ECAI 2008
M. Ghallab et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-891-5-234

Fighting Knowledge Acquisition Bottleneck with Argument Based Machine Learning

Martin Možina and **Matej Guid** and **Jana Krivec** and **Aleksander Sadikov** and **Ivan Bratko** [1]

Abstract. Knowledge elicitation is known to be a difficult task and thus a major bottleneck in building a knowledge base. Machine learning has long ago been proposed as a way to alleviate this problem. Machine learning usually helps the domain expert to uncover some of the more tacit concepts. However, the learned concepts are often hard to understand and hard to extend. A common view is that a combination of a domain expert and machine learning would yield the best results. Recently, argument based machine learning (ABML) has been introduced as a combination of argumentation and machine learning. Through argumentation, ABML enables the expert to articulate his knowledge easily and in a very natural way. ABML was shown to significantly improve the comprehensibility and accuracy of the learned concepts. This makes ABML a most natural tool for constructing a knowledge base. The present paper shows how this is accomplished through a case study of building a knowledge base of an expert system used in a chess tutoring application.

1 INTRODUCTION

Knowledge is a key component of every intelligent computer system. Knowledge acquisition is therefore one of the perennial tasks of artificial intelligence. Unfortunately, this task proves to be a very difficult one, especially so if the goal is to acquire the knowledge in a comprehensible form. In building expert systems, it is exactly this task that presents a major bottleneck [5]. The problem was addressed in various ways [1, 2, 4], proposing assorted cognitive techniques like interviews, observations, analogy, etc. to elicit as much knowledge from experts as possible. Nevertheless, the problem still remains largely unsolved [6].

Machine learning has long ago been proposed as an alternative way of addressing this problem [7]. While it was shown that it can be successful in building knowledge bases [8], the major problem with this approach is that automatically induced models rarely conform to the way an expert wants the knowledge organised and expressed. Models that are incomprehensible have less chance to be trusted by experts and users alike. In striving for better accuracy, modern trends in machine learning (e.g. support vector machines) do not seem to be doing anything to alleviate this problem.

A common view is that a combination of a domain expert and machine learning would yield the best results [14]. Most of the applications in the literature combine machine learning and the experts' knowledge in one of the following ways: (a) experts validate induced models after machine learning was applied, (b) experts provide constraints on induced models in the form of background knowledge, and (c) the system enables iterative improvements of the model, where experts and machine learning algorithm improve the model in turns. The last approach seems to be the best, however, it requires the most effort on the part of the expert. This calls for a method that allows the expert to express his or her knowledge in a most convenient way and at the same time allows for seamless interaction between the expert and the machine.

Argumentation based machine learning (ABML) [10] is a recent method that seems to have potential to accomplish just that. It is a natural fusion of argumentation and machine learning. The advantages over traditional machine learning methods are better accuracy and comprehensibility. Improvement in comprehensibility is especially important in light of knowledge extraction. Through argumentation, ABML enables the expert to articulate his or her knowledge easily and in a very natural way. Moreover, it prompts the expert to share exactly that knowledge that is most useful for the machine to learn, thus significantly saving the time of the expert.

The present paper describes a case study in building a knowledge base of an expert system used in a chess tutoring application [12] to demonstrate the power of ABML. First, we describe basics of ABML [10] and a procedure that interacts with ABML and the expert. Then, in Section 3 the case study is presented and in Section 4 its results are assessed and discussed. We finish the paper with conclusions.

2 ARGUMENT BASED MACHINE LEARNING

Argument Based Machine Learning (ABML) [10] is machine learning extended with some concepts from argumentation. Argumentation is a branch of artificial intelligence that analyzes reasoning where arguments for and against a certain claim are produced and evaluated [11]. A typical example of such reasoning is a law dispute at court, where plaintiff and defendant give arguments for their opposing claims, and at the end of the process the party with better arguments wins the case.

Arguments are used in ABML to enhance learning examples. Each argument is attached to a single learning example only, while one example can have several arguments. There are two types of arguments; positive arguments are used to explain (or argue) why a certain learning example is in the class as given, and negative arguments are used to explain why it should not be in the class as given. We used only positive arguments in this work, as negatives were not required. Examples with attached arguments are called *argumented examples*.

Arguments are usually provided by domain experts, who find it natural to articulate their knowledge in this manner. While it is generally accepted that giving domain knowledge usually poses a problem, in ABML they need to focus on one specific case only at a time and provide knowledge that seems relevant for this case and does not

[1] Faculty of Computer and Information Science, University of Ljubljana, Slovenia. Contact email: martin.mozina@fri.uni-lj.si

have to be valid for the whole domain. The idea can be easily illustrated with the task of commenting chess games. It would be hard to talk about chess moves in general to decide precisely when they are good or bad. However, if an expert is asked to comment on a particular move in a given position, he or she will be able to offer an explanation and provide relevant elements of this position. Naturally, in a new position the same argument could be incorrect.

An ABML method is required to induce a theory that uses given arguments to explain the examples. Thus, arguments constrain the combinatorial search among possible hypotheses, and also direct the search towards hypotheses that are more comprehensible in the light of expert's background knowledge. If an ABML method is used on normal examples only (without arguments), then it should act the same as a normal machine learning method. We will use method ABCN2 [10], an argument based extension of the well known method CN2 [3], that learns a set of unordered probabilistic rules from argumented examples. In ABCN2, the theory (a set of rules) is said to explain the examples using given arguments, when there exists at least one rule for each argumented example that contains at least one positive argument in the condition part. This definition is a bit simplified, since it omits the use of negative arguments, as they are not relevant for this paper[2].

In addition to rules, we need an inference mechanism to enable reasoning about new cases. In rule induction community this problem is known as rule classification and several approaches can be found in the literature[9]. In this study, we will use a simple algorithm; among all relevant rules the best rules (with the highest predicted class probability) for each class are selected and the probability of the example's class is obtained by normalising predicted probabilities of selected rules.

2.1 Interactions between expert and ABML

In ABML, experts are asked to provide their prior knowledge in the form of arguments for the learning examples rather than the general domain knowledge. However, asking experts to give arguments to the whole learning set is not likely to be feasible, because it would require too much time and effort. The following loop describes the skeleton of the procedure that picks out critical examples - examples that ABML can not explain without some help:

1. Learn a hypothesis with ABML using given data.
2. Find the most critical example and present it to the expert. If a critical example can not be found, stop the procedure.
3. Expert explains the example; the explanation is encoded in arguments and attached to the learning example.
4. Return to step 1.

To finalise the procedure we need to contemplate the following two questions:

- How do we select "critical" examples ?
- How can we achieve to get all necessary information for the chosen example?

2.1.1 Identifying critical examples

The main property of critical examples is that the current hypothesis can not explain them well, or, in other words, it fails to predict their class. Since ABCN2 gives probabilistic class prediction, we define the most critical example as the example with the highest probabilistic error. The probabilistic error can be measured in several ways. We use a k-fold cross-validation repeated n times (e.g. $n = 4, k = 10$), so that each example is tested n times. The most critical example is thus the one with highest average probabilistic error.

2.1.2 Are expert's arguments good or should they be improved?

Here we describe in details the third (3) step of the above algorithm, where the expert is asked to explain the critical example. Using expert's arguments, ABML will sometimes be able to explain the critical example, while sometimes this will still not be entirely possible. In such cases, we need additional information from expert. The whole procedure for one-step knowledge acquisition is described with the next 5 steps:

Step 1: Explaining critical example. In this step, the expert is asked the following question: "Why is this example in the class as given?" The answer can be either "I don't know" (the expert is unable to explain the example) or a set of arguments $A_1, \ldots, A_k$ all confirming the example's class value can be given. If the system gets the answer "don't know", it will stop this procedure and try to find another critical example.

Step 2: Adding arguments to example. Arguments A_i are given in natural language and need to be translated into domain description language (attributes). Each argument supports its claim with a number of reasons. When a reason is simply an attribute value of the example, then the argument is simply added to the example. On the other hand, if reasons mention other concepts, not currently present in the domain, these concepts need to be included in the domain as new attributes before the argument can be added to the example.

Step 3: Discovering counter examples. Counter examples are used to spot if arguments suffice to successfully explain the critical example or not. If ABML fails to explain the example, then the counter examples will show where the problem is. Here, ABML is first used to induce a hypothesis H_1 using previous learning data only and H_2 using learning data together with new arguments. A counter example is defined as: it has a different class value from the critical example, its probabilistic error increases in H_2 with respect to H_1, and H_2 mentions arguments (given to the critical example) while explaining the counter example.

Step 4: Improving arguments. The expert needs to revise the initial arguments with respect to the counter example. This step is similar to steps 1 and 2 with one essential difference; the expert is now asked "Why is critical example in one class and why counter example in the other?" The answer is added to the initial argument.

Step 5: Return to step 3 if counter example found.

3 CASE STUDY: BAD BISHOP

As a case study, we considered the elicitation of the well-known chess concept of bad bishop. There is a general agreement in the chess literature and among chess players about the intuition behind this concept. However, the formalisation of this concept is difficult even for chess experts, which served as the motivation for choosing this concept for the ABML-based knowledge-elicitation process.

Watson [13] gives the following definition as traditional: a bishop that is on the same colour of squares as its own pawns is bad, since

[2] Due to space limitations, we will only roughly describe some of the mechanisms of ABML (see [10] or/and its website *www.ailab.si/martin/abml* for precise details).

its mobility is restricted by its own pawns and it does not defend the squares in front of these pawns. Moreover, he puts forward that centralisation of these pawns is the main factor in deciding whether the bishop is bad or not.

In the experiments, the dataset for learning consisted of 200 middlegame positions from real chess games where the black player has only one bishop[3]. These bishops were then a subject of evaluation by the experts[4]. In 78 cases, the bishops were assessed as bad. Each position had also been statically evaluated (i.e. without applying any search) by the evaluation function of the well-known open source chess program CRAFTY, and its positional feature values[5] served as attribute values for learning. We randomly selected 100 positions for learning and 100 for testing (stratification was used, preserving the proportion of positive and negative examples).

In the first iteration of the previously mentioned process, only CRAFTY's positional features were used and no arguments have been given yet. ABCN2 induced all together 4 rules achieving 72% classification accuracy on the test set. Figure 1 shows the first critical example, automatically selected by our algorithm.

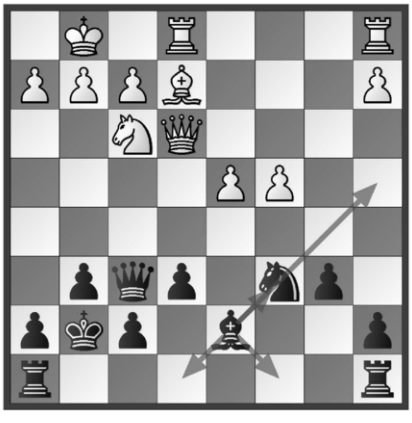

Figure 1. *Why is the black bishop not bad?* The experts used their domain knowledge to produce the following answer: "The black bishop is not bad, since its mobility is not seriously restricted by the pawns of both players."

The initial rules failed to classify this example as "not bad", as was previously judged by the experts. The following question was given to the experts: "Why is the black bishop not bad?" It turned out that the concept mentioned by the experts (see the caption in Figure 1) was not yet present in the domain attributes - the only CRAFTY's positional feature that could potentially describe bishop's mobility, BLACK_BISHOPS_MOBILITY, expresses the number of squares that the bishop attacks, but doing so takes into account all pieces (not only pawns) that block the bishop's diagonals, restricting its mobility. A new attribute, IMPROVED_BISHOP_MOBILITY, was therefore programmed and included into the domain. It is the number of squares accessible to the bishop, taking into account only own and opponents pawn structure. Based on the experts' explanation, the argument "IMPROVED_BISHOP_MOBILITY is high" was added to this example.

Taking only the bishop's mobility into account turned out not to be enough for ABCN2 to determine the goodness of the bishop. Also, the method, which at the time only had CRAFTY's attributes and the newly included attribute at its disposal, failed to find additional restrictions to improve the experts' argument. The ABML method then presented the experts with a counter example shown in Figure 2. This example is classified as "bad", although the value of the attribute IMPROVED_BISHOP_MOBILITY is high.

Figure 2. *Why is the black bishop bad, comparing to the one in Figure 1?* The experts' explanation was: "The important difference between the two examples is the following: in the example in Figure 2 there are more pawns on the same colour of squares as the black bishop, and some of these pawns occupy the central squares, which further restricts the bishop's possibilities for taking an active part in the game."

The experts were now asked to *compare* the black bishops in the two examples: "Why is the black bishop in Figure 2 bad, and the bishop in Figure 1 is not?" Again, the experts have been asked to give a description based on their knowledge in the presented domain. Based on this description (given in Figure 2), another attribute, BAD_PAWNS, was included into the domain. This attribute evaluates pawns that are on the colour of the square of the bishop ("bad" pawns in this sense). With some help of the experts, a look-up table with predefined values for the pawns that are on the same colour of squares as the bishop was designed in order to assign weights to such pawns. According to the previously mentioned Watson's definition, centralisation of the pawns was taken into account. The argument given to the example shown in Figure 1 was then extended to "IMPROVED_BISHOP_MOBILITY is high AND BAD_PAWNS is low," and with this argument the method could not find any counter examples any more. The new rule covering the critical example is:

if IMPROVED_BISHOP_MOBILITY$\geq$4 and BAD_PAWNS$\leq$32
then BISHOP=NOT_BAD; *class distribution [0,39]*

The above rule evidently uses given argument in its condition. The method operationalised the first condition of the argument as IMPROVED_BISHOP_MOBILITY$\geq$4 ($\geq$4 stands for high here), while in the second it decided that the value of 32 is critical for attribute BAD_PAWNS to distinguish a bad and a not bad bishop. The rule covers 39 learning examples (out of 100) and all of them are from class NOT_BAD, which suggests that the rule is good indeed.

The arguments can consist of both newly included attributes and/or existing ones. During the process, after they were given another critical example selected by the method, the experts expressed the following commentary: "The bishop is not bad, since the pawns that are on the same square colour are not sufficiently blocked by opponent's pawns and pieces." Their domain knowledge

[3] The learning data set and a detailed explanation of domain's attributes can be found at: *http://www.ailab.si/matej/*.

[4] The chess expertise was provided by woman grandmaster Jana Krivec and FIDE master Matej Guid.

[5] CRAFTY's evaluation function uses about 100 positional features.

was again translated into domain description language - attribute BLOCKED_BAD_PAWNS was added to the domain. As in the previous example, the method selected the position shown in Figure 2 as the most appropriate counter example. The "bad" black pawns in this position are also not blocked by opponent's pawns and pieces, but the bishop is regarded as bad anyway. The experts' explanation of the crucial difference between the two examples was the same as above in this case. The existing attribute BAD_PAWNS was therefore used to improve the argument to "BLOCKED_BAD_PAWNS is low AND BAD_PAWNS is low". The method was in this case able to induce the rest of the rule:

if BLOCKED_BAD_PAWNS$\leq$3
and BAD_PAWNS$\leq$26
and IMPROVED_BISHOP_MOBILITY$>$1
then BISHOP=NOT_BAD; *class distribution [0,19]*

Figure 3. *Why is the black bishop bad?* The following commentary was given: "The black bishop is bad, since both of its diagonals are blocked by its own pawns."

The ABML-based knowledge-elicitation process was used to induce rules to determine both good (i.e. not bad) and bad bishops. The automatically selected critical example shown in Figure 3 represents an example with other class value than the previous ones. The experts were in this case asked to describe why the black bishop is *bad*. Based on their answer (see Figure 3), another attribute was introduced into the domain: BLACK_PAWN_BLOCKS_BISHOP_DIAGONAL, which takes into account own pawns that block the bishops diagonals. The argument "BLACK_PAWN_BLOCKS_BISHOP_DIAGONAL is high" was added to the example, however a counter example presented in Figure 4 was found by the method and was shown to the experts. The question was: "Why is the bishop in Figure 4 not bad, and the bishop in Figure 3 bad?"

In this case, the experts were unable to express the crucial differences between the selected examples regarding the goodness of the bishop in a way that would enable to translate her description into domain description language. The description (see Figure 4), although completely relevant in the given position, is practically impossible to convert into appropriate attributes, since it would require several very sophisticated attributes to describe the dynamic factors expressed in the experts' commentary. In such a case (i.e. when the expert is unable to provide an argument that could be translated into domain description language), the ABML method searches for another counter

Figure 4. *Why is the bishop not bad, comparing to the bishop in Figure 3?* The experts: "The black bishop is not bad, since together with the black queen it represents potentially dangerous attacking force that might create serious threats against the opponent's king."

example (if available). In this case, the example in Figure 5 was given to the experts as a counter example to the one in Figure 3.

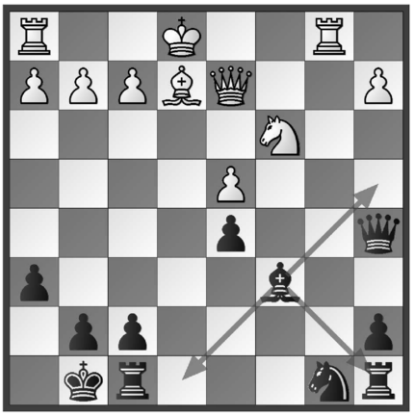

Figure 5. *Why is the bishop not bad, comparing to the bishop in Figure 3?* The experts described the difference: "The black bishop is not bad, since its mobility is not seriously restricted, taking the pawn structure into account."

Based on the experts' commentary (see Figure 5), the existing attribute IMPROVED_BISHOP_MOBILITY was used to improve the argument to "BLACK_PAWN_BLOCKS_BISHOP_DIAGONAL is high AND IMPROVED_BISHOP_MOBILITY is low". The following rule, explaining this critical example, can be found in the new set of induced rules:

if BLACK_PAWN_BLOCKS_BISHOP_DIAGONAL$\geq$20
and IMPROVED_BISHOP_MOBILITY$\leq$3
then BISHOP=BAD; *class distribution [18,0]*

In total, there were eight critical examples presented to the experts, however, due to space restrictions we were able to describe only three of these examples. The final model scored 95% accuracy on the test set.

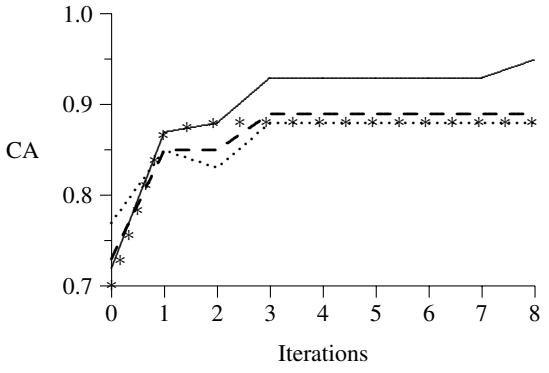

Figure 6. Progress of classification accuracies (CA) through iterations for ABCN2 (solid line), logistic regression (stars ∗), C4.5 (dashed line) and classic CN2 (dots).

4 ASSESSMENT AND DISCUSSION

The ABML-based knowledge-elicitation process presented in our case study consisted of eight (8) iterations. During the process, seven (7) arguments were attached to automatically selected critical examples, and five (5) new attributes were included into the domain. After each iteration, the obtained rules were evaluated on the test dataset. The improvement of the model is evident: from the initial 72% classification accuracy (Brier score 0.39, AUC 0.80), the final 95% accuracy (Brier score 0.11, AUC 0.97) was achieved after the end of the process.

The question is, whether these improvements were mainly due to the addition of new attributes or were the arguments also just important? The Figure 6 shows that the arguments also mattered significantly. We compared the progressions of classification accuracies of ABCN2 with some other ("non-ABML" - using only newly added attributes) machine learning algorithms, namely logistic regression, decision trees (C4.5), and the classic CN2. The accuracies of all methods improved during the process, however ABCN2 (which also used the arguments given by the experts) outperformed all the others. The obtained results suggest that the performance of other algorithms could also be improved by adding appropriate new attributes. However, using arguments is likely to lead to even more accurate models.

The main advantage of ABML over classical machine learning is the ability to take advantage of expert's prior knowledge in the induction procedure. This leads to hypotheses comprehensible to experts, as it explains learning examples using the same arguments as the expert did. In our case study this was confirmed by chess experts. According to them, the final set of rules are more alike to their understanding of the bad bishop concept than the initial rules were. Furthermore, the final rules were also recognised to be in accordance with the traditional definition of a bad bishop.

Our domain experts clearly preferred the ABML approach to manual knowledge acquisition. The formalisation of the concept of bad bishop turned out to be beyond the practical ability of our chess experts (a master and a woman grandmaster). They described the process as time consuming and hard, mainly because it is difficult to consider all relevant elements. ABML facilitates knowledge acquisition by fighting these problems directly. Experts do not need to consider all possibly relevant elements, but only elements relevant for a specific case, which is much easier. Moreover, by selecting only critical examples, the time of experts involvement is decreased, making the whole process much less time consuming.

5 CONCLUSION

In this paper, we introduced a new approach to knowledge elicitation based on the ABML type of machine learning. We studied the effectiveness of this approach in a case study that involves the concept of bad bishop in chess. This concept requires subtle expert judgement that is very hard to formalise. Given our experimental findings with ABML-based knowledge acquisition in this domain, we believe that this approach will be most helpful in general in areas that require subtle expert judgement, such as medical decision making or aesthetic evaluation of a painting or a piece of music.

In addition to the case study that illustrates the effectiveness of ABML-based knowledge acquisition, the paper makes the following new contributions:

1. The idea of counter examples and a mechanism for their detection.
2. The interactive procedure between the expert and ABML during knowledge acquisition.

ACKNOWLEDGEMENTS

This work was partly funded by the X-Media project (www.x-media-project.org) sponsored by the European Commission as part of the Information Society Technologies (IST) programme under EC grant number IST-FP6-026978, and Slovene Agency for Research and Development (ARRS).

REFERENCES

[1] J. H. Boose, 'A survey of knowledge acquisition techniques and tools', *Knowledge Acquisition*, **1**(1), 3–37, (1989).
[2] Timothy Chklovski, *Using Analogy to Acquire Commonsense Knowledge from Human Contributors*, Ph.D. dissertation, MIT Artificial Intelligence Laboratory, 2003.
[3] Peter Clark and Robin Boswell, 'Rule induction with CN2: Some recent improvements', in *Machine Learning - Proceeding of the Fifth Europen Conference (EWSL-91)*, pp. 151–163, Berlin, (1991).
[4] Nancy J. Cooke, 'Varieties of knowledge elicitation techniques', *Int. J. Hum.-Comput. Stud.*, **41**(6), 801–849, (1994).
[5] Edward A. Feigenbaum, 'Knowledge engineering: the applied side of artificial intelligence', in *Proc. of a symposium on Computer culture: the scientific, intellectual, and social impact of the computer*, pp. 91–107, New York, NY, USA, (1984). New York Academy of Sciences.
[6] Edward A. Feigenbaum, 'Some challenges and grand challenges for computational intelligence', *Source Journal of the ACM*, **50**(1), 32–40, (2003).
[7] Richard Forsyth and Roy Rada, *Machine learning: applications in expert systems and information retrieval*, Halsted Press, New York, NY, USA, 1986.
[8] Pat Langley and Herbert A. Simon, 'Applications of machine learning and rule induction', *Commun. ACM*, **38**(11), 54–64, (1995).
[9] Tony Lindgren, 'Methods for rule conflict resolution', in *In Proceedings of the 15th European Conference on Machine Learning (ECML-04)*, pp. 262–273, Pisa, (2004). Springer.
[10] Martin Možina, Jure Žabkar, and Ivan Bratko, 'Argument based machine learning', *Artificial Intelligence*, **171**(10/15), 922–937, (2007).
[11] Henry Prakken and Gerard Vreeswijk, *Handbook of Philosophical Logic, second edition*, volume 4, chapter Logics for Defeasible Argumentation, 218–319, Kluwer Academic Publishers, Dordrecht etc, 2002.
[12] Aleksander Sadikov, Martin Možina, Matej Guid, Jana Krivec, and Ivan Bratko, 'Automated chess tutor', in *Proceedings of the 5th International Conference on Computers and Games*, (2006).
[13] John Watson, *Secrets of Modern Chess Strategy*, Gambit Publications, 1999.
[14] Geoffrey I. Webb, Jason Wells, and Zijian Zheng, 'An experimental evaluation of integrating machine learning with knowledge acquisition', *Mach. Learn.*, **35**(1), 5–23, (1999).