Cooperative Case Bartering for Case-Based Reasoning Agents

Santiago Ontañón and **Enric Plaza**
IIIA, Artificial Intelligence Research Institute
CSIC, Spanish Council for Scientific Research
Campus UAB, 08193 Bellaterra, Catalonia (Spain).
{santi,enric}@iiia.csic.es, http://www.iiia.csic.es

Abstract

Multiagent systems offer a new paradigm to organize AI Applications. We focus on the application of Case-Based Reasoning to Multiagent systems. CBR offers the individual agents the capability of autonomously learn from experience. In this paper we present a framework for collaboration among agents that use CBR. We present explicit strategies for case bartering in order improve individual case bases and reduce bias is the case bases. We also present empirical results illustrating the robustness of the case bartering process for several configurations of the multiagent system.

Introduction

Multiagent systems offer a new paradigm to organize AI applications. Our goal is to develop techniques to integrate CBR into applications that are developed as multiagent systems. CBR offers the multiagent system paradigm the capability of autonomously learn from experience. In this paper we present a framework for collaboration among agents that use CBR and some experiments illustrating how they can improve its performance using case bartering strategies.

The individual case bases of the CBR agents are the main issue here, if they are not properly maintained, the overall system behavior will be suboptimal. In a real system, there will be agents that can very easily obtain certain kind of cases, and that will very costly obtain other types of cases, and for sure that other agents in the system will be in the inverse situation. It will be beneficial for both agents if they reach an agreement to trade cases. This is a very well known strategy in the human history called *bartering*. Using case bartering, agents that have a lot of cases of some kind will give them to another agents in return to more interesting cases for them.

Our research focuses on the scenario of separate case bases that we want to use in a decentralized fashion by means of a multiagent system, that is to say a collection of CBR agents that manage individual case bases and can communicate (and collaborate) with other CBR agents. In this paper we focus on case bartering. We present two protocols for case bartering that improve the overall performance of the system and of the individual CBR agents without compromising the agent's autonomy. This protocols will try to minimize the individual case base bias (how far is a case base of being a good sample of the overall distribution).

The structure of the paper is as follows. First, we present the collaboration scheme that the agents use, then the individual case base bias measurement is introduced. After that, the case bartering mechanism, including the bartering protocols is presented. Finally, The experiments are explained and the paper closes with related work and conclusion sections.

Collaboration Scheme

A multiagent CBR ($\mathcal{M}$AC) system $\mathcal{M} = \{(A_i, C_i)\}_{i=1...n}$ is composed on n agents, where each agent A_i has a case base C_i. In the experiments reported here we assume that initially case bases are disjunct ($\forall A_i, A_j \in \mathcal{M} : C_i \cap C_j = \emptyset$), i.e. initially there is no case shared by two agent's case bases. In this framework we restrict ourselves to analytical tasks, i.e. tasks (like classification) where the solution is achieved by selecting from an enumerated set of solutions $K = \{S_1 \ldots S_K\}$. A case base $C_i = \{(P_j, S_k)\}_{j=1...N}$ is a collection of pairs problem/solution.

When an agent A_i asks another agent A_j help to solve a problem the interaction protocol is as follows. First, A_i sends a problem description P to A_j. Second, after A_j has tried to solve P using its case base C_j, it sends back a message that is either `:sorry` (if it cannot solve P) or a solution endorsement record (SER). A SER has the form $\langle\{(S_k, E_k^j)\}, P, A_j\rangle$, where the collection of *endorsing pairs* (S_k, E_k^j) mean that the CBR method of the agent A_j has found E_k^j cases in case base C_j endorsing solution S_k— i.e. there are a number E_k^j of cases that are relevant (similar) for endorsing S_k as a solution for P. Each agent A_j is free to send one or more endorsing pairs in a SER record.

Voting Scheme

The voting scheme defines the mechanism by which an agent reaches an aggregate solution from a collection of SERs coming from other agents. The principle behind the voting scheme is that the agents vote for solution classes depending on the number of cases they found endorsing those classes. However, we want to prevent an agent having an unbounded number of votes. Thus, we will define a normalization function so that each agent has one vote that can be for a unique solution class or fractionally assigned to a number of classes depending on the number of endorsing cases.

Formally, let $\mathcal{A}^t$ the set of agents that have submitted their SERs to the agent A_i for problem P. We will consider that $A_i \in \mathcal{A}^t$ and the result of A_i trying to solve P is also reified as a SER. The vote of an agent $A_j \in \mathcal{A}^t$ for class S_k is

$$Vote(S_k, A_j) = \frac{E_k^j}{c + \sum_{r=1\ldots K} E_r^j}$$

where c is a constant that on our experiments is set to 1. It is easy to see that an agent can cast a fractional vote that is always less than 1. Aggregating the votes from different agents for a class S_k we have ballot

$$Ballot^t(S_k, \mathcal{A}^t) = \sum_{A_j \in \mathcal{A}^t} Vote(S_k, A_j)$$

and therefore the winning solution class is the class with more votes in total, i.e.

$$Sol^t(P, \mathcal{A}^t) = arg \max_{k=1\ldots K} Ballot(S_k, \mathcal{A}^t)$$

This voting scheme can be seen as a variation of *Approval Voting* (Brams & Fishburn 1983). In *Approval Voting* each agent vote for all the candidates they consider as posible solutions without giving any weight to its votes. In our scheme, *Approval Voting* can be implemented making $Vote(S_k, A_j) = 1$ if $E_k^j \neq 0$ and 0 otherwise.

There are two differences between the standard *Approval Voting* and our voting scheme. The first one is that in our voting scheme agents can give a weight to each one of its votes. The second difference is that the sum of the votes of an agent is bounded to 1. Thus we can call it *Bounded-Weighted Approval Voting* (BWAV). In the experiments section we will show some experiments illustrating the effect of changing the voting scheme.

We will show now the *Committee* collaboration policy that uses this voting scheme (see (Ontañón & Plaza 2001) for a detailed explanation and comparison of several collaboration policies).

Committee Policy

In this collaboration policy the agent members of a $\mathcal{M}$AC system $\mathcal{M}$ are viewed as a committee. An agent A_i that has to solve a problem P, sends it to all the other agents in $\mathcal{M}$. Each agent A_j that has received P sends a solution endorsement record $\langle\{(S_k, E_k^j)\}, P, A_j\rangle$ to A_i. The initiating agent A_i uses the voting scheme above upon all SERs, i.e. its own SER and the SERs of all the other agents in the multiagent system. The problem's solution is the class with maximum number of votes.

Notice that the agents participating in the Committee Policy have no reason or incentive to lie when providing a SER. First of all, it is rational for an agent to participate in the Committee Policy because it improves the accuracy of the agent itself in classification. Secondly, once an agent has joined the Committee Policy there is no incentive to cheat the others (there is no benefit in the others being worse). On the contrary, if agents start to cheat causing the Committee Policy accuracy to diminish, the agents would decide simply to leave the Committee Policy. Thus, it is rational to participate in the Committee Policy and cheating provides no immediate or long term benefit.

Notice that the agents have no incentive to lie in its SERs, since their only goal is to improve accuracy by cooperating. When an agent A_i asks counsel of another agent A_j, A_j has no incentive to lie because the outcome of the voting scheme will only be used by A_i. Moreover, A_j in the future may need the help of A_i expecting a sincere answer from him. Therefore, the agents will only obtain some benefit of the collaboration with other agents if all them are sincere.

Case Base Bias

In a previous work (Ontañón & Plaza 2001) we showed how agents can obtain better results using the *Committee* collaboration policy that working alone. However, in those experiments we assumed that every agent had a representative sample of cases in its case base. When an agent has a case base that is not representative of the overall distribution, we say that the agent has a biased case base.

In this section we are going to define a measure of the degree of biasing of an individual case base (*ICB* bias or *Individual Case Base bias*), then we will show how the performance of the *Committee* degrades as the *ICB* bias of the agents grow. Later sections introduce bartering policies to improve the *Committee* performance.

Individual Case Base Bias

Let be $d_i = \{d_i^1, \ldots, d_i^K\}$ the individual distribution of cases for an agent A_i, where d_i^j is the number of cases with solution S_j in the the case base of A_i. Now, we can estimate the overall distribution of cases $D = \{D^1, \ldots, D^K\}$ where D^i is the estimated probability of the class S_i,

$$D^j = \frac{\sum_{i=1}^{n} d_i^j}{\sum_{i=1}^{n}\sum_{l=1}^{K} d_i^l}$$

To measure how far is the case base C_i of a given agent A_i of being a representative sample of the overall distribution we will define the *Individual Case Base* (*ICB*) bias, as the square distance between the distribution of cases D and the (normalized) individual distribution obtained from d_i:

$$ICB(C_i) = \sum_{l=1}^{K}\left(D^l - \frac{d_i^l}{\sum_{j=1}^{K} d_i^j}\right)^2$$

In order to see how the *ICB* bias affects the performance of the system, Table 1 shows the accuracy of several multiagent systems with increasing *ICB* bias (the $\mathcal{M}$AC *ICB* bias is calculated as the mean of all the *ICB* bias of the agents in the system). There we can see that when the agents have case bases that are not representative (those with a high *ICB*) the agents using the *Committee* policy obtains lower accuracies. In the following sections, we will show how case bartering improves accuracy by lowering the individual biases.

$\mathcal{M}$AC ICB	*3 Ag.*	*5 Ag.*	*8 Ag.*	*10 Ag.*
0.0	88.36%	88.12%	87.50%	86.75%
0.1	86.07%	87.50%	85.35%	85.00%
0.2	81.46%	83.53%	83.00%	82.00%

Table 1: Classification accuracy for the marine sponge classification problem for systems with several mean Individual Case Base bias.

Case Bartering

In the physical world, bartering involves the interchange of two goods. But as our agents will barter with cases (that are just information) they will only send a copy of the cases to the other agents without losing them. It's a matter of the internal case deletion policy of each agent if a case must be forgotten or not. Deletion policies have been studied (Smyth & Keane 1995) but we will not be considering them in these experiments.

In this section, we are going to present the Case Bartering protocol that the agents use in order to improve the overall performance.

Case Bartering Mechanism

To reach a bartering agreement for bartering between two agents, there must be an offering agent A_i that sends an offer to another agent A_j. Then A_j has to evaluate whether the offer of interchanging cases with A_i is interesting, and accept or reject the offer. If the offer is confirmed, we say that A_i and A_j have reached a bartering agreement, and they will interchange the cases in the offer.

Formally an offer is a tuple $o = \langle A_i, A_j, S_{k_1}, S_{k_2} \rangle$ where A_i is the offering agent, A_j is the receiver of the offer, and S_{k_1} and S_{k_2} are two solution classes, meaning that the agent A_i will send one of its cases with solution S_{k_2} and A_j will send one of its cases with solution S_{k_1}.

Making and accepting offers

The Case Bartering Protocol is not restrictive in how many offers can an agent send at a time. So, many strategies can be used here, but in our experiments, the agents use a very simple one to choose which are the most interesting offers, as follows for a given agent A_i:

- For each solution class $S_{k_1} \in \{S_1 \dots S_K\}$
- A_i looks if increasing by one its number of cases with solution S_{k_1} will decrease its *ICB* bias.
- If so, A_i chooses which agent A_j of the others is the best one to ask for cases of solution S_{k_1} (Currently the chosen A_j is the one with more cases of the solution class S_{k_1}).
- Now A_i determines which is its best class S_{k_2} (the class for which it has more cases), and makes the offer $o = \langle A_i, A_j, S_{k_1}, S_{k_2} \rangle$, i.e. A_i offers to A_j a case of solution S_{k_2} if A_j gives one of solution S_{k_1} to A_i.

When an agent receives a set of offers, it has also to choose which of these offers to accept and which not. In our experiments the agents use the simple rule of accepting every offer that reduces its own *ICB* bias. Thus, we will define the set of interesting offers $Interesting(O, A_i)$ of a set of offers O for an agent A_i as those offers that will reduce the *ICB* bias of A_i. Moreover, an agent cannot send twice the same case to the same agent. So, the agents will only accept those interesting offers that can satisfy (i.e. can provide a new case for interchanging).

Case Bartering Protocol

We are going to present two different protocols for Case Bartering, both synchronous (i.e. there are preestablished stages ("rounds") where the agents can send their offers, then the protocol moves to the next stage, etc). The first one is called the *Simultaneous Case Bartering Protocol*, and the second one the *Token-Passing Case Bartering Protocol*.

When an agent member of the $\mathcal{M}$AC wants to enter in the bartering process, is sends an initiating message to all the other agents in the $\mathcal{M}$AC. Then all the other agents answer whether or not they enter the bargaining process. This initiating message contains a parameter t_R, corresponding to the time that each round of the protocol will last.

Simultaneous Case Bartering Protocol In this protocol,in every round all the agents send their offers simultaneously. When all the offers have been sent, all the agents send a message for the offers they accept.

1. Each agent A_i broadcasts its individual distribution d_i.
2. Each agent computes the overall distribution estimation D.
3. The agents send their bartering offers.
4. Each agent chooses a subset of accepted offers from the set of received offers from the other agents and sends accept messages.
5. When the maximum time t_R is over, all the unaccepted offers are considered as rejected.
6. Each agent that has some bartering agreements sends the cases to interchange to the corresponding agents.
7. Each agent broadcasts its new individual distribution d_i.
8. If there have been no interchanged cases, the Case Bartering Protocol ends, otherwise go to 3.

Token-Passing Case Bartering Protocol The main difference between this protocol and the previous one is the introduction of a Token-Passing mechanism, so that only the agent who has the Token can make offers to the others.

1. Each agent broadcasts its local statistics d_i.
2. Each agent computes the overall distribution estimation D.
3. Each agent computes the *ICB* bias of all the agents taking part in the bartering (including itself), and sorts them. This defines the order in which the Token will be passed through.
4. The agent with higher *ICB* bias is the first to have the Token.
5. The agent who has the Token sends its bartering offers.

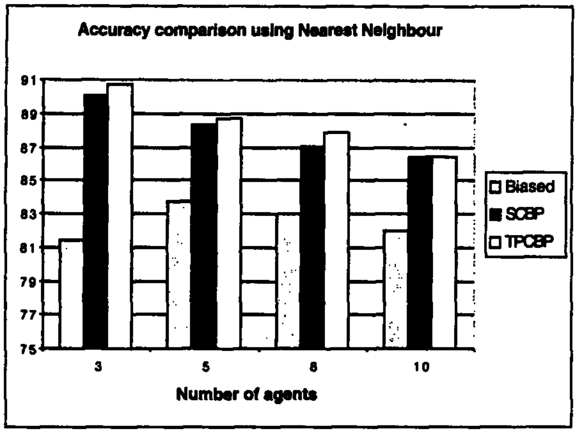

Figure 1: Accuracy comparison of systems where the agents use nearest neighbor with and without using case bartering

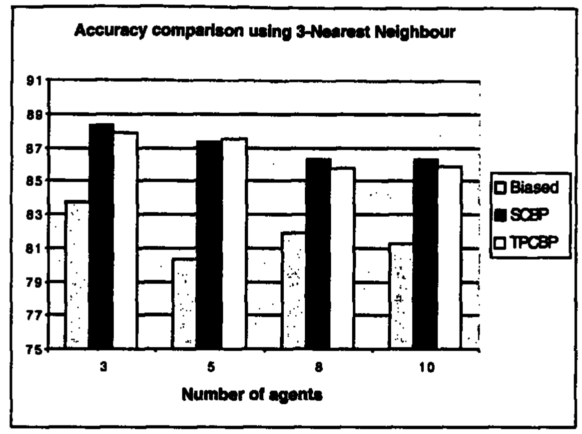

Figure 2: Accuracy comparison of systems where the agents use 3-nearest neighbor with and without using case bartering

6. Each agent chooses a subset of accepted offers from the set of received offers from owner of the token and sends accept messages.
7. When the maximum time t_R is over, all the unaccepted offers are considered as rejected.
8. Each agent that has some bartering agreements sends the cases to interchange to the corresponding agents.
9. Each agent broadcasts its new individual distribution d_i.
10. If the Token belongs to the last agent, go to 11, otherwise the Token is given to the next agent and we go to 5.
11. If there have been no interchanged cases, the Case Bartering Protocol ends, otherwise go to 3.

Protocol discussion

In both protocols, if an offer is not accepted neither rejected within the period time t_R, the offer is considered as rejected, and the protocol moves to the next round.

To ensure the convergence of both protocols, we have only to have in mind the only restriction that we have imposed: an agent cannot send twice the same case to the same agent. With this restriction it's easy to see that both protocols cannot run indefinitely, because each agent has a limited number of cases to trade with. So, we can say that in a bounded number of rounds both protocols will end.

Comparing the protocols, we can see that the *Simultaneous* protocol has the problem that an agent has to decide if accept offers or not without knowing if its own offers are going to be accepted. The *Token-Passing* protocol tries to solve this problem by letting only one agent to send offers at a time.

Experimental results

In this section we want to show how the classification accuracy of the agents improve using the case bartering protocols with respect to systems where the agents do not use them. We also show results concerning case base sizes after

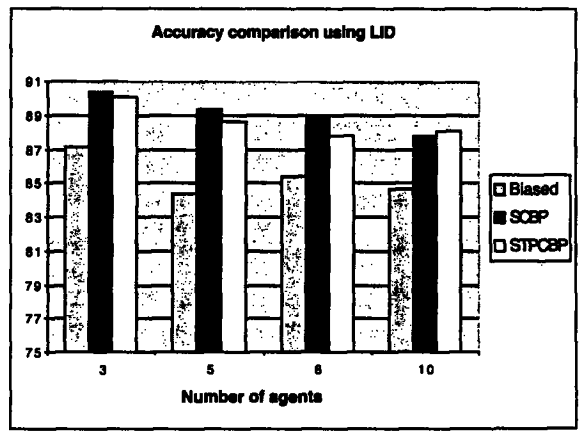

Figure 3: Accuracy comparison of systems where the agents use LID with and without using case bartering

the bartering and the number of rounds needed to converge to a stable case distribution.

We use the marine sponge identification (classification) problem as our test bed —this data set was also used to assess the relational inductive method INDIE in (Armengol & Plaza 2000). Sponge classification is interesting because the difficulties arise from the morphological plasticity of the species, and from the incomplete knowledge of many of their biological and cytological features. Moreover, benthology specialists are distributed around the world and they have experience in different benthos that spawn species with different characteristics due to the local habitat conditions.

In order to show the improvements obtained in the system when the agents use case bartering, we have designed an experimental suite with a case base of 280 marine sponges pertaining to three different orders of the *Demospongiae* class (*Astrophorida*, *Hadromerida* and *Axinellida*). In an experimental run, cases are randomly distributed among the agents (e.g. if the training set is composed of 252 cases and we have

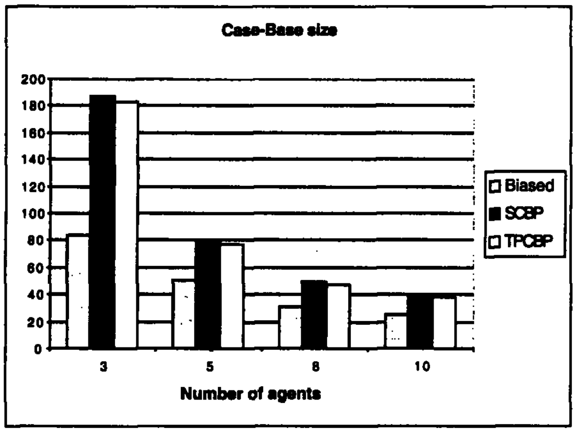

Figure 4: Comparison of the case base size before and after the bartering process

a 4 agents system, each agent will receive about 63 cases). In the testing phase, problems arrive randomly to one of the agents. The goal of the agent receiving a problem is to identify the correct biological order given the description of a new sponge. Once an agent has received a problem, he will use the *Committee* collaboration policy to obtain the prediction.

For experimentation purposes, we force biased case bases in every agent. Specifically, we increase the probability of each agent to have cases of some classes and decrease the probability to have cases of some other classes. For example, in the 3 agent scenario, the 70% of the cases for the class *Astrophorida* in the training set are in the individual case base of Agent 1, and the other two agents only have a 15% of them. Analogously, the 70% of the cases for the classes *Hadromerida* and *Axinellida* are in the case bases of the Agents 2 and 3 respectively. This process increases the individual case-base bias of the agents in the $\mathcal{M}$AC; the first row of Table 2 shows the average over Individual Case-Base (ICB) biases for the agents in the experiments.

Table 2 also shows the average *ICB* biases for the agents in the experiments after the bartering process. We can see that both protocols are able to reduce the ICB bias to very small values. This shows that the bartering protocols effectively interchange cases until all agents drastically reduce their *ICB* bias; only then the process ends and the overall accuracy has indeed improved to the level we expected. Finally, notice that when agents have a greater volume of cases to barter (e.g. in the 3 agents scenario) the *ICB* bias obtained after bartering is one order of magnitude lower than when the agents have fewer cases (from 0.00003 in 3 agents scenario to 0.0003 in the 10 agents scenario).

In order to test the generality of the protocols, we have tested them using systems with 3, 5, 8 and up to 10 agents, and using several CBR methods: nearest neighbor, 3-nearest neighbor and LID (Armengol & Plaza 2001). The results presented here are the average of 5 10-fold cross validation runs.

$\mathcal{M}$AC *ICB* bias	*3 Ag.*	*5 Ag.*	*8 Ag.*	*10 Ag.*
Before	0.2	0.2	0.23	0.15
After SCBP	0.00004	0.0003	0.0004	0.0004
After TPCBP	0.00003	0.0002	0.0002	0.0003

Table 2: $\mathcal{M}$AC *ICB* biases of the multiagent systems used in the experiments before and after the case bartering process.

	3 Agents	*5 Agents*	*8 Agents*	*10 Agents*
SCBP	79.4	24.9	14.0	12.9
TPCBP	212.0	101.1	97.2	99.4
*SCBP*n*	238.2	124.5	112.0	129.0

Table 3: Number of rounds need to converge in the case bartering protocols.

The figures 1, 2 and 3 show the results of applying the two case bartering protocols. Three bars are shown for each scenario, the ***biased*** results represent the average accuracy obtained by the $\mathcal{M}$AC without using case-bartering with biased individual case bases; and the *SCBP* and *TPCBP* results represent the average accuracy obtained by the $\mathcal{M}$AC after using the *Synchronous Case Bartering Protocol* and *Token-Passing Case Bartering Protocol* respectively. We can see in those figures that in all the scenarios, the $\mathcal{M}$AC systems using case bartering obtain a significative gain in accuracy than those systems that do not use case bartering. This shows the independence of the bartering protocols from the CBR method used by the individual agents. Those figures also show that case bartering is robust even when the size of the case bases decreases and the number of cases an agent can barter is very small, as we can see for the 10 agents scenario where each agent has only about 25 cases (i.e. less than 9 cases per class).

Comparing the accuracy obtained by the two protocols *SCBP* and *TPCBP* we see that both have nearly the same accuracy in all the scenarios. We can see that there is never a difference greater than 1% between the results of the Simultaneous protocol and the results of the Token-Passing protocol. Therefore no bartering protocol is significantly better than another but both are significantly better than using no bartering protocol.

Figure 4 shows the case base sizes reached after case bartering. We see that the agents stop interchanging cases before each agent acquires all known cases in the system. Moreover, except in the 3 agents scenario, the case base sizes do not increase very much. The 3 agents scenario is special because the initial case bases of the agents are quite big, and to repair their *ICB* biases the number of cases needed to be bartered is much greater than in the 5, 8 or 10 agent scenarios. We also see that the case base sizes obtained using the Token-Passing protocol are slightly smaller than the ones obtained using the Simultaneous protocol.

Concerning to the convergence of the protocols, they always converge. Table 3 shows the average number of rounds need to converge in both protocols. We can see that the Simultaneous protocol is much faster than the Token-Passing one (taking only in consideration the number of rounds need

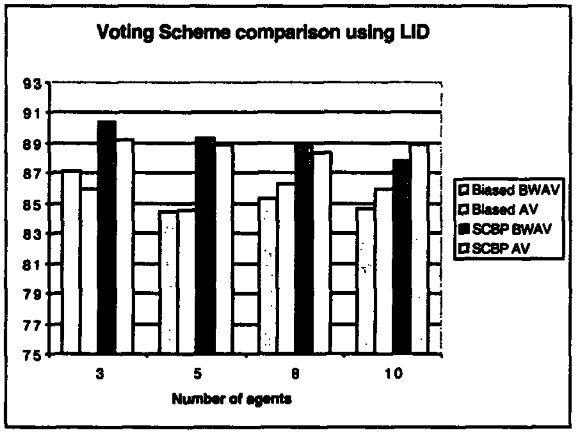

Figure 5: Comparison between *Bounded-Weighted Approval Voting* and standard *Approval Voting* for agents using LID.

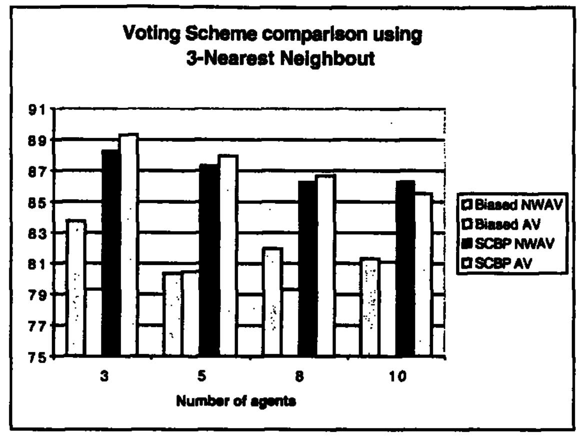

Figure 6: Comparison between *Bounded-Weighted Approval Voting* and standard *Approval Voting* for agents using 3-nearest neighbour.

to converge). This is as expected, since in the Token-Passing protocol only one agent can make offers each round, so *TPCBP* needs about n times more rounds (being n the number of agents in the system) than the Simultaneous protocol. The third row of table 3 shows the number of rounds of the Simultaneous protocol multiplied by n. We see that the number of rounds needed by the Token-Passing protocol is always a bit less than those numbers.

For comparison purposes Figures 5 and 6 show some results where the agents use standard *Approval Voting* instead of the *Bounded-Weighted Approval Voting*. These figures show a comparison between the two voting schemes for two different scenarios: in the first one the agents do not use case-bartering, and in the second one they use the *SCBP*. The results show the accuracy for LID and 3-Nearest Neighbour (since in 1-Nearest Neighbour agents vote for only one class, there is no difference between *AV* and *BWAV*). Figures 5 and 6 show that there is no significant difference between the two voting schemes. When the agents use LID *BWAV* works better for systems where there are fewer agents (and thus more cases per case-base). But when the agents use 3-Nearest Neighbour this difference is not so clear. When the case-bases are biased, standard *AV* is worse than *BWAV* with 3-Nearest Neighbour (specially in the 3 and 8 agents scenario). However, after the bartering process (when *ICB* bias is low), both voting schemes obtain nearly the same result. Sumarizing, both voting schemes behave similarly, but *BWAV* is more robust with higher biased conditions.

Related Work

Several areas are related to our work: multiple model learning (where the final solution for a problem is obtained through the aggregation of solutions of individual predictors), case base competence assessment, and negotiation protocols. Here we will briefly describe some relevant work in these areas that is close to us.

A general result on multiple model learning (Hansen & Salamon 1990) demonstrated that if uncorrelated classifiers with error rate lower than 0.5 are combined then the resulting error rate must be lower than the one made by the individual classifiers. The BEM (*Basic Ensemble Method*) is presented in (Perrone & Cooper 1993) as a basic way to combine continuous estimators, and since then many other methods have been proposed: *Stacking generalization* (Wolpert 1990), *Cascade generalization* (Gama 1998), *Bagging* (Breiman 1996) or *Boosting* (Freund & Schapire 1996) are some examples. However, all these methods do not deal with the issue of "partitioned examples" among different classifiers as we do—they rely on aggregating results from multiple classifiers that have access to *all* data. Their goal is to use multiplicity of classifiers to increase accuracy of existing classification methods. Our intention is to combine the decisions of autonomous classifiers (each one corresponding to one agent), and to see how can they cooperate to achieve a better behavior than when they work alone. A more similar approach is the one proposed in (Vuurpijl & Schomaker 1998), where a MAS is proposed for pattern recognition. Each autonomous agent being a specialist recognizing only a subset of all the patterns, and where the predictions were then combined dynamically.

Learning from biased datasets is a well known problem, and many solutions have been proposed. Vucetic and Obradovic (Vucetic & Obradovic 2001) propose a method based on a bootstrap algorithm to estimate class probabilities in order to improve the classification accuracy. However, their method does not fit our needs, because they need the entire testset available for the agents before start solving any problem in order to make the class probabilities estimation.

Related work is that of case base competence assessment. We use a very simple measure comparing individual with global distribution of cases; we do not try to assess the aeras of competence of (individual) case bases - as proposed by Smyth and McKenna (Smyth & McKenna 1998). This work focuses on finding groups of cases that are competent.

In (Schwartz & Kraus 1997) Schwartz and Kraus discuss negotiation protocols for data allocation. They propose two protocols, the sequential protocol, and the simultaneous protocol. These two protocols can be compared respectively to our *Token- Passing Case Bartering Protocol* and *Simultaneous Case Bartering Protocol*, because in their simultaneous protocol, the agents have to make offers for allocating some data item without knowing the other's offers, and in the sequential protocol, the agents make offers in order, and each one knows which were the offers of the previous ones.

Conclusions and Future Work

We have presented a framework for cooperative Case-Based Reasoning in multiagent systems, where agents use a market mechanism (bartering) to improve the performance both of individuals and of the whole multiagent system. The agent autonomy is maintained, because if an agent do not want to take part in the bartering, he just has to do nothing, and when the other agents notice that there is one agent not following the protocol they will ignore it during the remaining iterations of the bartering process.

In this article we have shown a problem arising when data is distributed over a collection of agents, namely that each agent may have a skewed view of the world (the individual bias). Comparing empirical results in classification tasks we saw that both the individual and the overall performance decreases when bias increases. The process of bartering shows that the problems derived from distributed data over a collection of agents can be solved using a market-oriented approach. Each agent engages in a barter only when it makes sense for its individual purposes but the outcome is an improvement of the individual and overall performance.

The naive way to solve the ICB bias problem could be to centralize all data in one location or adopt a completely cooperative multiagent approach where each agent sends its cases to other agents and they retain what they want (a "gift economy"). The problem with the completely cooperative approach is that the outcome improves but redundancy also increases and there may be scaling up problems; the bartering approach tries to interchange cases only to the amount that is necessary and not more.

In the experiments reported in this paper, the agents use strategies for choosing which offers to generate and send to other agents and for choosing which offers to accept from other agents. Currently, both strategies try to minimize the *ICB* bias measure. The *ICB* bias estimates the difference between the individual and global case distribution over the classes. However, we plan to study other kinds of biases that may characterize the individual agents' case base. In order to compute these new bias measures, the agents may need to make public more information. Thus, a modification in the bartering protocols would be needed to manage the information required.

We have focused on bartering for agents using lazy learning but future work should address the usefulness of bartering for eager (inductive) learning techniques.

Acknowledgements

The authors thank Josep-Lluís Arcos and Eva Armengol of the IIIA-CSIC for their support and for the development of the Noos agent platform and the LID CBR method respectively. Support for this work came from CIRIT FI/FAP 2001 grant and projects TIC2000-1414 "eInstitutor" and IST-1999-19005 "IBROW".

References

Armengol, E., and Plaza, E. 2000. Bottom-up induction of feature terms. *Machine Learning Journal* 41(1):259–294.

Armengol, E., and Plaza, E. 2001. Lazy induction of descriptions for relational case-based learning. In *12th European Conference on Machine Learning*.

Brams, S. J., and Fishburn, P. C. 1983. *Approval Voting*. Birkhauser, Boston.

Breiman, L. 1996. Bagging predictors. *Machine Learning* 24(2):123–140.

Freund, Y., and Schapire, R. E. 1996. Experiments with a new boosting algorithm. In *Proc. 13th International Conference on Machine Learning*, 148–146. Morgan Kaufmann.

Gama, J. 1998. Local cascade generalization. In *Proc. 15th International Conf. on Machine Learning*, 206–214. Morgan Kaufmann, San Francisco, CA.

Hansen, L. K., and Salamon, P. 1990. Neural networks ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (12):993–1001.

Ontañón, S., and Plaza, E. 2001. Learning when to collaborate among learning agents. In *12th European Conference on Machine Learning*.

Perrone, M. P., and Cooper, L. N. 1993. When networks disagree: Ensemble methods for hydrid neural networks. In *Artificial Neural Networks for Speech and Vision*. Chapman-Hall.

Schwartz, R., and Kraus, S. 1997. Bidding mechanisms for data allocation in multi-agent environments. In *Agent Theories, Architectures, and Languages*, 61–75.

Smyth, B., and Keane, M. T. 1995. Remembering to forget: A competence-preserving case deletion policy for case-based reasoning systems. In *IJCAI*, 377–383.

Smyth, B., and McKenna, E. 1998. Modelling the competence of case-bases. In *EWCBR*, 208–220.

Vucetic, S., and Obradovic, Z. 2001. Classification on data with biased class distribution. In *12th European Conference on Machine Learning*.

Vuurpijl, L., and Schomaker, L. 1998. A framework for using multiple classifiers in a multiple-agent architecture. In *Third International Workshop on Handwriting Analysis and Recognition*.

Wolpert, D. H. 1990. Stacked generalization. Technical Report LA-UR-90-3460, Los Alamos, NM.