A Two-Tiered Reasoning and Learning Architecture

Gennady Agre
Institute of Information Technologies - Bulgarian Academy of Sciences
Acad. G. Bonchev St. Block 29A, 1113 Sofia, Bulgaria
E-mail: agre@iinf.acad.bg

Abstract

The paper is an attempt to summarize the previous works of the author on integrating deductive and abductive reasoning paradigms for solving the classification task. A two-tiered reasoning and learning architecture in which Case-Based Reasoning (CBR) is used both as a corrective of the solutions inferred by a deductive reasoning system and as a method for accumulating and refining knowledge is briefly described. As illustrative examples the applications of the approach for problems of the case-based maintenance of rule-based systems and for case-based refinement of neural networks are presented.

Introduction

It has been widely recognized in AI literatures that the main problem in applying the pure deductive reasoning for classifying natural concepts is the inadequacy of representing them by context-independent, logic-style way. The approaches to solve the problem by applying the pure abductive reasoning (e.g. CBR) are based on the exemplar view for concept representation which is mainly criticized for lack of explicit meaningful definitions of the concepts to be classified. Michalski (Michalski 1990) proposed the two-tiered (TT) representation for describing such concepts which is an attempt to integrate two mentioned above extremes. A concept is described both by its ***base representation (BCR)*** explicitly defining main concept characteristics (e.g. in a form of rules) and by its inferential interpretation (ICR) (in a form of so called flexible matching procedure) implicitly defining the concept boundaries. Michalski also proposed a hybrid method for using TT representation in which the solution is searched either by applying deductive reasoning to BCR or by abductive reasoning to ICR.

This paper attempts to summarize some previous results of the author on the problem of learning the TT domain representation and to present them in a form of a two-tiered reasoning and learning architecture (TTRLA) in which CBR is used both as a corrective of the solutions inferred by a deductive reasoning system and as a method for accumulating and refining knowledge. As illustrative examples the applications of the approach for problems of the case-based maintenance of rule-based systems and for case-based refinement of neural networks are presented.

Two-Tiered Reasoning and Learning

The TTRLA structure (see Figure 1) consists of three main parts: a Deductive Reasoning System (DRS), an Inferential Concept Representation Learner (ICRL) and a Case-Based Reasoner (CB-Reasoner). It should be mentioned that we consider the problem of finding the TT domain representation as a useful metaphor for the problem of improving the classification accuracy of a deductive system. From this point of view we assume that the first tier along with ***its interpretation*** are fixed and implemented as a given DRS. Thus the problem of learning the first tier (BCR) is left out of the scope of our interest at present. A direct consequence of this is the possibility of knowledge included into the BCR to be not only incomplete but inconsistent as well.

Another consequence of the TT metaphor is a ***consequent character*** of the TT reasoning in the proposed TTRLA. The solution of a new example is initially searched by a deductive reasoning applied to the first knowledge tier. The inferred solution is considered as a general or typical one which needs to be further refined based on the system own problem solving experience accumulated during its training or/and maintenance phases . This experience along with the knowledge of how to utilize it do form the second - ICR tier of the TTRLA.

The Inferential Concept Representation Learner

To learn the second tier ICRL uses domain knowledge, training examples and the DRS problem-solving model. The tier is represented as a memory of cases

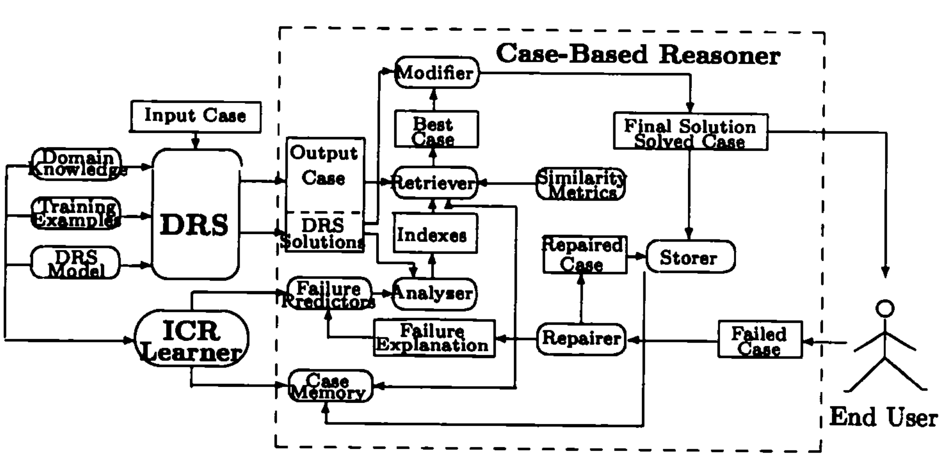

Figure 1: Two-Tiered Reasoning and Learning Architecture.

connected via indexes to the knowledge of the first tier. A case is represented as a single information structure containing the case name, a list of case features (attribute - value pairs) and the case solution. The Learner should decide i) what training examples are worth to be transformed and stored into the memory as useful past cases, ii) how the memory should be indexed to retrieve only relevant to the current situation cases and iii) how to recognize potentially "dangerous" situations in which the DRS solutions need to be modified. The Learner should also be able to choose an appropriate metrics for measuring similarity between the problem at hand and the retrieved past cases.

The Case-Based Reasoner

The Reasoner processes the example to be solved (which description may be augmented during the deductive problem solving) and its solution found by the DRS. The CB-reasoner architecture is an instance of the general architecture of a case-based planer adapted for solving the classification task (Hammond 1989; Agre 1995). The crucial points in its operation are the failure prediction, case retrieval and conflict reconciling.

Failure Prediction The process of searching a solution by means of the second tier starts only when the current situation is recognized as "dangerous", i.e. if there is a sufficiently convincing evidence that the solution inferred by DRS is wrong. This prediction process is based on comparing the DRS solution to the set of failure predictors formed by ICRL and extended by the Repairer. The roles of such predictors are played by the DRS solutions of the training examples included in the case memory (if such examples are available) and/or by the TTRLA solutions recognized as wrong by the user. In other words, the fact that the DRS has inferred once an erroneous solution is considered as a sufficiently convincing evidence for suspecting the system each time it infers the same solution of another problem. Conversely, if a DRS solution of a new problem is different from the known failure predictors, there are no reasons to "suspect" the system and its solution is accepted without any doubt.

Case Retrieval and Conflict Reconciling The success of integrated approaches combining several problem-solving paradigms crucially depends on the scheme for reconciling the conflicts between them. Most of the proposed methods (see e.g. (Golding & Rosenbloom 1991) for integration of CBR with rule-based reasoning) use a threshold scheme in which the threshold values are selected *ad hoc*. We overcome this deficiency by forming a proper set of cases to be retrieved. In this set we include not only the cases rejecting the proposed DRS solution but also the cases confirming it. In such a way the retrieval set reflects the TTRLA problem solving experience and restricts the search space only to the relevant (from the system point of view) cases.

Case-Based Maintenance of Rule-Based System

The first application of the described above TTRLA is to the task of improving a rule-base system (RBS) problem-solving behavior in the course of its operation (Agre 1995) - an important part of the adaptive maintenance task (Coenen & Bench-Capon 1993). The main characteristic of the task is that the RBS training examples are assumed to be unknown.

The First Tier

The first tier knowledge is represented as flat nonprobabilistic rules directly associating conjunctions of the problem features with problem solutions. A feature is a nominal attribute-value pair.

The RBS problem-solving behavior is modeled as a hypothesis driven process in which forward chaining is used for generation of a list of differential hypotheses (possible solutions) and backward chaining - for testing them. A hypothesis is considered to be confirmed if there is a satisfied rule having the solution as its conclusion and to be rejected if all rules leading to it have failed. The system stops its operation either if a confirmed hypothesis has been found or if all generated hypotheses have been tested and rejected.

The Second Tier

The second tier consists of cases solved by the system in the course of its operation and a case matching procedure.

Indexing Scheme Cases are indexed by all possible roles they may play in the rule-based reasoning. Each solved case may be indexed as true or false negative by each hypothesis rejected during problem solving and as true or false positive by the rule inferred the solution. A special index - untested - is used to specify a failure caused by an inappropriate application of the domain knowledge rather than by its incorrectness. Such kind of failures may occur as a result of some deficiencies in the mechanism of formation of a list of differential hypotheses or because of erroneous termination of the process of hypotheses testing.

Case Matching The case matching procedure is organized as a Nearest Neighbor algorithm using a case weighted distance $\Delta(X(C), Y) = W_X * \delta(X(C), Y)$, where:

$$\delta(X(C), Y) = \sqrt{\frac{1}{k} \sum_{j \in Att(C)} w_j \, (x_j - y_j)^2}$$

$x_j - y_j = 1$ if $x_j \neq y_j$ and $x_j - y_j = 0$ otherwise. For missing values $(x_j - y_j)^2 = \frac{1}{L_j} * (1 - \frac{1}{L_j})$, where L_j is the number of possible values of j-th attribute.

$X(C)$ denotes a case belonging to class C and Y - a new example to be classified. $x_j(y_j)$ is the value of j-th attribute of X (Y). $Att(C)$ denotes a set of relevant attribute for class C which is defined as the union of all attributes which the rules corresponding to this class refer to. In such a way we avoid the influence of any redundant (for a particular class) features. Weight w_j denotes the importance of j-th attribute of the case and is calculated as the ratio of the number of all rules containing this attribute to the whole number of rules.

Weight W_X of a stored case X is defined as a value reciprocal of the case typicality and calculated as proposed in (Zhang 1992). When more than one best case belonging to different classes are found the most typical one (i.e. with the minimum value of W) is preferred.

Selection of Cases to be Retrieved The retrieval set is determined by comparing the main characteristics of the current situation (the rule inferred the solution and the set of rejected hypotheses) with indexes connecting them to the stored cases. The set is formed by the exceptional cases rejecting the particular rule and by the cases confirming the rule. Since the set of such "confirming" cases may be empty the corresponding rule treated as a case is also retrieved.

In a situation when no rule-based solution is found all rules associated with the rejected hypotheses are retrieved along with all solved cases uncovered by the corresponding rules.

Learning the Second Tier All cases erroneously solved by the system are stored. Such cases are indexed as false positive or true negative w.r.t. the faulty system solution and false negative, true positive or untested (depending on the results of the failure analysis made by the CB-reasoner) w.r.t. the real solution of the problem. The solved case is also indexed as true negative w.r.t. all hypotheses tested and rejected by the system during problem solving session.

A case successfully solved by the system is stored only if its solution has been found as a result of reconciling a conflict between the paradigms and an identical case has not been stored. Such cases confirm the correctness of applying the concrete rule or using the concrete case to obtain the problem solution. For each newly stored case the value of weight W reflecting the case typicality is calculated.

Empirical Evaluation

The approach was implemented in the experimental system CoRCase (**C**o**r**recting **R**ules by **Case**s) and tested on two medical domains - prognosis of breast cancer recurrence (BC) and location of primary tumor (PT) - well known in ML community benchmark data bases [1]. The main characteristics of these databases are summarized in Table 1. (Both bases contain examples with missing values of some attributes).

The results were evaluated using the random subsampling strategy (Weiss & Kulikowski 1990). Each

[1]The data was prepared by M. Zwitter and M. Soklic from the University Medical Center, Institute of Oncology, Ljubljana, Slovenia.

Domain	Examples	Classes	Attrs	Vals/Attr
BC	286	2	9	5.8
PT	339	22	17	2.2

Table 1: Main characteristics of the databases used in empirical evaluation.

Domain	RBR (AQ)	TC-search	CBR	CoRCase
BC	62.0 ± 0.6	70.2 ± 2.3	70.6 ± 1.4	71.3 ± 3.0
PT	34.7 ± 4.2	33.5 ± 1.8	33.1 ± 2.0	37.0 ± 1.6

Table 2: Classification accuracy (%) of the tested problem-solving methods.

database was randomly split into two non-overlapping subsets, one for training (70% of examples) and one for testing (30% of examples). The experiments were repeated ten times for different splits and the results were averaged.

The training sets were used to induce the corresponding sets of rules. Two algorithms of different types were used to simulate the rules. The first one was an ID3-like algorithm inducing discriminating rules and the second one was an AQ-type one producing covering rules.

To evaluate the contribution of CBR and RBR to the final classification accuracy of the system four different algorithms for problem solving were tested. The first one was pure RBR. In the second algorithm the solution was searched by matching a problem at hand against the rules treated as cases. In this algorithm (named TC-search) no testing cases had been stored. The third algorithm was an incremental extension of the second one. In this algorithm each solved case had been stored and then used for searching solution of the next problem. The algorithm may be seen as a naive CBR method for problem solving with exhaustive search in the case space. And the last algorithm was an implementation of the TT reasoning method described in this section. The average accuracy and standard deviations obtained are presented in Table 2[2].

The results of the experiments indicate that the best accuracy is achieved by the proposed scheme for combination of rules and cases. It is particularly interesting that on the PT database both the TC-search and the naive CBR methods used separately have worse performance than RBR. It proves once again the effectiveness of the indexing scheme used in the proposed algorithm which allows to retrieve for matching comparatively relevant cases.

[2]See (Agre 1995) for more detailed description and evaluation of the experimental results.

Case-Based Refinement of Neural Networks

The second application of the proposed architecture was to the problem of case-based refinement of neural networks (NN) (Agre & Koprinska 1996). Our main goal was to study the potential of CBR for further improvement of a trained NN. That is why in that application only a part of the CB-reasoner architecture was used (Analyzer, Retriever and Modifier).

The First Tier

The first tier is represented as a trained NN. In order to not restrict the applicability of the proposed approach we consider the net as a "black box" described only by its input-output behavior. Thus the set of NN training examples along with the information how they have been solved by NN after completion of its training phase are used as an implicit problem-solving model of the network.

The Second Tier

The second tier is represented as a case base consisting of the NN training examples. A case is represented by a list of (normalized) attribute-value pairs and its real solution (classification). Each case is indexed both by its real and NN solutions. If an example has been correctly classified it is considered as a "typical" case, otherwise - as an "exceptional" one.

The case matching procedure is implemented as a Nearest Neighbor algorithm which uses the weighted Euclidean distance:

$$\Delta(X,Y) = \sqrt{\sum_{i=1}^{n} w_i \left(\frac{x_i - y_i}{max_i - min_i}\right)^2}$$

where n is the number of attributes describing a case, x_i and y_i stand for the values of i-th attribute for cases X and Y, and max_i and min_i - for the maximal and minimal values of i-th attribute.

The attribute weights w_i are calculated by using a variant of the *ReliefF* algorithm (Kononenko & Robnik-Sikonia 1996).

Empirical Evaluation

The approach was implemented in an experimental system **CorNCase2** (**Correction Network by Cases** - version 2). The system was tested on four well-known ML benchmark databases described by numerical attributes - glass (GL), diabetes (DB), breast cancer (BC)[3] and iris (IR) (Merz & Murphy 1996) (see Table 3).

[3]Missing values were replaced with the most frequently occurring values for the respected attributes.

Domain	Examples	Atts.	Classes	Missing (%)
BC	699	10	2	2.2
DB	145	5	3	0.0
GL	214	9	6	0.0
IR	150	4	3	0.0

Table 3: Main characteristics of the databases used in the empirical evaluation.

Domain	TBRBF	1NN	ReliefF-1NN	CorNCase2
BC	95.5 ± 1.2	95.6 ± 1.1	95.8 ± 1.2	96.3 ± 1.2
DB	91.9 ± 4.1	93.9 ± 4.4	94.4 ± 3.3	92.3 ± 4.0
GL	70.2 ± 4.6	69.5 ± 6.5	70.6 ± 6.2	75.0 ± 6.8
IR	97.1 ± 2.3	95.3 ± 1.9	96.4 ± 2.1	97.3 ± 1.7

Table 4: Classification accuracy (%) on the test data bases used in the experiments.

Each database was randomly split into two non-overlapping subsets, one for training (70% of examples) and one for testing (30% of examples). The experiments were repeated ten times for different splits and the results were averaged.

The TB-RBF system (Kubat & Ivanova 1995) was selected as an instance of NN. TB-RBF uses domain knowledge in the form of decision trees (or rules) to define the topology of a radial basis function network. Each class is mapped to an output node, each attribute - to an input node and the hidden nodes correspond to the branches of the tree. To evaluate the CorNCase2 behavior we also tested weighted (ReliefF-1NN) and unweighted variants of the Nearest Neighbor (1NN) algorithm working on the TB-RBF training set. The average accuracy and standard deviations obtained are presented in Table 4.

It can be seen that CorNCase2 outperforms TB-RBF in all databases. Moreover, in three of them (BC, GL and LD) the proposed integrated method outperforms both problem-solving paradigms used in the integration. These results prove the effectiveness of the proposed scheme for indexing and retrieving cases.

Discussion and Related Works

The idea of TT concept representation proposed by Michalski was further elaborated (Zhang 1991; Bergadano *et al.* 1992) and applied not only for rules but also for decision trees (Kubat 1996) and neural networks (Sun 1995). From this point of view the proposed architecture may be seen as a tool for incremental learning the TT domain representation. The main differences from other methods solving similar tasks are as follows:

First, in our approach there is no need to learn the first tier, as a consequence it may contain not only incomplete but incorrect knowledge as well.

Second, in contrast to some other approaches using rules for representing the first tier our case matching procedure does not need the training examples the rules were generated from. The proposed similarity measure uses only the domain knowledge encoded into the rules. This allows to apply the method to improve performance of classification rules independently on the way of their construction.

Third, the approach is incremental enabling to refine the domain description by storing some of newly solved cases as a part of the second tier. These cases along with the indexing information reflecting the roles they played in the problem solving, may be further used for real refinement of the first tier.

The proposed architecture may be considered in a broader context of hybrid systems integrating different problem solving and learning paradigms (e.g. (Plaza *et al.* 1993; Domingos 1996) etc). From this point of view the main characteristics of the approach are the consequent organization of integrated reasoning in which CBR is used for correcting solutions inferred by a deductive system. The method for conflict reconciling may be considered as an extension of the one proposed in (Golding & Rosenbloom 1991) and based on an available model of DRS. The necessity of using heuristically defined threshold values playing an important role for determining the CBR solution in such kind of methods is avoided by a proper selection of the retrieval set containing cases not only rejecting a deductively inferred solution but confirming it as well.

Finally, the architecture may be viewed as an approach for building a new generation knowledge-based systems (KBS). In the current stage of the information technology the great part of the international experience in the KBS development is available in an electronic form as i) databases with unexplained examples of problems solved in a concrete domain; ii) knowledge acquired during building a concrete KBS iii) KBS shells intended to solve some fixed task(s) based on a concrete formalism of knowledge representation and a model of knowledge processing (KBS model). The availability of such knowledge components radically changes the modern technology of KBS development. The emphasis now is on analyzing, identifying and adapting the already developed components rather then on the development of a KBS "from scratch".

The proposed architecture allows us re-using available knowledge components by adapting them to an implicit requirement for advanced KBS - a possibility to operate in a real (open) environment.

Conclusion and Future Work

In this paper a two-tiered reasoning and learning architecture is presented. The integrated reasoning process is sequential - the initial solution is inferred by a deductive reasoning system which plays a role of the first tier, and then this solution is modified (when necessary) by the second tier designed as a case-based reasoning system. Both tiers are connected by an indexing schema based on an available problem-solving model of the first tier. Two applications of the architecture - for improving the classification behavior of a rule-based system and a neural network - are briefly described.

The problem of providing a more tight integration of neural networks with CBR is still one of the future directions of our work. The solution will be searched for exploring more sophisticated problem-solving model of a NN. Regarding the classification on problems for multiple classifier combination (Xu, Krzyzuk, & Suen 1992) our current approach has solved the first type combination problem i.e. a combination based on the abstract level of classifier output information. In the future we intend to design an integration, based on the classifier output information of the measurement level (the combination problem of the third type).

Acknowledgment

This work has been partially supported by National Science Foundation grant number I-606/96. The author thanks Dr. Irena Koprinska for providing the results of the experiments with TB-RBF system.

References

Agre, G., and Koprinska, I. 1996. Case-based refinement of knowledge-based neural networks. In *Proceedings of the International Conference "Intelligent Systems: A Semiotic Perspective "*, volume II: Applied Semiotics, 221–226.

Agre, G. 1995. KBS maintenance as learning two-tiered domain representation. In *Lecture Notes in Artificial Intelligence*, volume 1010. Springer. 109–120.

Bergadano, F.; Matwin, S.; R.S., M.; and Zhang, J. 1992. Learning two-tiered description of flexible concepts: the POSEIDON system. *Machine Learning* 8:5–43.

Coenen, F., and Bench-Capon, T. 1993. *Maintenance of Knowledge-Based Systems: Theory, Techniques and Tools*. Academic Press.

Domingos, P. 1996. Unifying instance-based and rule-based induction. *Machine Learning* 24(2):141–168.

Golding, A., and Rosenbloom, P. 1991. Improving rule-based systems through case-based reasoning. In *Proceedings of the National Conference on Artificial Intelligence*, 22–27. Anaheim: MIT Press.

Hammond, K. 1989. *Case-Based Planning: Viewing Planning as a Memory Task*. Academic Press.

Kononenko, I., and Robnik-Sikonia, M. 1996. Relieff for estimation and discretization of attributes in classification, regression and ILP problems. In Ramsay, A., ed., *Artificial Intelligence: Methodology, Systems, Applications*, 31–40. IOS Press.

Kubat, M., and Ivanova, I. 1995. Initialization of RBF networks with decision trees. In *Proceedings of the 5th Belgian-Duch Conference on ML*, 61–70.

Kubat, M. 1996. Second tier for decision trees. In Saitta, L., ed., *Proceedings of the Thirteen International Conference on Machine Learning*, 293–301.

Merz, C., and Murphy, P. 1996. UCI repository of machine learning databases.

Michalski, R. 1990. Learning flexible concepts: Fundamental ideas and a method bases on two-tiered representation. In Kodratoff, Y., and Michalski, R., eds., *Machine Learning*, volume III. Morgan Kaufmann Publ. 63–111.

Plaza, E.; Aamodt, A.; Ram, A.; van de Velde, W.; and van Someren, M. 1993. Integrated learning architectures. In Brazdil, P., ed., *Proceedings of Machine Learning: ECML-93*, 429–441. Springer-Verlag.

Sun, R. 1995. Robust reasoning: integrating rule-based and similarity-based reasoning. *Artificial Intelligence* 75:241–295.

Weiss, S., and Kulikowski, C. 1990. *Computer Systems That Learn*. Morgan Kaufmann.

Xu, L.; Krzyzuk, A.; and Suen, C. 1992. Methods of combining multiple classifiers and their application to handwriting recognition. *IEEE Transaction on Systems, Man, and Cybernetics* 22(3):418–435.

Zhang, J. 1991. Integrating symbolic and subsimbolic approaches in learning flexible concepts. In Michalski, R., and Tecuci, G., eds., *Proceedings of the First International Workshop on Multistrategy Learning*, 289–304.

Zhang, J. 1992. Selecting typical instances in instance-based learning. In Sleeman, D., and Edwards, P., eds., *Machine Learning - Proceedings of the Ninth International Workshop*, 470–479. San Mateo, CA: Morgan Kaufmann.