Learning to Learn Decision Trees

Vlad G. Dabija

Knowledge Systems Lab., Comp. Sci. Dept.
Stanford University, Stanford, CA-94304
vlad@cs.stanford.edu

Katsuhiko Tsujino **Shogo Nishida**

System 4G, Central Res. Lab., Mitsubishi Electric Corp.
8-1-1 Tsukaguchi-Honmachi Amagasaki, Hyogo 661 JAPAN
{tsujino,nishida}@sys.crl.melco.co.jp

Abstract[1]

Decision trees are widely used in machine learning and knowledge acquisition systems. However, there is no optimal or even unanimously accepted strategy of obtaining "good" such trees, and most of the generated trees suffer from improprieties, i.e. inadequacies in representing knowledge. The final goal of the research reported here is to formulate a theory for the decision trees domain, that is a set of heuristics (on which a majority of experts will agree) which will describe a good decision tree, as well as a set of heuristics specifying how to obtain optimal trees. In order to achieve this goal we have designed a recursive architecture learning system, which monitors an interactive knowledge acquisition system based on decision trees and driven by explanatory reasoning, and incrementally acquires from the experts using it the knowledge used to build the decision trees domain theory. This theory is also represented as a set of decision trees, and may be domain dependent. Our system acquires knowledge to define the notion of good/bad decision trees and to measure their quality, as well as knowledge needed to guide domain experts in constructing good decision trees. The partial theory acquired at each moment is also used by the basic knowledge acquisition system in its tree generation process, thus constantly improving its performance.

Introduction

Inductive systems based on decision trees formation are commonly used in machine learning and knowledge acquisition. Therefore, an important amount of research has been devoted to designing and refining algorithms for learning decision trees. However, when a human expert inspects the results of such an algorithm, he will usually point out some inadequacies (which we will call improprieties) in the structure of the generated trees. By interpreting the global situation in which each of these improprieties arises, the expert can recommend corrective actions, according to his particular domain of expertise.

The problem of generalizing both the interpretation of improprieties, and the actions recommended by experts to be taken in order to eliminate these improprieties, is equivalent to formulating a theory of the decision trees domain; and in its turn, acquiring this theory amounts to learning efficient ways to learn "good" decision trees, that is trees which satisfy the expert's knowledge, experience and aesthetical feelings about his own domain of expertise, criteria which can vary slightly from one domain to another. Although a little surprising, representing this theory itself using decision trees came very natural in our work.

This work was inspired by the need to provide an interactive knowledge acquisition tool that stimulates the expert in specifying the appropriate knowledge in the appropriate form. The idea of such a tool is not new at all (one of the first such tools was the TEIRESIAS system (Davis 1979) designed to help the implementation of EMYCIN based consultation systems (vanMelle 1980)), and following it there have been a number of such systems providing different degrees of help for the expert. However, our approach tries to advance one step further in helping the expert articulate the underlying knowledge of his domain, by integrating knowledge acquisition and machine learning through combining domain knowledge acquisition with the use of examples.

Basic Framework

Our knowledge acquisition system named KAISER - a Knowledge Acquisition Inductive System driven by Explanatory Reasoning - (figure 1) (Tsujino, Takegaki, & Nishida, 1990), inductively learns classification knowledge in the form of a decision tree and analyzes it using its domain knowledge to detect improper conditions and mismatches between theory and induction results. These improper conditions are then used to guide the expert in changing and augmenting the knowledge base to eliminate them, and to continue the induction cycle. This process supplies a mental stimulus for the expert to help him to refine and better organize his experience and knowledge of his domain. KAISER's impropriety knowledge

[1]This work was done while the first author was a visiting researcher at System 4G, Central Research Laboratory, Mitsubishi Electric Corp., Amagasaki, Japan

represents both what makes a decision tree to be good or bad, and how to improve such a tree. Similar knowledge is issued by constructive induction techniques (Matheus 1989, 1991; Pagallo 1990) which deal with equivalent questions like when should new features be built or old ones discarded, and what constructive operators to apply to which features. The main difference, however, is that KAISER aims at interactive knowledge acquisition while constructive induction aims as automatic feature generation. Therefore, KAISER generates explanations to help the expert in understanding the problem, and offers a wide variety of impropriety elimination actions in addition to constructing new features - e.g. by pruning, asking for new examples and domain knowledge, redefining the problem (its classes and attributes), etc.

It accepts examples in OAV form (i.e., class, attributes and values) and domain knowledge (e.g., abstract rules that examples must/may satisfy). Decision trees are learned inductively from the current set of examples (using an ID3-like algorithm (Quinlan 1986)) and then evaluated based on domain knowledge to detect 'improprieties'. The key idea of KAISER is that (i) this evaluation process uses a fair amount of heuristic knowledge about decision trees that constitutes a qualitative measure of the goodness/badness of a decision tree, and therefore (ii) by defining an appropriate set of improprieties, their explanations and elimination actions, we can make an intelligent and apt interview to acquire more refined domain knowledge from the expert. Domain knowledge includes (possible incomplete/incorrect) knowledge about (i) the relation between classes and the attribute values (i.e., the value of attribute 'eyes' must be 'oval' if the class is 'cat'), (ii) ordinal relationships among attribute values (i.e., the value 'yellow' of attribute 'color-of-warning-lamp' is between the values of 'red' and 'green') and (iii) derivation knowledge of an attribute value from other attribute values (i.e., the value of attribute 'weight' can be calculated from the values of 'mass' and 'density' by multiplying them together). By referring this domain knowledge, the impropriety detector discovers improprieties such as noisy or unreliable conditions, similar structure, or mismatches between domain knowledge and the decision tree. The impropriety interpreter combines them together to derive more specialized and adequate improprieties. The impropriety selector chooses the most important one to be treated next. To eliminate the impropriety, KAISER prints the associated explanation and actions, which usually prompt the expert for new examples and domain knowledge.

In order to acquire "good" knowledge, several researchers have attempted to integrate induction which guarantees the operationality of this knowledge and deduction which validates its reliability. Nunez proposed his EG2-like algorithm for decision tree induction (Nunez 1991), which adopts a hierarchy of attribute values and the measurement of attributes cost as background knowledge to learn more understandable trees than ID3, but it does not handle the abstract relationship between classes and attributes which is necessary to acquire reliable and expert interpretable knowledge. Pazzani proposed an induction system named FOCL (Pazzani & Kibler 1990), which accepts this knowledge in the form of Horn clauses and evaluates and refines them by induction. Although KAISER is mainly based on induction, it also provides deductive explanation-based optimization of induced knowledge by using domain knowledge. Instead of refining all this knowledge automatically, KAISER tries to conduct an interview with the expert and obtain it based on the heuristic knowledge expressed as improprieties. Pazzani also suggests using such heuristics to achieve interactive knowledge acquisition in the KR-FOCL (Pazzani & Brunk 1991) system. The heuristics of KR-FOCL largely correspond to KAISER's improprieties and their elimination actions.

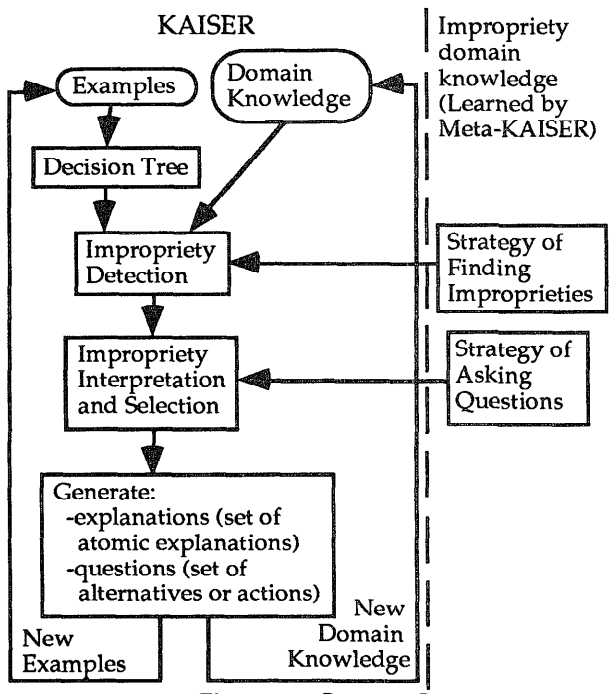


Figure 1. System Structure

A Domain Theory for Decision Trees

In the original implementation of KAISER, the knowledge used to interpret tree improprieties and to suggest corrective actions was specified and

hand-coded by us through an introspection process. As we have expected, some of the experts using KAISER to specify knowledge in different domains (gas composition analysis of electric transformers, diagnosis of electric motors failures, tap changers for electric transformers), expressed some disagreement with the system's interpretation of certain combinations of improprieties or with its recommendations in some particular cases. Thus, we realized that the impropriety domain theory specified by us was neither complete, nor totally consistent, and therefore was not well enough suited to its purpose of providing an efficient tool for an easy improvement of the domain knowledge base. It became clear that the process of defining the improprieties and their associated explanations and suggested corrective actions is a complex heuristic process. The improprieties domain is best characterized as common sense knowledge, possible domain dependent. It requires learning heuristics about detecting and using improprieties (which are themselves heuristics about incoherences in decision trees), and thus it is part of Heuretics - the study of heuristics as defined by Polya (Polya 1945). What we actually needed was a theory of the domain of classification trees composed of a set of heuristics which specify: (i) WHAT means to have a good/bad decision tree for a given domain (by identifying possible improprieties), (ii) WHY a decision tree is good/bad (by supplying appropriate explanations to all possible combinations of improprieties), and (iii) HOW to obtain a good decision tree (by suggesting the best corrective actions to eliminate these improprieties). The formation of this theory is the ultimate goal of our research, and the tool to formulate it is described in the next section.

The primitives used to formulate the theory for the domain of decision trees are the impropriety, the atomic explanation and the alternative.

An impropriety represents anything that the expert believes is not right about a decision tree. It may be something definitely wrong about it (like a node which has examples belonging to conflicting classes), or something that the expert finds strange (like a path which appears to lack some essential condition), or even something that may be fine in general, but in this particular case the expert would prefer it otherwise (like two sibling subtrees which are not identical but consist of similar conditions). Appendix 1 presents the improprieties which actually formalize these informal descriptions and their associated explanations and actions, as they have been acquired by our system. By interpreting the reason why such an impropriety appears, the expert can recommend corrective actions (usually new examples and/or new constraints according to his domain knowledge). Each impropriety is associated with a node of the tree (if the impropriety refers to a subtree, then it will be associated with the root of that subtree). One node of the tree may have, simultaneously, a combination of improprieties, which taken together may have different properties than a simple union of the properties of each of these improprieties. Each impropriety and combination of improprieties has an associated explanation (of why it occurred and what it represents), and an associated action (representing a set of possible corrective measures that can be taken to eliminate the impropriety)[2].

An atomic explanation is an arbitrary large piece of an explanation. It may, but need not, constitute an explanation by itself. The explanations associated with the improprieties are made up of an ordered list of atomic explanations.

An alternative is a piece of action that can be taken to eliminate the impropriety (or combination of improprieties) with which it is associated. Each alternative may constitute an action, and each action associated with improprieties is made up of an ordered list of alternatives. These are ordered according to their efficiency in eliminating the impropriety (i.e. less efficient alternatives will eliminate the current impropriety but may create a new, less "important" one), and are not necessarily ordered according to their frequency of use. Therefore there is no statistical measure to be used in learning or in ordering them. Also, though ideally we would like to obtain an autonomous system which

---

[2]The set of improprieties and their combinations can be totally ordered according to their gravity, and this way we can propose a theoretical measure of the quality of a decision tree, by taking into account the combined effects described by the discovered improprieties. Let g(i) be the gravity of the impropriety of each node on a scale of 0 to 1, and N the number of nodes in the tree. This quality may then be defined as:

$$Q(T) = 1 / [\sum_{i=1}^{N} (1 + g(i))]$$

Using this measure, we can analyze and compare different decision trees for the same domain, and can improve them by first eliminating the most damaging improprieties. We have experimented with a number of such expressions, and although this is the closest approximation to the concept of the quality of a decision tree as described here, it is still only of theoretical value, and not suitable for implementation in our system; a single formula seems too rigid to encompass by itself the complexity of the problems that may appear in forming decision trees. Therefore, our system uses a set of qualitative heuristic rules for deciding the next impropriety to reduce. An example of such a rule is: "An impropriety that needs a tree modification takes precedence over one that requires domain knowledge refinement", and their complete set is listed in [Tsujino et. al., 1990].

may apply a single alternative for each combination of improprieties detected, this is clearly not possible, and the final result will be a list of the best alternative actions to be taken for each impropriety, ordered by their efficiency in obtaining a "better" tree, and ask the expert to choose one of them.

A Learning Apprentice for the Decision Trees Domain Theory

The decision trees domain theory consists of the strategy of finding improprieties and the strategy of asking questions. Their initial specification in KAISER proved itself incomplete and difficult to use, and therefore we decided to try to specify a comprehensive theory of the impropriety domain.There are different possible ways to build such a theory. One way would have been by searching the space of such heuristics, approach similar to (Lenat 1982, 1983). However, an automatic evaluation criteria for the generated heuristics was difficult to define, and we felt that an interactive system would have better chances of success.Thus, our first step was to provide a tool for the analysis of the decision trees domain, as well as for the systematic incremental accumulation of knowledge about this domain, while in the same time using this already acquired knowledge to help other domain experts in already deriving better decision trees for their particular tasks.The most natural way for us to build such a tool was to employ a recursive architecture, by bootstrapping the learning system (Pat Langley, personal communication, March 1991) using KAISER to acquire knowledge in the domain of decision trees improprieties from human experts' experience and skills, knowledge which is in the same time applied by KAISER to generate its explanations and questions. We gave the name Meta-KAISER to the system used to acquire impropriety domain knowledge, since the kind of knowledge it acquires is meta-knowledge from KAISER's point of view, being used by it to acquire other specialized domain knowledge. One further difficulty was the absence of any actual expert in this domain, (the domain itself was defined by us). However, each expert in a field can also be used as an impropriety domain expert, since he is usually able to articulate his disagreement or discontent with the current classification tree generated by KAISER. This way, the Meta-KAISER system was born, to run in parallel with the domain acquisition system (KAISER) and to acquire knowledge, whenever the opportunity presents itself, for the decision tree improprieties domain. Meta-KAISER is a learning apprentice system which accumulates this knowledge from every expert that uses KAISER in his own domain, and it is our hope that it will eventually converge towards a unified theory of the decision trees domain (maybe application-domain dependent), which will in turn enable KAISER to more efficiently help experts in generating specific domain dependent knowledge bases.

It is already known that the performance of a theory formation system is heavily dependent on representation, on the relationship between form and content (Lenat & Brown 1983). Given the specifications of the KAISER system used for this purpose, we have decided to also use a decision tree representation for the heuristics describing the decision trees domain theory.

For the strategy of asking questions, we use a multiple class decision tree for each place in the ordered list of alternatives which represents an action, and another tree for each place in an explanation. The classifying (binary valued) attributes are the detected improprieties. Each such decision tree will indicate, for a detected set of improprieties, the best alternative (resp. atomic explanation), if any, to be given to the expert on the corresponding place in a question (resp. explanation). Appendix 2 presents such a tree for the alternatives on the first position in an action, together with a partial list of improprieties and alternative corrective actions currently known to the system. Any kind of representation would be used, it would introduce some kind of bias on the learning process in the improprieties domain. After experimenting with several kinds of such representations, we concluded that this one is the best suited for both the domain to be learned and the system used towards this goal. Though the representation is clearly inconvenient for human interpretation, a simple procedure translates the domain theory it describes into human readable form, generating a set of heuristics which completely describe Meta-KAISER's knowledge of the impropriety domain at any current stage, in a form easily understandable and interpretable (and therefore modifiable) by the human expert.

From the expert's reaction to the way KAISER treats improprieties in the expert domain's decision tree, Meta-KAISER generates new positive and negative examples in the decision trees domain to support the expert's point of view and uses these examples to refine its impropriety domain knowledge.The expert is allowed, at any time during the process of knowledge acquisition and decision tree formation in his own domain of expertise, to add or delete an alternative or explanation for a certain combination of improprieties, or to modify the order in which the system suggests explanations or actions for a combination of improprieties. He may also define a new basic impropriety, or a new alternative or atomic explanation. The system generates all pertinent examples suggested by the current situation by adding to the current set of examples new positive examples for the added

alternatives and negative examples for the deleted ones, and the appropriate type of examples in the case of an order modification. In each case, the system checks for possible example conflicts and manages them in a consistent manner. When the expert defines a new concept (impropriety) by specifying a detection procedure for it, the system also automatically generates a set of plausible examples corresponding to the expert's definition of the concept, and will refine them during subsequent interaction with the expert. After accepting any changes indicated by the expert in the impropriety domain knowledge, Meta-KAISER uses all the examples and domain knowledge acquired so far regarding decision trees improprieties to rebuild the affected classification trees which represent the heuristics for the impropriety domain theory. If necessary, the original KAISER mechanism conducts a dialogue with the expert to resolve any improprieties in these trees.

After this process is concluded, the newly generated trees will immediately constitute KAISER's new strategy of asking questions. This way, KAISER continuously modifies itself (using Meta-KAISER) to improve its performance in assisting the domain experts in specifying their own domain knowledge. Appendix 1 presents an example of a heuristic modified by one of the experts that have worked with our system.

Due to the uncertainty and complexity of the impropriety domain, it is possible that, at some point, the expert contradicts one of his previous requests or recommendations regarding KAISER's treatment of the improprieties in the decision tree, or that he contradicts the opinion of a previous expert in a similar situation. Therefore, Meta-KAISER remembers the history of changes made to the decision trees domain (for each combination of improprieties) by different experts. If such a case happens, the system points out the contradiction and supplies the expert with all the information about that case, allowing the expert to make an informed decision.

For the heuristics used by the strategy of finding improprieties we have also employed a decision tree representation and an identical knowledge acquisition method. There is one decision tree for each impropriety type,using as attributes different kinds of tree characteristics: local to a node (like the number of examples supporting the class of a node) or global to the tree (like the existence of two similar subtrees with the same parent). The expert may again define new improprieties or characteristics to consider, or may require modifications to this strategy, similarly to what he can do for the strategy of asking questions. Meta-KAISER generates new examples to support the expert's requests, and then creates new decision trees which represent the current set of acquired impropriety detection heuristics.

We specified an initial impropriety domain knowledge for Meta-KAISER, again using the basic KAISER system. From previous observations of the system behavior, we have defined a basic domain theory for improprieties and have extracted a set of examples which were used by KAISER to produce and refine the set of decision trees which represent the initial heuristics describing the impropriety domain knowledge. (Dabija, Tsujino, & Nishida 1992) presents this initial domain knowledge and an example of a decision tree generated by KAISER for it.This initial knowledge base is currently refined by each expert using the KAISER system in his own domain of expertise, as Meta-KAISER is designed to always work in parallel with KAISER and to respond to any discontent voiced by the expert with regard to KAISER's suggested explanations and/or actions.

Different criteria for judging the quality of decision trees are applied by experts in different domains, and our mechanism is able to accomodate them by learning domain dependent impropriety theories. This way, when KAISER is used to learn decision trees in a given domain, the impropriety detection and interpretation knowledge to be used will be the one acquired by Meta-KAISER in previous applications for the same domain. Further studies are needed to determine whether we can isolate a basic general theory (perhaps as intersection of all domain dependent ones) and whether we can specify the domain dependent theories as variations from the basic one.

Conclusions

Decision trees are a powerful knowledge acquisition tool, but the algorithms used to build them usually produce improper trees. While implementing a system that aids experts in improving the decision trees generated for their domains, we have found out that the domain of decision trees is in itself poorly understood and lacks a unified theory. The final objective of this research is to provide a theory of what a good decision tree is, and a practical tool for obtaining them, through a complete understanding of the classification trees domain, their inefficiencies and improprieties. However, we believe that this goal can be achieved only by a prolonged and constant analysis of a considerable number of decision trees, applied in different domains. Since there is no actual expert in this domain, the acquisition of this theory must be done from different examples which appear during the knowledge acquisition processes for other domains. The process of articulating the knowledge for the impropriety domain theory seems almost impossible when it is first presented to any person. However, when such situations are actually

encountered by experts during their attempts to specify knowledge in their own domains, the process of correcting these particular situations seems very natural to the experts. Thus, our first step was to provide a tool (Meta-KAISER) for the analysis of the decision trees domain, as well as for the systematic incremental accumulation of knowledge about this domain, while in the same time using (through KAISER) this already acquired knowledge to help domain experts in already deriving better decision trees for their domains. We have integrated these systems in a recursive, self-bootstrapping architecture which employs a unitary knowledge representation paradigm using decision trees to represent the decision trees domain heuristics themselves. Meta-KAISER is designed to always run in parallel with KAISER, with two advantages: (i) it can use every expert working with KAISER as its own domain expert and will add the opinions of these experts incrementally, and (ii) the partial theory it has acquired at every moment will be used by KAISER in its own operation, thus continually improving its own performance. Moreover, the decision trees domain theory it acquires and uses may be domain dependent, and therefore can be fine-tuned for each particular application domain. Based on this theory we will be able to design a system that analyzes the improprieties in the trees it generates, and is able to recommend (or whenever possible to take by itself) the best suited actions in order to improve these trees.

We expect the process of gathering and stabilizing the knowledge for the decision trees domain to be a long one, requiring the use of the KAISER system in the knowledge acquisition process for many particular domains and the interaction of the respective experts with Meta-KAISER. However, the results obtained so far make us believe that eventually, after a large number of runs, Meta-KAISER will gather generally accepted sets of heuristics to describe the decision tree domain theory. While the first experts using KAISER had more trouble in adjusting this knowledge, Meta-KAISER is less and less frequently invoked as it approaches a generally accepted basic theory for decision trees. Appendix 2 presents part of this theory as generated by Meta-KAISER after a few runs. Preliminary results show a clear improvement in KAISER's performance as confirmed by the experts using it, but further experiments are needed in order to refine the theory learned by Meta-KAISER and to determine the suitability of domain-dependent vs. general decision trees theory. The initially generated trees, representing the decision trees domain theory, were already complex although the heuristics they represent were defined by us in an introspective way. But these heuristics were clearly incomplete, and we expect the corresponding theory to become much more complex by the consistent use of Meta-KAISER together with every application of the KAISER system.

It is possible that eventually, after a large number of Meta-KAISER runs, the decision tree domain theory it develops will converge to a quasi-stable set of heuristics. However, total convergence to a perfectly stable theory is probably not possible and even not necessarily desirable, particularly since this theory may have a component dependent on the particular domain for which the decision trees are used. This view of learning systems which do not converge to a perfectly stable state, but may oscillate among a number of partially satisfactory states, has been encountered in other domains too (Dabija 1990), and is also acknowledged by other researchers (Pat Langley, personal communication, June 1991). Our framework is particularly suited for acquiring and developing specific domain dependent theories of decision trees, sensitive to the particularities of different application domains.

Acknowledgement

The authors want to express their thanks to Pat Langley for several very helpful comments on an earlier draft of this paper.

Appendix 1

The following dialogue was conducted by KAISER (and Meta-KAISER) while acquiring diagnosis knowledge for the domain of oil isolated electric transformers. The problem at hand is to classify a faulty transformer into one of three classes of faults (Arc, Over Heat and Partial Discharge) based on the composition of the dissolved gas such as H2, CH4 and C2H6. Underlined strings show the responses of the expert.

impropriety detected: **Explicable-jumble-leaf**:

This leaf contains **three** examples of **two** classes (**Arc** and **OverHeat**) that cannot be separated because their attribute values are exactly same. Domain knowledge **Over-Heat-02** suggests that the class may be **OverHeat**.

(1) Accept the suggestion, and change it into a leaf of class **OverHeat**.
(2) Change it into a leaf of major class **OverHeat**.
(3) Merge it with its largest siblings of class **Arc**.
(4) Name the class.
(5) Try further separation.
(6) Modify the alternatives.

Choose: 1

It was changed into a leaf of class **OverHeat**.

impropriety detected: **Near-miss-explanation**:

A leaf of class **Arc** failed in explanation because one and only one condition **CH4-C2H6** of domain knowledge **Arc-01** was not satisfied.

(1) Graft the condition **CH4-C2H6** and expand the tree.
(2) Give new counter examples that arrives to this leaf but the class is not **Arc**.
(3) Modify the alternatives.

Choose: 1

CH4-C2H6 will be grafted and the tree expanded.

impropriety detected: **Twin-nodes**:

Two siblings **Node23** and **Node25** refer to the same attribute (**H2-C2H2**), and the major class in their examples is the same (**Arc**).

(1) Merge the twins and generate a new subtree from the union of their examples.
(2) Merge the twins and ignore the smaller sibling (Node25).
(3) Merge the twins and ignore the larger sibling (Node23).
(4) Modify the alternatives.

Choose: 4

Calling Meta-KAISER...

Meta-KAISER:

(1) Define new impropriety.
(2) Define new alternative.
(3) Add alternative to impropriety list.
(4) Delete alternative from impropriety list.
(5) Modify order of alternatives for impropriety list.
(6) Return to previous menu.

Choose: 1

New Impropriety name: **Twins-not-in-order**

Current known alternatives are:

1 - Name the class of a leaf.
2 - Give new examples that will arrive to the current node.

: :

List the alternatives (in proper order) to be used with this impropriety: **9 10 16**

Please contact the system developer if you need help during the following session:

This new impropriety must be (1) detected or (2) derived? **2**

Current known improprieties are:

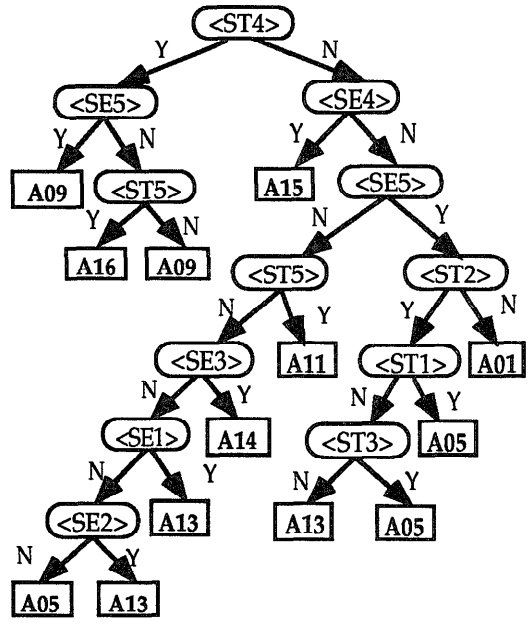
1- NIL-leaf
2- Noisy-node

: :

List the improprieties from which this new one should be derived: **14 and not11**

Appendix 2

Here is an example of a decision tree learned by Meta-KAISER to describe the impropriety domain theory. Each leaf specifies the first alternative in each action suggested by KAISER, when the combination of improprieties in the nodes on the path to this leaf is found in an application domain decision tree.



The improprieties tested for in the nodes are (Tsujino, Takegaki, & Nishida, 1990):

(ST1) *Noisy node* impropriety: a node has few examples, e.g., less than half of the average leaf size. This is a typical clue for pruning.

(ST2) *NIL leaf* impropriety: the class of a leaf cannot be determined by induction because of the lack of examples. This often arises when we use multiple valued (not binary) attributes.

(ST3) *Inseparable examples* impropriety: some examples can not be separated only with given attributes. This impropriety possesses features of both ST1 and ST2.

(ST4) *Similar node* impropriety: two brother nodes refer the same attribute, their entropies are near, and they consist of similar component classes. This is a structural clue to generalize the attribute of their father by merging the links to them.

(ST5) *Similar class* impropriety: more than one node tries to separate the same set of classes at different places in a decision tree. This is a primitive clue for a new attribute for separating the conflicting classes, which is represented by a subtree induced from the subset of examples that belong to the conflicting classes.

(SE1) *Contradictory explanation* impropriety: the conditions to a leaf are explicable by the domain knowledge that belongs to a different class. A primitive elimination action of this impropriety is to specialize the conditions of the miss-matched domain knowledge, and/or to generalize the conditions of the domain knowledge that should match and get higher support factors.

(SE2) *Multiple explanation* impropriety: more than one explanation is suggested by domain knowledge. This impropriety is a clue to relax a condition to lessen the support factor of a piece of knowledge that is not so important, and/or add some weak conditions to strengthen the factor of a preferable piece of knowledge.

(SE3) *Near-miss explanation* impropriety: one and only one condition is missing to explain a leaf. This impropriety is a strong clue for over-fitting. A primitive elimination action is to add the missing condition and expand the tree.

(SE4) *Twin immediate siblings* impropriety: the immediate siblings of a node belong to the same class. This impropriety is a clue for noisy examples. A primitive elimination action is to change the node between the siblings into a leaf of the siblings' class.

(SE5) *No explanation* impropriety: no explanation is given. It is a clue to generalize the conditions of a piece of knowledge that should match, and/or ask for a new piece of domain knowledge for the leaf.

The alternative actions recommended on leaves are (Dabija, Tsujino, & Nishida, 1992]:

(A01) Name the class of a leaf.

(A02) Give new examples that will arrive to the current node.

(A03) Give the domain knowledge that will explain the current node (or one of its predecessors).

(A04) Merge current node with one of its siblings.

(A05) Change the node into a leaf of its major class.

(A06) Remove the examples of the minor class.

(A07) Further separate the class of a node.

(A08) Give new attributes to separate the class of a node.

(A09) Merge two siblings and generate a subtree from the union of their examples

(A10) Merge two siblings and ignore the examples of one of them.

(A11) Generate automatically a new attribute to discriminate a pair of classes by generating a separate tree for the examples belonging only to this pair of classes.

(A12) Merge two classes into a single one and rebuild the tree.

(A13) Refine domain knowledge.

(A14) Graft a condition to node and expand the tree.

(A15) Accept the suggestion of class for node.

(A16) No action (cancel the impropriety).

References

Dabija,V.G. 1990. Learning by Experimentation in Uncertain Environments. In Proceedings of the 4th Australian Joint Conference on Artificial Intelligence, 24-37, Perth, Australia.

Dabija,V.G.; Tsujino,K.; and Nishida,S. 1992. Theory Formation in the Decision Trees Domain. *Journal of the Japanese Society for Artificial Intelligence*, 7(3).

Davis,R. 1979. Interactive Transfer of Expertise: Acquisition of New Inference Rules. *Artificial Intelligence,* 12(2):121-157.

Lenat, D.B. 1982. The Nature of Heuristics. *Artificial Intelligence*, 19(2): 189-249.

Lenat,D.B. 1983. Theory Formation by Heuristic Search. The Nature of Heuristics II: Background and Examples. *Artificial Intelligence*, 21(1-2):31-59.

Lenat,D.B. and Brown,J.S. 1983. Why AM and Eurisko Appear to Work. In Proceedings of the Fourth National Conference on Artificial Intelligence, 236-240.

Matheus,C.J., and Rendell,L.A. 1989. Constructive Induction On Decision Trees. In Proceedings of the 11-th International Joint Conference on Artificial Intelligence, 645-650.

Matheus,C.J. 1991. The Need for Constructive Induction. In Proceedings of the Machine Learning Conference, ML'91, 173-177.

Nunez,M. 1991. The Use of Background Knowledge in Decision Tree Induction. *Machine Learning*, 6(3):231-250.

Pagallo,G. and Haussler,D. 1990. Boolean Feature Discovery in Empirical Learning. *Machine Learning* 5:71-99.

Pazzani,M.J. and Kibler,D. 1990. The Utility of Knowledge in Inductive Learning, Technical Report, TR-90-18, University of California at Irvine.

Pazzani,M.J. and Brunk,C.A. 1991. Detecting and Correcting Errors in Rule-Based Expert Systems: An Integration of Empirical and Explanation-based Learning. *Knowledge Acquisition* 3(2):157-173 .

Polya, G. 1945. *How to solve it.* Princeton Univ. Press, Princeton, NJ.

Quinlan,J.R. 1986 Induction of Decision Trees. *Machine Learning* 1(1):81-106.

Tsujino,K., Takegaki,M., and Nishida, S. 1990. A Knowledge Acquisition System that Aims at Integrating Inductive Learning and Explanation-Based Reasoning. In Proceedings of the First Japanese Knowledge Acquisition for Knowledge-Based Systems Workshop,175-190.

vanMelle,W. 1980. A Domain-Independent System that Aids in Constructing Consultation Programs. Technical Report, STAN-CS-80-820, Stanford Univ.