ISSN:0975-9646

E-mail Spam Filtering Using Genetic Algorithm: A Deeper Analysis

Mandeep Chowdhary , V. S. Dhaka

Jaipur National University,
Jaipur, India

***Abstract*— In the present scenario, Electronic mail holds an intrinsic and an inevitable aspect of all. This attunes spammers on a positive note to utilizing electronic mails to selling their products. Some of the email providers share email data, thus helping spammers to send e-mails in consonance with the end user needs and interests. Henceforth, mailbox gets flooded with spam mails and removal of such mails gets very annoying for the users. A mechanism is needed wherein spam mails can be 'prevented' with a simple procedure. This paper details a genetic algorithm based email filtering process which is evolutionary in nature. The process detailed in the paper allows end users to inhibit spam mails on a single click thereby automatically blocking emails with a similar content. Moreover, various variations are also done for deeper analysis of the algorithm.**

***Keywords*— E-mail, SPAM, HAM, Genetic Algorithms.**

I. INTRODUCTION

Genetic Algorithm is known to be an evolutionary process wherein a population of solutions evolves over a sequence of new generations. Subsequent generations re-evaluate the fitness of each of the solutions thereby 'picking up' the better solutions from the previous ones for reproduction, based on their fitness values. Selections follow the principle of 'Survival of the fittest' owing which 'Good' solutions are selected from 'bad' ones. 'Goodness' of a solution is determined from its fitness value [1].

The selected solutions then undergo recombination subjected to crossover and mutation and sometimes flip operators. The point of emphasis over here is that the genetic representation differs considerably from the natural form of the parameters of the solutions. Fixed, variable length and binary encoded strings for the representation of solutions have dominated GA research because they provide the maximum number of schemata and are amenable to simple implementations. This work accounts for various variations of the GA.

II. GENETIC ALGORITHM DISSECTION

'Crossover' is the epicenter from which the power of GA emanates. Crossover causes a structured exchange of genetic material between solutions transforming 'good' solutions into better ones.

In Mutation, modification of the value of each 'gene' of a solution with some probability say 'p', termed as the 'mutation probability' is done. Mutation has played a vital role in GA in that of restoring lost or unexplored genetic material into the - suboptimal solutions. Auxiliary operations are common in GA. Scaling involves a readjustment of fitness values of the solutions so as to sustain a steady selective pressure in the population and also to prevent the premature convergence of the population into suboptimal solutions.

In uni-modal optimization, it is important that the GA should be able to converge to as fewer a generations as possible. Specific to the case of multimodal functions, the algorithm is able to locate the region in which the global optimum exists and then it converges to the optimum solution.

Also, the algorithm possesses hill-climbing properties essential for multimodal function optimization; however, the properties themselves are vulnerable to getting stuck at a local optimum (notably for smaller population size).

A. Crossover variation

This section discusses the role of the parameter '*p'*, which accounts for the probabilities of crossover and mutation and controls the behavior of the GA. In the past, extensive research has been done in controlling GA performance and has long been acknowledged in GA research [2]. In the past, several studies, both empirical [3], [4] and theoretical [5][6] have been done to identify optimal parameter values for GA. The crossover probability ' *p'*, controls the rate at which solutions are subjected to crossover. Higher value of '*p*', quickly introduces newer solutions into the population. However, as the value of '*p*' increases, solutions are akin to be disrupted faster than what the selections can exploit them.

Typical values of '*p'* lie in the range of (0.5-2.0). Mutation acts only as a secondary operator to restore genetic material. Nevertheless the selection of '*p*', is critical to the performance of Genetic Algorithm and has been emphasized in DeJong's inceptional work [6]. Larger values of *p*, transform the GA into a purely random search algorithm, while some mutation is required to prevent the premature convergence of the GA. '*p*', is typically chosen to lie within the range of (0.005-0.30).

B. Mutation variation

Mutation probability (or ratio) measures the likeliness of the random elements of the chromosome being flipped into something else [7,8]. As an example if the chromosome gets encoded into a binary string of length 100 having a % mutation probability, it implies that 1 out of 100 bits (on an average) shall be randomly picked and flipped.

III. E-MAIL FILTERING PROCESS

Presently filtering technology is divided into two types, one that filters the e-mail addresses while the other that filters the e-mail content. However both the approaches lack intelligence as well as adaptability for the newer emerging spam mails. With the spammers and means of diversification sprouting up, the traditional filter based on the previous technique is difficult to adapt to the newer spam mails. The study of the email structure according to the network information as well as the information on the transmission, to add on to the others, identifies the characteristic features of the spam.

Genetic algorithm can be used as spam classifiers. The collection of the e-mails is termed as 'corpus'. Spam mails comprising the corpus are encoded into a class of chromosomes that undergo genetic operations of crossover, mutation and fitness function. The rules set for spam mails is developed using the genetic algorithm.

A. Rules for classifying the emails:

The weight of the words of 'gene' in the test mail and the weight of words of 'gene' in the spam mail prototypes are compared and the matching gene is found. If the gene matched is greater than some number say 'x' then mail is considered to be the spam mail.

B. Fitness Function:

$$F = \begin{cases} 1 & \text{SPAM mail} \\ 0 & \text{Ham mail} \end{cases}$$

The basic idea lies in finding SPAM and HAM mails from among the mails arriving in the mail box, as the fitness function is itself problem dependent and cannot be fixed initially in the SPAM email filtering.

For the evolution of the fitness function we carried out experiments and found that the minimum score point for the available 1346 SPAM mails was 3. Hence, the' fitness function' was developed:

$$F = \begin{cases} 1 & \text{Score point } \geq 3 \\ 0 & \text{Score point } < 3 \end{cases}$$

A general email mainly consists of three parts; the header, the subject and the body. In Genetic Algorithm based method, the body of the email is scanned, and words are extracted. However, in the extraction process involving some intelligent methods, prepositional and common words like "is, am, are, the" and similar other words are discarded. During the extraction process generally numbers are also discarded. However the exception to the rule is 'pornographic material', in which numbers can also be included, like: (18 years).

C. Procedure:

An email primarily comprises of 'header' and 'message' or 'body'. The header portion has the fields, 'From', 'To', 'CC' (carbon copy), 'BCC' (black carbon copy) and 'Subject'. Genetic algorithm treats header as irrelevant and takes into account only the body, followed by extraction of words from the body. The extraction however excludes articles like "a, an, the, for" and also the numerical numbers. In the first place, genetic algorithm is subjected to the database that classifies spam and ham emails, furthermore classifying the database into several categories. The point of emphasis over here is that as the size of the database increases, the number of words in the data dictionary also correspondingly increases. The selection of categories depends on the classifications of the emails. However the decreasing number of categories is still apt to identify spam mails. This constraint however increases the possibilities of false positive/negatives.

Our experiment in particular considered database of 2448 emails, out of which 1346 were SPAMS and the rest 1102 were HAMS. The data-dictionary in particular considered 421 words which were further divided into seven categories. The data dictionary is presented in [9,10]. The procedure of calculating weights for a particular word of a particular group is mentioned below:

As an example let an email consist of four words namely 'sex', 'nude', 'free' and 'game'. Out of these four words 'sex' and 'nude' fall in the category C_1 while 'Free' and 'Game' to the category C_3 [9,10].

Let us consider an email with 1103 words, out of which 997 have' sex', 'nude', 'free' and' game' as keywords with frequencies of occurrence as '113', '23',' 694' and '167' respectively. These words are taken so large in number so as to ensure that the considered mail is a spam mail as the spam database is very small comprising of only '421' words. To start with, the words extracted from the emails are checked for as to whether they belong to any spam database category or not. In case the words in the email match with the words in the spam data dictionary then the probability of obtaining a word from the spam database is obtained by dividing the frequency of a 'spam' word by the total number of words in data dictionary.

In our case the frequency of occurrence of the word "nude' is 23; hence the probability of getting the word 'nude' is 23/421=0.268.

The weight of the word (W_w) is calculated under the formula

$$W_w = \frac{F_w / T_{WD}}{\sum p_W} \times \frac{S_{WM}}{T_{WM}}$$, where

F_w **:** Frequency of the spam word

T_{WD} **:** Total number of words in the data dictionary

S_{WM} **:** Total number of spam words in the e-mail

T_{WM} **:** Total number of words in the e-mail

$\sum p_W$: Probability of getting a word

The p_w for the word 'sex' is

$$W_w = \frac{F_w / T_{WD}}{\sum p_W} \times \frac{S_{WM}}{T_{WM}}$$

$$W_w = \frac{113/421}{0.268+0.055+1.648+0.397} \times \frac{997}{1103}$$

$$W_w = 0.102$$

The weight of the category is calculated by taking the average of the category.

As an example the weight of category C_1 is (0.102+ 0.021)/2=0.062.

Thus the obtained weight for each word is tabulated in the given Table

Table I:
CALCULATION OF WEIGHTS UNDER AVERAGE WEIGHTAGE METHOD

Group	Word	Frequency	Probability of getting a word	Weight of word	Weight of group
C_1	Sex	113	0.268	0.102	0.062
C_1	Nude	23	0.055	0.021	
C_3	Free	694	1.648	0.63	0.391
C_3	Game	167	0.397	0.151	

Then after normalization the weights are converted to fall in the range of 0.000 to 1.000. Using the hex representation we have

The weight of the gene can be encoded as

Binary 0000000000 represents weight 0.000

Binary 0000000001 represents weight 0.001

Binary 0000000010 represents weight 0.002

……

…….

Binary 1111100111 represents weight 0.999

Binary 1111111000 represents weight 1.000

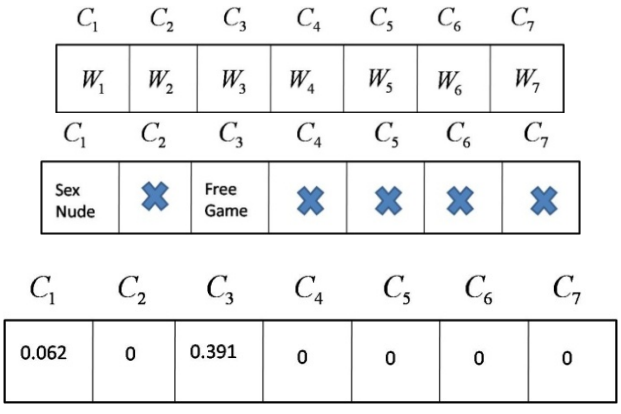

Fig. 1 SPAM chromosomes prototype

As discussed above, each mail is encoded into chromosomes consisting of 70 bits, which are further sub divided into 7 equal groups. Each group of 10 bits represents the hex number of the probability of the words lying in a particular group. As shown below in the figure 2.

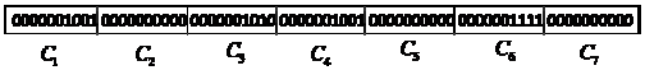

Fig.2 Chromosomes construction for a SPAM mail

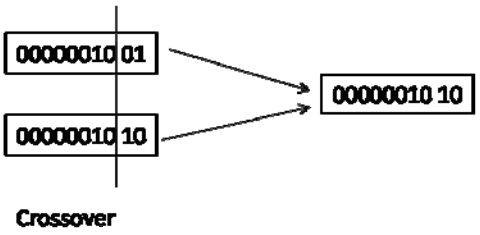

Fig.3 One point crossovers in chromosomes

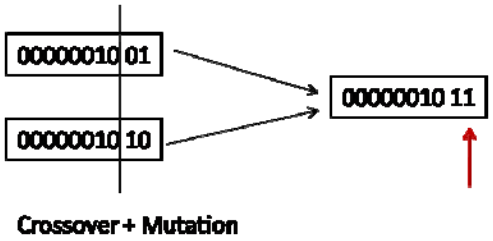

Fig.4 One point crossover and mutation in chromosomes

The process of Genetic Algorithm starts and the cross over takes place once the chromosomes are constructed for all the mails. As discussed above there are various ways in which the cross-over operation can be performed. Crossover is only allowed for a bit of gene in a particular category only.

In our algorithm, both multi-point and single point crossover is done and the positions of the bits are selected randomly. In each generation of chromosomes only 12% of the bits are crossed.

Next follows the process of mutation. (Figure 4) so as to recover some of the lost genes. In our example only 3 % of genes are mutated.

The process starts by comparing the weight of the words of the gene in the test mail with those of the gene in the spam mail prototype so as to find the matching gene. If number of matched genes is greater than or equal to the numerical value three, than spam mail prototype receives a single score point.

If the score point happens to be greater than say some threshold score point than the mail is considered as a spam mail. However, the threshold point can be manually attuned to obtain the appropriate results. It must however be kept in mind that we have used the fitness function on the basis of our experimental results. There lies no doubt that the fitness function is depends on the parameters presented in Table 6.1; however to guess an appropriate fitness function is quite an ardent task.

Our experiments conducted above used two mail spools; The first one comprised of SPAM mails,1346 in number, while the other consisted of 1102 HAM messages. It is noticeable that the evolutionary process takes a long time to execute thus promoting an increased size of the selected database.

IV. RESULTS

A. Experiment 1

During the first experiment, a SPAM corpus of 1346 mails is taken into account, while on the other hand 1102 mails are tested with the developed algorithm. The size of the chromosome is kept to be fixed and is of 70 bits. The different variations of cross-over (Uniform, Non-uniform, permutations and Gaussian) are henceforth applied, varying the probability from 5% to 15%. The obtained results are presented in Table I. As is evident from the Table, when the corpus size is large there is no effect of the cross-over type and cross-over probability and efficiency of SPAM mail identification stands to be 81.67.

Table: I Crossover vs. efficiency (Corpus 1346 mails chromosomes size 7 total numbers of bits 70 with crossover variation)

Cross-over type	Probability	Efficiency
Uniform	5	81.67
Non uniform	5	81.67
permutation	5	81.67
Gaussian	5	81.67
Uniform	10	81.67
Non uniform	10	81.67
permutation	10	81.67
Gaussian	10	81.67
Uniform	15	81.67
Non uniform	15	81.67
permutation	15	81.67
Gaussian	15	81.67

These stand as the expected results. As already stated, owing to the large mail corpus size, there will minimal or altogether no effect of cross-over type and probability, it eventually converges to the best optimum result.

B. Experiment 2

In the second experiment, the mail corpus sums to 2448 mails (1346 + 1102). The chromosome length is kept consistent to be seven; total numbers of bits are seventy (Table II) . The crossover probability is kept at 15%, while mutation is varied from 2% to 5%. Again the efficiency is calculated to be 81.67%.

Table: II Mutation vs. efficiency (Corpus 1346 mails chromosomes size 7 total numbers of bits 70 with mutation variation)

Mutation type	Probability	Efficiency
Uniform	2	81.67
Non uniform	2	81.67
permutation	2	81.67
Gaussian	2	81.67
Uniform	3	81.67
Non uniform	3	81.67
permutation	3	81.67
Gaussian	3	81.67
Uniform	5	81.67
Non uniform	5	81.67
permutation	5	81.67
Gaussian	5	81.67

To re-enforce, it is very obvious from the above results that when corpus size is large (or population is sufficient) enough, there is no effect of crossover and mutation on the overall efficiency.

C. Experiment 3

In the third variation, we have kept the corpus as it is, with the crossover and mutation stable at 15% and 3% respectively, varying the chromosome length from 4 to 8. It is evident from the table that the efficiency increases as the length of the chromosomes increases from 4 to 8. When, the length of the chromosome is 4, the efficiency is only 54.11% and when the length of the chromosome is 7, the efficiency is 81.67%.Chromosomes of length 8 depict somewhere around an efficiency of 82% Table III.

Table: III Chromosomes size vs. efficiency (Corpus 1346 mails crossover 15% and Mutation 3%)

Chromosome size	Efficiency
4	54.11
5	56.23
6	68.17
7	81.67
8	81.7

In the next variation the size of the SPAM database has been varied from 100 mails to 1346 mails with a gap of 100 emails. Here and hence it is clear from the table above that the efficiency is very less when the number of mails is less; in-fact for 100 mails the efficiency is only 31.33% (Table IV). Similarly for 500 mails, the efficiency is 60.34%. However, as we approach 900 mails the efficiency touches 80% mark and adheres somewhere around it for a higher number of mails.

Table: IV SPAM mail database size vs. efficiency (Chromosomes size 7, crossover 15% and Mutation 3%)

SPAM Mails database Size	Efficiency
100	31.33
200	40.15
300	50.67
400	55.31
500	60.34
600	70.17
700	78.77
800	79.80
900	80.17
1000	81.23
1100	81.60
1200	81.66
1346	81.67

In our previous results we discovered that, if number of words in the mail is large enough, then more correct a classification is possible. Our algorithm has been checked upon a large corpus and found that nearly 82% mails have been correctly classified using the method. The score point varies from 5 to 173; therefore this method is much better in comparison to the previous methods.

V. CONCLUSIONS

In this paper, various variations of the genetic algorithm have been tried and it has been found that the variation in genetic algorithm parameters have a greater influence on the overall performance of the Genetic Algorithm. It has also been discovered that the length of the chromosome does have an impact on the efficacy and efficiency of the algorithm and that the length of the chromosome should not be less than the certain minimum length. Paradoxically, if mail the corpus contains larger number of mails (> 500) then the type and percentage of crossover and mutation does not have any impact on the overall efficiency. Moreover, the size of corpus is not detrimental to the overall efficiency of the GA; on the contrary the efficiency increases as the number of mails in the corpus increases. However, this improvement in efficiency is not progressive, and gets saturated if the sufficient numbers of mails are available in the mail corpus.

REFERENCES

[1] S. Haykin, "Neural Networks: A Comprehensive Foundation", MacMillan College Publishing Company, New York, 1995.

[2] J. D. Schaffer and A. Morishma, "An adaptive crossover mechanism for genetic algorithms," in Proc. Second Int. Con$ Genetic Algorithms,1987, pp. 3640.

[3] L. Davis (Ed.), Genetic Algorithms and Simulated Annealing, London:
Pitman, 1987

[4] L. Davis, "Adapting operator Probabilities in genetic algorithms," Proc.Third Int. Genetic Algorithms, 1989, pp. 61-69. - (Ed), "Handbook of Genetic Algorithms," Van Nostrand Reinhold, 1991.

[5] K. A. DeJong, "Genetic algorithms: A 10 year perspective" in ***Proceedings***of an International Conference of Genetic Algorithms and their***Applications,*** **(J** Greffenstette, editor), Pittsburgh, July 24-26, 1985, pp.

[6] K. A. Ddong, "Adaptive system design: a genetic approach," ***ZEEE Trans Syst, Man, and Cybernitics,*** vol. 10, No. 9, pp. 566-574, Sep.1980.

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

[8] Tang K.S. et.al. 1996. Genetic Algorithm and Their Applications, IEEE Signal Processing magazine, pp.22-37.

[9] Sanpakdee U. et.al. 2006. Adaptive Spam Mail Filtering Using Genetic Algorithm.

[10] Spam Assassin, http://spamassassin.org.