Canadian AI Conference, Workshop on What is Inductive Learning?
Toronto, Ontario, May 1996.

# Inductive Learning and Case-Based Reasoning

Igor Jurisica[1]
Department of Computer Science
University of Toronto, Toronto, ONT M5S 1A4
Tel.: (416) 978-7589, email: juris@ai.utoronto.ca

**Abstract**

This paper describes an application of an inductive learning techniques to case-based reasoning. We introduce two main forms of induction, define case-based reasoning and present a combination of both. The evaluation of the proposed system, called ***TA3***, is carried out on a classification task, namely character recognition. We show how inductive knowledge improves knowledge representation and in turn flexibility of the system, its performance (in terms of classification accuracy) and its scalability.

## 1. Introduction

Inductive learning is a process of generalizing specific facts or observations [MCM86]. It is a basic strategy by which one can acquire knowledge. There are two main forms associated with inductive learning:

1. **Instance-to-class induction,** where the learning system is presented with independent instances, representing class and the task is to induce a general description of the class.
2. **Clustering** problem arises when several objects or situations are presented to a learner and the task is to find classes that group objects usefully.

A **case-based reasoning** [Kol93] relies on the idea that similar problems have similar solutions. Facing a new problem, a case-based system retrieves similar cases stored in a case base and adapts them to fit the problem at hand. Informally, a case comprises an input (the problem), an output (the solution) and feedback (an evaluation of the solution).

Considering the case-based reasoning cycle (storing, retrieving and adapting cases), one can say that the more similar the cases are the less adaptation is necessary and, consequently, the more accurate the solution is. An important task is to measure the case relevance to a given problem, i.e., similarity or closeness between the problem and a case in a case base. The similarity measure should be flexible enough to allow for retrieving only highly relevant cases, even if the user preferences are changed or a different task is under consideration. Usually, it is better to retrieve fewer cases, or none, than to retrieve less useful cases that would result in a poor solution.

The **classification task** involves associating cases to particular classes. Based on the case description, a classification system determines whether a given case belongs to a specified class. In general, this process consists of: (1) generating a set of classes and (2) classifying cases into the created classes.

[1] This research was supported by the Information Technology Research Centre of Ontario. Author is indebted to Janice Glasgow and John Mylopoulos for helpful comments and suggestions.

If the classes are known from the domain knowledge then the process of associating instances with classes is called a classification process. If the system is supposed to generate classes then we use the term (conceptual) clustering [SM86].

Various techniques have been utilized for the classification task, including neural networks [SMT91], genetic algorithms [FS91], inductive and instance-based learning [Tur95,SG95] and case-based reasoning [PBH90,AB94]. Individual approaches are comparable based on the method they deploy, the accuracy they achieve and the complexity of the algorithm used. Classification rules are usually extracted from training examples during a learning process. These rules are then used for the classification of unseen instances. Case-based systems, however, store whole cases during the learning process and use the similarity between a given problem and a case in a case base to determine a proper class for the problem [Jan94,GB95]. Clustering algorithms can be divided into two basic approaches: (1) Traditional numeric methods, where classes are created by minimizing similarity of objects in different classes and maximizing inter-class similarity. Usually, Euclidean measure of distance is used. (2) Conceptual clustering methods which use structured data and they determine clusters not only by attribute similarity but also by conceptual cohesiveness.

Inductive learning can be deployed in case-based reasoning systems in two ways:

1. To produce an induction tree which can be used during retrieval to speed up the process of retrieving similar cases.
2. To construct important features that characterize classes which allows for more precise retrieval.

Next, we shall introduce the ***TA3*** CBR system, describe a combination of CBR and inductive learning and present an evaluation of ***TA3*** on the character recognition task.

## 2. The *TA3* System

The ***TA3*** case-based reasoning system is intended for intelligent decision support tasks. As in any other knowledge-based system, it is desired that knowledge is represented in a form appropriate for the intended user (either novice or expert user) and that the representation is rich enough to allow for complex, yet efficient processing. In general, a case corresponds to a problem, solution and feedback. Cases in ***TA3*** are represented as structured objects with relations [MBJK90]. Thus, a case is a finite set of attribute/value pairs with relations defined both between individual attributes and between cases. In addition, attributes can be grouped into one or more categories [Jur94]. This grouping allows for ascribing different constraints to different groups of attributes, based on their relevancy to a given problem. It allows for more flexible reasoning as well as for more comprehensible presentation of complex information. As a result, attributes irrelevant for a given problem may be ignored during retrieval and similarity assessment which enhances quality of the result. However, there is an obvious problem, namely, how to identify appropriate categories.

During case-based reasoning process, a partial equality exists if attribute values of a case $C_1$ match to only a subset of values of a case $C_2$. In order to control the partial matching, we define **context**. The goal is to retrieve not only exact matches (equivalent cases) but partial matches (similar cases) as well. In short, context is a parameter of **similarity relation** which maps a case base onto a set of similar (in terms of context) cases. Context is a view of a case, i.e., a case can be interpreted in a given context. This allows for controlling what can and what cannot be considered as a partial match. More formally, a context is defined as a finite set of attributes with associated constraints on the attribute values. Context specifies important attributes and how "close" attribute value must be. Context and similarity allows for uncertainty representation – by allowing for imprecise and incomplete specification of the problem and by supporting not only exact but similar matches as well. We say that a case satisfies a particular context, if for each attribute specified in the context, the value of that attribute in the case satisfies the constraint. Thus, the matching process can be described as a constraint satisfaction process [THNG90,Gaa93].

One of the resulting problems is selecting a method for specifying a context, i.e., specifying which attributes are needed for accurate classification, and what range or similarity should be allowed for attribute values (i.e., what constraints should be posed on values). The system should discover appropriate interpretation for input cases, so that case-based reasoner retrieves only highly relevant cases. There are several possible scenarios for creating the initial context. Their advantages/disadvantages depend on the actual implementation, as discussed in [JG96,JMG96]. Using context allows for diminishing the problem of instance-based learning algorithms [AB94], namely performance degradation with increased number of irrelevant attributes.

Pattern **discovery** in ***TA3*** involves collecting cases sharing some features, for example using an IVF case [JS95], all women that have pregnancy with complications are in their third or fourth cycle. However, the pattern identification is not sufficient – patterns need to be described: given a set of cases labeled by class.

In [Ort95] it was shown that *m_of_n* matches allow for improved performance if *m* is reasonably selected. Our approach of representing cases as sets of categories, each comprising a set of tuples, allows for multiple levels of *m_of_n* matching. Thus, important attributes may require *n_of_n* matches for a given category, and less important attributes may allow for *(n-k)_of_n* matches. The problem is to find these attribute groupings.

***TA3*** allows for discovering useful interpretation of cases. This helps during problem solving for improving prediction accuracy as well as scalability. The function for discovering context is an attribute-oriented clustering algorithm. The function maps a set of cases and a case base into a context that guarantees the relevance of cases, i.e., organizes the case base into classes of semantically similar cases – equivalence classes. Since various organizations are possible in general, the system supports controlled explanation: A user-specified coverage *T* is used as a termination criteria. *T* specifies the percentage of cases that have to be relevant in order to accept particular context. The task is to find the minimal context which satisfies the given threshold *T*. The algorithm uses factor analysis [Har76], discriminant analysis [Lac75] and is similar to the *CDP* algorithm [AIS93] as described in [JMGSC96].

The system uses factor analysis [Har76] to locate relevant attributes (characteristic features). Factor analysis is a statistical technique used to identify a reasonable small number of factors that can be used to represent relationships among sets of interrelated variables. After cases can be described by their characteristic features, the case base is dynamically organized into particular classes.

If there is a high variability of values for a particular attribute, then the attribute is not a good predictor for a particular class. Low variability of attribute values suggests a predictive feature. When all attributes are processed, the system groups similar attributes into categories, i.e., predictive attributes are separated from attributes unsuitable for classification. This technique is simple enough that it scales up for large case bases, yet it allows for improved reasoning as will be shown in the next section. This organization also allows for reducing the negative effect of irrelevant attributes on the performance. It is expected to see other improvements by applying discovered knowledge during performance evaluation: (1) appropriate grouping of attributes improves *m_of_n* and (2) knowledge about values helps the similarity-assessment algorithm to decide which values can be considered "close" to each other.

## 3. Evaluation

Case-based classification assigns new problems (cases without a solution) to classes, based on their similarity to cases with known class membership. In ***TA3***, given a case base CB, a set of relevant features with constraints assigned to them (context CTX) and a problem P, ***TA3*** returns a pair <P, $class_i$> if and only if there exist a case C in CB, such that P is more similar to C than to other cases in a case base, and that C belongs to a $class_i$ :

$$\mathit{classify}\ (P_{CTX}) \rightarrow \langle P_{CTX}, class_i \rangle \leftrightarrow \exists\, C \in CB : \mathit{similar}(P_{CTX}, C_{CTX}) \wedge \langle C, class_i \rangle.$$

It is important to note that both P and C are first interpreted in a given context, denoted $P_{CTX}$ and $C_{CTX}$, and only afterwards they are compared for similarity. For different contexts, different cases may be similar to a given problem. Clearly, the context can be used as a form of bias, since its modifications can control quality and quantity of retrieved cases and thus classification result. If more cases are similar to a problem and they belong to different classes, the closer match is preferred, i.e., the match were for a given context, the value differences on individual attributes are smaller.

We have evaluated ***TA3***'s performance on several domains [JG95,JS95,Jur95,JG96,JMG96,JMGSC96]. Here we report on the letter classification task, previously used for testing machine learning algorithms [FS91,Aha92].[2] The data set consists of 20,000 instances, described by 17 attributes. Based on simple cross-validation, the basic ***TA3*** system achieved comparable classification accuracy to other systems. Attributes were grouped into categories for selective cardinality relaxation/restriction, i.e., selective *m_of_n* matching [Ort95]. We have tested several relaxation strategies and attribute-grouping approaches that yielded various accuracy results, using 95% confidence intervals. We have also tested the effect of the case base size on the classification accuracy achieved for individual situations.

| ***Method*** | ***Accuracy (20K cases)*** | ***Accuracy (2K cases)*** |
|---|---|---|
| Back-propagation | -- | 91.9 ± 0.6% |
| IB1 | 95.7 ± 0.4% | 81.7 ± 0.7% |
| CN2 | 87.9 ± 0.8% | 68.7 ± 1.0% |
| C4 | 86.4 ± 0.7% | 67.4 ± 0.8% |
| Genetic Algorithm | 82.7 | -- |
| ***TA3_1*** | 90% ± 0.1461 | 80% ± 0.1461 |
| ***TA3_2*** | 100% | 85% ± 0.1742 |
| ***TA3_3*** | 100% | 95% ± 0.1001 |
| ***TA3_4*** | 100% | 100% |

In ***TA3_1*** we used a modified nearest-neighbor approach – attributes were grouped into categories for selective cardinality relaxation/restriction. Simple instance-based learning algorithms achieved similar classification accuracy as ***TA3_1***. ***TA3_2*** was obtained by using a value relaxation first (i.e., relaxing attribute values to include values of immediate neighbors). If more cases were needed, cardinality relaxation was performed. ***TA3_3*** was obtained by using a combination of previous approaches with equal voting for each of them. ***TA3_4*** was obtained by using different category groupings for attributes. In the previous test, categories

---

[2] Letter recognition data and classification accuracy obtained by other machine learning algorithms are available at ftp:ics.uci.edu:pub:/machine-learning-databases/letter-recognition.

were created sequentially, each group having four attributes (letter attribute was an extra category). In the latter test, we used an induction to find out the most relevant attributes in recognizing individual letters and we grouped attributes accordingly. The process can be described as conceptual clustering, since the algorithm generates clusters of cases based on their important features. Induced features can then be used during case retrieval, which would improve the performance, provided that the future cases will have same important features.

Similarly, in [JG95,JS95] we report on how ***TA3***'s representation scheme, context-based similarity assessment and inductive learning allow for flexible computation. By giving or limiting computational resources in [JG95] the precision of the results is controlled – if enough time is available for computation, more accurate result can be returned. In [JS95,Jur95], the quantity and quality of retrieved items is controlled analogously and the system supports both recall-oriented retrieval (the system retrieves all relevant items, even is some irrelevant are included) and precision-oriented retrieval (the system retrieves only relevant items, even is some relevant items are not found).

## 4. Conclusions

This paper presents an approach of combining inductive learning and case-based reasoning. Applying inductive knowledge improves both task and time performance of CBR. By considering only relevant attributes during problem solving, more accurate results are provided and the reasoning process is more efficient. As an evaluation of the proposed system, we have compared standard CBR approach with that using an inductive learning on the classification task, namely character recognition. The results show how induction improves the representation of cases and in turn system's performance. These results complement previous experience reported in [JMG96,JMGSC96], where scalability of ***TA3***'s incremental context modification algorithm, used during retrieval, was evaluated. In the future, we plan to apply ***TA3*** to more complex tasks and more structured domains.

## 5. References


[AB94] Aha, D. W. and Bankert, R. L. Feature selection for case-based classification of cloud types: An empirical comparison. In *AAAI-94 Workshop on Case-Based Reasoning*, Seattle, WA, 1994.

[Aha92] Aha, D. W. Generalizing from case studies: A case study. In Proc. of the *9th International Conference on Machine Learning*, pages 1-10, Aberdeen, Scotland, 1992.

[AIS93] Agrawal, R., Imielinski, T., and Swami, A. Database mining: A performance perspective. *IEEE Transactions on Knowledge and Data Engineering*. Special issue on Learning and Discovery in Knowledge-Based Databases, 5(6):914-925, 1993.

[FS91] Frey, P. W. and Slate, D. J. Letter recognition using Holland-style adaptive classifiers. *Machine Learning*,6(2), 1991.

[Gaa91] Gaasterland, T. Restricting query relaxation through user constraints. In Proc. on International Conference on Intelligent and Cooperative Information Systems, pages 359-366, Rotterdam, The Netherlands, 1993.

[GB95] Griffiths, A. D. and Bridge, D. G. Inductive bias in case-based reasoning systems. Technical Report YCS 95/259, University of York, York, UK, 1995.

[Har76] Harman, H. *Modern Factor Analysis*. University of Chicago Press, Chicago, IL, 1976.

[Jan94] Jantke, K. P. Nonstandard concepts of similarity in case-based reasoning. In Proc. of the *17th Annual Conference of the Gesellschaft fur Klassifikation*., Kaiserslautern, Springer Verlag, 1994.

[JG95] Jurisica, I. and Glasgow, J. Applying case-based reasoning to control in robotics. In *3rd Robotics and Knowledge-Based Systems Workshop*, St. Hubert, Quebec, 1995.

[JG96] Jurisica, I. and Glasgow, J. Case-based classification using relevance assessment in the ***TA3*** system.1996. Submitted.

[JMG96] Jurisica, I., Mylopoulos, J. and Glasgow, J. Performance analysis of an incremental algorithm for the CBR system. 1996. Submitted.

[JMGSC96] Jurisica, I., Mylopoulos, J., Glasgow, J., Shapiro, H. and Casper, R.F. Case-based reasoning in IVF: Prediction and knowledge mining. *AI in Medicine. Special issue on CBR in Medicine* , 1996. Submitted.

[JS95] Jurisica, I. and Shapiro, H. A computer model for case-based reasoning in IVF. In *The 51st Conference of the American Society for Reproductive Medicine* , Seattle, Washington, 1995.

[Jur94] Jurisica, I. How to retrieve relevant information?. In. *Proc. of the AAAI Fall Symposium Series on Relevance*., New Orleans, Louisiana, pp 101—104, 1994. Extended version available as the Technical Report, DKBS-TR-94-5, University of Toronto, Department of Computer Science.

[Jur95] Jurisica, I. A similarity-based retrieval tool for software repositories. *In The 3rd Workshop on AI and Software Engineering: Breaking the Mold. IJCAI-95* , Montreal, Quebec, 1995.

[Kok95] Kokeny, T. Constraint satisfaction problems with order-sorted domains. *International Journal on Artificial Intelligence Tools*. Vol. 4, Nos. 1-2, pp. 55--72, 1995.

[Kol93] Kolodner, J. L. *Case-Based Reasoning*. Morgan Kaufmann, San Mateo, CA, 1993.

[Lac75] Lachenbruch, P. *Discriminant Analysis*. Hafner Press, New York, NY, 1975.

[MBJK90] Mylopoulos, J., Borgida, A., Jarke, M., and Koubarakis, M. Telos: Representing knowledge about information systems. *ACM Transactions on Information Systems*, 8(4):325-362, 1990.

[MCM86] Michalski, R., Carbonell, J., and Mitchell, T., editors. *Machine Learning. An artificial intelligence approach*, Volume 2. Morgan Kaufmann Publishers, Los Altos, CA, 1986.

[Ort95] Ortega, J. (1995). On the informativeness of the DNA promoter sequences domain theory. *Journal of Artificial Intelligence Research* , 2:361-367. Research Note.

[PBH90] Porter, B. W., Bareiss, E., and Holte, R.C. (1989). Knowledge acquisition and heuristic classification in weak-theory domains. Technical Report AI89-96, AI Laboratory, University of Texas, Austin.

[SG95] Schuurmans, D. and Greiner, R. Learning to classify incomplete examples. *In Computational Learning Theory and Natural Learning Systems: Addressing Real World Tasks*. MIT Press, 1995.

[SM86] Stepp, R. E., Michalski, R.S. Conceptual clustering: Inventing goal-oriented classifications of structured objects. In [MCM-86], pp. 471-498, 1986.

[SMT91] Shavlik, J., Mooney, R., and Towell, G. Symbolic and neural learning algorithms - An experimental comparison. *Machine Learning*, 6(2):111-143, 1991.

[THNG90] Thagard, P. R., Holyoak, K. J., Nelson, G., and Gotchfeld, D. Analog retrieval by constraint satisfaction. *Artificial Intelligence* , 46:259-310, 1990.

[Tur95] Turney, P. D. Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm. *Journal of Artificial Intelligence Research* , 2:369-409, 1995.