Neural Networks. *Working Notes on the IJCAI'95 WS on Comprehensibility in ML*, 61-75.

[5] Demšar, J., Curk, T., Erjavec, A. (2013). Orange: Data Mining Toolbox in Python. *Journal of Machine Learning Research,* 14 (August), 2349 - 2353.

[6] Domingos, P. (1999). The role of occam's razor in knowledge discovery, *Data Mining and Knowledge Discovery,* 3, 409–425.

[7] Elomaa, T. (1994). In Defense of C4.5: Notes on learning onelevel decision trees. *In: Proceedings of 11th International Conference on ML*, 62-69.

[8] Freitas, A. A. (2013). Comprehensible classification models - a position paper. *ACM SIGKDD Explorations,* 15 (1) 1-10.

[9] Giraud-Carrier, C. (1998). Beyond predictive accuracy: what? *Proceedings of the ECML-98 Workshop on Upgrading Learning to Meta-Level: Model Selection and Data Transformation*, 78-85.

[10] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I. H. (2009). The WEKA Data Mining Software: An Update. *SIGKDD Explorations,* 11 (1).

[11] Houy, C., Fettke, P., Loos, P. (2012). Understanding Understandability of Conceptual Models – What Are We Actually Talking about? *Conceptual Modeling - Lecture Notes in Computer Science Volume* 7532, 64-77.

[12] Huysmans, J., Dejaeger, K., Mues, C., Vanthienen, J., Baesens, B. (2011). An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. *Decision Support Systems,* 51 (1) 141-154.

[13] Kodratoff, Y. (1994). The comprehensibility manifesto, KDD Nuggets (94:9).

[14] Kohavi, R. (1996). Scaling Up the Accuracy of Naive-Bayes Classifiers: a Decision-Tree Hybrid. *In:* Proceedings of the 2nd International Conference on KD and DM, 202-207.

[15] Maimon, O. O., Rokach, L. (2005). Decomposition Methodology for Knowledge Discovery and Data Mining: Theory and Applications, World Scientific Publishing Company.

[16] Martens, D., Vanthienen, J., Verbeke, W., Baesens, B. (2011). Performance of classification models from a user perspective. *Decision Support Systems,* 51 (4) 782- 793.

[17] Michalski, R. (1983). A theory and methodology of inductive learning, *Artificial Intelligence* 20, 111–161.

[18] Pazzani, M. (1991). Influence of prior knowledge on concept acquisition: experimental and computational results. *Journal of Experimental Psychology. Learning, Memory, and Cognition* 17, 416–432.

[19] Quinlan, J. R. (1999). Some elements of machine learning. *Proc. 16th Int. Conf. on Machine Learning (ICML-99)*, 523-525.

[20] Sommer, E. (1995). An approach to quantifying the quality of induced theories. *Proceedings of the IJCAI Workshop on Machine Learning and Comprehensibility*.

[21] Zhou, Z.-H. (2005). Comprehensibility of data mining algorithms. *Encyclopedia of Data Warehousing and Mining*, 190-195, Hershey.