

Combining Nonmonotonic Inference and Case-Based Reasoning to Solve Practical Problems

Hai Lin^{1a} and Baoliang Mu^{1b}

¹College of Software, Shenyang Normal University 253 Huanghe North Street, Shenyang, 110034, China

^ajl_linhai@163.com, ^b87090110@qq.com

Keywords: nonmonotonic reasoning, case-based reasoning, machine learning

Abstract. This paper aims to establish the connection between case-based reasoning and nonmonotonic reasoning. In particular, we suggest to make nonmonotonic inference using case-based reasoning. The key idea of this work is that nonmonotonic reasoning should be considered as an inductive process. When provided with incomplete knowledge, our method turns to previous cases and reuses these cases when they are found to be similar enough. We show that many benchmark problems in nonmonotonic reasoning literature can be solved using our proposed method.

Introduction

The problem of reasoning has been studied extensively in the AI literature [1,2,3,4]. The task of reasoning is to derive conclusions from some preconditions. In this paper, we suggest to capture nonmonotonic reasoning using case-based reasoning. The key idea of this work is that we should consider reasoning as an inductive process. We believe that the reasoning patterns that human use in our daily life are very flexible. When provided with incomplete knowledge, our method turns to previous cases and reuses these cases when they are found to be similar enough. And this similarity is defined by the learning process.

Traditionally learning and reasoning are treated separately. An agent learns some rules from its observations and uses these rules, possibly together with some other observations, to reason. The reasoning patterns are captured using logic, and are totally separated from learning. However, our method combines learning and reasoning in a single framework: The conclusions drawn are determined by the most similar case, which is in turn affected by the learning process. In our method, the reasoning patterns are implicitly extracted from past cases.

The rest of this paper is structured as follows. We present our method for nonmonotonic inference in the next section. And then we show how nonmonotonic reasoning can be performed using some benchmarks. Finally, we make some concluding remarks.

Our Method

In this section, we present our method for nonmonotonic inference using case based reasoning. We use the same method in [5] to represent cases. Cases are denoted as a vector $X=\{x_1, x_2, \dots, x_n\}$ of variables, possibly including unknown attributes. Each variable represents a world's feature and take value 1 (true) or 0 (false) or * (unknown). The agent learns weights of features as well as default values from the cases. When a feature's value is unknown, we use the feature's default value. For example, if the default value of "penguin" is 0.05, that means penguin's value can be assumed to be approximately the same as "false" when its value is unavailable. The reasoning task is thus performed by finding the most similar case and reusing this case.

From the point of view of case-based reasoning: In the RETRIEVE process, the most similar case is found using the weights of features and default values; In the REUSE process, we copy the solution; In the REVISE process, the conclusion is revised when it is not correct; In the RETAIN process, we simply store problem-solving experience as cases.

It is important to decide the similarity between two cases. Here we use weight-based similarity.

In particular, we extend the method in [6] to deal with unobserved attributes. The agent learns corresponding weight of each feature, as well as a default value ranging from 0 to 1 for each feature. The similarity between two cases X1 and X2 is computed by the following formula.

$$S(X_1, X_2) = \frac{1}{D(X_1, X_2)}$$

$$D(X_1, X_2) = \left(\sum_{f \in F} w(f) d(X_{1f}, X_{2f})^2 \right)^{\frac{1}{2}}$$

where F denote the set of features.

$$d(X_{1f}, X_{2f}) = |X_{1f} - X_{2f}|$$

where X_f denote the value of feature f, or the default value of feature f when it is unknown.

The learning process proceeds as follows: The algorithm updates feature weights by a fixed amount Δ1, as well as default value by a fixed amount Δ2, after classifying each training example. Feature weights are updated using the following strategy: Correct classifications cause the weights of matching (mismatching) features to be incremented (decremented). Incorrect classifications cause the weights of mismatching (matching) features to be incremented (decremented). Default values are also updated in a way that correct classifications are awarded and incorrect classifications are punished: Correct (incorrect) classifications cause the distance between features increased (decreased) by updating the corresponding default value by a fixed amount Δ2.

Test on Some Benchmarks

In this section, we consider some typical nonmonotonic reasoning problems from [5]. Using some past cases, these problems are solved by learning feature weights and default values, finding a most similar case, finally reusing the case.

Example 1 (Basic Example) Consider a query whether Tweety can fly. All we know is that Tweety is a bird. Assume that a reasoner have the following past cases.

- (bird = 1, penguin=1, fly=0)
- (bird = 1, fly = 1, red = 1)
- (bird = 1, fly = 1, red = 0)
- (bird = 1, penguin = 0, fly = 1, has_beak = 1)
- (bird = 1, fly = 1, has_beak = 1)
- (bird = 1, penguin = 1, fly = 0, has_beak = 1)

Given these cases, the following feature weights and default values can be learned using the method described earlier.

	bird	penguin	red	has_beak
feature weights	0.9	0.9	0.1	0.1
default values	0.5	0.1	0.5	0.5

Figure 1

Intuitively, the results imply that “bird” and “penguin” is much more important than “red” and “has_beak” for determining similarity. In addition, the value of “penguin” can be assumed to be approximately the same as “false” when its value is unavailable. Given these results, the most similar case found is the second case.

Example 2 (Specificity) We add into the above example more specific information that Tweety is penguin. In this case, the value of “penguin” is known, so we don’t have to use its default value. And the most similar case found is the first case, which yields the answer that Tweety cannot fly.

Example 3 (Irrelevance i) Still consider the above example. This time the query is whether a red bird Tweety can fly. The feature “red” is not a crucial feature in determining similarity since its weight is low. The most similar case found is still the second case. Hence the conclusion is that

Tweety can fly.

Example 4 (Irrelevance ii) Still consider the above example and the query is whether a penguin has beak. The following feature weights and default values can be learned. The most similar case found is the sixth case and the conclusion is that the penguin has beak.

	bird	penguin	red	fly
feature weights	0.9	0.1	0.1	0.1
default values	0.5	0.5	0.5	0.5

Figure 2

Conclusions

In this paper, we treat the nonmonotonic reasoning problem an inductive process and propose to use case-based learning to make the inference. This method differs from both traditional logic-based method. The common feature between nonmonotonic reasoning and machine learning is that they both get to conclusions that do not, in a strict sense, logically follow from premises. Future work includes investigating and finding case-based learning algorithms suitable for nonmonotonic reasoning. No method can be considered practical until it can be used to solve large-scale problems. So another future work is to test our method against large-scale benchmarks.

Acknowledgement

This work is supported by Liaoning Provincial Natural Science Foundation under grant 201202202, Scientific Research Foundation of Liaoning Provincial Education Department under grant L2012388.

References

- [1] Aamodt, A. & Plaza, E. (1994). Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications*, 7(1), pp. 39-59.
- [2] Ray, O (2009). "Nonmonotonic Abductive Inductive Learning", *Journal of Applied Logic*, pages 329-340.
- [3] Reiter, R. (1980). A logic for default reasoning. *Artificial Intelligence*, 13:81-132.
- [4]Rokka, J. (2012), "Agent-Based Argumentative Diagnostic Reasoning in Dementia".
- [5] Roth, D. (1995). Learning to reason: the non-monotonic case. In *Proceedings of the International Joint Conference of Artificial Intelligence*, pp. 1178-1184.
- [6] Salzberg, S.L. (1991). A nearest hyperrectangle learning method. *Machine Learning*, 6:251-276.