ECAI 2010
H. Coelho et al. (Eds.)
IOS Press, 2010

doi:10.3233/978-1-60750-606-5-85

Learning better together

Gauvain Bourgne[1] and **Henry Soldano** [2] and **Amal El Fallah Seghrouchni**[3]

Abstract. This article addresses collaborative concept learning in a MAS. In a concept learning problem an agent incrementally revises a hypothetical representation of some target concept to keep it *consistent* with the whole set of examples that it receives from the environment or from other agents. In the program SMILE, this notion of consistency was extended to a group of agents. A surprising experimental result of that work was that a group of agents learns better the difficult boolean problems, than a unique agent receiving the same examples. The first purpose of the present paper is to propose some explanation about such unexpected superiority of collaborative learning. Furthermore, when considering large societies of agents, using pure sequential protocols is unrealistic. The second and main purpose of this paper is thus to propose and experiment broadcast protocols for collaborative learning.

1 INTRODUCTION

This article deals with the problem of collaborative learning [9] in a multi-agent system (MAS). More precisely, we are concerned with the extension of incremental (i.e. online) concept learning from examples, a simple model of supervised learning that outputs a hypothesis that covers positive examples and rejects negative examples of some target concept, to a collaborative setting. A motivation for collaborative concept learning was reported in a work concerning single agent learning in an intentional MAS using a BDI formalism [7]. In that work, agents shared plans, each of them being associated with a context defining the conditions in which the plan can be triggered. Then, the agents had to revise the triggering context of their plans depending on the failure or success of their execution and were equipped for that purpose with communication and hypothesis revision capabilities. A collaborative concept learning protocol has been further proposed and investigated resulting in the SMILE implementation [1]. In SMILE, autonomous agents are organized in a fully connected MAS, and each agent stores information, i.e. non-revisable factual knowledge received from the environment or from other agents. Each agent also stores and shares with the other agents, a set of common and revisable beliefs, forming the current *theory* in the MAS. When an agent receives some contradictory information, it has to revise the current theory in order to keep it *consistent* with its own information memory. However, as it has also to keep the hypothesis consistent with the whole information in the MAS (*mas-consistency*), a set of interactions with the other agents is necessary. During these interactions, the revising agent plays the role of the *learner* while the other agents act as *critics*. However agents can in turn be learners or critics, none of them being kept to a specific role. A very surprising result of the experiments on boolean learning presented in [1], is an unexpected increase in the accuracy, measured by predicting the function value on test examples, when comparing the results of a n-agents MAS with those of a single agent equipped with the same learning mechanism.

The revision protocol in SMILE, as presented in [1], is purely *sequential* i.e. when an agent acts as a learner, it sends the revised hypothesis to one critic and waits for the end of interaction (during which the hypothesis has possibly been again revised) before proposing it to another critic. The main purpose of this paper is to propose a *broadcast* variant of this protocol and experiment it with small to large number of agents. There are several motivations for that:

- Broadcast protocols should allow a quicker, and more flexible, revision process, as all critics can analyze and criticize a revised hypothesis in parallel. In large societies of agents, time constraints can make sequential protocols inefficient or even inapplicable. More precisely, the consistency maintenance process in SMILE relies on a *slow learning* assumption: in the experiments, examples are sent sequentially to random agents in such a way that a whole revision process can be achieved before a new example arrives. Broadcast protocol respects the SMILE's slow learning assumption with much more frequent example arrivals. Furthermore, broadcasting revised knowledge in a large group seems more realistic and desirable than a sequential propagation.
- Between broadcast and sequential protocols, there is room for a range of collective behaviors, either depending on some structured organization of agents or selective gossip-based propagation. A variant of SMILE has been recently introduced to cope with MAS organized in a network. In that case, each agent only communicates with a set of direct neighbors [2], and the proposed protocols for propagating the revisions are sequential. In this paper, we propose to investigate a broadcast protocol in a fully connected MAS as a first step towards more sophisticated protocols for collective knowledge revision and propagation in networks of agents.

It raises several questions. The first one is regarding the cost of the parallelism. The sequential protocol in SMILE was supposed to minimize the information storage by ensuring that only useful information, i.e. the information that enforces revision, was communicated. Obviously, broadcast cannot ensure such a property. So, an extra communication and storage cost could be expected, and it should be experimented. The second question is related to the accuracy improvement noticed in SMILE w.r.t. the single agent learning: where does this improvement come from? is it related only to the sequential propagation or is it preserved when broadcasting the information?

The paper is organized as follows. Section 2 recalls sequential collaborative concept learning as experimented in SMILE and summarizes the results. Section 3 experiments two variants of the sequential protocol in order to investigate the reasons of the accuracy improvement. In section 4 we present our *broadcast protocol* together with a

[1] N.I.I, 2-1-2 Hitotsubashi, Chiyodaku, Tokyo 1018430, Japan
[2] L.I.P.N, UMR-CNRS 7030, Univ. Paris-Nord, 93430 Villetaneuse, France
[3] LIP6, Univ. P. et M. Curie 104, Av. du Pr. Kennedy, 75016 Paris, France

new behavior of agents, suggested by the results of the above analysis and denoted as *Forgetness*. In section 5 we discuss how Broadcast and Forgetness affect the collaborative learning. Finally, section 6 presents some related work and concludes.

2 COLLABORATIVE LEARNING IN SMILE

2.1 Summary

In SMILE, a MAS with n agents, denoted n-MAS, is represented as a set of agents r_1, ..., r_n. Each agent r_i has a belief set B_i consisting of all the revisable knowledge it has. A *common* part B_C of its knowledge is shared with other agents. If an agent r_i revises its belief set B_i to B'_i, changing in the process B_C into B'_C, all other agents r_k must then revise their belief set B_k to $B'_k = (B_k - B_C) \cup B'_C$. Moreover, each agent r_i owns some information K_i representing the observed facts, taken as being true, and which can possibly contradict B_i. Some *consistency* property $Cons(B_i, K_i)$ is supposed to be maintained by the agent. We have then the following definition:

Definition 1 *An agent r_i is* a-consistent *iff $Cons(B_i, K_i)$ is true. An agent r_i is* mas-consistent *iff $Cons(B_i, K)$ is true, where $K = \cup_{j\in\{1,..,n\}} K_j$ is the information stored in the n-MAS. A n-MAS is* consistent *iff all its agents r_i are mas-consistent.*

Consistency of the agents is supposed to be *additive*, meaning that whenever $Cons(B_i, K_1)$ and $Cons(B_i, K_2)$ hold, $Cons(B_i, K_1 \cup K_2)$ also holds. Furthermore, B_C is assumed to be independent from the remainder of B_i : $Cons(B_i, K_i)$ iff $Cons(B_i - B_C, K_i)$ and $Cons(B_C, K_i)$.

M denotes an *internal revision* mechanism that is applied whenever the agent r_i receives a contradictory piece of information k[4]. It changes B_i into a new belief set $B'_i = M(B_i)$ that is consistent with its new knowledge, thus maintaining the a-consistency of the agent. In the same way, the *mas-consistency* of a *revision* mechanism M_s requires that the agent stays consistent with the whole information stored in the MAS. Finally M_s is *strongly mas-consistent* iff, whenever M_s is applied by an agent, the whole MAS is made consistent.

A strongly mas-consistent revision mechanism M_s is constituted of reiterated applications by the *learner* agent r_i of its *internal* a-consistent revision mechanism M, followed by some interactions between r_i and the other agents, until r_i regains its mas-consistency. The mechanism is triggered by an agent r_i that, upon receipt of a *contradictory* piece of information k, revises B_C to B'_C. An interaction $I(r_i, r_j)$ between the *learner* agent r_i and another agent r_j, acting as *critic* is as follows:

1. Agent r_i sends the revision B'_C to r_j.
2. Agent r_j checks the revision B'_C. If this modification preserves its a-consistency, r_j sends an *acceptance* of B'_C to r_i, else it sends a contradictory piece of information k' : $Cons(B'_j, k')$ is false.

An iteration of M_s is then composed of an internal revision performed by the *learner* agent r_i, followed by a sequence of interactions $I(r_i, r_j)$. If a counter example k' is transmitted to r_i, this triggers a new iteration, starting with a new revision of the learner to reestablish its consistency. Otherwise, when all the critics have sent an acceptance of the proposed hypothesis B'_C, r_i has restored its mas-consistency. It then notifies the other agents, who *adopt* the new hypothesis B'_C. This ensures that, at the end of the revision process, all the agents share the same hypothesis B'_C.

In [1], the revision mechanism M_s described above was proved as strongly mas-consistent when $Cons$ is additive.

Incremental collaborative concept learning

Single agent learning. The mechanism mentioned above was applied to incremental MAS concept learning. In this context a hypothesis is a monotone DNF, *i.e.* a disjunction of terms, each represented as a conjunction of positive literals from a set of atoms A. An example is an interpretation together with a label $+$ or $-$. A hypothesis H *covers* an example e whenever e satisfies (is a model of) H[5]. Given a set of positive and negative examples $E = E^+ \cup E^-$, a hypothesis is *complete* when it covers all the positive examples in E^+, and is *coherent* when it covers no negative examples in E^-. To learn Boolean formulae, negative literals are represented by additional atoms, like $not - a$[6]. Given a current hypothesis H, a memory $E = E^+ \cup E^-$ filled with the examples previously received by the agent, and a new example e that falsifies either completeness or coherence of H (i.e. e *contradicts* H), an internal revision mechanism M produces a revised hypothesis H' that is complete and coherent with respect to the new memory state $E \cup e$.

The internal revision mechanism M implemented in SMILE performs a minimal revision of H as follows: if H does not cover $e = e^+$ (e^+ is a *positive counter example*), H is revised either by minimally generalizing some term or by adding e^+ as a new term. If H covers $e = e^-$ (e^- is a *negative counter example*), each term h covering e^- is discarded from H and replaced by a set of new terms $\{h'_1, ..., h'_n\}$. Terms of the resulting hypothesis that are less general than the others are discarded.

Collaborative learning. If H is the current hypothesis, E_i the current example memory of agent r_i and E the set of all the examples in the system, the notation of section 2.1 becomes $B_i = B_C = H$, $K_i = E_i$ and $K = E$. $Cons(H, E_i)$ states that H is complete and coherent with E_i. The piece of information k received by agent r_i is an example e. As the revision mechanism M we have described, is a-consistent, M_s as described in Section 2.1 is strongly mas-consistent: upon reception of a new example in the MAS by an agent r, a set of interactions between r and the other agents results in a new hypothesis, shared by all the agents, which is complete and coherent with the set E of all the examples in the MAS.

2.2 Experiments

We briefly describe here the results of the experiments performed on collaborative concept learning. An experiment is typically composed of 30 to 50 trials. Each trial corresponds to a sequence of m examples that are incrementally learned by a n-MAS. During these runs, measurements are taken, each time a given number of examples are received by the system. A trial begins by sending an example to a random agent who restores the MAS consistency. Another example is then sent to the MAS and again mas-consistency is restored and so on. Experiments were performed both on a set of boolean problems including Multiplexer-11 ($M11$) and a xor function ($Xor3\_25$), and on various real-world problems from the UCI database [4].

[4] so turning K_i into $K'_i = K_i \cup k$ such that $Cons(B_i, K')$ is false.

[5] e is a model of H whenever there is a term t in H *more general than* e, i.e. such that t is included in e.

[6] A target formula as for instance $f = (a \wedge b) \vee (b \wedge \neg c)$ would be represented as $(a \wedge b) \vee (b \wedge not - c)$. The positive example $\{not - a, b, not - c\}$ is a model of f.

Execution time and Example Redundancy. The *execution time* represents the whole computation and communication activity in the MAS. The results showed that it linearly depends on the number of agents. *Redundancy* depends on n_e, the total number of examples received from the environment in the MAS, and is written as $R_S = n_S/n_e$, where n_S is the sum of the sizes of the agents' example memories E_i. Redundancy reaches a peak when learning is most active, and then slowly decreases towards its minimal value 1.

A n-MAS selects a simpler and more accurate solution than a single agent. This improvement of accuracy (i.e. the ratio of correct classification of a set of test examples) was not expected, because whether there are one or n agents in the MAS, when n_e examples are given to the MAS, it has access to the same amount of information and maintains only one ongoing hypothesis. The improvement, which is impressive on some difficult boolean functions, is however only observed in a few cases in ML database problems. As we will see in the next sections, this improvement is mainly due to a better exploration of the search space, while in many UCI noisy problems, enhancing the exploration does not help.

3 WHY DO N AGENTS LEARN BETTER THAN ONE?

In the following, we discuss experiments on two boolean formulas that are a difficult test for learning methods (see [5]). The *multiplexer-11* (M11) formula, built on 11 Boolean attributes, has 8 conjunctive 4-length terms. The *Xor*3_25 problem is a member of the Xorp_m family : there must be an odd number of value 1 in the first p attributes of the $p + m$-size instance, for it to be positive.

Regarding the reasons why n agents learn better than one, first consider that each learner agent revises the current hypothesis using the examples stored in its own memory i.e. the examples it received from the environment and the *selected* examples it received from critics. To evaluate the related *selection effect*, we have implemented a first variant of SMILE, denoted as Smile$_V$. Secondly, during a revision each learner will have to produce possibly many hypotheses in order to maintain the MAS consistency. The overall effect is an *extensive* exploration of the search space. Also, remember that the current hypothesis is, in a sense *minimal*, as it contains no such term which subsumes another of its terms. This means that simpler hypotheses are favored, and are more likely to be encountered during an extensive exploration. In order to evaluate the overall effect of *selection* plus *extensive exploration* of the search space, we have implemented a second variant denoted as Smile$_O$ similar to SMILE, except that the examples are not randomly distributed to agents.

1. *Smile$_V$*: $n-1$ examples are sequentially sent to the agent r_1. The agent r_2 is then added to form the MAS $\{r_1, r_2\}$. As the last example is sent to r_2, r_2 starts from the *null* hypothesis and applies M_s. This program allows us to test separately the effect of selection of counter examples by a critic on the learning process.
2. *Smile$_O$*: n examples are sequentially sent to the n agents $r_1 \dots r_n$ of a MAS, i.e. agent r_i receives only example e_i from the environment, and applies M_s to the current hypothesis previously adopted from agent r_{i-1}. In this program the extensive exploration effect is added to the mere selection effect of *Smile$_V$*.

Hereunder we give the accuracy results for these experiments, compared with those of the original SMILE (with n agents MAS) and with Smile-1 (the single agent learner receiving the n examples). In these results, n is set to 200 for M11 and to 100 for Xor3_25. These values correspond to the maximal increase in accuracy obtained by a MAS with respect to a single agent.

Measure	Smile-1	Smile$_V$	Smile$_O$	SMILE
Accuracy (M11)	0.874	0.902	0.948	0.950
Accuracy (Xor3_25)	0.63	0.76	0.919	0.952

The standard deviation on accuracy ranges from 0.04 to 0.06 (M11, decreasing from left to right) and from 0.08 to 0.14 (Xor3_25). It appears that:

Part of the accuracy improvement with respect to Smile-1 is due to the selection effect. When applying Smile$_V$ to M11, the learner agent r_2 receives about 70 counter examples and the resulting hypothesis outperforms Smile-1 by $\simeq 0.028$. Regarding Xor3_25, selection alone increases the accuracy by $\simeq 0.13$,

A larger part of the accuracy increase is due to the extensive exploration effect. Smile$_O$ outperforms Smile$_V$ by $\simeq 0.046$ on M11 and by $\simeq 0.16$ on Xor3_25.

So, neglecting the residual effect of randomness (which causes SMILE to slightly outperform Smile$_O$ on Xor3_25 problem), we argue that the accuracy increase regarding the (hard) Boolean problems, is mainly due to the selection of counter examples by critics, and because of the larger exploration of the search space.

4 BROADCAST AND FORGETNESS

If time is crucial, then the sequential nature of SMILE is a big flaw. It does not take advantage from the possible simultaneous actions of the agents. In what follows, we propose a protocol in which the learner agent *broadcasts* the revised hypothesis to critics, so speeding-up the revision process. Note that whereas the sequential protocol ensures that each counter example sent by a critic provokes a new revision, this is not the case with broadcast protocols, thus suggesting a higher redundancy. As experiments suggested that high redundancy reduces the accuracy gain, it would be important to avoid the duplicating examples as much as possible. *Forgetness* will provide a way to ensure that redundancy is at most local and temporary.

Broadcast Broadcasting is obtained by replacing the bilateral $1-1$ interactions of the sequential revision process by $1 - n$ interactions. Each time it modifies its hypothesis, the learner agent broadcasts it to all other agents, and gets answers from each of them: either *accept* or all the *counter examples* of the hypothesis it possesses. The learner then uses all the received counter examples at once for revising its hypothesis. As a new hypothesis is produced, the process iterates, until all the critics reply with an *accept*. Such a *broadcast protocol* reduces the number of internal revisions (while extending their magnitude). Here is a small example which illustrates broadcast revision.

Example 1 *There are three agents in the MAS. A positive example e_{10}^+ is collected by agent r_1, and e_{10}^+ is not covered by the current hypothesis H_0. The revision H_1 is then proposed to agents r_2 and r_3. r_3 then answers by the negative counter example e_3^- (e_3^- is covered by H_1) while r_2 accepts the revised hypothesis. r_1 then revises H_1, and so H_2 is proposed to r_2 and r_3. r_2 accepts H_2 but r_3 sends e_9^+ and e_7^- as counter examples to r_1. r_1 then revises H_2 and proposes H_3 to r_2 and r_3 who both accept it. Now H_3 has been accepted by the two critics, so r_1 notifies mas-consistency of H_3 to r_2 and r_3 who then both adopt H_3 as their new current hypothesis. r_1, r_2 and r_3 are now all mas-consistent and so the whole MAS is consistent.*

Forgetness Both the sequential and the broadcast protocols can be modified to ensure that there is no redundant example in the system after a revision. All examples received from other agents during the revision (given as *counter examples* of the proposed hypothesis) are only stored in a temporary memory and forgotten at the end of the revision. Each agent thus only retains in its memory, the examples it received from the environment.

Protocols summary We denote sequential protocols by 'S', broadcast ones by 'B', and add 'f' when Forgetness is used. When referring to a protocol in a society of n agents, we will add '-n' to its name. We shall thus discuss four protocols, which are all strongly mas-consistent (proof similar to that in [1]):
Sequential (S), the sequential protocol (SMILE)
Forgetness (Sf), the sequential protocol with Forgetness
Broadcast (B), the broadcast protocol
Broadcast+Forgetness (Bf), the broadcast protocol with Forgetness

Using the same method and conditions as in 2.2, we tested these protocols with societies of 10, 20, 50 100 and 200 agents on several problems: the boolean ones such as Multiplexer-11, its variant with 9 irrelevant attributes (M11, M11-9), and different Xorp_m problems (Xor3-25, Xor5-5 and Xor5-15) as well as 5 UCI problems (heart-statlog, kr-vs-kp, breast-w, tic-tac-toe, voteMp).

Redundancy Protocols with Forgetness ensure that redundancy equals 1 after each revision. So here, we only consider the two protocols without Forgetness. Figure 1 depicts a *normalized redundancy*, i.e. the ratio of redundancy to the number n of agents, when learning the Xor5-15 problem with various n-MAS.

Note that the broadcast protocol exhibits a slightly inferior highest redundancy than the sequential variant, but a greater final one. Here, an important observation is that *broadcast does not significantly*[7] *increase redundancy during the most active period of learning*.

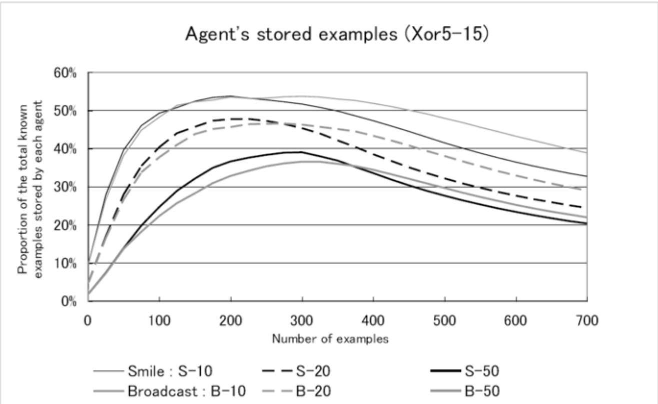
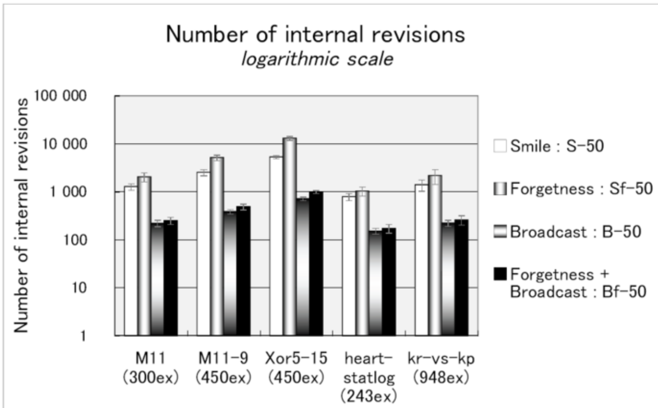


Figure 1. Comparison of normalized redundancies when learning with broadcast and sequential protocols

Computations In order to roughly estimate the level of exploration of the search space, we compare the four protocols on the basis of the *total number of internal revisions* that have been produced by the MAS during learning, as shown in Figure 2(top) . We can see that, on one hand, Forgetness increases the number of internal revisions quite drastically, whereas on the other hand, *broadcast* reduces it in important proportions. When both are combined,it seems that the factorization operated by Broadcast balances the increase of internal revisions caused by the rediscovery of forgotten counter examples. Hence, the combination of Broadcast with Forgetness leads to a low number of internal revisions.



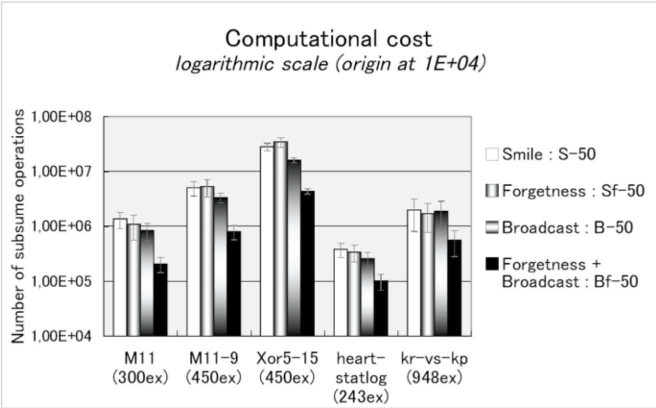


Figure 2. Comparison of the four protocols: total number of internal revisions (top) and computational cost (bottom)

Now, we consider as computational cost, the total number of basic subsumption operations made in the MAS, resulting from both internal revisions and criticisms. Figure 2(bottom) shows the results for three boolean problems and two UCI ones, for each protocol, with 50 agents. It appears that *Broadcast+Forgetness does significantly less overall computations than the other three*. This observation may be explained as follows. Even though the broadcast protocol B does not make a lot of internal revisions, it tests each hypothesis against the example memory of each critic (this memory fills quickly during the active part of the training), a costly process. It is thus one of the most expensive protocols. However note that a large part of these computations, namely those resulting from criticisms, are performed in parallel. Sequential protocols S and Sf solicit agents as critics one by one, so resulting in a smaller total number of criticisms because a hypothesis stops being criticized as soon as a counter example is

[7] even in the case where there a statistically relevant difference (unpaired t-test), the ratio is less than 1.33 in all problems except breast-w and voteMp.

found, but they perform much more internal revisions, and as a result have also rather bad performance. As it has no redundancy, *Broadcast+Forgetness* avoids the very high cost of criticisms of protocol B, and given its small number of internal revisions, is clearly the most computationally efficient protocol. Here again some of these computations are performed in parallel.

Communications Regarding communications, we measure both the total number of messages that are sent and, since in broadcast variants a single message can contain several counter examples, the total size of data that is exchanged between agents. Figure 3 shows the results on total size of data exchanged for each protocol with 10, 50 and 200 agents MAS on the Multiplexer-11 problem (200 examples are sent to the MAS). Only the results on the number of messages are discussed. Overall, *Forgetness* increases the communication cost for both measures: as counter examples are forgotten, they might have to be communicated again. The communication cost of Sf soon becomes prohibitive as the number of agents increases. The Bf protocol suffers much less than Sf from the increase in communication cost since Broadcast greatly reduces the number of produced hypothesis, and its overall cost is only marginally higher than that of the sequential protocol. Furthermore, as it factorizes the messages by transmitting several counter examples at once, it is even better regarding the number of messages.

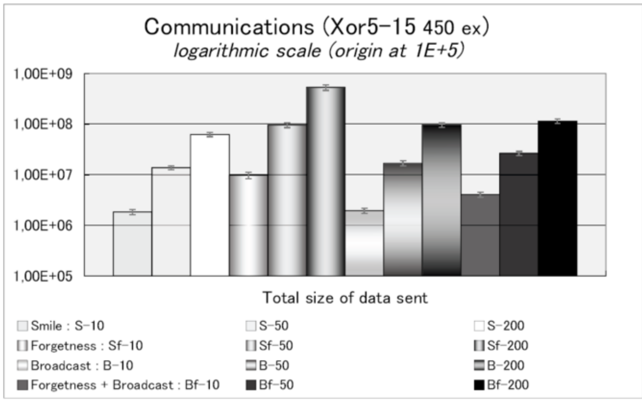


Figure 3. Communication Cost

Accuracy Regarding accuracy, we present here the main observations resulting from the experiments. They are verified on most boolean problems.Tentative explanations are presented in the next section. We made four main observations:

Broadcast protocol (B) accuracy increases with the number of agents (O_1), as illustrated by Figure 4(top). It is significantly lower than SMILE with 10 or 20 agents, but becomes one of the most accurate protocols for high number of agents (see Figure 5(bottom)).

Sequential protocol with Forgetness (Sf) has an accuracy that decreases with the number of agents (O_2), (see Figure 4(bottom)). While very good with 10 agents (see Figure 5(top)), the accuracy quickly deteriorates though still better than that of a single agent.

Protocols with Forgetness (Sf, Bf) usually performs very well with few agents (O_3). They outperform the protocols without Forgetness when the number of agents is small(see Figure 5(top)).

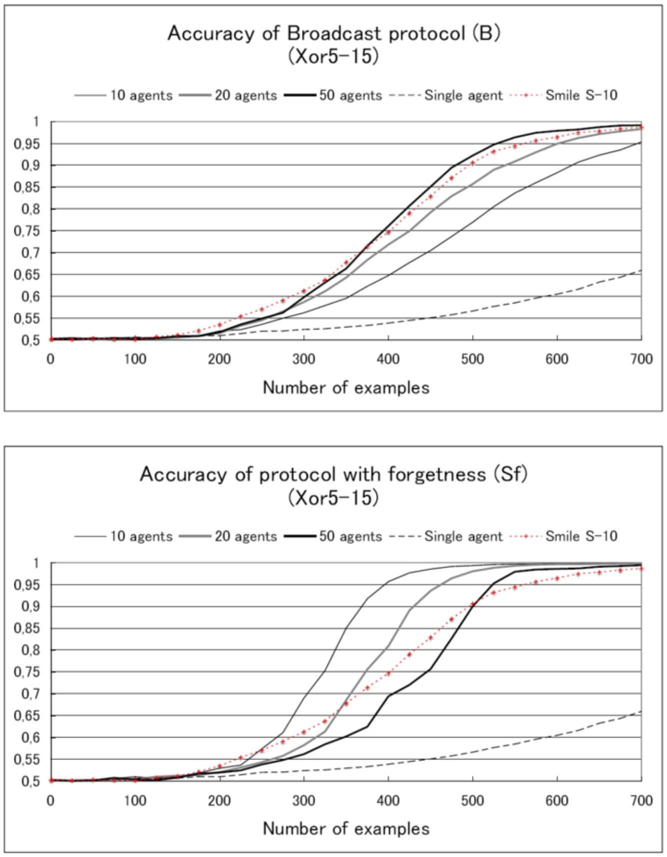


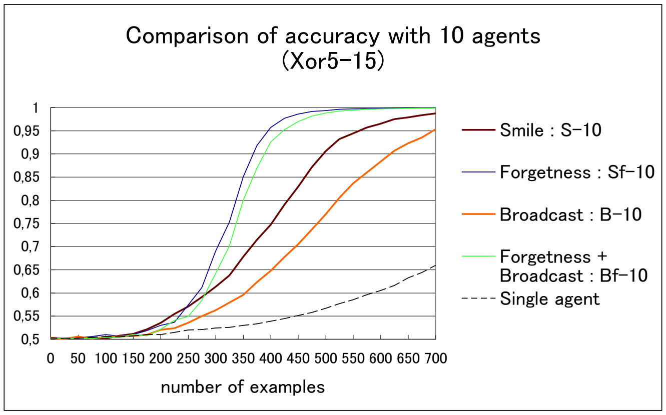
Figure 4. Accuracy : Broadcast protocol B (top) and Sequential+Forgetness Sf (bottom)

Broadcast protocols (B, Bf) usually perform very well when the number of agents is very high (O_4), as seen on Figure 5(bottom). Especially, the broadcast protocol with Forgetness benefits from both mechanisms, and performs consistently well on all problems.

5 HOW DO FORGETNESS AND BROADCAST AFFECT LEARNING?

In this section, we discuss how the Broadcast and Forgetness affect the learning process. *Forgetness* obviously affects the number of examples present in the memory of each agent. *Broadcast* deeply affects the next internal revision to be performed, as before each internal revision, all counter examples of the current hypothesis are gathered.

In section 3 we have related the observed accuracy improvement while learning together, to both the intensive exploration of the search space and a selection effect. Here we argue that, more precisely, the improvement is due to a *better* exploration of the search space both *quantitatively* (through many internal revisions) and *qualitatively* (through focussing on interesting hypotheses). For the increased computational resources to be efficient, internal revisions should, as much as possible, explore new areas of the search space rather than going over and over on the same trajectories. It is important to both (i) take benefit from previous exploration (*stability*) and (ii) allow enough variations to explore new trajectories (*variability*). There must be a good balance between stability and variability as too



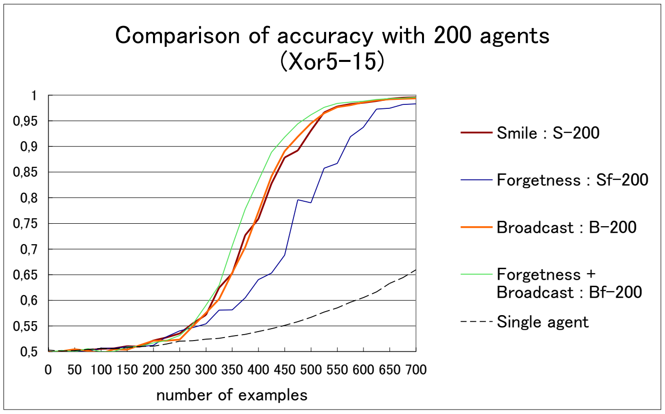


Figure 5. Accuracy : comparison of the four protocols with 10 agents (top) and 200 agents (bottom)

much variations might fully cancel the benefit of previous hypotheses (reducing the accuracy increase simply to a selection effect), while not enough variations might prevent the discovery of better hypotheses.

Then, we can interpret the results by considering each example as a constraint reducing the space of possible solutions. When the learner has only few examples, as it happens when redundancy is low, its internal revisions are less constrained, and the variability is high. Likewise, when it gathers counter examples one by one, it makes more internal revisions with only one more constraint between each revision (increased variability), whereas if it gets a batch of counter examples, it makes less revisions, each of them being more constrained (increased stability). To summarize, using Forgetness or increasing the number of agents increases variability, whereas using Broadcast or decreasing the number of agents increases stability.

It explains O_1, as Broadcast without Forgetness with few agents has an excess of stability, which is compensated when a greater number of agents increases variability. Likewise, with fewer number of agents (high stability), Forgetness is useful to add more variability (O_3). Reversely, excess of variability causes the sequential protocol with Forgetness to give bad results if the number of agents is high (O_2). The increased stability of the broadcast protocol is thus useful to counteract the variability caused by the large societies of agents (O_4). Using both Broadcast and Forgetness is thus a very good option as it introduces a good balance between stability and variability of the hypotheses regardless of the number of agents.

6 CONCLUSION

This article adresses the problem of collaborative learning in a MAS using interactions between agents [8], and more precisely it adresses collaborative concept learning[10, 3]. It builds up on some existing work that demonstrated an improvement of the accuracy with multiple agents in the case of sequential learning process [1]. Our work goes ahead and proposes to take full advantage of multi agency by considering our agents as autonomous interacting entities performing parallel actions while organised (or distributed) over a network. Thus, this paper introduces Broadcast and Forgetness mechanisms to add parallelism while keeping a low redundancy. It appeared that using the two mechanisms simultaneously (protocol Bf) provides an efficient process, minimizing the overall number of computations without using much more communications than SMILE (as a result, learning time is reduced even without considering the effect of parallelization). It can thus be used in larger societies of agents. Moreover, our Bf protocol fully preserves the accuracy improvement, and even improves it further on some difficult problems.

We have considered here the broadcast revision in a fully connected network. Previously, collaborative MAS learning has been extended to cope with situations in which agents only communicate with their neighbors [2], but only in a sequential way. A major perspective is then to extend the broadcast protocols to much more realistic networks (e.g. sensor networks). Note that in this case the broadcast protocol will result in having various current hypotheses propagating in the network and an agent will need to have some way to rank or merge the hypotheses, and so the interactions will have to be much more sophisticated. Finally, from a *computational learning theory* perspective, collaborative concept learning as proposed here has some links with the theory revision with queries[6]: the critics defined here may be seen as the *incomplete oracles* answering the *equivalence queries*. A deeper comparison should be conducted in this perspective.

REFERENCES

[1] G. Bourgne, A. ElFallah-Seghrouchni, and H. Soldano. Smile : Sound multi-agent incremental learning ;-). In *AAMAS'07*, pages 164–171. IFAAMAS, 2007.

[2] G. Bourgne, A. ElFallah-Seghrouchni, and H. Soldano. Learning in a fixed or evolving network of agents. In *ACM-IAT'09*. IEEE, 2009.

[3] W. Davies and P. Edwards. Distributed learning an agent-based approach to data-mining. In D. Gordon, editor, *Working Notes of the ICML '95 Workshop on Agents that Learn from Other Agents*. Tahoe City, CA, 1995.

[4] C. B. D.J. Newman, S. Hettich and C. Merz. UCI repository of machine learning databases, 1998.

[5] S. Esmeir and S. Markovitch. Lookahead-based algorithms for anytime induction of decision trees. In *ICML'O4*, pages 257–264. Morgan Kaufmann, 2004.

[6] J. Goldsmith, R. H. Sloan, B. Szorenyi, and G. Turan. Theory revision with queries: Horn, read-once, and parity formulas. *Artificial Intelligence*, 156(2):139–176, 2004.

[7] A. Guerra-Hernández, A. ElFallah-Seghrouchni, and H. Soldano. Learning in BDI multi-agent systems. In *CLIMA IV*, volume 3259, pages 218–233, 2004.

[8] S. Ontañón and E. Plaza. Case-based learning from proactive communication. In *IJCAI'07: Proceedings of the 20th international joint conference on Artifical intelligence*, pages 999–1004, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

[9] L. Panait and S. Luke. Cooperative multi-agent learning: The state of the art. *Autonomous Agents and Multi-Agent Systems*, 11(3):387–434, 2005.

[10] S. S. Sian. Extending learning to multiple agents: Issues and a model for multi-agent machine learning. In Y. Kodratoff, editor, *EWSL-91, LNCS 482*, pages 440–456. Springer, 1991.