A Multiobjective Evolutionary Algorithm for Spam E-mail Filtering

A.G. López-Herrera[1], E. Herrera-Viedma[2], F. Herrera[2]
1.Dept. of Computer Sciences, University of Jaén, E-23071, Jaén (Spain), aglopez@ujaen.es
2.Dept. of Computer Sciences and A.I., University of Granada, E-18071, Granada (Spain), {viedma, herrera}@decsai.ugr.es

Abstract

Unsolicited Commercial Email, also known as spam, has been a major problem on the Internet. In this paper a well known Multiobjective Evolutionary Algorithm, NSGA-II, is first time used for spam e-mail filtering. NSGA-II is adapted to use Genetic Programming components to achieve a set of filtering rules with different profiles.

1. Introduction

The Unsolicited Commercial Email, also known as spam, is commonplace everywhere in email communication. Spam is a major and growing problem. It is estimated that in the month of May 2006, for example, 86% of all e-mails sent were spam [1]. Spam is a costly problem and many experts agree it is only getting worse [2, 3, 4, 5, 6]. Because of the economics of spam and the difficulties inherent in stopping it, it is unlikely to go away soon. Consequently, a large amount of effort has been expended on devising effective filters to identify spam e-mails.

In recent years, personalized anti-spam filters of email client applications based on content filters have now become the standard for spam filters [7, 8]. Spam filters may be implemented using rule-based filters [9], nearest neighbor classifiers [10], decision trees [11] and Bayesian classifiers [12], etc.

E-mails are filtered inconsistently across different users irrespective of the user's interest. Since users have different interests or business needs, a good anti-spam filtering system should take into account of the different users' needs and interests into consideration and influence the overall decisions and behavior [13].

Spam filter design problem is naturally multiobjective. Given a spam filter, stopping as many spam e-mails as possible is in direct conflict with preventing the filtering of legitimate e-mails. In fact, the conflicting nature of decreasing the number of false positives (fraction labelled as spam from the non-spam class) and the increasing the number of true positives (fraction labelled correctly as spam) is a very general problem in pattern recognition. In other words, it is difficult to design a filter which simultaneously optimises the precision and recall.

The Spam Filtering (SF) problem can be considered as a specific Information Retrieval (IR) one. In IR there are usually two kinds of documents, relevant and non-relevant. From this IR point of view, spam e-mails can be treated as non-relevant documents, and non-spam e-mails as relevant ones.

Defining a query for information retrieval is analogous to build a rule for spam filtering. In IR, a query is used to get relevant documents, whereas, in SF a rule is set for blocking undesirable e-mails.

Evolutionary Algorithms (EAs) have been used for IR purposes [14], being the query definition problem one of the most studied one. Concretely, the use of Multiobjective EAs (MOEAs) in the query definition problem has been proved as advantageous, due to MOEAs can learn a set of queries with good precision-recall trade-off in a unique run [26]. This feature of MOEAs, applied to SF, would be used to get a set of filtering rules, each one defining a different profile, from very strong rule (high recall) to weak rules (high precision).

In this paper, we treat the SF problem using concepts from the query definition problem. A new MOEA for SF is presented in this paper. It is called SPAM-NSGA-II-GP, and it is built on the basis of a well known MOEA, the *Non-dominated Sorting Genetic Algorithm* (NSGA-II) [15]. The performance of SPAM-NSGA-II-GP in the automatic building rule problem is analyzed using a public spam dataset.

To do so, this paper is structured as follows. In Section 2, we first briefly describe the query definition problem. In Section 3, SPAM-NSGA-II-GP is introduced. Section 4 shows the experimental results. Finally, Section 5 summarizes several concluding remarks.

978-1-4244-2197-8/08/$25.00 ©2008 IEEE

2. The Query Definition Problem

This is the most extended group of applications of EAs to IR. Every proposal in this group use EAs either like a relevance feedback technique or like an Inductive Query by Example (IQBE) algorithm. The basis of relevance feedback lie in the fact that either users normally formulate queries composed of terms that do not match the terms used to index the relevant documents to their needs, or they do not provide the appropriate weights for the query terms.

The operation mode involving modifying the previous query adding and removing terms or changing the weights of the existing query term staking into account the relevance judgements of the documents retrieved by it, constitutes a good way to solve the latter two problems and to improve the precision, and especially the recall, of the previous query [16].

IQBE was proposed in [17] as "a process in which searchers provide sample documents (examples) and the algorithms induce (or learn) the key concepts in order to find other relevant documents". This way, IQBE is a process for assisting the users in the query formulation process performed by machine learning methods. It works taking a set of relevant documents (and optionally non-relevant documents) provided by the user and applying an automatic learning process to generate a query that describes the user information needs (represented by the previous set of documents). The query that is obtained can be executed to obtain new relevant documents.

Several IQBE EAs for different IR models have been proposed and revised in [9]. The most well known, in the context of Boolean IR [16], IQBE approach is that of Smith & Smith [18], which is based on Genetic Programming (GP) [19], with queries being represented by expression syntax trees and where the algorithms are articulated on the basis of the classic operators: cross, mutation and selection. It is called Boolean IQBE-GP, and it is able to derive Boolean queries. As it is usual in the field [14], this approach is guided by a weighted fitness function combining the classical retrieval accuracy criteria, precision and recall [16]. In [20], Kraft et al. propose an IQBE technique to learn the whole composition of extended Boolean queries for Fuzzy IR systems. The algorithm is based on GP and the fuzzy queries are encoded in expression trees, whose terminal nodes are query terms with their respective weights and whose inner nodes are the Boolean operators.

In this paper, the IQBE paradigm will be used, with spam e-mails playing the role of non-relevant documents.

3. A new MOEA for Spam Filtering

Following sections introduce: the classical performance measures used in SF, the objectives and essential notions of pareto set used in this paper, and the new MOEA for SF proposed.

3.1. Performance Measures

There are several ways to measure the quality of a SF system, such as the system efficiency and effectiveness, and several subjective aspects related to user satisfaction. Traditionally, the filtering effectiveness is based on the e-mail judgement with respect to the user's needs or interests. There are different criteria to measure this aspect, but Precision (P) and Recall (R) [16] are the most used. Precision is the ratio between the spam e-mails filtered by the SF system in response to a filtering rule and the total number of e-mails filtered, whilst recall is the ratio between the number of spam e-mails filtered and the total number of spam e-mails in the user's inbox. The mathematical expression of each of them is:

$$P = \frac{E_{sf}}{E_{tf}};\ R = \frac{E_{sf}}{E_{ts}}$$

where E_{sf} is the number of spam e-mails filtered, E_{tf} is the total number of e-mails filtered and E_{ts} is the total number of spam e-mails in the user's inbox. P and R are defined in [0,1], being 1 the optimal value. We notice that the only way to know all the spam e-mails existing in an inbox (value used in the R measure) is to evaluate all e-mails. Due to this fact and tacking into account that spam labelling is subjective, there are some public spam datasets: LingSpam [21], Spambase [22], SpamAssassin [23] and PU1 [24, 25], each one with a set of e-mails being labelled as spam or non-spam, so that they can be used to verify the new proposals in the field. In this paper, we use the PU1 dataset.

3.2. Objectives and Pareto Set

In multiobjective optimization problems, the definition of the quality concept is substantially more complex than in singleobjective ones, since the optimization processes imply several different objectives.

The key concepts to evaluate MOEAs are the dominance relation and the Pareto sets.

The MOEA presented in this paper assume the two classical criteria to evaluate SF systems, i.e., Precision and Recall whose expressions are introduced in Subsection 3.1. The proposed MOEA assumes that all objectives have to be maximized. The solutions are represented by objective vectors, which are compared according to the dominance relation defined below and displayed in Figure 1.

Definition 1. *(Dominance relation) Let* $f, g \in \mathbb{R}^m$. *Then* f *is said to dominate* g, *denoted as* $f \succ g$, *iff*

1. $\forall i \in \{1, \cdots, m\} : f_i \geq g_i$,
2. $\exists j \in \{1, \cdots, m\} : f_j > g_j$.

Based on the concept of dominance, the Pareto set can be defined as follows.

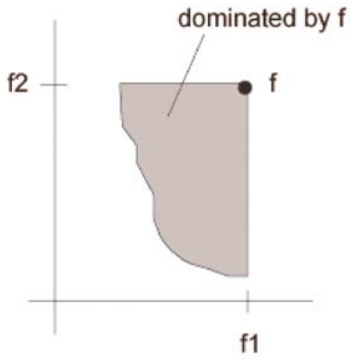

Figure 1: Concept of dominance.

Definition 2. *(Pareto set) Let* $F \subseteq \mathbb{R}^m$ *be a set of vectors. Then the Pareto set (see Figure 2)* F^* *of* F *is defined as follows:* F^* *contains all vectors* $g \in F$ *which are not dominated by any vector* $f \in F$, *i.e.:*

$$F^* := \{g \in F | \nexists f \in F : f \succ g\}.$$

3.3. SPAM-NSGA-II-GP

NSGA-II [15] is a MOEA that incorporates a preservation strategy of an elite population and uses an explicit mechanism (crowded comparison operator) to preserve diversity.

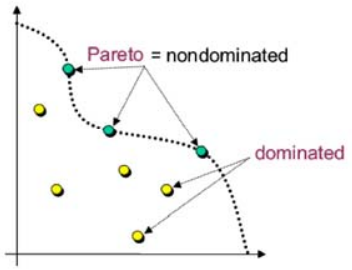

Figure 2: Concept of pareto set.

NSGA-II works with an offspring's population Q_t, which is created using the predecessor population P_t. Both populations (Q_t and P_t) are combined to form an unique population R_t, with a size 2·M, that is examined in order to extract the front of the Pareto. Then, an arrangement on the non-dominated individuals is done to classify the R_t population. Although this implies a greater effort compared with the arrangement of the set Q_t, it allows a global verification of the non-dominated solutions, that belong as well as to the population of offsprings or to the one of the predecessors.

Once the arrangement of the non-dominated individuals finishes, the new generation (population) P_{t+1} is formed with solutions of the different non-dominated fronts ($F_1, \ldots, F_m$), taking them alternatively from each of the fronts. It begins with the best front of non-dominated individuals and continues with the solutions of the second one, and so on.

Since the R_t size is 2·M, it is possible that some of the front solutions have to be eliminated to form the new population.

In the last states of the execution, it is usual that the majority of the solutions are in the best front of non-dominated solutions. It is also probable that the size of the best front of the combined population R_t is bigger than M. It is then, when the previous algorithm assures the selection of a diverse set of solutions of this front by means of the crowded comparison operator (the NSGA-II procedure is shown in Figure 3). When the whole population converges to the Pareto-optimal frontier, the algorithm continues, so that the best distribution between the solutions is assured.

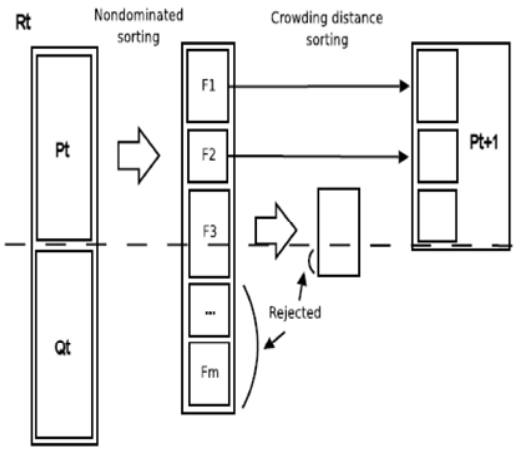

Figure 3: NSGA-II procedure.

In this paper NSGA-II will be adapted to use GP components. It will be denoted as SPAM-NSGA-II-GP.

The SPAM-NSGA-II-GP proposed in this paper has the following components:

- **Codification scheme**: Boolean rules are encoded in expression syntax trees, whose terminal nodes are terms and whose inner nodes are the Boolean operators AND, OR and NOT. Hence, the natural representation is to encode the rule within a tree and to work with a GP algorithm [19] to evolve it, as done by previous approaches devoted to the derivation of Boolean queries in the field of IR [18, 26]. An example of rule can be seen in Figure 4.

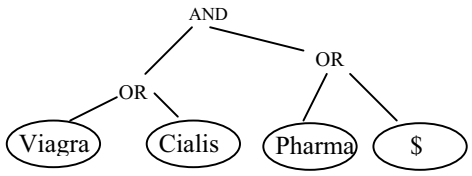

Figure 4. Example of spam filtering rule.

- **Crossover operator**: Subtrees are randomly selected and crossed-over in two randomly selected rules.

- **Mutation operator**: A randomly selected term or operator is changed in a randomly selected tree.

- **Initial population**: All individuals of the first generation are generated in a random way. The population is created including all the terms in the spam e-mails given by the user. Those that appear in more spam e-mails will have greater probability of being selected.

4. Experimental Study

This subsection is divided into two parts according to the following contents: experimental framework and experimental results.

4.1. Experimental Framework

The experimental study has been developed using the PU1 dataset [24]. PU1 is a mixture of 481 spam messages and 618 legitimate messages received by a particular user, after replacing each token (i.e., word, number, punctuation mark, etc.) by a unique number throughout the corpus. Only the earliest five legitimate messages of each sender are retained. Attachments, HTML tags, and duplicate spam messages received on the same day are not included. For more information see [25].

The role of the user who provides spam e-mails will be played by different sets of spam e-mails. In PU1 each e-mail (spam or legitimate) is stored in a separated file. Filenames with the form **spmsg*.txt* are spam messages. Files whose names have the form **legit*.txt* are legitimate messages. The first number in each filename is random; it was originally[1] used to shuffle the messages. The second number was the initial identifier of the message; it does not reflect the order in which the messages were received. To bypass privacy issues, the messages are "encoded", as explained in the paper [25]. The second part in the filename of spam files has an initial letter (*A*, *B* or *C*). Three spam groups have been created. Filenames with letter *A* are associated to group A, those with letter *B* are colleted by the group B, and finally, filenames with letter *C* are grouped by group C. The size of these groups is 166, 146 and 169 respectively. In this way, for example, if there are 166 spam e-mails in group A, this situation will mimic a situation in which the user provides 166 spam e-mails related with his/her undesirable commercial e-mails. Besides, the remaining 933 e-mails (1099 - 166) will be considered as non-spam e-mails for the IQBE process.

[1] This is not used in this paper.

SPAM-NSGA-II-GP generates a set of rules from the spam and the non-spam e-mails sets.

The SPAM-NSGA-II-GP in this contribution has been run 30 times for each spam group (a total of 90 runs) with different initializations. The number of chromosome evaluations is 50.000 per run.

A 2.0GHz Pentium Core 2 Duo computer with 1Gb of RAM was used. The common parameter values considered are shown in Table 1.

Table 1. Common parameters.

Parameter	Value
Cross probability	0.8
Mutation probability	0.2
Maximum nodes per rule	19
Population size (M)	800

4.2. Experimental Results

In Table 2 we present the average number of different rules (in the decision space) for the 30 runs for each spam group for the PU1 dataset. We can observe SPAM-NSGA-II-GP achieves an extra-ordinarily number of different rules in a unique run.

Table 2. Average number of different rules (in the decision space) for each spam group on PU1 dataset.

Group	Rules
A	270,2
B	328,3
C	232,2

In Figure 5, we can see the Pareto in a representative run of SPAM-NSGA-II-GP. We can appreciate that SPAM-NSGA-II-GP achieves 27 filtering rules profiles, from very strong filtering rules (high recall and low precision) to weak rules (high precision and low recall). A set of intermediate profile is also get. With these different filtering rules profiles, user can decide how strong the SF systems will be set. For example, using strong filtering rules, the SF system will block as many spam e-mails as possible, although a high number of legitimate e-mails will be also labelled as spam (false positives). The inverse situation can be also set, i.e., a minimum portion of spam e-mails will be labelled.

5. Concluding Remarks

In this contribution, a MOEA for spam filtering purposes has been developed. This MOEA has been called SPAM-NSGA-II-GP.

It has been tested using a public spam dataset, and the benefits of using MOEAs in the SF process under the IQBE parading have been also proved.

SPAM-NSGA-II-GP provides a flexible way to set a filtering rule profile in a SF system. User can decide the "level of anti-spam security" it is desirable in his/her e-mail client. SPAM-NSGA-II-GP would be continuously learning from the user's e-mails, getting new filtering rules.

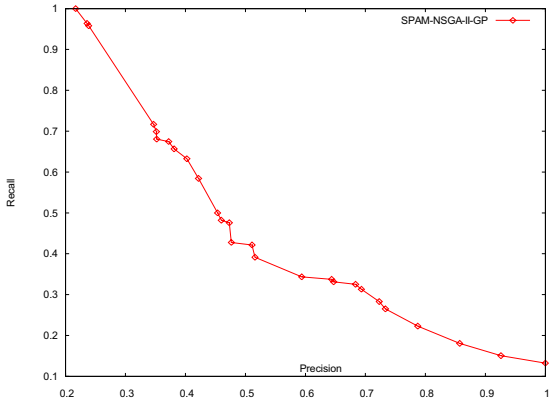

Figure 5. Example of the Pareto front along one representative run of SPAM-NSGA-II-GP on PU1 (group A).

Acknowledgments

This paper has been supported by the projects: FUZZY-LING, Cod. TIN2007-61079 and Proyecto de Excelencia de la Junta de Andalucía SAINFOWEB, Cod. 00602.

References

[1] G. J. Koprowski. Spam accounts for most e-mail traffic, Tech News World (2006). Available: http://www.technewsworld.com/story/51055.html

[2] L. F. Cranor and B. A. LaMacchia. Spam! CACM, 41:8 (1998), pp. 74-83.

[3] S. Machlis. Uh-oh: Spam's getting more sophisticated. Computerworld, Jan 17 2003.

[4] J. Gleick. Tangled up in spam. New York Times, Feb 9 2003. Available: http://www.nytimes.com/2003/02/09/magazine/09SPAM.html.

[5] K. Schneider. Fighting spam in real time. In Proceedings of the 2003 Spam Conference, Jan 2003. Available:

http://www.brightmail.com/press/2003_MIT_ Spam_Conference/.

[6] L. Weinstein. Inside risks: Spam wars. CACM, 46:8 (2003), pp. 136.

[7] S. Hinde. Spam: the evolution of a nuisance. Computer Security 22(6) (2003), pp. 474–478.

[8] Z. Gyongyi, H. Garcia-Molina. Spam: it's not just for inboxes anymore. IEEE Computer 38:10 (2005), pp. 28–34.

[9] Mason J. The SpamAssassin homepage. (2004). Available: http://www.Spamassassin.org/index.html.

[10] G. Sakkis, I. Androutsopoulos, G. Paliouras, V. Karkaletsis, C. Spyropoulos, P. Stamatopoulos. A memory based approach to anti-Spam filtering for mailing lists. Information Retrieval 6 (2003), pp. 49–73.

[11] X. Carreras, L. Marquez. Boosting trees for anti-spam email filtering. In: Proceedings of RANLP-01, 4th international conference on recent advances in natural language processing (2001).

[12] Androutsopoulos I, Koutsias J, Chandrinos KV, Paliouras G, Spyropoulos CD (2000) An evaluation of naive Bayesian antispam filtering. In: Proceedings of the workshop on machine learning in the new information age. 11th European Conference on Machine Learning, Barcelona, Spain (2000), pp. 9-17.

[13] X. Yue, A. Abraham, Z.X. Chi, Y.Y. Hao, H. Mo. Artificial immune system inspired behavior-based anti-spam filter. Soft Computing 11:8 (2007), pp.729-740.

[14] O. Cordón, E. Herrera-Viedma, C. López-Pujalte, M. Luque, and C. Zarco, A review on the application of evolutionary computation to information retrieval, International Journal of Approximate Reasoning 34 (2003), pp. 241-264.

[15] K. Deb, A. Pratap, S. Agrawal, and T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2002), pp. 182-197.

[16] C.J. van Rijsbergen, Information retrieval, Butterworth, 1979.

[17] H. Chen, G. Shankaranarayanan, L. She, and A. Iyer, A machine learning approach to inductive query by example: An experiment using relevance feedback, ID3, genetic algorithms, and simulated annealing, Journal of the American Society for Information Science 49:8 (1998), pp. 693-705.

[18] M. P Smith and M. Smith, The use of genetic programming to build Boolean queries for text retrieval through relevance feedback, Journal of Information Science 23:6 (1997), pp. 423-431.

[19] J. Koza, Genetic programming. on the programming of computers by means of natural selection, The MIT Press, 1992.

[20] D.H. Kraft, F.E. Petry, B.P. Buckles, and T. Sadasivan. Genetic algorithms and fuzzy logic systems, ch. Genetic algorithms for query optimization in information retrieval: relevance feedback, pp. 155-173, Sanchez, E. and Shibata, T. and Zadeh, L.A., Eds., (World Scientific), 1997.

[21] LingSpam dataset. Donwloadable from http://www.aueb.gr/users/ion/data/lingspam/public.tar.gz

[22] Spambase dataset. Donwloadable from http://www.ics.uci.edu/~mlearn/MLRepository.html.

[23] SpamAssasin dataset. Donwloadable from http://spamassassin.apache.org.

[24] PU1 dataset. Donwloadable from http://www.iit.demokritos.gr/skel/i-config/downloads/ pu1_encoded.tar.gz

[25] I. Androutsopoulos, J. Koutsias, K.V. Chandrinos, C.D. Spyropoulos. An experimental comparison of naive Bayesian and keyword-based anti-spam filtering with personal e-mail messages. Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval (2000), pp. 160-167.

[26] O. Cordón, E. Herrera-Viedma, and M. Luque. Improving the learning of boolean queries by means a multiobjective IQBE evolutionary algorithm, Information Processing & Management 42:3 (2006), pp. 615–632.