A Distributed Solution to the PTE Problem

Ignacio Giráldez
giraldez@ia.uc3m.es
Universidad Carlos III de Madrid
Avda. Universidad 30, 28911 Leganés,
Madrid, Spain

Charles Elkan
elkan@cs.harvard.edu
Harvard University,
ESL 401, 40 Oxford Street,
Cambridge, MA 02138, USA

Daniel Borrajo
dborrajo@ia.uc3m.es
Universidad Carlos III de Madrid,
Avda. Universidad 30,
28911 Leganés, Madrid, Spain

Abstract

A wide panoply of machine learning methods is available for application to the Predictive Toxicology Evaluation (PTE) problem. The authors have built four monolithic classification systems based on Tilde, Progol, C4.5 and naive bayesian classification. These systems have been trained using the PTE dataset, and their accuracy has been tested using the unseen PTE1 data set as test set. A Multi Agent Decision System (MADES) has been built using the aforementioned monolithic systems to build classification agents. The MADES was trained and tested with the same data sets used with the monolithic systems. Results show that the accuracy of the MADES improves the accuracies obtained by the monolithic systems. We believe that in most real world domains the combination of several approaches is stronger than the individuals.

Introduction

The Predictive Toxicology Evaluation (PTE) Challenge (Srinivasan *et al.* 1997) was devised by the Oxford University Computing Laboratory to test the suitability of machine learning programs for carcinogenesis prediction. Machine learning systems use information related to the structure of chemical compounds, and the output of various biological tests, and relate it to the carcinogenic activity of each chemical compound. Several data sets were made available for this purpose. We used the PTE data set (300 chemicals) for training, and the PTE1 data set (39 chemicals) for testing. We also used available background knowledge to create additional representations.

Experiments with monolithic machine learning systems

As a first approach to solve the classification task, we trained and tested four different monolithic machine learning systems: Tilde (Blockeel & Raedt 1997), Progol (Muggleton 1995), C4.5 (Quinlan 1993) and a naive bayesian classification (Smith 1988) system.

system	accuracy	std. dev.
C4.5	51.0%	8.0%
Progol	58.9%	7.9%
Naive Bayes	64.0%	7.7%
Tilde	71.8%	7.2%

Table 1: individual accuracies

In order to evaluate the accuracy of the monolithic systems and whether these accuracies differ significantly, one needs an estimate of the standard errors of the observed accuracies. Suppose we view each test example as a single binary trial, where the probability of getting the example right is p. Then if n is the number of examples, the expected number of right examples is np with standard deviation $\sqrt{np(1-p)}$. Let the observed accuracy be our estimate of p. The results obtained using the monolithic systems appear on table 1.

Because the test set is so small, only the difference in accuracy between C4.5 and Tilde is statistically significant. The analysis of the results reveals that each individual classification system is right on a different set of chemicals, we call this set the competence region of the system. We performed some experiments to determine in an approximate way what is the competence region of every system. For this purpose, we used the distributed reinforcement learning of competencies method as it appears on (Giráldez & Borrajo 1998). This method provides an estimate of the suitability of problem solvers (e.g. classification systems in this task) for the satisfactory solution of different classes of problem instances. Applied to the classification of chemicals as either carcinogenic or non carcinogenic, this method gives a measure of how suitable every system is for the classification of various types of chemicals. The application of the method to carcinogenesis prediction can be summarised as follows:

1. The space of chemicals S is partitioned into s subsets that should contain chemicals with similar structure and properties: $S = \bigcup_{i=1}^{s} S_i$ such that $S_i \bigcap S_j = \emptyset$ when $i \neq j$. A set of 38 attributes was selected, then for every possible 38-tuple a subset was formed that contains all the drugs that share that attribute values. For the train set used only 256 subsets were actually needed, since for many 38-tuples there were not any drug in the train set with that attribute values.

2. A reinforcement table is associated to every subset. This table has an entry for every classifier system, which reflects the cumulative reinforcement obtained by the system as a result of the decisions it made on the elements of the subset.

3. A writing counter is associated to every reinforcement table (and hence to every subset) to keep track of how many times the table has been modified. This counter is a measure of the representativeness of the subset, and is also a measure of the coarseness of the grain used in the partition.

4. For every chemical in the training set all the classifiers produce a classification, and the subset S_i of the partition that contains this chemical is determined.
   (a) A positive reinforcement is distributed among the systems that produced right classifications, and is added to the cumulative reinforcement of the system associated to the subset S_i.
   (b) A negative reinforcement is distributed among the systems that produced wrong classifications, and is added to the cumulative reinforcement of the system associated to the subset S_i.
   (c) Classification systems that refrained from making a decision due to low confidence factors are not affected at all.
   (d) The writing counter of S_i is properly updated.

The approximate competence region of a system is the union of all the subsets that have reinforcement tables associated to them where the entry of the system is greater than zero. The extent of the competence regions of the systems can be shown by the number of subsets contained within their competence regions (CR). The number of subsets where the score of a given system is better than any of the others (sole wins), or equal to the best score (shared wins) measures how much the system complements the other systems. This information appears on the following table:

The analysis of the table 2 shows that:

system	subsets in CR	sole wins	shared wins
C4.5	87	10	99
Progol	196	41	194
Naive Bayes	162	0	156
Tilde	192	24	189

Table 2: size of CR's and complementation measures

- the competence regions of the monolithic systems are different. This fact is clearly evidenced by the disparate sizes shown in the second column.
- *C4.5*, *Progol* and *Tilde* have each a sole wins counter greater than zero. This means that each of them has some know-how not shared by any other of the four systems and also that they can complement any other system.
- *Naive Bayes* has a sole wins counter of zero. This means that there is not any kind of chemicals that are only correctly classified by it.

A Distributed System for Decision Making and Learning

If the monolithic classifiers were combined appropriately, then accuracy better than that of any individual classifier might be achieved. Our method of combination is a distributed system that we call a Multi Agent Decision System (MADES), a special kind of Multi Agent System. A MADES is a Multi Agent System built for decision making, where a single decision is output by the system as a group, although internally multiple decisions may be made locally by the component agents. The model we used for the construction of the MADES is the IAO (Intelligent Agents Organization) model (Giráldez & Borrajo 1997). This model provides: a distributed system architecture, a communication protocol used by the agents, a distributed decision making algorithm, a distributed learning algorithm and a conflict resolution mechanism.

MADES architecture

Figure 1 shows a high level view of this architecture. In the IAO model there are various different roles for agents:

- One agent, known as the *referee*, is in charge of the overall system control. It broadcasts problem instance descriptions (service requests), and control signals to the rest of the team. It then receives the respective replies from the rest of the agents.

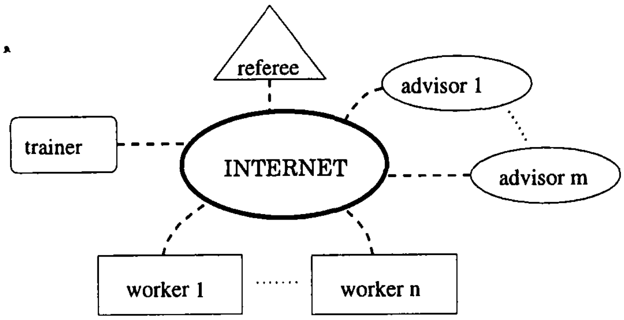

Figure 1: A schematic view of the IAO Architecture.

These replies may be either advice, or problem solving proposals. The relationship among the referee and the rest of the agents can be regarded as a client-server relationship. The referee (client) sends service requests to the other agents (servers), that reply with the required information. The services the referee may request to an agent are: solution proposal synthesis (only to worker agents), execution of a learning session (if the agent has learning capability) and advice request (again, only to selected agents). These service requests are scheduled in a way that maximizes parallellism (every agent runs on a different machine), so the MADES response time is minimized.

The referee is in charge of organizing the worker agents, with the aid of the advisor agents, in a way that maximizes cooperation to attain the common goal.

- The *worker agents* (Beyer & Śmieja 1993) receive problem descriptions from the referee, and reply with solution proposals. They work in parallel on a solution proposal to the same problem instance, are capable of autonomous decision making, and may have learning capabilities. Any of them could be the basis of a monolithic system aimed to solve each problem.
- Several agents may play the role of *advisors*. When the referee broadcasts a problem description, the worker agents start working on proposals to solve the problem. Meanwhile, the advisor agents synthesize advice that will inform the referee about which worker agent the advisors think is the most competent for solving this particular problem instance. The referee will use this advice when one of the proposals the worker agents provide has to be selected.
- A *trainer* agent produces problem instances that are used for training and testing. Problem generation can be made either randomly, or using an "ad hoc" scheme. The criteria for problem synthesis affects the success of the learning effort, as it is widely known.

IAO Decision Making

When a problem instance arrives at the referee, it consults the advisors to determine whether any worker agent is expected to solve that instance of the problem satisfactorily. In that case, the proposal that this agent provides will be given a privileged status when it has to compete with the proposals of its fellow workers. The problem instance description is broadcasted to the workers, so that they can work on it, and reply with a solution proposal. One advantage IAO presents, is that the advisors and the workers work in parallel, so the IAO response time is a very small overhead longer than the one that would be obtained from a monolithic system built from the most time consuming IAO worker.

When the referee receives the proposals of all the worker agents, and the advice from its advisors, it starts its own decision process. Now, the referee has to decide which proposal to use (most of the times this proposals will be incompatible and contradictory). The referee uses a poll mechanism for conflict resolution: the proposal that gets the greatest support is the one the referee will follow. The advisors' candidates receive extra votes in this poll, so they have some advantage over less credited workers.

Experiments with a Multi Agent Decision System

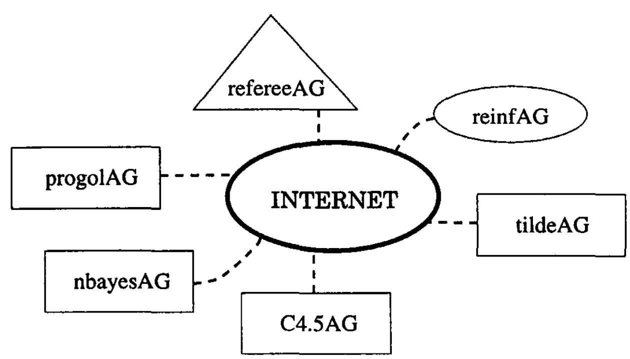

Figure 2: A schematic view of the EICAP application.

The EICAP (Environmentally Induced CAncer Prevention) application was built using the IAO model for the construction of a MADES. A worker agent was built from each previous monolithic classifier. An advisor agent, known as *reinfAD*, was built for learning the competencies of the worker agents by means of distributed reinforcement learning. Based on this information it can provide advice to any requesting agent on which is the worker agent that is expected to provide the most trustworthy classification for a given drug. The referee agent is in charge of controlling the execution of the distributed decision making algorithm, and uses the information provided by *reinfAD* for conflict resolution.

EICAP was trained using the PTE dataset, and tested with the unseen PTE1 data set. The accuracy obtained is 84.6% with a standard deviation of 5.8%, which is an encouraging preliminary result.

Conclusions

The results provide an experimental evidence that under certain conditions, the IAO model can take advantage of heterogeneous monolithic decision making systems, and integrate them in a MADES, so that the joint performance is better than the best individual performance. The necessary conditions for this to happen are: first that the worker agents must complement each other, and second, that the algorithm for learning the worker competencies must be able to approximate the actual competence regions close enough so that complementation opportunities are recognized in the conflict resolution phase.

References

Beyer, U., and Śmieja, F. 1993. Learning from examples, agent teams and the concept of reflection. Technical Report 766, GMD.

Blockeel, H., and Raedt, L. D. 1997. Top-down induction of logical decision trees. In *Proceedings of the 9th Dutch Conference on Artificial Intelligence NAIC97.*

Giráldez, J. I., and Borrajo, D. 1997. A distributed model for the combination of heterogeneous knowledge based agents. In *J. Hunt, R. Milnes (eds.) "Research and Development in Expert Systems XIV".* Proc. of ES97, pp175-184, SGES Publications.

Giráldez, J. I., and Borrajo, D. 1998. Distributed reinforcement learning in multi agent decision systems. In Coelho, H., ed., *Progress in Artificial Intelligence*, number 1484 in Lecture Notes in Artificial Intelligence. Springer.

Muggleton, S. 1995. Inverse entailment and progol. *New Gen. Comput.* 13:245–286.

Quinlan, J. R. 1993. *C4.5: Programs for Machine Learning.* San Mateo, CA: Morgan Kaufmann.

Smith, J. 1988. *Decision Analysis, a Bayesian Approach.* Chapman and Hall.

Srinivasan, A.; Muggleton, S.; King, R.; and Sternberg, M. 1997. The predictive toxicology evaluation challenge. In *Proc. of IJCAI 97.*