From JoJo to Frog: Extending a bi-directional Search Strategy to a more flexible three-directional Search

Dieter Fensel and Markus Wiese
Institut für Angewandte Informatik und Formale Beschreibungsverfahren,
Universität Karlsruhe (TH), 76128 Karlsruhe, Germany
e-mail: {fensel | wiese}@aifb.uni-karlsruhe.de

Abstract: ***Learning from examples*** is a field of research in machine learning where class descriptions, like decision trees or implications (production rules or horn clauses) are produced using positive and negative examples as information. To solve this task many different heuristic search strategies have been developed, so far. The search by ***specialization*** is the most widely used search strategy, whereas other approaches use a search by ***generalization*** only. ***JoJo*** is an algorithm that combines both search directions into one search procedure. According to the estimated quality of the currently regarded rule either a generalization or specialization step is carried out by deleting or adding one premise to the conjunction part of the rule. But, to create an even more flexible (and faster) algorithm, it should be possible to delete or add more than just one premise at a time. Relaxing this restriction of JoJo led to the new highly flexible algorithm ***Frog*** that additionally uses a third search direction.

Introduction

One broad field of interest in machine learning is concerned with what is called *learning from examples*. A set of examples is used to learn a classifier that describes the examples in a more compact way and that can also be used to classify new (unknown) cases. The ID3 ([Qui84]) and C4 ([Qui90b]) algorithms use a set of positive and negative examples for a class to derive a *decision tree* describing the class. The algorithms AQ ([MMH86]), CN2 ([ClN89], [ClB91]), JoJo ([Fen93], [FeW93]) and PRISM ([Cen87]) deal with the same task producing a *set of production rules* as classifier.

In this paper we discuss the ***heuristic*** search strategies of the algorithms JoJo and Frog that both produce a set of rules to solve this classification problem. Because the problem of finding minimal descriptions for a class using examples as input is NP-complete, several heuristic search strategies have been developed in the last years. In chapter two, we discuss the ***bi-directional search strategy*** of JoJo ([Fen93], [FeW93], [FKN93]) that integrates generalization and specialization into one flexible algorithm. But even JoJo´s very general search strategy has a serious limitation. In every generalization or specialization step just one premise is deleted from or added to a rule. This restriction does not hold for the new algorithm ***Frog*** that is introduced in chapter three. Frog uses the bi-directional search strategy of JoJo but generalizes it by allowing the deletion or addition of several premises in one step. Additionally, Frog uses ***Sidesteps*** as third search direction by deleting and adding the same number of premises to a rule at a time. Therefore, the algorithm Frog ***jumps*** in the lattice of rules instead of performing a number of single steps.

1. Generalization as Search

Machine learning from examples tries to generate a more general description from a set of examples. These examples are either positive or negative examples for a class. The target description of a class is a ***minimal sufficient set*** of ***minimal correct rules*** that covers all positive examples but no negative example. A single rule is an ***implication*** that has attributes as premises and a class as its conclusion.

A rule ***covers*** an example if the attribute values of this example is a super-set of the premises of the rule. If a rule covers only positive examples the rule is said to be ***correct***. Furthermore, a ***minimal correct rule*** is a rule which has a minimal set of premises, i.e. the rule would become incorrect if any of its premises were deleted. A set of correct rules is ***sufficient*** if every positive example is covered by at least one of its rules.[1] Such a rule set is ***minimal*** if the deletion of any rule causes some positive examples to be not covered at all. If the set of attributes is given a partial ordering, "≤" can be defined for these rules which can be used to induce a lattice. E.g., let two rules

r_1: $p_{11} \wedge p_{12} \wedge ... \wedge p_{1m} \rightarrow c$ and
r_2: $p_{21} \wedge p_{22} \wedge ... \wedge p_{2n} \rightarrow c$

with the same conclusion be given.

$r_1 \leq r_2$ holds if every model of $p_{11} \wedge p_{12} \wedge ... \wedge p_{1m}$ is also a model of $p_{21} \wedge p_{22} \wedge ... \wedge p_{2n}$
$\Leftrightarrow p_{11} \wedge p_{12} \wedge ... \wedge p_{1m} \models p_{21} \wedge p_{22} \wedge ... \wedge p_{2n}$.
$\Leftrightarrow \{p_{21}, p_{22}, ..., p_{2n}\} \subseteq \{p_{11}, p_{12}, ..., p_{1m}\}$.

Therefore, a rule can be ***specialized*** by adding a premise and be ***generalized*** by deleting a premise.[2]

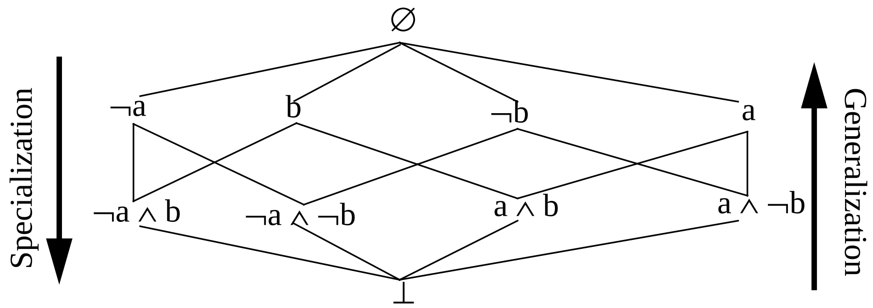

Figure 1. Lattice for two premises.

"The above generalization problem is essentially a search problem."[Mit81]. The ***version space algorithm*** (cf. [Mit81]) is a classical algorithm that learns rules from a set of examples by using a ***dual*** search strategy. But, since it performs an ***exhaustive search*** it can be applied only to small data sets as it is impossible to find the minimal hypothesis in polynomial time (cf. [BEH87]).[3]

2. Bi-directional Heuristic Search: JoJo

Since it cannot be decided that either specialization or generalization should be prefered (cf.[Fen93]) in any general case, we developed the algorithm JoJo that integrates both search dirctions into one bi-directional process using a heuristic search strategy. Starting from a chosen (arbitrary) point in the lattice of rules JoJo generalizes and specializes the currently regarded rule as long as the quality or correctness of the rule can be improved, i.e. until a local optimum

1. The handling of noise and uncertain rules is discussed in [FKN93].

2. Defining the ordering at the implications leads to the reverse direction:
$a \wedge b \models a$ but $a \rightarrow c \models a \wedge b \rightarrow c$.

3. A further and mathematically well-defined procedure is contained by the formal concept analysis, which produces iteratively all pseudo-contents of a lattice (cf. [Gan87], [Wil87]).

is found.

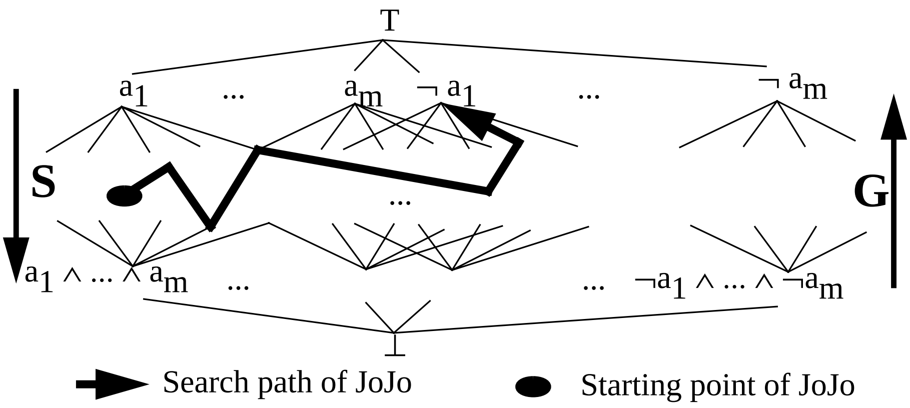

Figure 2. JoJo

Integrating both search directions gives JoJo two considerable advantages:

- Depending on one´s preference that is determined by domain and task-specific circumstances, the procedure can utilize the advantages of both search strategies.
- The search direction can be switched during the search and both directions are therefore more flexibly integrated than in rule induction using specialization with additional pruning.

An evaluation of JoJo and its extension to a four-step incremental procedure to refine, complete, reduce, and minimize a set of rules according to new examples is given in [FeW93]. In the following, we only sketch the main ideas of JoJo.

2.1 Choice of the Starting Points

Contrary to algorithms that use just one search direction, JoJo is able to start at any arbitrary point in the lattice because using both search directions makes it possible that every rule in the lattice can be reached.[4] Heuristics for choosing appropriate starting points are discussed in [Fen93].[5] Furthermore, it is possible to carry out several program runs with different starting points. In the case of incremental refinement rules that have already been produced by JoJo or other algorithms can be used as starting points for further refinement and improvement of the rules.

2.2 Search Process in the Lattice

The core of JoJo consists of three components: a ***generalizer***, a ***specializer***, and a ***scheduler***. The ***generalizer*** and the ***specializer*** compute, validate and order the descriptions that can be reached by the next generalization or specialization step using a predefined strategy and a predefined preference criterion. An example for a simple generalizer is H-RELAX (cf. [FeK92]):

- Conjunctions are generalized by deleting a premise.

4. Algorithms that work by specialization only have to start with the most general description so that they don‘t overlook possible solutions while only generalizing algorithms have to start with most specific descriptions

5. E.g., rules are produced randomly for every length and the distribution of their quality is used to decide the starting points.

- The applied g-preference is:[6]

$$1 - \frac{\text{number of covered negative examples} + 0{,}5}{\text{number of covered positive examples}}$$

An example of a simple specializer is:

- Conjunctions are specialized by adding a premise.
- The applied s-preference is equal to the g-preference.

Other generalizers or specializers with different strategies and preference criteria are possible.

The ***Scheduler*** selects the search direction in the lattice of rules by using a predefined (total) t-preference. An example of a simple scheduler is:

- Specialize if the error rate of the rule is higher than the specified threshold.
- Otherwise, choose the best generalization if a possible generalization exists, i.e. a generalization with allowable error rate.
- Otherwise stop.

The scheduler would prefer most-general (but correct) descriptions.

The third main feature of JoJo, its immediate integration of an incremental learning procedure, is discussed in [FeW93]. Rules which are produced by JoJo or other algorithms can be refined by JoJo according to new examples which are introduced. The rules are used as starting points for the search for the refined rules.

3. Frog: Three-directional Search by Jumps

The strong restriction of JoJo to add or relax just one premise per step leads to a long search time to leave an unfavourable region that is especially annoying if a too specific or too general starting point was chosen. A strategy which can save time is allowing to add or delete several premises in one step.

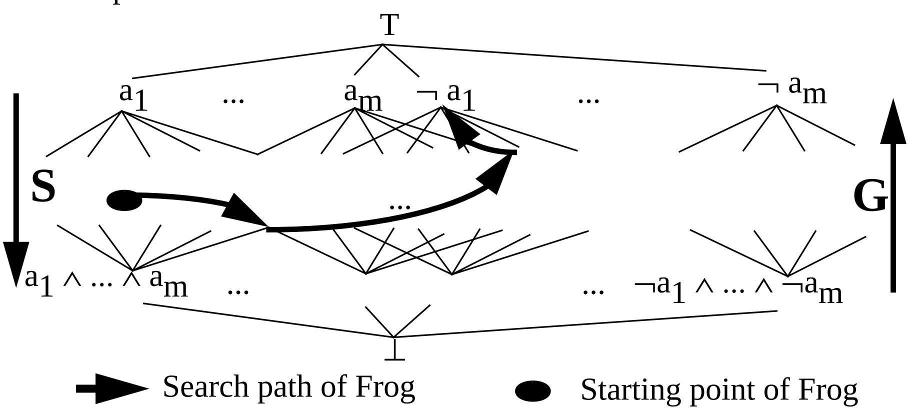

Figure 3. Frog

According to the quality of a current rule and the given search amount per step, several alternative rules can be evaluated in the environment of the current rule and used as the rule which is regarded as starting point for a further search. In the following, we sketch the Frog algorithm:

6. In the case of no covered positive example the g-preference prefers rules that cover less negative examples over rules that cover a greater number of negative examples. But, rules that cover at least one positive example are prefered over rules that cover only (but possibly a smaller number(!) of) negative examples in any case.

```
current rule := init;
while current total search amount < total search amount treshold
do
        successor rule := current rule
        initialize current search amount
        while current search amount < search amount threshold
        do
                compute a new successor rule of the current rule
                /* store the new successor as a rule which has already been regarded
                to prevent repetition. */7
                if quality(successor rule) < quality(new successor rule)
                then successor rule := new successor rule endif
                increase current search amount
        enddo
        current rule := successor rule
        increase total search amount
enddo
```

Besides generalization and specialization steps, Frog uses also Sidesteps to derive a successor rule from the current rule. These might be very useful especially for incorrect rule because the semantical change of a sidestep can correct the rule without changing the number of premises. Generally, there exist two possibilities to perform a sidestep. First, one premise just changes its attribute value. Second, a new premise is introduced to the conjunction part of the rule while an existing premise is being deleted. The following two examples correspond to sidesteps in the lattice of all possible rules:

- $a_1 = 3 \wedge a_2 = 0 \rightarrow h$ [8] is replaced by $a_1 = \boldsymbol{2} \wedge a_2 = 0 \rightarrow h$
- $a_1 = 3 \wedge a_2 = 0 \rightarrow h$ is replaced by $a_2 = 0 \wedge \boldsymbol{a_3 = 1} \rightarrow h$

The new rule is neither a generalization nor a specialization of the given rule.

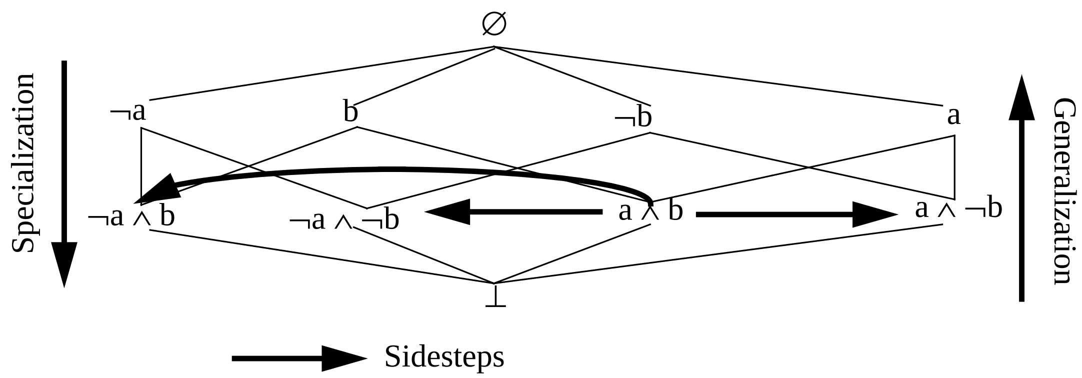

Figure 4. Sidesteps.

But, since the new rule can cover both fewer negative examples and more positive examples it can have a higher quality as the current one. Therfore, Frog does not only examine generalizations or specializations of a current rule, but also examines rules that are derived from the current rule by changing some of the premises. The rate of change used to derive the new rule decreases as the quality of the current rules increases, i.e. the “length” of the jumps becomes smaller during the search process. In general, allowing bigger jumps in the lattice of rules doesn‘t only mean to have the advantage of saving search effort, but gives Frog also the possibility of leaving a local optimum.

---

7. If the new successor rule is not marked as an already regarded rule the danger of repetition and loops arise. Otherwise, the demand on storage and time increases if the algorithms tries to prevent repetition.

8. Ordinal attributes are used in this example.

Conclusion

The paper shows bi-directional search strategies that can be applied to the task of learning from examples. The algorithm JoJo was developed to integrate both specialization and generalization into one bi-directional search strategy as one cannot generally be prefered over the other. Since JoJo still has some shortcomings, a three-directional search strategy using jumps has been developed. The algorithm Frog can delete, add or change several premises in one step so that it requires less search time.

References

[BEH87] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth: Ocam´s Razor, *Information Processing Letters*, vol 24, 1987, pp. 377-380.

[Cen87] J. Cendrowska: PRISM: An Algorithm for Inducing Modular Rules, *Int. J. Man-Machine Studies*, vol. 27, 1987, pp. 349-370.

[ClN89] P. Clark and T. Niblett: The CN2 Induction Algorithm, *Machine Learning*, vol. 3, no. 4, 1989, pp. 261-283.

[ClB91] P. Clark and R. Boswell: Rule Induction with CN2: Some Recent Improvements, *Proceedings of the European Workshop on Machine Learning (EWSL´91)*, March 6-8, Porto, Portugal, 1991, pp. 151-163.

[FeK92] D. Fensel and J. Klein: Solving a Generalization Task by Generalization: RELAX, H-RELAX, and I-RELAX. Three Algorithms for Rule Induction and Pruning. Institut für Angewandte Informatik und Formale Beschreibungsverfahren, University of Karlsruhe, research report, no 232, Januar 1992.

[Fen93] D. Fensel: JoJo: Integration of Generalization and Specialization, *Proceedings of the Workshop Knowledge and Data Engineering*, Atelier d´Ingenierie des Connaissances et des Donees, A.I.C.D., Strasbourg, France, January 25-27, 1993. Also appeared as: JoJo, Institut für Angewandte Informatik und Formale Beschreibungsverfahren, University of Karlsruhe, research report, no 261, Januar 1993.

[FeW93] D. Fensel and M. Wiese: Incremental Refinement of Rule Sets with JoJo, *Proceedings of the European Conference on Machine Learing ECML-93*, April 5-8, Vienna, Austria, Lecture Notes in AI, no 667, Springer-Verlag, Berlin, 1993.

[FKN93] D. Fensel, J. Klein, and U. Neubronner: RJ: An Environment for Learning from Example, *Proceedings of the 13th International Conference Expert Systems and Their Applications*, 24-28 Mai, Avignon, 1993.

[Gan87] B. Ganter: Algorithmen zur Formalen Begriffsanalyse, B. Ganter et al. (eds.), *Beiträge zur Begriffsanalyse*, B.I.-Wissenschaftsverlag, Mannheim, 1987, pp. 241-54.

[Mit81] T.M. Mitchell: Generalization as Search, B. Webber et al. (eds), *Readings in Artificial Intelligence*, Tioga Publishinh Co., Palo Alto, 1981.

[MMH86] R. S. Michalski, I. Mozetic, J. Hong, and N. Lavrac: The Multi-Purpose Incremental Learning System AQ15 and its Testing Application to Three Medical Domains, *Proceedings of the 5th National Conference on AI (AAAI-86)*, Philadelphia, August 11-15, 1986, pp. 1041-1045.

[Qui84] J.R. Quinlan: Learning Efficient Classification Procedures and Their Application to Chess End Games, R.S. Michalski et al. (eds.), *Machine Learning. An Artificial Intelligence Approach, vol.1*, Springer-Verlag, Berlin, 1984, pp. 463-482.

[Qui90b] J.R. Quinlan: Probabilistic Decision Trees, Y. Kodratoff et al. (eds.), *Machine Learning. An Artificial Intelligence Approach, vol. III*, Morgan Kaufmann Publisher, San Mateo, 1990.

[Wil87] R. Wille: Bedeutung von Begriffsverbänden, B. Ganter et al. (eds.), *Beiträge zur Begriffsanalyse*, B.I.-Wissenschaftverlag, Mannheim, 1987, pp. 161-212.