Collaborative Agent-Based Learning for Brain Tumour Diagnosis

Xavier RAFAEL-PALOU [a,b], Michael ROVATSOS [b], Mariola MIER-PEREZ [b], and Magi LLUCH-ARIET [a]

[a] *Microart, Parc Cientific de Barcelona, 08028 Barcelona (ES)*

[b] *School of Informatics, The University of Edinburgh, Edinburgh EH8 9AB (UK)*

Abstract. This paper presents the collaborative agent-based learning subsystem of HealthAgents, a multiagent distributed decision support system for brain tumour diagnosis. The subsystem aims to boost the performance of the independent and heterogeneous classifiers in spite of the limited data transfer conditions prevailing in the system. The susbsystem is composed by local autonomous agents which are interacting among them, following an existing collaborative learning model. The different aspects and decisions dodged during the adaptation of this model are described in addition to the results of its initial evaluation with the data of HealthAgents. Significant increments of classification performance attained by the learning agents demonstrate the potential benefits of this subsystem.

Keywords. Distributed data mining, Collaborative Learning, MultiAgent systems

1. Introduction

The HealthAgents (HA) system [1] attempts to give support in the decision-making process of medical experts in the diagnosis of distinct types of brain tumours. This task is traditionally done using established but partially subjective criteria from the histological examination of a brain biopsy. Recent technical advances have improved the diagnosis using non-invasive methods like image radiology or magnetic resonances spectroscopy (MRS). Thus, HealthAgents coalesces these technical advances with advanced data mining techniques to assist clinicians in their complex tasks.

For this purpose HealthAgents aims to join different clinical centers in an open but secure network in order to gather large quantities of data (brain tumour cases) for better results in data mining techniques. However, the data transference in medical domains is constrained by the clinical centers since those maintain restrictive data security policies, due to the importance in the privacy of the patient data and since this data constitute the asset of these centers for present and future researches. Thus, the data mining process of HealthAgents presents an isolated distributed design i.e. different producer nodes (where the machine learning tools are built) may exists and those may have distinct data permissions since the data rights are always managed by the local data centers. This condition entails the outcome of multiple heterogeneous classifiers from different producers made from different quantities of data, using different types of data, or running different learning algorithms which may discriminate among different tumour classes.

Although the data sharing restrictions, there is room for improvement in the classification accuracy of independent classifiers through the use of complex distributed data mining techniques which may combine, integrate or reuse the predictive models of the system. This paper focuses in this issue and presents the adaptation of an existing agent-based learning model [6] into the specifications of the HealthAgents system.

This paper is organised as follows. Section 2 describes some general solutions to our distributed data mining problem stressing in the solution provided for HealthAgents. Section 3 describes the adaptation and implementation of this solution in HealthAgents. Section 4 presents the evaluation of the solution. Finally, section 5 draws conclusions from the analysis of results and suggests future research directions.

2. Related Work

This work is concerned with one of the distributed data mining interests like the achievement of best classification rates in distributed environments[8]. This task implies obtaining classification models as accurate as possible. Common solutions in distributed environments for data classification tasks consist on gathering all data in a central data-warehouse and subsequently apply traditional machine learning techniques to infer accurate predictive models. Nevertheless, data centralised strategies are not always feasible in any environment due to local data are quickly changing, too complex to communicate, too large or local sites may not be willing to reveal private data even they are cooperative overall.

Multiagent technology fits in this sort of distributed and open domains offering capabilities which improve traditional approaches, i.e. autonomy of data nodes, reuse of data knowledge or self-directedness of the learning process in heterogeneous environments. Existing Multiagent learning approaches in this direction are [2,3,4], those systems use distributed technologies for applying data mining algorithms to learn global models from local learning processes. However these approaches overlook the autonomy of local learning processes, the decentralisation of system control, and the local learning heterogeneity of the processes.

A recent multiagent learning framework (MALEF)[7] attempts to cover these aspects. In MALEF the learning process is defined as the iteration of learning steps (t) in which each step provides an opportunity to engage in communication with a different learning process before initiating next step. Thus the learning step is defined as a tuple $l_t = \langle D_t, H_t, f_t, h_t, g_t \rangle$, where D represents the data training set, H the hypothesis space, f the training function and its parametrisation, h the learning hypothesis (for the purposes of this paper, a classifier), and g the quality function which evaluates the current performance of the classifier.

The role of an agent is seen as an entity which performs the iterative learning process which in each step the agent(i) establishes a collaboration with other agent (j) through the transference from the contacted agent(j) of some kind of learning knowledge and the integration of that kind of knowledge within the knowledge of the initiator agent(i).

A continuation of this work is found in [6] where the notion of learning steps (l_t) is kept although the sense of collaboration among agents is refined by specifying a model of behaviour for the learning agents. Four distinct intuitive actions compose the new collaborative model: *Neighbour Selection, Knowledge Integration, Performance Evaluation* and *Learning Update*. Furthermore in this study specific operations for knowledge integration were defined for environments where data sharing is restricted or even prohibited. An example is the *merge tree* operation in which the learners integrate their hypothesis. The evaluation of this approach shows a dramatic increase of classification accuracies on the local learners being nearby to the performances of centralised solutions.

3. Collaborative agent-based learning for HealthAgents

Following the positive results obtained in [6,7] by using decentralised, autonomous and collaborative agents for learning in open, distributed and heterogeneous environments, we have adapted this approach into the HealthAgents system to attempt the increment of the classification performance of the distributed heterogeneous classifiers in the network.

3.1. Design overview

We have defined a new type of agents (*learners*) for the collaborative learning subsystem of HealthAgents. These agents are in charge of performing the collaborative learning model defined in [6]. This learning process entails several interactions with this type of agents, but also with the rest of the agents in the system.

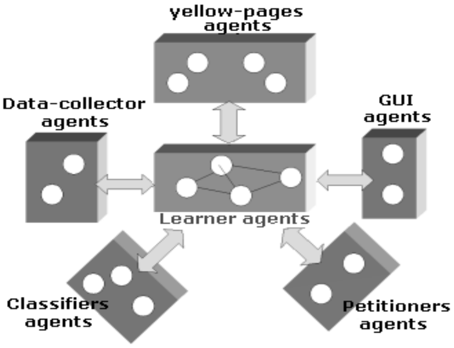


Figure 1. Learner agents interactions

In the figure 1, we can see how learners interact with the yellow pages agents. Those agents provide a repository with information (properties and capabilities) of the running agents. Through the Yellow pages, other agents can search-for and retrieve location and properties from the published agents. The learners also interact with other learner, classifier and data collector agents along the collaborative learning process as will be detailed in the next section. Finally the learners call to classifier, petitioner and GUI agents for the classification of instances.

We may remark the different role carry out by classifier and learner agents in HA. The first ones come from DM analysis and implement specific discriminant models. The learners correspond to a particular classifier with whom share initially the same *classification properties* (e.g. type of input data, question to solve, learning technique and classification performance) but with collaborative learning capabilities.

All the communications performed by the learners with other agents of the system are performed through the HealthAgents Language, HAL [1]. The HAL is a communication language that uses an ontology to define the constructs (words) that can be used in messages (sentences) sent around the network. By imparting the ontology on the agent communication, the agents are able to interact with one another. In the figure 2 we show an example of a partial communication among learner agents.

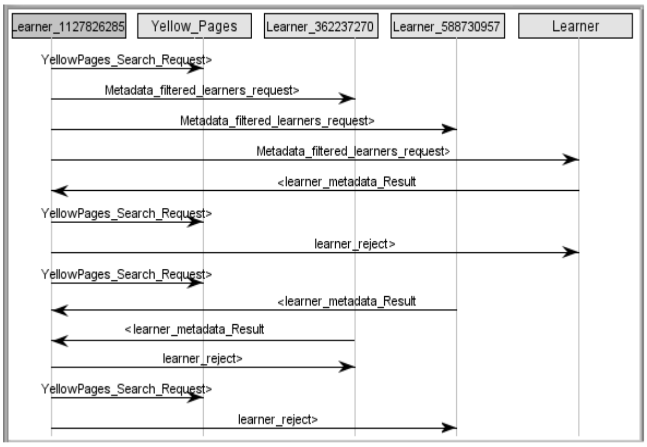

Learner_1127826285
Yellow_Pages
Learner_362237270
Learner_588730957
Learner
YellowPages_Search_Request>
Metadata_filtered_learners_request>
Metadata_filtered_learners_request>
Metadata_filtered_learners_request>
<learner_metadata_Result
YellowPages_Search_Request>
learner_reject>
YellowPages_Search_Request>
<learner_metadata_Result
<learner_metadata_Result
learner_reject>
YellowPages_Search_Request>
learner_reject>

Figure 2. Snapshot of interactions performed by a learner agent during its collaborative learning process

3.2. Functionalities

Following we describe the main functionalities of this subsystem:

1. *Learner agent startup*. This functionality permits to create a new learner agent and starts its learning behaviour. The creation of a learner occurs automatically and after the creation of a classifier. In this manner, the launch of the learner requires no intervention of the user. Two different actions compose this functionality:

   (a) Learner agent setup. This process initialises and registers in the YP the meta-information of the learner to perform its learning behaviour. This meta-information will be the *classification properties* from the classifier which comes from, the *configuration properties* to parametrise the collaborative learning(neighbour selection, integration method, update criteria and data test set), the *state* of the learner(current version, accuracy, list of the ensemble of classifiers) and finally the *array of classification outputs* from the test set.

   (b) Startup the agent's behaviour. The learner creates an asynchronous and iterative process in charge of performing the collaborative learning.

2. *Learner classification*. Any learner is able to classify new instances. Thus, the learner will predict the class of a given instance from calling to classify that instance to all different classifiers the learner has found to be combined with, and merging all the outputs using the integration operation, which was previously configured.

3.3. *Learner agent's behaviour*

The next flow diagram (fig.3) specifies the implementation of the collaborative learning model [6] done for HealthAgents:

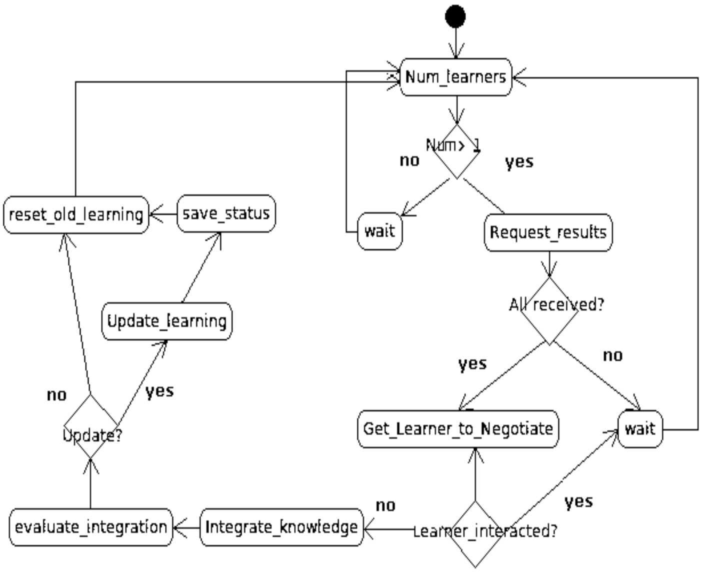


Figure 3. Learning step implementation for a learner agent of HealthAgents

3.3.1. *The neighbour selection stage*

Each learner will attempt to achieve the properly agent to interact with. Thus, each learner asks for (Num_learners method) the available learners in the system through a yellow pages HAL request message: *Yellow_Pages_search_request* (fig.2). The Yellow Pages obtains all learners which satisfies the conditions of having *same type of training data*, *output classes* and *same integration operation*, to apply the integration operations previously defined.

When the learner receives the list of suitable learners to interact with, (Request_result method), the learner will ask to them for their learning state information, with the *metadata_filtered_learners_request* HAL message. In the meantime, the learner will be waiting during a certain period to receive all the results. Alternatively, some learners at that stage may have initiated the communication with others learner agents to change their internal state. This could make that the metadata they initially send would not be correct once obtained by the receiver learner. Therefore for avoiding this inconvenient, we have specified an internal attribute (state) for controlling multi access to shared information.

Once the metadata from different learners arrives to the initiator learner, a local search criteria is performed, (Get_learner_to_negotiate method), for obtaining the learner to interact with. In particular, the *greedy accuracy-based search* has been taken as the best results in short time it seems to achieve. This strategy looks for the best learner (l_k)

in terms of highest classification accuracy(Acc) from all participants (L). The accuracy estimation will be computed as follows:

$$Acc = \frac{|correctly\ classified\ cases|}{|all\ cases|} \quad (1)$$

Thereupon, the learner send a HAL message (*learner_reject*) to all participants to conclude the current collaboration allowing them to engage in new collaborations.

3.3.2. The integration stage

The integration operations for merging classifiers has been constrained to the HealthAgents specifications of heterogeneity in classifiers types and restrictions in data transferring. Furthermore, we have looked common, diverse and relatively fast operations which does not impact dramatically on the efficiency of the system.

We have implemented different integration methods based on *arithmetical combination of posterior probabilities[5]* (maximum, minimum and average). Regarding the average that is computed as follows:

$$\mu_j(x) = \frac{1}{n}\sum_{i=1}^{n} d_{i,j}(x)$$

where $d_{i,j}$ is the posterior probability of the classifier i over the class j. Additionally, we have implemented integration methods based on voting like *majority voting* [5] and also a simple evolution of this one, *weighted majority voting* by adding a discriminant function for class j obtained through weighted voting:

$$g_i(x) = \sum_{i=1}^{L} b_i d_{ij}$$

where b_i is a coefficient for classifier i and g_i the resulting class for the instance x. We have defined the value of the coefficient as the accuracy of the classifier. For convenience we have normalised the coefficients, $\sum_{i=1}^{L} b_i = 1$.

The implementation of these operations in the learners has been done by transferring and storing arrays of classification results from the classifiers to belong to the ensemble. This implementation is in accordance with HealthAgents security policies since local raw data is not needed to be transferred. However this implementation constraints the evaluation to a same test dataset for all learners and adds some overloading due to the transfer of classification outputs.

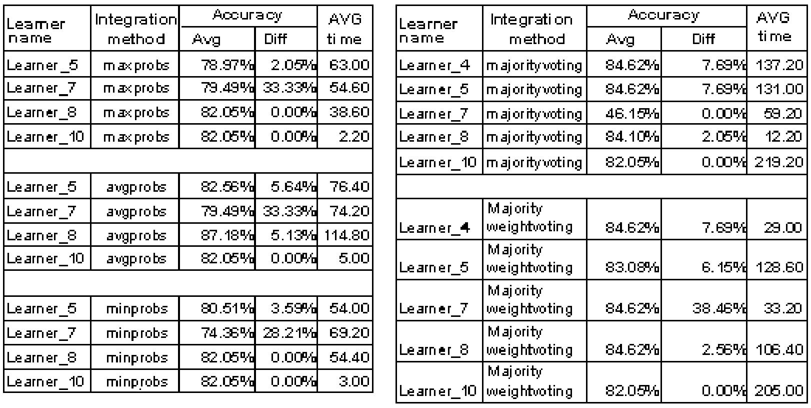
3.3.3. The evaluation and updating stages

After the knowledge integration stage, it is necessary to evaluate the performance of the resulting classifier. For this, we use the formula presented in section 3.3.1 for which it is needed the *integration operation* of the learner, their distinct arrays of *classification outputs* of the ensemble of classifiers and an independent data test set in order to avoid biased classification accuracy measurements.

In order to decide whether to update the new learning knowledge, we have applied a simple greedy in accuracy mechanism: $g_{i+1} > g_i$. If the condition is satisfied a new learner state is created updating the current accuracy of the learner, the learners visited and the list of arrays of output classification results of the ensemble of the learner. Otherwise, the learning information received from the selected learner will be removed.

4. Results

To test the collaborative learning subsystem, we used an independent test dataset (40 instances) which represented 1/5 of the size of the training set for building the classifiers of HealthAgents. All learners were configured with the methods described in previous section. However, since we proposed several integration operations, we conducted different experiments changing this operation in the learner configurations. The learners were created for five distinct HA classifiers. Four of them (5,7,8 and 10) came from linear discriminants (LDA) techniques and they had an initial classification accuracy of (76.92%, 46.15%, 82.05%, 82.05%). The last classifier (4) came from nearest neighbour (K-NN) technique and which had 76.92% of performance. Since this last classifier didn't outcome posterior probabilities, we prepared two experimental scenarios, one testing integration operations based on posterior probabilities, and the second using all available classifiers but with integration operations based on voting methods. The experiments were repeated 10 times in order to assure consistency in the results.



Leamer name	Integration method	Accuracy		AVG time
		Avg	Diff	
Learner_5	maxprobs	78.97%	2.05%	63.00
Learner_7	maxprobs	79.49%	33.33%	54.60
Learner_8	maxprobs	82.05%	0.00%	38.60
Learner_10	maxprobs	82.05%	0.00%	2.20
Learner_5	avgprobs	82.56%	5.64%	76.40
Learner_7	avgprobs	79.49%	33.33%	74.20
Learner_8	avgprobs	87.18%	5.13%	114.80
Learner_10	avgprobs	82.05%	0.00%	5.00
Learner_5	minprobs	80.51%	3.59%	54.00
Learner_7	minprobs	74.36%	28.21%	69.20
Learner_8	minprobs	82.05%	0.00%	54.40
Learner_10	minprobs	82.05%	0.00%	3.00

Learner name	Integration method	Accuracy		AVG time
		Avg	Diff	
Learner_4	majorityvoting	84.62%	7.69%	137.20
Learner_5	majorityvoting	84.62%	7.69%	131.00
Learner_7	majorityvoting	46.15%	0.00%	59.20
Learner_8	majorityvoting	84.10%	2.05%	12.20
Learner_10	majorityvoting	82.05%	0.00%	219.20
Learner_4	Majority weightvoting	84.62%	7.69%	29.00
Learner_5	Majority weightvoting	83.08%	6.15%	128.60
Learner_7	Majority weightvoting	84.62%	38.46%	33.20
Learner_8	Majority weightvoting	84.62%	2.56%	106.40
Learner_10	Majority weightvoting	82.05%	0.00%	205.00

Figure 4. Summary of learners performance using different merging operations

The figure 4 shows the results regarding the tests done in the two different scenarios. In these tables we can observe for each learner in the experiment its integration operation, the average of classification accuracy and the difference in average of classification performance, and finally the average in time since the last interaction performed by the learner.

From those results, we may appreciate that most of the learners achieve substantial increments of classification performance regarding their initial performance, e.g. 38.46% with *majority weighted voting* or 33.33% with *average of the posterior probabilities*. This is specially rellevant in the learner_7 which initally had the lowest performance but after its collaborative process it reaches dramatical improvements. However, the learners with best initial accuracy never experiment improvements due to no better collaborations are obtained with the rest of the classifiers. Nevertheless, we must hightlight that learner_8 using the *avgprobs* operation achieves 87.18% of classification accuracy which overcomes the accuracy of the best classifiers.

The figure 5 sumarises the increments in the classification accuracy carry out by the learners. Also, we can observe how the collaborative agent-based learning allows to

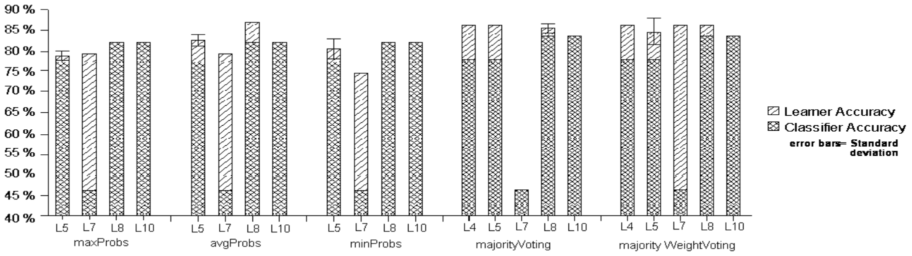


Figure 5. Learners Classification accuracy achieved using different merging operations

achieve a more uniform and effective distributed classification system since the learners raises up the performances of low accurate classifiers. This fact supports the argument that adding autonomy and decentralised learning into machine learning techniques may help to improve the performance of distributed decision support systems.

5. Conclusions

In this paper we have presented the collaborative agent-based learning subsystem of HealthAgents. This subsystem take into account the data sharing restrictions in the environment and propose the use of autonomy and communication among new data miners agents (learners) for boosting their initial performance. Those learners implement an established collaborative agent based learning model, although some particularities have been done in order to adapt it into HealthAgents. The empirical evaluation of this subsystem, using the classifiers of the system, indicates that most of the local learners improve their initial accuracy in short time, even under conditions with small number of classifiers. Although great results were achieved, further and more exhaustive evaluations must be conducted as well as the development of more complex decision making criteria would also be recommended.

References

[1] H. G.-V. et al. HealthAgents: Distributed multi-agent brain tumor diagnosis and prognosis. *Appl. Intell.*, 30(3):191–202, June 2009.

[2] S. B. et al. Papyrus: a system for data mining over local and wide area clusters and super-clusters. In *Proc. 1999 ACM/IEEE conf. on Supercomputing*, page 63, New York, USA, 1999. ACM.

[3] S. S. et al. Jam: Java agents for meta-learning over distributed databases. In *Proc. 3rd Intl. Conf. Knowledge Discovery and Data Mining*, pages 74–81. AAAI Press, 1997.

[4] H. Kargupta, B. hoon Park, D. Hershberger, and E. Johnson. Collective data mining: A new perspective toward distributed data mining. In *Advances in Distributed and Parallel Knowledge Discovery*, pages 133–184. AAAI/MIT Press, 1999.

[5] L. I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, 2004.

[6] X. Rafael and M. Rovatsos. Collaborative agent-based learning with limited data exchange(short paper). In *Proc. 8th Intl. joint conf. on Autonomous agents and multiagent systems*, New York,USA, 2009. ACM.

[7] J. Tozicka, M. Rovatsos, and M. Pechoucek. A framework for agent-based distributed machine learning and data mining. In *Proc. 6th Intl. joint conf. on Autonomous agents and multiagent systems*, pages 1–8, New York, USA, 2007. ACM.

[8] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2 edition, 2005.