Data Mining II, C.A. Brebbia & N.F.F. Ebecken (Editors)
© 2000 WIT Press, www.witpress.com, ISBN 1-85312-821-X

A visual data mining tool to support cooperative learning

J.G. le Roux & H.L. Viktor
Department of Informatics, University of Pretoria, South Africa.

Abstract

A cooperative data mining team consists of two or more diverse data mining tools that cooperate to learn. The data mining tools learn from one another by accessing the other team member's knowledge, generating new data and thus adapting its own perspective of the problem at hand. By combining the complementary learning approaches followed by diverse tools, new insights into the problem domain are obtained.

This paper describes the ViziMine cooperative data mining tool that acts as a personal assistant, assisting domain experts to explore and increase their expertise regarding a problem domain. The tool provides a uniform interface that incorporates a number of data mining tools that express their knowledge as rules. Results indicate that the approach successfully facilitates the cooperative learning process and eases the transfer of knowledge between learners. This is especially evident in an environment where the human expert actively participates in cooperative learning.

1 Introduction

When presented with exactly the same set of training examples, two data mining tools seldom produce exactly the same sets of concept descriptions. This is due to the *a priori* assumptions that each tool makes when learning. The *a priori* assumptions are called the inductive bias of an approach [6]. If data mining tools can cooperate during the learning process, the methods can utilize the diverse results to produce fused sets of concept descriptions that surpass the individual results.

This paper introduces the ViziMine cooperative learning tool in which two or more data mining tools, modeled as learners, co-exist in a cooperative learning team. Cooperation involves the active improvement of all the

Data Mining II, C.A. Brebbia & N.F.F. Ebecken (Editors)
© 2000 WIT Press, www.witpress.com, ISBN 1-85312-821-X

members' individual results. This is facilitated by means of a rule combination, data generation and rule pruning algorithm. In this way, the team members learn from one another, obtaining new insights into the problem at hand.

The ViziMine tool provides a uniform interface, thus acting a personal assistant to the human user aiding the human to improve his perspective of the problem domain [5]. The tool is used to model the transfer of knowledge through cooperative learning. This is especially useful in an environment where the human expert acts as a learner that actively participates in the cooperative learning team.

The ViziMine tool consists of two components, namely the cooperative data mining team back-end, together with the graphical user interface. Section 2 introduces the cooperative learning team concept by discussing the team member's architecture and the learning program followed by each learner. Section 3 describes the ViziMine interface and Section 4 concludes the article. The illustrative example used throughout the paper concerns the well-known Iris benchmarking problem. Three types of Irises (Versicolor, Setosa or Virginica) are classified by considering their sepal-length, sepal-width, petal-length and petal-width.

2 Cooperative data mining team

The Oxford dictionary defines cooperation as *"to act or work with others for a common purpose"*. In a data mining environment, cooperation involves the co-existence of two or more data mining techniques that work together to solve the data mining problem at hand.

This section describes the learners that form part of a cooperative learning team [9]. Section 2.1 describes the architecture of each cooperating data mining technique or *learner*. This is followed, in Section 2.2, by a description of the learning program followed by each learner.

2.1 Learner architecture

A learner consists of five components, as depicted in Figure 1. The **learning element** contains the data mining tool. In the current implementation, the learning element contains one of the following approaches: The CN2 rule induction program [3], the C4.5 decision tree and rule generation program [7] or the ANNSER approach that extracts rules from artificial neural networks [8]. The **performance element** monitors and guides the progress of the learning element. It controls the learning process and is thus responsible for improving the output of the learning element. The task of the **critic** is to evaluate the learner's results against a set of predetermined quality measures. The **data generator** is used to generate new examples that are based on the rules obtained from other learners. These examples are used for future training. Each learner contains a **knowledge-base** that is used

Data Mining II, C.A. Brebbia & N.F.F. Ebecken (Editors)
© 2000 WIT Press, www.witpress.com, ISBN 1-85312-821-X

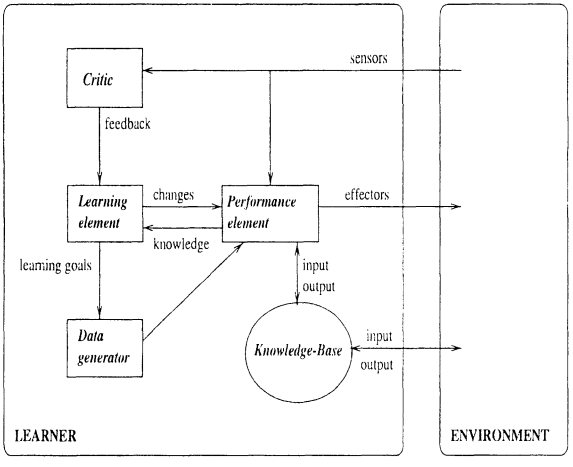


Figure 1: The learner architecture

for inter-learner communication. A learner accesses the current knowledge of another by querying the other learner's knowledge-base. Similarly, it responds to the queries of other learners by reporting the requested knowledge as contained in the original knowledge-base.

2.2 Learning program

Each member of the data mining team completes the learner program, which consists of two phases. The first phase involves **individual learning**. All of the team members use the same training set to form an individual set of rules that is placed in its individual knowledge-base. The individual learning process is controlled by the performance element. Learner A receives a preprocessed training data set from the environment. The learning element, through a process of parameter adaptation, produces sets of initial rules. The critic evaluates the quality of each of these rule sets and the best rule set is selected and is placed in the individual knowledge-base. This rule set has the highest overall accuracy against the previously unseen test data set.

At the completion of the first phase, the team members set a rule accuracy threshold. This threshold is used for cooperative learning. Here, the team members calculate the average rule accuracy of their rules against the test data set. Rules with an accuracy equal to or higher than the threshold are considered to be of a high quality. Each learner removes the low quality rules from its knowledge-base and initiates the cooperative learning phase. The aim of the second **cooperative learning** phase is the creation of individual high quality rule sets together with a team knowledge-base that

© 2000 WIT Press, www.witpress.com, ISBN 1-85312-821-X

contains the high quality rules produced by the individual learners. The quality of the results of cooperative learning is tested against a validation set.

The cooperative learning phase consists of the following steps.

Step 1. Querying the knowledge base. Each learner queries the knowledge-bases of the other team members to obtain the high quality rules that it may have missed. These rules are compared with the rules contained in its rule set. In this way, a NewRule list of relevant rules is produced.

Step 2. Rule comparison. The NewRule list is compiled by identifying the following relationships between the rules [9], where R1 denotes a high quality rule contained in the learner's rule set and R2 denotes a high quality rule created by another team member:

Two rules R1 and R2 are *similar* if two conditions hold. Firstly, the training examples covered by rule R1 is a subset of those covered by R2. Secondly, the rules contain the same attributes with similar values. *For example, the attribute-value test (petal-width $>$ 49.0) is similar to the test (petal-width $>$ 49.5).* If R1 and R2 are similar, it implies that the learner has already acquired the knowledge as contained in rule R2 and that R2 should not be added to the NewRule list.

Two rules *overlap* when they contain one or more attribute-value tests that are the same. *For example, rule Rl with attribute-value tests (petal-length $>$ 7.5) and (petal-width $<$ 47.5) and rule R2 with attribute-value tests (petal-length $>$ 7.5) and (sepal-length $<$ 35.5) overlap.* Rule R2 is placed on the learner's NewRule list.

A rule R2 *subsumes* another Rl if and only if they describe the same concept and the attribute-value tests of R2 is a subset of that of rule Rl [2]. In other words, rule R2 is more general than R1. If R2 subsumes rule Rl, it is placed on the NewRule list.

Step 3. Rule combination. The rules as contained in the NewRule list are used to form new rules. The attribute-value tests of the rules are combined with the attribute-value tests of the rules in the learner's rule set to form a new set of rules. Each of these new rules is evaluated against the test set. Those new high quality rules, which are not similar to, overlap with, or are subsumed by existing rules, are retained on the NewRule list. These rules act as input to the data generation step.

Step 4: Data generation. The data generator uses each of the rules in the NewRule list to generate a new set of training instances [9]. The newly generated training instances are added to the original training set and the learner re-iterates the individual learning phase. In this way, a new training set that is biased towards the particular rule is generated. This process is constrained by ensuring that the distribution of the data, as contained in the original training set, is maintained. A detailed description of this process is beyond the scope of this paper. Interested readers are referred to [9] for a detailed description of the process.

Steps 1 to 4 are re-iterated until no new rules can be generated. Next,

the pruning algorithm is initiated.

Step 5: Pruning. The individual rule set may contain duplicates, rules that are too specific, or rules that are redundant with respect to the classification of concepts. For example, a rule containing three attribute-value tests may be reduced to a rule containing one test without any reduction in its accuracy. The reduced error pruning (REP) scheme, as discussed by [6] and [7], is applied to the rule set. Results indicate that this method achieves high quality results in terms of overall quality [6, 9]. This pruning algorithm removes redundant attribute-value tests and rules.

Finally, the resulting rule set is placed in the learner's knowledge-base and a team knowledge-base is constructed. The individual rule sets are combined into one and pruned using the REP approach. The results of this phase are individual rule sets that contain the knowledge as derived from combining the individual learner's expertise, together with the team knowledge-base that contains the fused results of cooperative learning. Finally, the quality of these individual and team results are evaluated against a validation data set that contains previously unseen examples. The next section shows how the cooperative learning process is visualized by means of the ViziMine interface.

3 ViziMine interface

The main aim of the ViziMine interface is to illustrate the cooperative data mining process and to provide a tool used by the domain expert to incorporate his knowledge into the system.

The ViziMine interface consists of three components. The main learning team interface provides the user with a detailed description of the current state of each team member. This interface also allows the domain expert to incorporate his knowledge into the system by participating as a learning team member. The rule building interface visualizes the cooperative learning phase as described in Section 2. The third component of the tool provides an option to visualize the data by means of a scatter diagram.

3.1 Main cooperative learning team interface

The main interface contains the details regarding the learning team members, as depicted in Figure 2. The figure shows a learning team consisting of three learners.

Note that the human expert may be explicitly modeled as part of the learning team. His knowledge is added to the system by the creation of a new Expert learner, using the (*Learner Add Expert*) option. The user is subsequently prompted to enter a set of rules that is placed in the Expert learner knowledge-base, to be used during cooperative learning.

The team members are indicated by buttons located on the left hand side of the window. The information regarding the current state of a learner

Data Mining II, C.A. Brebbia & N.F.F. Ebecken (Editors)
© 2000 WIT Press, www.witpress.com, ISBN 1-85312-821-X

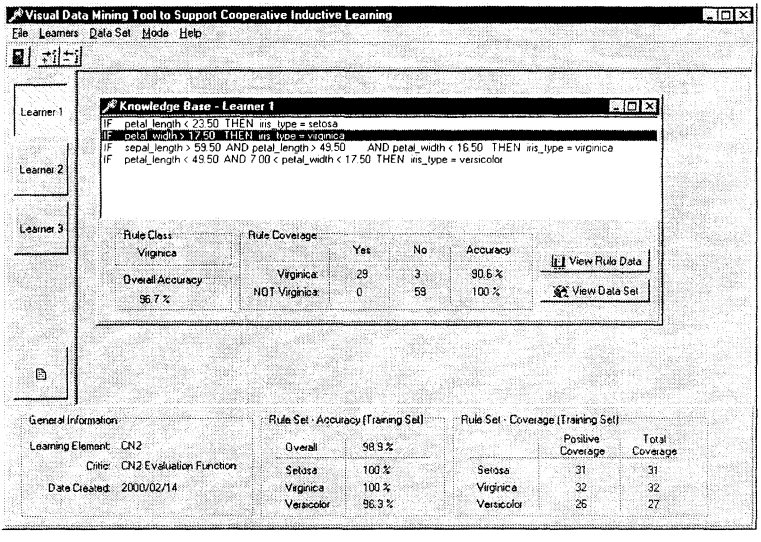


Figure 2: The ViziMine user interface containing the team member's current information

is displayed by selecting the appropriate button. The windows regarding more than one team member are tiled for easy accessibility to the team's knowledge.

The active learner is described as follows: The lower portion of the window shows the contents of the learning element, the critic and the date created. The rule set accuracy and coverage denotes the quality of the rule set. This information depicts the overall accuracy of the rule set against the particular data set under consideration, together with the individual accuracies against the output classes. *For example, the figure shows that the rule set is 98.9% accurate against the training set. The classification of the Setosa and Virginica Irises are 100% correct and that the classification of the Versicolor Irises are 96.3% accurate. The coverage information describes the number of cases covered by the rule set. In the figure, 26 of the 27 Versicolor Irises are correctly covered.* The accuracy and coverage of each of the rules as contained in the knowledge-base are also displayed. *For example, the second rule was 96.7% accurate and misclassified 3 of the 29 Virginica Irises.*

The user also has the option to view the data underlying a selected rule or the selected data set, as described next.

3.2 Graphical representation of rules and data

The data visualization component uses a scatter diagram to identify interesting subsets of the data set. A scatter diagram is a simple data visualization technique that can provide users with a wealth of possible information regarding the problem domain [1, 2]. In this approach, two input values are plotted in a Cartesian space, thus identifying possible clusters of data points. This tool can convey enormous amounts of information in a compact representation [1].

The user is able to view the training, test or validation data set by making a projection along any two input dimensions. The domain expert uses this tool to obtain a general idea of the contents and quality of the data set. However, the main objective of this component is to visualize the portion of the data set covered by a particular rule. *Figure 3 shows how the rule (IF petal-width > 17.50 THEN Iris = virginica) is visualized by means of a scatter diagram. The diagram depicts the petal-width and petal-length input dimensions. The scatter diagram shows the Virginica examples covered by the rule, together with those examples that were not covered. The diagram indicates that the petal-lengths of most Virginica Irises are larger than 46.50.* Note that any two of the inputs can be depicted on the X-axis or Y-axis. This may lead to the identification of new clusters that are not described in the rule. *For example, a new cluster identifying a relationship between the sepal-widths and petal-widths of Virginicas is identified. This information is subsequently used to form a new rule to be used for further training.*

The Rule Builder is used to construct new rules to be used during cooperative learning, as described next.

3.3 Rule Builder

The Rule Builder models a learner's participation during the cooperative learning phase, as discussed in Section 2. Note that the learning team operates in one of two modes. In automatic mode, the cooperative learning phase is dynamically re-iterated until no new rules can be created. That is, the rule combining, data generation and rule pruning steps are completed without any feedback from the user. Here, the human expert acts as a spectator, viewing the learning team's cooperation. This is especially useful when the user wishes to trace the learning process. In manual mode, the user activity participates in the learning process. He monitors and guides the re-iteration of new rules and the data generation process. Importantly, the user can aid the learning process by removing or adapting rules. In this way, the user guides the learning process by incorporating his domain knowledge into the system.

The Rule Builder interface is shown in Figure 4. The top part of the interface displays the rules as generated by the active learner. The current rule accuracy threshold, which is equal to the average rule accuracy, is

Data Mining II, C.A. Brebbia & N.F.F. Ebecken (Editors)
© 2000 WIT Press, www.witpress.com, ISBN 1-85312-821-X

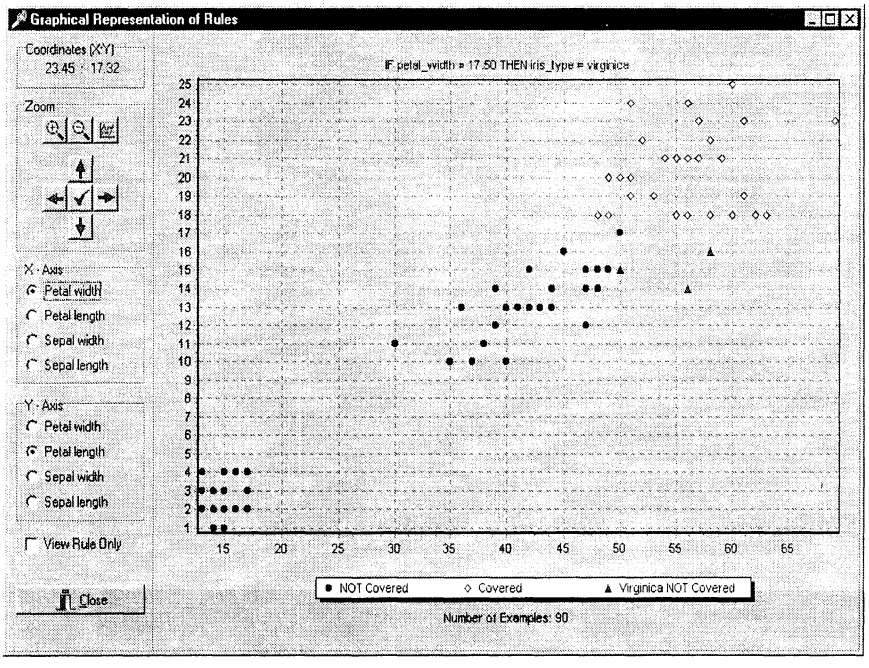


Figure 3: Visualization of data covered by a rule

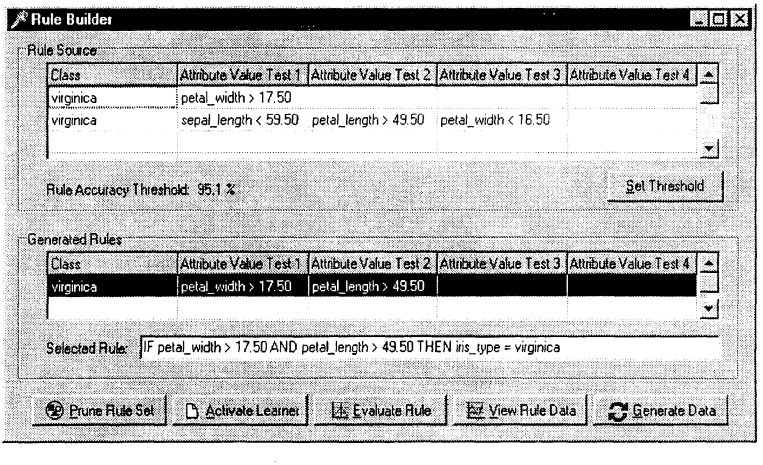


Figure 4: The Rule Builder interface

displayed next. Recall that this value is used to distinguish between high and low quality rules. The window also displays the newly generated rules. These rules are generated using the rule combination algorithm described in Section 2.

The interface contains a number of user options, which activates the following tasks.

- **Modify Selected Rule**. A selected rule is modified by adapting the attribute-value of a test, *(e.g. changing a test from (petal-width > 17.0) to (petal-width > 14.0)*. Alternatively, new attribute-value tests may be added to or removed from a rule. In this way, the domain expert's domain knowledge is used to focus the data mining process.

- **Evaluate Rule.** The quality of a selected rule is evaluated against the training, test or validation set, as described in Section 3.1. The accuracy and coverage of a newly created rule are evaluated, both prior to data generation and after the re-iteration of the individual learning phase.

- **View Rule Data**. The currently selected rule is visualized, as described in Section 3.2. This aids the user in determining the quality of the rule.

- **Generate Data.** New data, based on the selected rule, are automatically generated, following the data generation process as described in Section 2.

- **Activate Learner**. The individual learning phase of the learner is activated to create a new set of rules based on the data generated from a selected rule.

- **Prune Rule Set**. The current rule set is pruned using the REP rule pruning algorithm.

The Rule Builder interface models the cooperative data mining process, thus providing insights into the learning process followed by each learner. This interface is used to determine whether a particular data mining tool is suited for the task at hand. Note that the failure of a learner, due to its inductive bias, to generate high quality rules can thus be detected. Such a learner should be removed from the cooperative data mining team, since it is ill-suited for the current problem domain.

4 Conclusion

The use of a visual cooperative data mining tool to act as a personal assistant to a human domain expert provides a promising approach to solving the well-known knowledge acquisition bottleneck. In the cooperative data mining team approach, two or more data mining tools are incorporated into

Data Mining II, C.A. Brebbia & N.F.F. Ebecken (Editors)
© 2000 WIT Press, www.witpress.com, ISBN 1-85312-821-X

a cooperative learning team. The members of the learning team cooperate by sharing their unique perspectives with one another. The learning process, as well as the results produced by the learning team, are visualized using the ViziMine tool. The user has access to the knowledge of the data mining team and is thus able to adapt his perspective of the problem at hand. In addition, the human domain expert's knowledge can be incorporated in the learning system by explicitly modeling the expert as a member of the learning team. In this way, the domain expert obtains insights into the data mining process, which may lead to a better understanding of the problem domain. The data mining results are dynamically updated as new rules are produced through automated data generation. Experience shows that this approach leads to increasing domain experts' trust in the data mining effort, especially in difficult to learn domains [10].

References

[1] Adriaans, P. & Zantine, X., Data mining, Addison Wesley, The Netherlands, 1998.

[2] Amant, R.E. & Cohen, P.R., A Planner for Exploratory Data Analysis, *Readings in Agents*, Morgan Kaufmann, San Francisco, pp. 131-138, 1998.

[3] Clark, P. & Niblett, T., The CN2 rule induction algorithm *Machine Learning*, 3, pp. 261-283, 1989.

[4] Dix, A., Finlay, J., Abowd, G. & Beale, R., *Human-Computer Interaction*, Prentice-Hall, New York, 1993.

[5] Huhns, M.N. & Singh, M.P., Agents and Multiagent Systems: Themes, Approaches and Challenges, *Readings in Agents*, Morgan Kaufman, San Francisco, pp. 1-24, 1998.

[6] Mitchell, T. *Machine Learning*, McGraw-Hill, New York, 1997.

[7] Quinlan, R., *C4.5: Programs for Machine Learning*, Morgan Kaufman, California, 1994. 1996.

[8] Viktor, H.L., Engelbrecht, A.P. & Cloete, I., Rule extraction from Artificial Neural Networks using Sensitivity Analysis, *Proc. International Conference on Neural Networks*, Perth: Australia, pp. 1-12, 1995.

[9] Viktor, H.L., The CILT multi-agent learning system, *South African Computer Journal*, 24, pp. 176-184, 1999.

[10] Viktor, H.L., Arndt, H. & Oberholzer, M.O., Enhancing government decision making through knowledge discovery from data, *Proc. 8th European Conference on Information Systems (ECIS'00)*, Austria, 2000.