I.J. Intelligent Systems and Applications, 2012, 9, 61-67
Published Online August 2012 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2012.09.08

The Impact of Feature Selection on Web Spam Detection

Jaber Karimpour
Dept. of Computer Science, University of Tabriz, Tabriz, Iran
karimpour@tabrizu.ac.ir

Ali A. Noroozi
Dept. of Computer Science, University of Tabriz, Tabriz, Iran
aliasghar.noroozi@gmail.com

Adeleh Abadi
Dept. of Computer Science, University of Tabriz, Tabriz, Iran
adeleh.abadi@gmail.com

Abstract— Search engine is one of the most important tools for managing the massive amount of distributed web content. Web spamming tries to deceive search engines to rank some pages higher than they deserve. Many methods have been proposed to combat web spamming and to detect spam pages. One basic one is using classification, i.e., learning a classification model for classifying web pages to spam or non-spam. This work tries to select the best feature set for classification of web spam using imperialist competitive algorithm and genetic algorithm. Imperialist competitive algorithm is a novel optimization algorithm that is inspired by socio-political process of imperialism in the real world. Experiments are carried out on WEBSPAM-UK2007 data set, which show feature selection improves classification accuracy, and imperialist competitive algorithm outperforms GA.

Index Terms— Web Spam Detection, Feature Selection, Imperialistic Competitive Algorithm, Genetic Algorithm

I. Introduction

With the explosive growth of information on the web, it has become the most successful and giant distributed computing application today. Billions of web pages are shared by millions of organizations, universities, researchers, etc. Web search provides great functionality for distributing, sharing, organizing, and retrieving the growing amount of information [1]. Search engines have become more and more important and are used by millions of people to find necessary information. It has become very important for a web page, to be ranked high in the important search engines' results. As a result many techniques are proposed to influence ranking and improve the rank of a page. Some of these techniques are legal and are called Search Engine Optimization (SEO) techniques, but some are not legal or ethical and try to deceive ranking algorithms. These spam pages try to rank pages higher than they deserve [2].

Web spam refers to web content that get high rank in search engine results despite low information value. Spamming not only misleads users, but also imposes time and space cost to search engine crawlers and indexers. That is why crawlers try to detect web spam pages to avoid processing and indexing them.

Content-based spamming methods basically tailor the contents of the text fields in HTML pages to make spam pages more relevant to some queries. This kind of spamming is also called term spamming. There are two main content spamming techniques, which simply create synthetic contents containing spam terms: repeating some important terms and dumping many unrelated terms [3,4].

Link spamming misuses link structure of the web to spam pages. There are two main kinds of link spamming. Out-link spamming tries to boost the hub score of a page by adding out-links in it pointing to some authoritative pages. One of the common techniques of this kind of spamming is directory cloning, i.e., replicating a large portion of a directory like Yahoo! in the spam page. In-link spamming refers to persuading other pages, especially authoritative ones, to point to the spam page. In order to do this, a spammer might adopt these strategies: creating a honey pot, infiltrating a web directory, posting links on user-generated content, participating in link exchange, buying expired domains, and creating own spam farm [2].

Hiding techniques are also used by spammers who want to conceal or to hide the spamming sentences, terms, and links so that web users do not see those [3]. Content hiding is used to make spam items invisible. One simple method is to make the spam terms the same color as the page background color. In cloaking, spam web servers return an HTML document to the user and a different document to a web crawler. In redirecting, a spammer can hide the spammed page by automatically redirecting the browser to another URL as soon as the page is loaded. In two latter techniques, the spammer can present the user with the intended content and the search engine with spam content [5].

Various methods have been proposed to combat web spamming and to detect spam pages. One important and basic type of methods is considering web spam detection as a binary classification problem [4]. In this kind of methods, some web pages are collected as training data and labeled as spam or non-spam by an expert. Then, a classifier model is learned from the training data. One can use any supervised learning algorithm to build this model. Further, the model is used to classify any web page to spam or non-spam. The key issue is to design features used in learning. Ntoulas et al. [4] propose some content-based features to detect content spam. Link-based features are proposed for link spam detection [6,7]. Liu et al. [8] propose some user behavior features extracted from access logs of web server of a page. These features depict user behavior patterns when reaching a page (spam or non-spam). These patterns are used to separate spam pages from non-spam ones, regardless of spamming techniques used. Erdelyi et al. [9] investigate the tradeoff between feature generation and spam classification accuracy. They conclude that more features achieve better performance; however, the appropriate choice of the machine learning techniques for classification is probably more important than devising new complex features.

Feature selection is the process of finding an *optimal* subset of features that contribute significantly to the classification. Selecting a small subset of features can decrease the cost and the running time of a classification system. It may also increase the classification accuracy because irrelevant or redundant features are removed [10]. Among the many methods proposed for feature selection, evolutionary optimization algorithms such as genetic algorithm (GA) have gained a lot of attention. Genetic algorithm has been used as an efficient feature selection method in many applications [11,16].

In this paper, we incorporate genetic algorithm, and imperialist competitive algorithm [12] to find an optimal subset of features of the WEBSPAM-UK2007 data set [13,14]. The selected features are used for classification of the WEBSPAM-UK2007 data.

The rest of the paper is organized as follows. Section 2 gives a brief introduction of the imperialist competitive algorithm (ICA). Section 3 describes the feature selection process by ICA and GA. Experimental results are discussed in section 4, and finally, section 5 concludes the paper.

II. Imperialistic Competitive Algorithm

The imperialist competitive algorithm is inspired by imperialism in the real world [12]. Imperialism is the policy of extending the power of a country beyond its boundaries and weakening other countries to take control of them.

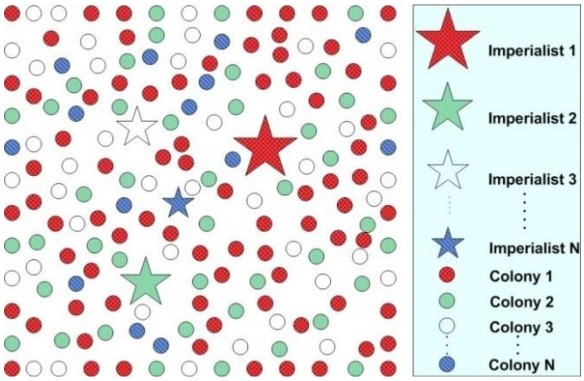

Fig. 1 Initialization of the empires: The more colonies an imperialist possesses, the bigger is its ★ mark [12]

This algorithm starts with an initial society of random generated countries. Some of the best countries are selected to be *imperialists* and others are selected to be *colonies* of these imperialists. The power of an empire which is the counterpart of fitness value in genetic algorithms, is the power of the imperialist country plus a percentage of mean power of its colonies. Figure 1 depicts the initialization of the empires.

After assigning all countries to imperialists, and forming empires, colonies start moving towards the relevant imperialist (Assimilation). Then, some countries randomly change position in the search space (Revolution). After assimilation and revolution, a colony may get a better position in the search space and take control the empire (substitution for the imperialist).

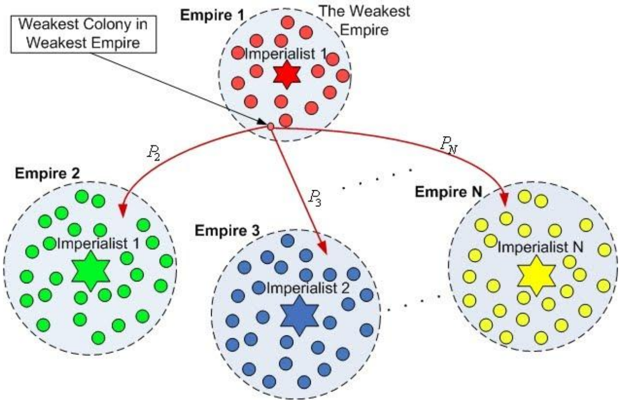

Fig. 2 Imperialistic competition: The weakest colony of the weakest empire is possessed by other empires [12]

Then, imperialistic competition begins. All empires try to take control of the weakest colony of the weakest empire. This competition reduces the power of weaker empires and increases the power of the powerful ones. Any empire that cannot compete with other empires and increase its power or at least prevent decreasing it, will gradually collapse. As a result, after some iterations, the algorithm converges and only one imperialist remains and all other countries are colonies of it. Figure 2 depicts the imperialistic competition. The more powerful an empire is, the more likely it will take control of the weakest colony of the weakest empire.

The pseudo code of ICA is as follows:

1. Initialize the empires
2. Assimilation: Move the colonies toward their relevant imperialist
3. Revolution: Randomly change the characteristics of some colonies
4. Exchange the position of a colony and Imperialist. If a colony has more power than that of imperialist, exchange the positions of that colony and the imperialist
5. Compute the total power of all empires
6. Imperialistic competition: Give the weakest colony from the weakest empire to the empire that has the most likelihood to possess it
7. Eliminate the powerless empires
8. If there is just one empire, stop, else, go to 2

III. Feature Selection

WEBSPAM-UK2007 data set contains 96 content based features. We use the imperialist competitive and genetic algorithms to optimize the features that contribute significantly to the classification.

A. *Feature Selection Using ICA*

In this section, the steps of feature selection using ICA are described.

1) Initialize the empires

In the genetic algorithm, each solution to an optimization problem is an array, called *chromosome*. In ICA, this array is called *country*. In feature selection, each country is an array of binary numbers. When country[i] is 1, the i^{th} feature is selected for classification, and when it is 0, the i^{th} feature is removed [15]. Figure 3 depicts the feature representation as a country.

F_1	F_2	F_3	...	F_{n-1}	F_n

Country

1	**0**	**1**	**…**	**1**	**0**

Feature subset = $\{F_1, F_3, ..., F_{n-1}\}$

Fig. 3 Feature representation as a country in ICA [15]

The power of each country is calculated by *F-score*. F-score is a commonly used measure in machine learning and information retrieval [3,10]. The confusion matrix of a given a classifier is considered as table 1.

Table 1. Confusion Matrix

	Classified spam	Classified non-spam
Actual spam	A	B
Actual non-spam	C	D

F-score is determined as follows

$$\text{F-score} = 1 / (1 / \text{Recall} + 1 / \text{Precision}) \qquad (1)$$

Where Recall, and Precision are defined as follows

$$\text{Recall} = C / (C + D) \qquad (2)$$

$$\text{Precision} = B / (B + D) \qquad (3)$$

The algorithm starts by randomly initializing a population of size N_{pop}. N_{imp} of the most powerful countries are selected as imperialists and form the empires. The remaining countries (N_{col}) are assigned to empires based on the power of each empire. The normalized power of each imperialist is defined by

$$NP_{\text{n}} = \frac{P_n}{\sum_{i=1}^{N_{\text{imp}}} P_i} \qquad (4)$$

Where P_n is the power of $\text{country}_{\text{n}}$.

The initial number of colonies of empire_{n} will be

$$NC_{\text{n}} = round\{NP_{\text{n}} * N_{\text{col}}\} \qquad (5)$$

To assign colonies to empires, NC_{n} of the colonies is chosen randomly and assigned to $\text{imperialist}_{\text{n}}$. These colonies along with the $\text{imperialist}_{\text{n}}$ will form empire_{n}.

2) Assimilation

In this phase, colonies move towards the relevant imperialist. Since feature selection is a discrete problem, we use following operator for assimilation [15]

For each colony

- Create a binary string and assign a random generated binary to each cell
- Copy the cells of the relevant imperialist, corresponding to the location of "1"s in the binary string, to the same positions in the colony

3) Revolution

The purpose of revolution is preserving and introducing diversity. It allows the algorithm to avoid local minimum. Revolution occurs according to a user defined revolution probability. For each colony, some cells are selected randomly and their containing binary is inverted ("1" is inverted to "0", and "0" is inverted to "1").

4) *Exchange the positions of a colony and imperialist*

After assimilation and revolution, a colony may gain more power than that of imperialist. As a result, the best colony of an empire and its imperialist exchange positions. Then, the algorithm will continue by the imperialist in a new position.

5) *Compute the total power of empires*

The total power of an empire is mainly affected by the power of its imperialist. Another factor in computing the total power of an empire is the power of colonies of that empire. Of course, the main power is by the power of the imperialist, and the power of colonies has less impact. As a result, we define the total power of $empire_n$ is defined

$$TP_n = power(imperialist_n) + \xi mean\{power(colonies\ of\ empire_n)\} \quad (6)$$

Where ξ is a positive factor which is considered to be less than 1. Decreasing the value of ξ increases the role of the imperialist in determining the total power of an empire and increasing it will increase the role of the colonies.

6) *Imperialistic Competition*

In this important phase of the algorithm, the empires compete to take control of the weakest colony of the weakest empire. Each empire has a likelihood of possessing the mentioned colony. The possession probability of $empire_n$ is obtained by

$$P_{emp_n} = \frac{TP_n}{\sum_{i=1}^{N_{imp}} TP_i} \quad (7)$$

As you can notice, the most powerful empire does not take possession of the weakest colony of the weakest empire, but it will be more likely to possess the mentioned colony.

7) *Eliminate the powerless empires*

Imperialistic competition causes some empires to lose power and gradually collapse. When an empire loses all its colonies, we assume it is collapsed and eliminate it. The imperialist of this powerless empire is possessed by other empires as a colony.

8) *Convergence*

As a result of imperialistic competition and elimination of powerless empires, the algorithm will converge to the most powerful empire and all the countries will be under the control of this empire. The imperialist of this empire will determine the optimal subset of features selected for classification, because this imperialist is the most powerful of all countries.

B. Feature selection using GA

In the genetic algorithm, each solution to the feature selection problem is a string of binary numbers, called *chromosome*. When chromosome[i] is 1, the i^{th} feature is selected for classification, and when it is 0, the i^{th} feature is not selected [11,16].

The fitness function is considered the accuracy of the classification model. In this research, we calculate the fitness value of each chromosome by F-score. F-score was described in the previous section.

The algorithm starts by randomly initializing a population of size N_{pop}. Then, crossover and mutation are done.

Crossover allows the generation of new chromosomes by combining current best chromosomes. To do crossover, single point crossover technique is used, i.e., one crossover point is selected, binary string from beginning of chromosome to the crossover point is copied from one parent, the rest is copied from the second parent. Figure 4 shows how children are generated from each pair of chromosomes by crossover.

Mutation is similar to revolution in ICA. It maintains genetic diversity and allows the algorithm to avoid local minimum. To do mutation, in each chromosome, a random cell is selected and its containing bit in inverted ("1" is inverted to "0", and "0" is inverted to "1").

Mutation and crossover occur according to a previously defined mutation and crossover probability.

Genetic algorithm iterates for some user defined number of generations.

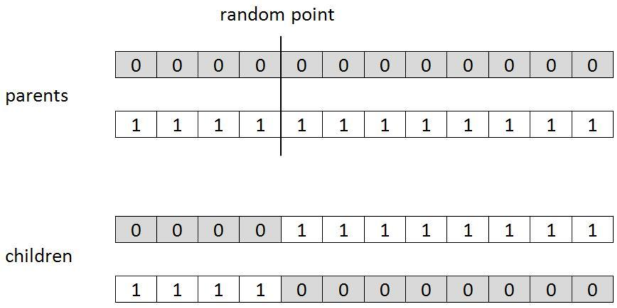

Fig. 4 how children are generated from parents by crossover [17]

IV. Experimental Results

In order to investigate the impact of feature selection on web spam classification, WEBSPAM-UK2007 data are used. It is a publicly available web spam data collection and is based on a crawl of the .uk domain done in May 2007 [13, 14]. It includes 105 million pages and over 3 billion links in 114529 hosts.

The training set contains 3849 hosts. This data set contains content and link based features. In our experiments, we used only content based features because they were enough to meet our purposes. The selected data set contains 3849 data, with 208 spam and 3641 non-spam pages. We partitioned this data set to two disjoint sets: training data set with 2449 data, and test data set with 1000 data. After performing feature selection using the training set, the test set was used to evaluate the selected subset of features.

The evaluation of the overall process was based on weighted f-score which is a suitable measure for the spam classification problem. It was also used as the power function in ICA and fitness function in GA.

Bayesian Network, Decision Tree (C4.5 algorithm), and Support Vector Machine (SVM) were chosen as learning algorithms to perform the classification and calculate the weighted F-score. These algorithms are powerful learning algorithms used in many web spam detection researches [4, 5, 18].

Following parameters were used for ICA

Number of countries = 100
Number of imperialists = 10
$\xi = 0.1$
Revolution rate = 0.01

Selected parameters for GA are as follows

Initial population = 100
Number of generations (iterations) = 100
Crossover rate = 0.6
Mutation rate = 0.01

Figure 5 depicts maximum and mean power of all imperialists versus iteration, using Decision Tress, SVM, and Bayesian Network classifiers, in ICA. As shown in this figure, by SVM and Decision Tree classifiers, the global maximum of the function (maximum power) is found in less than 5 iterations, while by Bayesian Network, it is found in 12th iteration.

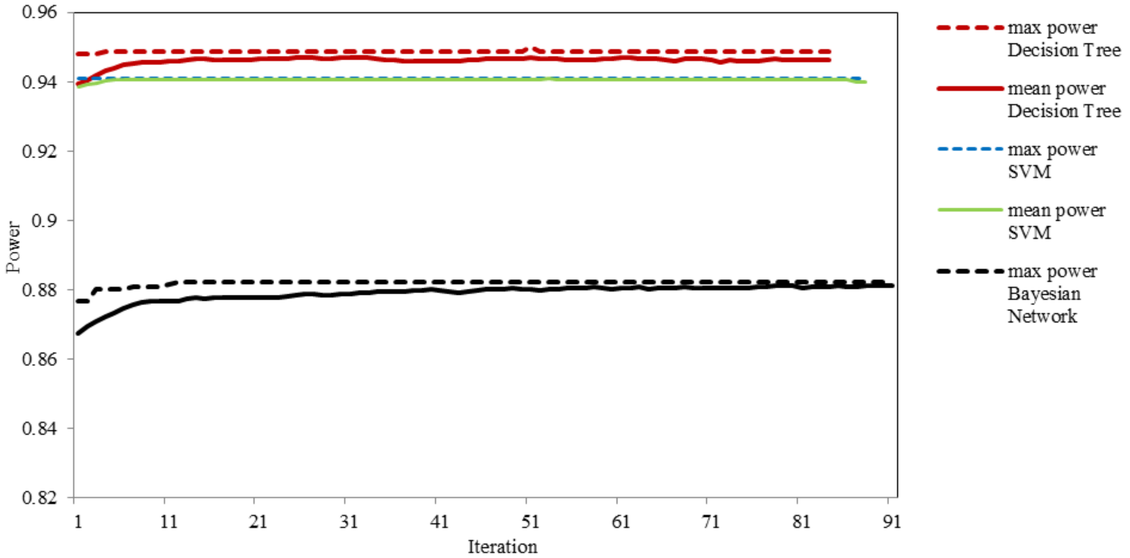

Fig. 5 Mean and maximum power of all imperialists versus iteration, using different classifiers, in ICA

Figures 6, and 7 compare ICA power function and GA fitness function versus iteration. Figure 6 shows the power (fitness) of best answer versus iteration (generation), using Bayesian Network classifier, in ICA and GA. As you can see, ICA converges faster than GA, and has more power than GA in all iterations. Another important point is that the initial value of f-score which is the result of random initialization of population in both algorithms, gets a higher increase by ICA over iterations. This point shows that imperialistic competition outperforms genetic evolution in the problem of spam classification.

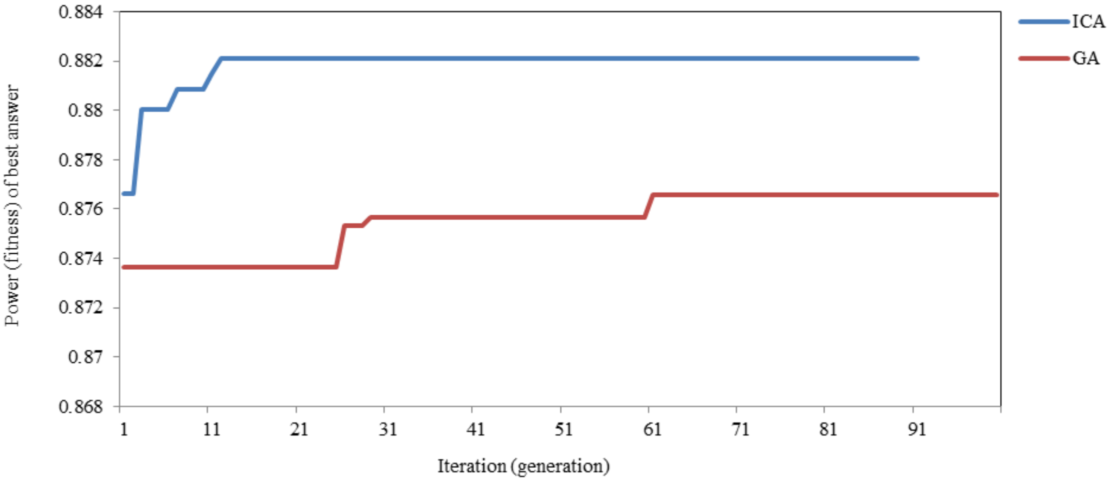

Fig. 6 Power (fitness) of best answer versus iteration, using Bayesian Network classifier

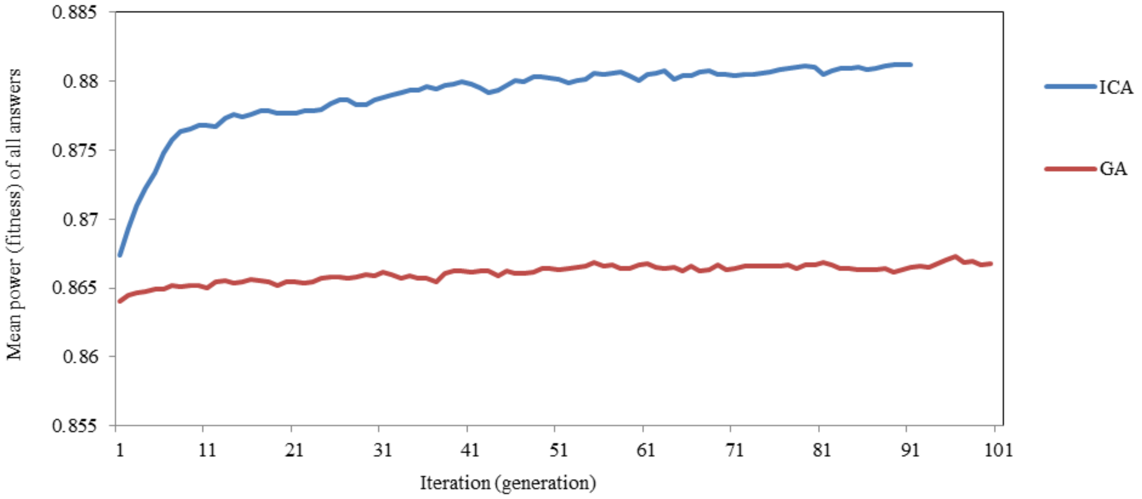

Fig. 7 Mean power (fitness) of all answers versus iteration, using Bayesian Network classifier

Figure 7 depicts mean power (fitness) of all answers versus iteration, using Bayesian Network classifier, in ICA and GA. As you can see, ICA gets a higher increase in mean power of all answers.

The optimal subset of features selected by ICA and GA are used to train a classification model. This model is evaluated by the test data set. Evaluation results obtained for Bayesian Network, Decision Tree, and SVM classifiers are shown in table 2. These results indicate that feature selection by both ICA and GA techniques improves web spam classification. Furthermore, ICA based feature selection outperforms GA based feature selection in the problem of web spam detection.

Table 2 The impact of ICA and GA based feature selection on web spam classification, using different classifiers

	Bayesian Network		Decision Tress		SVM	
	Number of features	F-score	Number of features	F-score	Number of features	F-score
All features	96	**0.854**	96	0.935	96	0.937
GA	48	**0.876**	49	0.950	56	0.939
ICA	41	**0.882**	56	0.950	61	0.940

V. Conclusion

In this paper, we studied the impact of feature selection on the problem of web spam classification. Feature selection was performed by Imperialist Competitive Algorithm and Genetic Algorithm. Experimental results showed that selecting an optimal subset of features increases classification accuracy, but ICA could find better optimal answers than GA. In fact, we observed that reducing the number of features decreases the classification cost and increases the classification accuracy.

Other optimization methods, such as PSO and ant colony can be used for feature selection and compared with ICA and GA in future works.

References

[1] Caverlee J, Liu L, Webb S. A Parameterized Approach to Spam-Resilient Link Analysis of the Web. IEEE Transactions on Parallel and Distributed Systems (TPDS), 2009, 20:1422-1438.

[2] Gyongyi Z,Garcia-Molina H. Web spam taxonomy. In: First internationalworkshop on adversarial information retrieval on the web (AIRWeb’05), Japan, 2005.

[3] Liu B. Web Data Mining, Exploring Hyperlinks, Contents, and Usage Data. Springer, 2006.

[4] Ntoulas A, Najork M, Manasse M, et al. Detecting Spam Web Pages through Content Analysis. In Proc. of the 15th Intl. World Wide Web Conference (WWW’06), 2006. 83–92

[5] Wang W, Zeng G, Tang D. Using evidence based content trust model for spam detection. Expert Systems with Applications, 2010. 37(8):5599-5606

[6] Becchetti L, Castillo C, Donato D, et al. Link-based characterization and detection of Web Spam. In Proc. Of 2nd Int. Workshop on Adversarial

Information Retrieval on the Web (AIRWeb'06), Seattle, WA, 2006. 1–8

[7] Castillo C, Donato D, Gionis A, et al. Know your neighbors: Web spam detection using the web topology. In Proc. Of 30th Annu. Int. ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR'07), New York, 2007. 423–430

[8] Liu Y, Cen R, Zhang M, et al. Identifying web spam with user behavior analysis. In Proc. Of 4th Int. Workshop on Adversarial Information Retrieval on the Web (AIRWeb'08), China, 2008. 9-16

[9] Erdelyi M, Garzo A, Benczur A A. Web spam classification: a few features worth more. In Proceedings of the 2011 Joint WICOW/AIRWeb Workshop on Web Quality2011, India, 2011. 27-34.

[10] Han J, Kaber M, Pei J. Data Mining, Concepts and Techniques. 3rd edn, Morgan Kaufman, 2011.

[11] Vafaie H, De Jong K. Genetic algorithms as a tool for feature selection in machine learning. In Proceedings of Fourth International Conference on Tools with Artificial Intelligence (TAI '92), 1992. 200-203.

[12] Atashpaz-Gargari E, Lucas C. Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition. IEEE Congress on Evolutionary Computation (CEC 2007), 2007. 4661-4667

[13] Castillo C, Donato D, Becchetti L, et al. A reference collection for webspam. SIGIR Forum, 2006, 40(2): 11–24

[14] Yahoo Research. Web Spam Collections. [cited 2011 May], Available from: http://barcelona.research.yahoo.net/webspam/datasets/, 2007

[15] Mousavi Rad S J, Mollazade K, Akhlagian Tab F. Application of Imperialist Competitive Algorithm for Feature Selection: A Case Study on Bulk Rice Classification. International Journal of Computer Applications, 2012. 40(16):41-48

[16] Yang J, Honavar V. Feature subset selection using a genetic algorithm. Intelligent Systems and their Applications, IEEE, 1998. 13(2):44-49.

[17] Eiben A E, Smith J E. Introduction to Evolutionary Computing, Springer, 2010.

[18] Araujo L, Martinez-Romo J. Web Spam Detection: New Classification Features Based on Qualified Link Analysis and Language Models. IEEE Transactions on Information Forensics and Security, 2010. **5**(3):581-590.

KARIMPOUR Jaber (1974－), male, Tabriz, Iran, Assistant Professor, his research directions include verification and formal methods.

NOROOZI Ali A. (1986－), male, Tabriz, Iran, Master of Science, his research directions include adversarial information retrieval and distributed systems.

ABADI Adeleh (1976 -) female, Tabriz, Iran, Master of Science, his research directions include verification and formal methods.