Agent-Based Learning Classifier Systems for Grid Data Mining

Manuel Filipe Santos
University of Minho
Campus de Azurém
4800 Guimarães, Portugal
+351 253 510 306
mfs@dsi.uminho.pt

Hélder Quintela
Polytechnic Institute
Av. do Atlântico
4900-348 Viana do Castelo, Portugal
+351 258 819 700
hquintela@estg.ipvc.pt

José Neves
University of Minho
Campus de Gualtar
4700 Braga, Portugal
+ 351 253 604 466
jneves@di.uminho.pt

ABSTRACT
Grid Data Mining tools must be able to cope with very large, high dimensional and, frequently heterogeneous data sets that are geographically distributed and stored in different types of repositories, produced from different devices and retrieved through different protocols. This paper presents an agent-based version of a Learning Classifier System. An experimental study was conducted in a computer network in order to determine the systems' efficiency. The results showed that the model is suitable to be applied in inherently distributed problems and is scalable, i.e., when the latency communication times are not considerable, the system obtains an interesting speedup.

Categories and Subject Descriptors
H.2.8 [**Information Systems**]: Database Management, Database Applications – *Data Mining*.

General Terms
Algorithms, Measurement, Performance, Design, Experimentation.

Keywords
Learning Classifier Systems, Grid Data Mining, Distributed Systems, Machine Learning, Agent-based Systems

1. INTRODUCTION
The use of Learning Classifier Systems (LCS) [1] [2] [3] [4], a machine learning paradigm, as a Data Mining (DM) technique is an attractive idea.

LCS are based on work done by Holland [5] latter refined by Holland and Reitman [6]. Since then, the basic architecture for LCS has not been too much improved, but learning algorithms and their integration with LCS evolved significantly. Today, we have not a unique standard definition for LCS (LCS is basically a concept). We can distinguish two main approaches: the Michigan approach and the Pittsburg approach. Various alternatives have been proposed: Precision Based LCS [7]; Anticipatory LCS [8][9]; Zero Level LCS [7]; Heterogeneous LCS [10]; Corporate LCS [11]; Organizational LCS [12]. The most important implementations, applications and systems are, to name a few: GABIL, NEWBOOLE, COGIN, GA-MINER, REGAL, ALECSYS, ZCS, XCS and DICE [3].

The canonical version of LCS presents some weaknesses on real sized problems. Learning efficiency, execution times and scalability constraints, are the critical aspects that should be addressed in future developments. The critical factors are associated to: the matching operation; the auction operation; the GA; and the conflict resolution operation.

On the other hand, learning in computational distributed environments (e.g., in the dynamic computational load balancing), is one of the areas of the knowledge where the LCS could be applied [3][4], if we will be able to adapt its conceptual structure to the resolution of problems in a cooperative and competitive form, not relinquishing problems of performance and complexity. A form to avoid these limitations encompasses the implementation of the LCS in parallel hardware (e.g., distributed systems, grid computing). This work explores the parallel and distributed implementation of the LCS model, modelling it as an Agent-Based system, designated by LCS_{AB}, in order to construct a software that works as an open platform

2. KDD and Grid Data Mining
The interest in Knowledge Discovery from Databases (KDD) and Data Mining (DM) arose due to the rapid emergence of electronic data management methods. Data Mining, also popularly known as Knowledge Discovery from Databases, refers to the nontrivial extraction of implicit, previously unknown and potentially useful information from data in databases [13]. While DM and KDD are frequently treated as synonyms, DM is actually part of the knowledge discovery process [14].

Datasets tend to be distributed in a geographical manner, because data are naturally collected in a distributed way, and there are technological limitations in the upper limit on the amount of storage it is reasonable to keep in one place. As storage volume grows, the cost per gigabyte drops – but an increasing amount of infrastructure is needed to maintain the whole system. Furthermore, increasing the size of storage necessary increases the average latency to fetch data. Grid Data Mining (GDM) tools

GECCO'06, July 8–12, 2004, Seattle, WA, USA.

must be able to cope with very large and high dimensional data sets that are geographically distributed and stored in different types of repositories [15].

3. Agent-Based Systems

The term agent is a metaphor allowing various definitions, interpretations and taxonomies. Actually, no one of them is universally accepted, despite this some positions are considered referential [16][17][18][19]. One of the most comprehensive definitions of agent was proffered by Jennings & Wolldridge [20]. Applications of agents are widespread and can be found in travel planning [21], e-commerce [22] and even as components of Decision Support Systems [23].

In the context of this work, the AIMA definition prosecuted by [24] was adopted, stating that an agent is an entity capable of perceiving the environment and actuating on that environment. From a software engineering point of view, an agent is an abstraction that allows the construction of more complex systems designated by Agent-Based Systems or Agencies.

4. LCS_{AB} System

Making use of a logical framework, that formally represents a multiagent system [25][26], LCS_{AB} is defined as an agency (Figure 1) composed by several semi-autonomous agents, which interact, to perform the operations inherent in a LCS. Formally, the system can be defined as a tuple $\Xi \equiv \langle C_{LCSAB}, \Delta_{LCSAB}, q_{ma}, q_{ca}, q_{ka1},..., q_{kam} \rangle$, m≥1, where: C_{LCSAB} is the context and corresponds to a logical theory; Δ_{LCSAB} is the set of bridge rules defining the interaction among the systems' components (the agents); q_{ca} is the control agent; q_{kai} corresponds to a knowledge agent, $i=1...m$; q_{ma} is the monitor agent that interacts with the users.

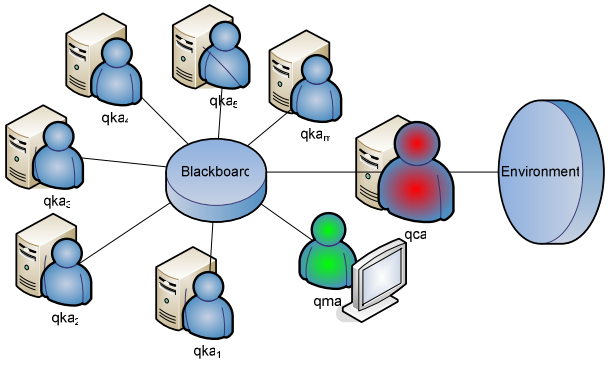

Figure 1. The LCS_{AB} architecture

The social model is static, pre-defined, and incorporated into the system by the developers. The intelligent behaviour, the accuracy, the robustness, the flexibility and efficiency of this system emerges from the agents and their interaction.

4.1 Description of the Agents

The Control Agent q_{ca}

The Control Agent (q_{ca}) is in charge of the system co-ordination process. The q_{ca} associated events are: *initialise*; *detection*; *post_messages; conflict_resolution; effectors; environment_reward; clearinghouse; new_cycle; finalise; message.*

The Knowledge Agents q_{kai}

The Knowledge Agents (q_{kai}) are a form of autonomous, co-operative and competitive entities that interact through a blackboard. The q_{ka}s are autonomous in the sense that they are individual processing units; co-operative in the sense that they make contributions to the solution of a common problem; competitive once their contributions are put into the *blackoard* in a competitive fashion. Any q_{ka} may execute the events: *initialisation*; *matching; auction; tax_collect; accounting; immigration; genetic_algoritm; finalise; message.*

The Social Environment

The interaction among the agents is represented in terms of the bridge rules Δ_{LCSAB} described in the Table 1 (where t stands for the system's time cycle).

Table 1. Some bridge-rules of the system

Bridge-Rule	Description
C_{qca}: *occurs(initialise, t)* ——— *occurs(initialise, t)*	Initialises all the system's agents when the q_{ca} is initialised.
…	
C_{qaki}: *occurs(auction, t)* ——— C_{qca}: *[occurs(post_messages, t) ∧ occurs(conflit_resolution, t) ∧ occurs(effectors, t) ∧ occurs(environment_reward, t) ∧ occurs(clearing_house, t) ∧ occurs(new_cycle, t)]*	After the q_{ka}'s auction phase, the q_{ca} will continue posting the winners classifiers' messages into the message list, and handles the conflict resolution process to choose a single classifier. Then, the action proposed by the selected classifier will be posted on the environment through the effectors and the system waits for a reward. The q_{ca} broadcasts a set of accounting orders to be evaluated by the q_{ka}s on the classifiers. Then, a new cycle may begin.

5. Experimental results

To evaluate the performance and efficiency of the LCS_{AB} system, was developed a prototype in the Sicstus Prolog environment making use of the blackboard toolkit for process communication. A set of tests was carried on a TCP/IP network containing eleven personal computers (PC_1,.., PC_{11}). The tests have been conducted in the following conditions:

- The problem considered is the character recognition (classification);
- Each test comprised the execution of ten complete system cycles ($CT=10$);
- The classifiers are structures containing a condition part formed by eighteen digits and an action part corresponding to a character (A to Z), in the form:

n:::[1,2,2,2,2,#,#,2,1,1,2,2,2,27,2,2,2]===>(eff,["A"]):::[10,0.11]

- The initial population of classifiers was randomly generated for each one of the tests and was distributed in an equitable manner by the Knowledge Agents;
- 42 different scenarios where considered in terms of the distribution of the classifiers to the Knowledge Agents (Table 2);
- The learning and GA parameters considered are presented in the Table 3.

Table 2. Number of Classifiers for each Knowledge Agent

Number of Classifiers	Number of Knowledge Agents				
	2	**4**	**6**	**8**	**10**
120	60	30	20	15	12
480	240	120	80	60	48
1020	510	255	170	127	102
4080	2040	1020	680	510	408
8160	4080	2040	1360	1020	8160
16320	8160	4080	2720	2040	1632
20400	10200	5100	3400	2550	2040

Table 3. Learning and GA Parameters

Parameter	Value	Parameter	Value
Time Limit	10	*Bad Reward*	0
Bid Coefficient	0.1	*Crossover Probability*	1.0
Bid Sigma Constant	0.075	*Mutation Probability*	0.02
Bid Tax	0.01	*Alpha Factor*	0.01
Life Tax	0.01	*Crowding Factor*	3
Bid Parameter 1	1.0	*Crowding Subpopulation*	3
Bid Parameter 2	0.0	*Number of replacements*	2
Effective Bid 1	1.0	*Message List Length*	1
Effective Bid 2	0.0	*Genetic Algorithm Period*	1
Good Reward	1	*Number of emigrants*	1

During the study, the execution time of several operations has been taken for comparison and to determine the speedups attained.

5.1 Global Speedup

The Global Speedup attained by the LCS_{AB} version comparatively to the monolithic version is presented in the Figure 2.

5.2 Efficiency Measurement

We made use of a measurement of the system efficiency, which implicitly incorporates the latency times relatively to the Global Execution Time. Table 4 presents the efficiency values attained by the system in the experiments carried out.

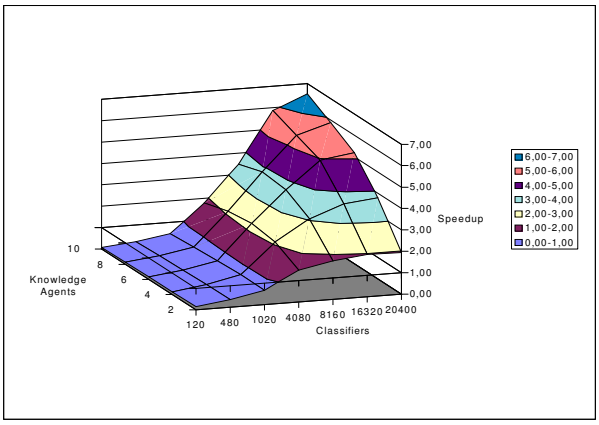

Figure 2. Global Speedup vs Number of Classifiers and Number of Knowledge Agents

Table 4. Efficiency Measurement

Knowledge Agents	Number of Classifiers						
	120	480	1020	4080	8160	16320	20400
2	7%	18%	34%	75%	90%	99%	97%
4	3%	9%	17%	52%	76%	91%	95%
6	2%	5%	10%	37%	61%	83%	87%
8	1%	3%	7%	26%	48%	68%	73%
10	1%	2%	5%	18%	35%	59%	65%

6. Discussion and Related Work

Computationally, the amplitude of the gains increases in the direction of a bigger number of classifiers, and a bigger number of Knowledge Agents. On the other hand it is visible that for a number of classifiers less than 2000 the use of system is not computationally advisable (Table 4). The variation of the speedup relatively to the number of classifiers and the number of Knowledge Agents points to a linear growth, what in a three dimensional referential is approximated by a pyramid.

In the Table 3, the values in the shading area (values above of the 50%), represent the zone of efficiency of the system. The system attained a value of 99% of efficiency when we considered two Knowledge Agents and a number of 16320 classifiers.

The GA operation is executed in parallel by the Knowledge Agents using the island paradigm [27], complemented with an immigration mechanism. This is very useful for homogeneous populations. For heterogeneous environments, where the datasets are of different nature, another solution should be addressed.

The registered AG Execution Times showed a substantial speedup for all the settings. This issue is very important due to the impact the GA has in the learning task, opening room for an effective application in GDM problems.

This work can be compared with the proposal made in [27], the biggest differences are positioned at the ML processes level. In the LCS_{AB} approach the GA and auction operations were transferred to the Knowledge Agents and implemented in a decentralized form.

7. Conclusions and further work

This paper presented the LCS_{AB} system, an agent-based model for LCS. To demonstrate the potentialities of the model, an experimental study was conducted in a computer network in order to determine the systems' efficiency. A measurement of efficiency was introduced to define the best scenarios where LCS_{AB} is more appropriated.

The main contribution of this paper focuses on the applicability and the scalability of the LCS paradigm in distributed DM problems. Further work will be concentrated on the adaptation of the system to Grid Computer environments and on the DM task, i.e., on the learn efficiency from homogeneous and heterogeneous distributed datasets.

8. REFERENCES

[1] Holmes, J., H.: Applying a Learning Classifier System to Mining Explanatory and Predictive Models from a Large Clinical Database. In Proceedings of the International Workshop on Learning Classifier Systems, Paris (2000)

[2] Lanzi, Pier L., Stolzman, W., Wilson, Stewart W.: Learning Classifier Systems From Foundations to Applications. Lecture Notes in Artificial Intelligence 1813, Springer, 2000.

[3] Santos, Manuel Filipe: Learning Classifier Systems in Distributed Environments. PhD Thesis, Universidade do Minho, Portugal (1999)

[4] Santos, M., Neves, J., Alves, V.: The inventive power of Learning Classifier Systems: a contribution to Data Mining. Proceedings of the Third International Conference on Data Mining, Bologna, Italy (2002)

[5] Holland, J.: Escaping Brittleness: The Possibilities of General-Purpose Learning Algorithms Applied to Parallel Rule-Based Systems. In R. Michalski, J. Carbonell, and T. Mitchell, editors, Machine Learning: In Artificial Intelligence Approach, vol. II. Morgan Kaufman Publishers, Inc., Los Altos, CA (1986)

[6] Holland, J., Reitman, J.: Cognitive Systems Based on Adaptive Algorithms. In D. Waterman and F. Hayes-Roth (ed.), Pattern Directed Inference Systems, Academic Press Inc., New York (1987)

[7] Wilson, S. W.: State of XCS Classifier System Research. In Proceedings of the Second International Workshop on Learning Classifier Systems, USA (1999)

[8] Soltzman, W.: Latent Learning in Khepera Robots with Anticipatory Classifier Systems. In Proceedings of the Genetic and Evolutionary Computation Conference, Orlando, Florida, Morgan Kaufmann Publishers, San Francisco California (1999)

[9] Hoffmann, J.: Probleme der Begriffsbildungsforchung: Von S-R Verbindungen zu S-R-K Einheitein. Sprache und Kognition, 11(4), (1992) 223-238

[10] Lattaud, C.: Non-Homogeneous Classifier Systems in a Macro-Evolution Process. In Proceedings of the Second International Workshop on Learning Classifier Systems, USA (1999)

[11] Tomlinson, A., Bull, L.: On Corporate Classifier Systems: Increasing the Benefits From Rule Linkage. In Banzhaf, W., Daida, J., Eiben, A. E., Garzon, M. H., Honavar, V., Jakiela, M., Smith, R. E., (ed.), Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-99), Morgan Kaufmann Publishers, USA (1999)

[12] Wilcox, J. R.: Organizational Learning Within A Learning Classifier System. Msc Thesis, University of Illinois, USA (1995)

[13] Fayyad U., Piatesky-Shapiro G., Smyth P.: From Data Mining to Knowledge Discovery: An Overview. In Fayyad et al. (eds) Advances in Knowledge Discovery and Data Mining. AAAI Press/The MIT Press, Cambridge MA, (1996) 471-493

[14] Zaïane Osmar R.: Principles of Knowledge Discovery in Databases. University of Alberta;, USA (1999)

[15] Brezany, P., Hofer, J., Tjoa, A.M., Wohrer, A.: Towards an Open Service Architecture for Data Mining on the Grid. 14th International Workshop on Database and Expert Systems Applications (DEXA'03) (2003)

[16] Jennings R., Wolldridge J.: Agent Technology Foundations, Applications and Markets. Springer, Berlin Heidelberg New York (1998)

[17] Nwana, H. S., Ndumu, D. T.: A Brief Introduction to Software Agent Technology. In Jennings, N. R., Wooldridge, M. J., eds, Agent Technology Foundations, Applications, and Markets, Springer (1998)

[18] Weiss, G.: Multiagent Systems A Modern Approach to Distributed Artificial Intelligence. MIT Press, Cambridge MA, 1999.

[19] Ferber J.: Multi-Agent Systems – An Introduction to Distributed Artificial Intelligence. Addison-Wesley (1999)

[20] Jennings R., Wolldridge J.: Agent Technology Foundations, Applications and Markets. Springer, Berlin Heidelberg New York (1998)

[21] Yim, H.S., Ahn, H.J., Kim,J.W., Park, S.J.: Agent-based adaptive travel planning system in peak seasons. Expert Systems with Applications, 27(2), (2004) 211-222

[22] Garcia-Serrano, A.M., Martinez, P., Hernandez, J.Z.: Using AI techniques to support advanced interaction capabilities in a virtual assistant for e-commerce. Expert Systems with Applications 26(3), (2004) 413-426

[23] Hess, T.J., Rees, L.P., Rakes, T.R.: Using autonomous software agents to create next generation of decision support systems. Decision Sciences, 31 (1), (2000) 1-31

[24] Russel, S., Norvig, P.: Artificial Intelligence – A Modern Approach 2nd Edition. Prentice Hall, New Jersey (2003)

[25] Cavedon, L., Tilhar, G.: A Logical Framework for Multi-Agent Systems and Joint Attitudes. In Proceedings of First Australian Workshop on Distributed Artificial Intelligence, Canberra, Australia (1995)

[26] Santos, M., Neves, J.: Modelling Learning Classifiers as Multiagent Systems. In Proceedings of II Iberoamerican Workshop on DAI and MAS, Toledo, Spain (1998)

[27] Dorigo, M. M. V.: Parallel Genetic Algorithms: Introduction and Overview of Current Research. In Parallel Genetic Algorithms, J. Stender Ed. IOS Press, USA (1993)