A Multi-Agent Recommender System

A. Jorge Morais[1,4,5], Eugénio Oliveira[1,3], Alípio Mário Jorge[2,4]

[1]Faculty of Engineering of the University of Porto
[2]Faculty of Science of the University of Porto
[3]Laboratory of Artificial Intelligence and Computer Science (LIACC)
[4]Laboratory of Artificial Intelligence and Decision Support (LIAAD – INESC Porto L. A.)
[5]Universidade Aberta (Portuguese Open University)

Abstract The large amount of pages in Websites is a problem for users who waste time looking for the information they really want. Knowledge about users' previous visits may provide patterns that allow the customization of the Website. This concept is known as Adaptive Website: a Website that adapts itself for the purpose of improving the user's experience. Some Web Mining algorithms have been proposed for adapting a Website. In this paper, a recommender system using agents with two different algorithms (associative rules and collaborative filtering) is described. Both algorithms are incremental and work with binary data. Results show that this multi-agent approach combining different algorithms is capable of improving user's satisfaction.

1 Introduction

Nowadays, most organizations have a Website, in order to easily deliver information to the general audience. When the size of the Website grows to a significant number of WebPages, the difficulty for users to find what they want also grows. This led organizations to become more concerned with the problem of organizing all the information efficiently, so that it may be easy to find every product or information a user is searching.

Dealing with large datasets is also the motivation for the area of Data Mining and Knowledge Discovery [1], which takes advantage of the large quantity of data from previous transactions that are kept in organizations, finding useful information that is not easily visible. This led to the development of several algorithms that automatically extract and discover important patterns, and represented them in an understandable way. Considering the large number of pages in the Web, it became natural to apply Data Mining and Knowledge Discovery to the Web scope, resulting in the new area of Web Mining [2][3]. Web Mining can be defined as the application of data mining algorithms to extract and discover useful information (documents and services) from the Web. The results may enable the Website owner to improve its structure by reorganizing the Website or providing navigation assistance to users.

The problem of Web adaptation is not new. Recommender systems [3] have had several improvements over the last decade. One of the current solutions that are being proposed for this problem is using autonomous agents. Multi-Agent Systems [4] is a research area that has been in great development over the last decade, and has some particular characteristics that fit in this problem. In fact, it was already proposed to use a multi-agent approach, because of its flexibility and its capability of dynamic adaptation to the Web applications needs [5]. Moreover, Multi-Agent Systems are already used for automatic retrieval and update of information in Websites [6]. An architecture proposal of a recommender system using this approach was already proposed in [7].

In this paper we present a multi-agent approach for Web adaptation, where different incremental algorithms based on binary data produce item-based recommendations and make bids to provide the next set of recommendations to the user. Agents are cooperative in the sense they base their bids on client's satisfaction instead of their own revenue and they share the same data. However, their results are not combined in order to provide recommendations. Our goal is to show that this approach is able to achieve better results than the individual algorithms.

The remaining of the paper starts by presenting previous approaches and applications in the area of recommender systems and multi-agent systems, followed by the description of our approach. The results of the tests with four datasets, and some conclusions and future work complete the paper.

2 Previous approaches and applications

A global vision on adaptive Web sites based on user interaction analysis is given in [8]. In fact, only less ambitious approaches were proposed, such as reorganization of the Website [9], use of recommendations in the pages [10], automatic categorization of user actions [11], or seek of relevant Web sequence paths using Markov models [12].

Recommendation systems include the combination of clustering with nearest neighbour algorithms [13], Markov chains and clustering [14], association rules [15], and collaborative and content-based filtering [16]. Web dynamics has been controlled, for instance, by efficient incremental discovery of sequential Web usage patterns [17], and on-line discovery of association rules [18]. Data-driven categorization of Website usability may be done by typical usage patterns visualization [11] or with objective metrics [19].

Some platforms, like WebWatcher, use previous users' knowledge to recommend links [20]. AVANTI implements an adaptive presentation based on a model constructed from user actions [21]. WUM infers a tree structure from log records enabling experts to find patterns with predefined characteristics [22]. In [23] it was proposed an integrated tool (HDM) to discover access patterns and association rules from log records in order to automatically modify hypertext organization.

In [5] a multi-agent platform was proposed for personalization of Web-based systems, given the flexibility of this approach and its dynamic adaptation to Website needs. Multi-agent approaches for developing complex systems, like Web adaptation, were defended in [24]. Intelligent agents may also be an important contribution for autonomic computing [25]. Such systems main characteristics are being complex systems with self-administration, self-validation, self-adjustment and self-correction. Web adaptation systems should also have these characteristics, because Website environment dynamics requires either a high degree of system automation or high allocation of human resources. Another important usage of multi-agent systems in this issue is the automatic collection and actualization of information in Websites [6].

In [7] it was presented an implemented web adaption platform [26] that was the basis for this work, with the posterior adaptations to our special needs. An implementation of collaborative filtering using an incremental approach was presented in [27].

3 Multi-Agent Approach

The multi-agent system recommender was implemented taking into account that agents should answer rapidly to any request from another agent and prepare in advance for the next request, and tasks that involve a large amount of time (like updating the model) should not interfere with the performance of the system.

Two recommender agents were created. The first one generates single-condition association rules and the second one uses a collaborative filtering algorithm. Since recommendations are meant to be fast in order to keep users interest and taking into account that each new response updates the recommendation models, these incremental approaches must be able to deliver a set of recommendation in a very small amount of time.

Therefore, both algorithms share a matrix $A_{n \times n}$, where n is the number of items (Webpages) and each $a_{ij} \in A$ registers the total number of co-occurrences of items i and j in the same session. The matrix is updated each time a session ends.

The single-condition association rules agent checks all possible rules $i \rightarrow j$, where i and j are items, taking into account two values (k number of sessions):

$$Support_{i \rightarrow j} = \frac{a_{ij}}{k}$$

$$Confidence_{i \rightarrow j} = \frac{a_{ij}}{a_{ii}}$$

Therefore, if a set of *n* recommendation is requested, the *n* best recommendations according to the confidence that satisfy minimum confidence and support requirements are proposed.

The collaborative filtering agent uses the same matrix to compute similarity, returning the top *n* most similar items:

$$sim(i,j) = \frac{a_{ij}}{\sqrt{a_{ii}}\sqrt{a_{jj}}}$$

Agent biddings are based on an accumulated score for each given item obtained from previous ratings – the best *N* are sorted and if the next selected item was in that set it receives a score *N*-*p*+1, where *p* is the ordered position of the item. To this score we add the percentage of the overall score to untie equal biddings:

$$Bid_{agent,item} = Score_{item} + \frac{1}{\#requests * N} * \sum_{i \in Items} Score_i$$

The multi-agent approach was implemented in Java, using the JADE platform [28]. The communication with the browser is implemented using AJAX [29], using XMLHTTPRequest interface, so that the user can consult the Web page without losing interest. The interaction between the user and the recommender system is presented in figure 1, and the architecture of the latter is shown in figure 2. Client agents behaviour is shown in figure 3, while the behaviour of recommender agents is shown in figure 4.

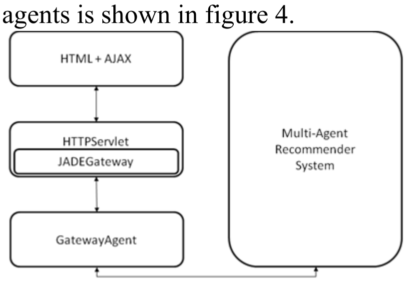

Fig. 1 – Interaction with the recommender system.

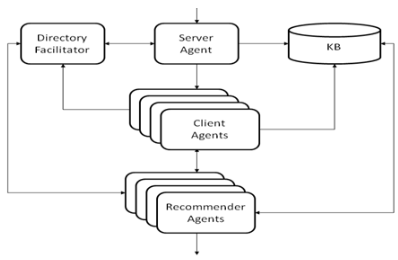

Fig. 2 – Multi-agent recommender system architecture.

```
// created for each client
while not end of session {
  receives request from server
  sends request to recommenders
  waits for responses
  determines winner
  sends results to providers
}
updates knowledge base
destroys itself
```

Fig. 3 – Client agents behaviour

```
receives request from client
sends bid to client
builds recommendation set
if wins bid
  sends recommendations
  //directly to GatewayAgent
updates knowledge base
sends results to provider
determines winner
sends results to provider
```

Fig. 4 – Recommender agents behaviour

4 Experimental Results

Experiments were undertaken offline and focused on four datasets (obtained from real Web data records). Since the application is able to perform several requests in parallel, a modification was made in order to process them sequentially – each request is only processed after the previous one is completed. In real-time, however, requests may be processed concurrently. Each time recommendations are made one of the following situations occur:

a. No item was followed – end of the session.
b. The set of recommendations was empty.
c. An item not in the recommendation set was followed.
d. One of the recommendations was followed – in the following request from the user the requested page was in the recommendation list.

When measuring the recommendation system, the first situation is discarded. The user left the Website for some reason and it is impossible to confirm if any of the recommendations would be useful.

For evaluation of performance there are several metrics. Since the algorithms are incremental, which means we do not have a fix split for train and test sets, the evaluation that fits better to our case is a per-user variant, where predictions are computed and the ranking metrics are calculated for each recommendation, and the average over all recommendations gives the final value [31].

There are two measures that we will use for evaluating recommendation: precision and recall [32]. Precision is the ratio of relevant items selected to number of items selected – it represents the probability that a selected item is relevant. Recall is the ratio of relevant items selected to total number of relevant items available. We will also use the F1 metrics [33], which combines precision and recall.

In our case, precision and recall are given by the following formulas (given N recommendations, and considering b, c and d of the list of possible situations above):

$$Recall = \frac{\#d}{\#b + \#c + \#d}$$

$$Precision = \frac{1}{N} Recall = \frac{1}{N} \frac{\#d}{\#b + \#c + \#d}$$

$$F1 = \frac{2\ Precision\ Recall}{Precision + Recall} = \frac{2 Recall}{N+1} \left(\text{since}\ Precision = \frac{1}{N} Recall \right)$$

This measure is also applied to the recommendation system, which combines agents' algorithms. When the recommendation set is incomplete or inexistent (because it is the first time the item appears, so there are no correlations yet), the system completes it with the most popular items. In table 1 we can see the main characteristics of the datasets and in figures 5a-d the distribution of session's sizes.

In tables 2 to 5, we present the results, for N=1 to 10 number of recommendations, with the evaluation metrics (EM) Recall (R), Precision (P), and F1, for association rules (AR), collaborative filtering (CF), and for the winners (W) of the auctions. The best results between AR and CF are boldface, and when the mutli-agent recommender system is better than both algorithms it is also boldface.

Table 1. Datasets characteristics.

Dataset	#items	# records	#sessions	#records/#session	#records/#items
e-com	335	1409	413	3.411622	4.20597
e-gov	133	4047	1244	3.253215	30.42857
pe100	100	6070	803	7.559153	60.7
pe200	200	2042	200	10.21	10.21

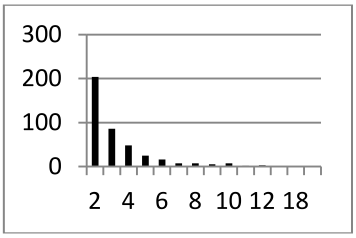

Fig. 5a – e-com sessions size distribution

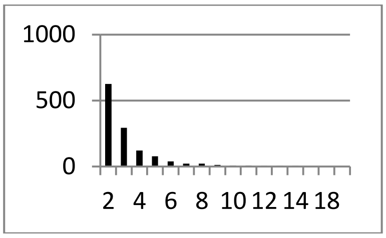

Fig. 5b – e-gov sessions size distribution

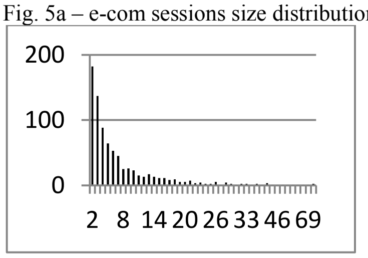

Fig. 5c – pe100 sessions size distribution

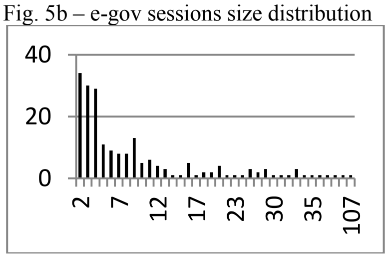

Fig. 5d – pe200 sessions size distribution

Table 2. Results for e-com

#R	EM	1	2	3	4	5	6	7	8	9	10
AR	R	**5.22%**	**9.24%**	**11.65%**	**13.25%**	**15.46%**	**16.67%**	**17.27%**	**18.27%**	**18.78%**	**19.38%**
	P	**5.22%**	**4.62%**	**3.88%**	**3.31%**	**3.09%**	**2.78%**	**2.47%**	**2.28%**	**2.09%**	**1.94%**
	F1	**5.22%**	**6.16%**	**5.82%**	**5.30%**	**5.15%**	**4.76%**	**4.32%**	**4.06%**	**3.76%**	**3.52%**
CF	R	4.82%	7.83%	9.84%	10.84%	12.35%	13.25%	14.66%	15.26%	15.96%	16.47%
	P	4.82%	3.92%	3.28%	2.71%	2.47%	2.21%	2.09%	1.91%	1.77%	1.65%
	F1	4.82%	5.22%	4.92%	4.34%	4.12%	3.79%	3.66%	3.39%	3.19%	2.99%
W	R	**6.02%**	**10.14%**	**13.55%**	**15.36%**	**17.97%**	**19.38%**	**20.78%**	**22.09%**	**23.19%**	**24.20%**
	P	**6.02%**	**5.07%**	**4.52%**	**3.84%**	**3.59%**	**3.23%**	**2.97%**	**2.76%**	**2.58%**	**2.42%**
	F1	**6.02%**	**6.76%**	**6.78%**	**6.14%**	**5.99%**	**5.54%**	**5.20%**	**4.91%**	**4.64%**	**4.40%**

Table 3. Results for e-gov

#R	EM	1	2	3	4	5	6	7	8	9	10
AR	R	**13.20%**	**19.51%**	**24.22%**	**27.68%**	**31.18%**	**33.93%**	**36.46%**	**38.67%**	**40.35%**	**42.06%**
	P	**13.20%**	**9.76%**	**8.07%**	**6.92%**	**6.24%**	**5.65%**	**5.21%**	**4.83%**	**4.48%**	**4.21%**
	F1	**13.20%**	**13.01%**	**12.11%**	**11.07%**	**10.39%**	**9.69%**	**9.12%**	**8.59%**	**8.07%**	**7.65%**
CF	R	11.95%	18.27%	23.40%	27.29%	30.11%	32.36%	34.86%	36.50%	38.21%	39.56%
	P	11.95%	9.13%	7.80%	6.82%	6.02%	5.39%	4.98%	4.56%	4.25%	3.96%
	F1	11.95%	12.18%	11.70%	10.92%	10.04%	9.25%	8.71%	8.11%	7.64%	7.19%
W	R	12.49%	19.27%	24.15%	**27.79%**	31.11%	33.61%	36.35%	38.35%	40.10%	41.96%
	P	12.49%	9.63%	8.05%	**6.95%**	6.22%	5.60%	5.19%	4.79%	4.46%	4.20%
	F1	12.49%	12.84%	12.08%	**11.12%**	10.37%	9.60%	9.09%	8.52%	8.02%	7.63%

Table 4. Results for pe100

#R	EM	1	2	3	4	5	6	7	8	9	10
AR	R	6.15%	10.14%	13.35%	16.46%	19.42%	21.95%	24.23%	26.28%	28.52%	30.43%
	P	6.15%	5.07%	4.45%	4.12%	3.88%	3.66%	3.46%	3.28%	3.17%	3.04%
	F1	6.15%	6.76%	6.67%	6.58%	6.47%	6.27%	6.06%	5.84%	5.70%	5.53%
CF	R	**7.23%**	**11.98%**	**15.63%**	**18.68%**	**20.96%**	**23.49%**	**25.78%**	**27.64%**	**29.49%**	**31.23%**
	P	**7.23%**	**5.99%**	**5.21%**	**4.67%**	**4.19%**	**3.91%**	**3.68%**	**3.46%**	**3.28%**	**3.12%**
	F1	**7.23%**	**7.99%**	**7.81%**	**7.47%**	**6.99%**	**6.71%**	**6.45%**	**6.14%**	**5.90%**	**5.68%**
W	R	**7.50%**	11.96%	**15.74%**	18.63%	**21.28%**	**24.09%**	**26.77%**	**29.07%**	**30.91%**	**32.73%**
	P	**7.50%**	5.98%	**5.25%**	4.66%	**4.26%**	**4.02%**	**3.82%**	**3.63%**	**3.43%**	**3.27%**
	F1	**7.50%**	7.97%	**7.87%**	7.45%	**7.09%**	**6.88%**	**6.69%**	**6.46%**	**6.18%**	**5.95%**

Table 5. Results for pe200

#R	EM	1	2	3	4	5	6	7	8	9	10
AR	R	**3.26%**	**5.92%**	**7.71%**	**9.34%**	10.53%	**12.21%**	13.36%	14.71%	16.02%	**17.54%**
	P	**3.26%**	**2.96%**	**2.57%**	**2.33%**	2.11%	**2.04%**	1.91%	1.84%	1.78%	**1.75%**
	F1	**3.26%**	**3.94%**	**3.85%**	**3.74%**	3.51%	**3.49%**	3.34%	3.27%	3.20%	**3.19%**
CF	R	2.93%	5.81%	7.60%	9.28%	**10.86%**	12.00%	**13.46%**	**14.98%**	**16.18%**	16.83%
	P	2.93%	2.90%	2.53%	2.32%	**2.17%**	2.00%	**1.92%**	**1.87%**	**1.80%**	1.68%
	F1	2.93%	3.87%	3.80%	3.71%	**3.62%**	3.43%	**3.37%**	**3.33%**	**3.24%**	3.06%
W	R	**3.58%**	**6.41%**	**8.63%**	**10.37%**	**12.00%**	**13.84%**	**15.47%**	**16.99%**	**18.24%**	**19.49%**
	P	**3.58%**	**3.20%**	**2.88%**	**2.59%**	**2.40%**	**2.31%**	**2.21%**	**2.12%**	**2.03%**	**1.95%**
	F1	**3.58%**	**4.27%**	**4.32%**	**4.15%**	**4.00%**	**3.96%**	**3.87%**	**3.78%**	**3.65%**	**3.54%**

5 Discussion and future work

Looking at the characteristics of the datasets, we can see that in the e-com and e-gov association rules algorithm (AR) has better results, while in pe100 collaborative filtering (CF) is the best. On the other hand, pe200 has 6 recommendation sizes where AR is better and 4 where CF is better. A possible explanation for AR success in the first two datasets is that in both cases, the percentages of 2-items sessions are around 50% (49.39% for e-com and 50.32% for e-gov), while in the others those values are below 25%.

Analysing the results, we can observe that in e-gov and pe200 datasets the multi-agent recommender system (MARS) outperforms the individual algorithms AR and CF. In pe100, the two only cases where that does not happen the differences to the recall value for the best of the individual algorithms are 0.02% and 0.05%.

The only dissonant case is the e-gov dataset, where the MARS is better only once, for N=4. For N=1, we have the highest difference to the best individual algorithm (AR), 0.71%, while all the other differences vary from 0.07% to 0.32%.

As we can observe, the MARS is able to outperform the individual algorithms in most cases. In the other cases, the results have less than 1% recall difference to the best of the individual algorithms.

As future work, we will perform an in-depth analysis of the e-gov dataset to discover what characteristics are beyond its performance behaviour and we will study new improvements to the MARS in order to improve its results.

Acknowledgements

This work was funded by Fundação para a Ciência e a Tecnologia, Ref. SFRH/BD/27571/2006.

References

1 Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R., editors, 1996. *Advances in Knowledge Discovery and Data Mining*. AAAI/MIT Press, Menlo Park, California.

2 Cooley, R., Mobasher, B., and Srivastava, J., 1997. Web mining: Information and patterns discovery on the world wide Web. In *Proceedings of the ninth IEEE International Conference on Tools with Artificial Intelligence*, pages 558–567, Newport Beach, California.

3 Jannach, D., Zanker, M., Felfernig, A., Friedrich, G., 2011. *Recommender systems: an introduction.* Cambridge University Press.

4 Wooldridge, M., 2002. *An Introduction to MultiAgent Systems*. John Wiley & Sons.

5 Ardissono, L., Goy, A., Petrone, G., and Segnan, M., 2005. A multi-agent infrastructure for developing personalized web-based systems. ACM Trans. Inter. Tech., 5(1):47–69.

6 Albayrak, S., Wollny, S., Varone, N., Lommatzsch, A., and Milosevic, D., 2005. Agent technology for personalized information filtering: the pia-system. In *SAC'05: Proceedings of the 2005 ACM symposium on Applied computing*, pages 54–59, New York, NY, USA. ACM Press.

7 Morais, A. J., 2009. A Multi-Agent Approach for Web Adaptation. In *Y. Demazeau et al. (Eds.): 7th International Conference on PAAMS'09*, AISC 55, pp. 349-355. Springer-Verlag Berlin Heidelberg.

8 Perkowitz, M. and Etzioni, O., 2000. Towards adaptive web sites: Conceptual framework and case study. *Artificial Intelligence*, 118(2000):245–275.

9 Ishikawa, H., Ohta, M., Yokoyama, S., Nakayama, J., Katayama, K., 2002. Web usage mining approaches to page recommendation and restructuring. In *International Journal of Intelligent Systems in Accounting, Finance & Management*, 11(3), Pages: 137-148.

10 El-Ramly, M., Stroulia., E., 2004. Analysis of Web-usage behavior for focused Web sites: a case study. In *Journal of Software Maintenance and Evolution: Research and Practice*, 16(1-2), Pages: 129-150.

11 Berendt, B., 2002. Using Site Semantics to Analyze, Visualize, and Support Navigation. In *Data Mining and Knowledge Discovery*, Volume 6, Issue 1, Pages 37 – 59.

12 Borges, J. L., 2000. A Data Mining Model to Capture User Web Navigation Patterns, PhD thesis, University College London, University of London.

13 Mobasher, B., Dai, H., Luo, T., Nakagawa, M., 2002. Discovery and Evaluation of Aggregate Usage Profiles for Web Personalization. In *Data Mining and Knowledge Discovery*, Kluwer Publishing, Vol. 6, No. 1, pp. 61-82.

14 Cadez, I., Heckerman, D., Meek, C., Smyth, P., White, S., 2003. Model-Based Clustering and Visualization of Navigation Patterns on a Web Site. In *Data Mining and Knowledge Discovery*, Volume 7, Issue 4, Pages 399 – 424.

15 Jorge, A., Alves, M. A., Grobelnik, M., Mladenic, D., and Petrak, J., 2003. Web Site Access Analysis for A National Statistical Agency. In *Data Mining And Decision Support: Integration And Collaboration*, Mladenic, D., Lavrac, N., Bohanec, M., Moyle, S., Kluwer Academic Publishers.

16 Basilico, J., Hofmann, T., 2004. Unifying collaborative and content-based filtering. In *Proceedings of ICML'04, Twenty-first International Conference on Machine Learning*. ACM Press, New York.

17 Masseglia, F., Teisseire,, M., Poncelet, P., 2003. HDM: A client/server/engine architecture for real time web usage mining. In Knowledge and Information Systems (KAIS), Volume 5, Number 4, pp 439 - 465.

18 Lin, W., Alvarez, S. A., Ruiz, C., 2002. Efficient Adaptive-Support Association Rule Mining for Recommender Systems. In *Data Mining and Knowledge Discovery*, 6:83-105.

19 Spiliopoulou, M., and Pohle, C., 2001. Data mining for measuring and improving the success of web sites. In Ronny Kohavi and Forest Provost, editors, *Journal of Data Mining and Knowledge Discovery, Special Issue on E-commerce*, volume 5(1-2), pages 85-114. Kluwer Academic Publishers.

20 Armstrong, R., Freitag, D., Joachims, T., and Mitchell, T., 1995. WebWatcher: A learning apprentice for the world wide web. In *Proceedings of the AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments*, pages 6–12, California.
21 Fink, J., Kobsa, A., and Nill, A., 1996. User-oriented adaptivity and adaptability in the AVANTI project. In *Designing for the Web: Empirical Studies*, Microsoft Usability Group, Redmond, Washington.
22 Spiliopoulou, M. and Faulstich, L. C., 1998. WUM: a tool for web utilization analysis. In *Proceedings of the International Workshop on the Web and Databases*, pages 184–203, Valencia, Spain.
23 Masseglia, F., Teisseire, M., and Poncelet, P., 2001. Real Time Web Usage Mining: a Heuristic Based Distributed Miner., In *Second International Conference on Web Information Systems Engineering (WISE'01)*, Volume 1, p. 0288.
24 Jennings, N. R., 2001. An agent-based approach for building complex software systems. In *Communications of the ACM*, vol. 44, no. 4, pp. 35-41.
25 Kephart, J. O., 2005. Research challenges of autonomic computing. In *ICSE '05: Proceedings of the 27th International Conference on Software Engineering*, pages 15–22, New York, NY, USA. ACM Press.
26 Domingues, M. A., Jorge, A. M., Soares, C., Leal, J. P., and Machado, P., 2007. A data warehouse for web intelligence. In *Proceedings of the 13th Portuguese Conference on Artificial Intelligence (EPIA 2007)*, 487–499.
27 Miranda, C., Jorge, A. M., 2009. Item-Based and User-Based Incremental Collaborative Filtering for Web Recommendations. In
28 JADE (Java Agent DEvelopment Framework) Website: http://jade.tilab.com. Access date: 14/10/2011.
29 Asynchronous Javascript And XML (AJAX), Mozilla Developer Center: https://developer.mozilla.org/en/ajax. Access date: 14/10/2011.
30 Apache Derby Website: http://db.apache.org/derby/. Access date: 14/10/2011.
31 Herlocker, J. L., Konstan, J. A., Terveen, L. G., Riedl, J. T., 2004. Evaluating collaborative filtering recommender systems. In *ACM Transactions on Information Systems*. 22 (1): 5–53.
32 Cleverdon, C., Kean, M., 1968. Factors Determining the Performance of Indexing Systems. *Aslib Cranfield Research Project*, Cranfield, England.
33 Sarwar, B. M., Karypis, G., Konstan, J. A., Riedl, J., 2001. Item-based collaborative filtering recommendation algorithms. In *WWW*, pp 285–295.