Improving Genetic algorithm for the web pages classification problem

Solmaz Hedayati

Department of Computer Engineering, College of Engineering and Technology, Kazerun Branch, Islamic Azad University, Kazerun, Iran
hedayati_solmaz@yahoo.com, hedayati.solmaz62@gmail.com

Abstract: In the past two decades, there have been proposed several algorithms for the web pages classification problem that majority of them are in the field of evolutionary algorithms such as Genetic algorithm. Moreover, Performance of Genetic algorithm associated with the web pages classification problem is significantly positive. In this paper, we try to improve both operators crossover and mutation of Genetic algorithm which have upgraded performance of this algorithm on the web pages classification problem. Then, We have attempted to find an appropriate adjustment of two operators: crossover and mutation and the most possible and proper examples in each generation, and then use the stated samples in subsequent generations. In other words, we have tried to maintain good features and good examples of each generation to pass on future generations. It can be called the exploitation of knowledge problem, expresses that we used the previous steps to take next steps. We also examine aspects of exploration and exploitation by setting the operators: crossover and mutation. Thus, the proposed algorithm has been moving in a right direction to find the best example or classifier. Furthermore, we have tested the proposed algorithm on benchmark datasets such as Conference, Course, Student. Our conclusion shows better performance of the proposed algorithm in comparison to basic Genetic algorithm, NB and KNN classifiers. The proposed algorithm shows 97% accuracy on a dataset Course.

[Hedayati S. **Improving Genetic algorithm for the web pages classification problem.** *Researcher* 2014;6(7):32-37]. (ISSN: 1553-9865). http://www.sciencepub.net/researcher. 5

Key words: Genetic algorithm, Operator crossover, Operator mutation, Explore, Exploit.

1. Introduction

The web pages classification problem can be stated as follows: determine whether a web page belongs to a group or groups of pre-defined or not. This inference process makes the learning model through a set of web pages that previously have been classified, and then the learned model are used to classify new web pages that have been unseen before(Choi & Yao, 2005; Qi & Davison, 2007; Zu Eissen & Stein, 2004). In the past two decades, In addition to well known and fundamental algorithms in the field of web pages classification, several algorithms have been proposed in various areas of science. Evolutionary algorithms such as Genetic algorithm including the algorithms used in this context, Which is used either alone or in combination with other methods to solve the problem of web pages classification.

Began applying Genetic algorithm to classify web pages since 2003 by Ribeiro et al(Riberio & Fresno & Garcia-Alegre & Guinea, 2003). They offered a web page classifier form "If condition then class i". The preceding condition include all of the words within feature vectors of the training set, which the degree of relevance of each word in terms was determined by fuzzy membership function such as "Medium Relevance", "Not Relevance" and "Extreme Relevance". Experimental results show that the proposed method in this context is suitable for resolve the ambiguity inherent problem of web pages classification. In 2004, a classifier web pages Based on combination of Genetic and k-means clustering algorithm was presented (Qi & Sun, 2004). In this study, for each group, a set of keywords from training web pages was produced, and the next step, for each keyword was considered an initial weight in each group. Then GA was used to optimize the weight words of each group. Test results indicated a good performance of Genetic Algorithms in the proposed method.

In 2007, the theory of rough sets and Genetic were used to reduce the dimension of feature vector, and then SVM was used to classify new web pages(Bia & Wang & Liao, 2007). High-dimensional feature vector web pages will affect on the speed and accuracy of the classification. Also configuration parameters in SVM, and select the prominent features other factors affecting on the accuracy of classification. The goal of the proposed method was mentioned to reduce the dimensions of feature vectors and optimize the parameters of SVM classifier to improve speed and accuracy of SVM. Feature vectors was reduced by the theory of rough sets and was selected, and then SVM parameters were optimized by GA. Experimental results show that this method has better performance than traditional SVM and other traditional classifier. In 2008, a Genetic called Olex-GA was introduced. The proposed algorithm was used a set of rules to classify web pages in different groups (Pietramala & Policicchio & Rullo & Sidhu, 2008). If document D includes terms t

$_1$ or ....or t_n , but does not include terms t_{n+1} or ..... or t_{n+m}, then the document D is classified in class C. Olex-GA was used the F-measure as the fitness function. Olex-GA performance was compared with those famous classifier such as Naive Bayes, Ripper, C4.5, and SVM. Experiments showed that this algorithm has achieved very good results.

In 2010, was proposed a web pages classification system based on Genetic algorithm, which was used the terms within each tag and HTML tags as features (Ayse Ozel, 2010). in existing GA-based classifiers, only HTML tags or terms in each tag are used as features, however in this study both of them were taken together and optimal weights for the features were learned by GA. The proposed algorithm was compared with algorithms such as Naive Bayes and kNN. Experimental results show that the proposed algorithm is more accurate than those based classifier. In 2011, was proposed a Genetic algorithm to select the best features for the problem of web pages classification to improve the accuracy and running time of the those classifiers such as KNN,NB and Decision Tree (Ayse Ozel, 2011). In this study, to reduce the feature space, a Genetic algorithm developed to determine the best features for a given set of web pages. In this algorithm was used all HTML tags and terms in each tag as features, which increases the accuracy of the classification. The best classifier found by Genetic was used in the process of classifying new web pages by KNN, NB and decision trees, and the best performance was observed in KNN classifier. In 2012, was proposed a classification system of Chinese web pages based on feature selection algorithm called CFS-GA (Correlation-based Feature Selection and Genetic Algorithm)(Chunpin & Xiaoxia, 2012). The purpose of this new algorithm was mentioned dimension reduction of feature space which this goal was achieved by using Genetic. This algorithm was used a subset of features to represent a chromosome, and then the chromosome was shown to be a binary code. CFS was used as the fitness function of Genetic algorithm to evaluate the chromosomes. The results show that this algorithm can effectively reduce the dimensions of feature space and increase the accuracy of classification.

2. Materials and Methods

2.1. Feature extraction and selection

Each term in its own tag in a web page is considered as a feature, It makes the choice of more distinctive and important features. This action makes the same terms in different tags have different values. Most important terms are observed in tags like <title>, <h1>, <h2>, <h3>, <a>, <em>, <strong>, <b>, <i>, <p>, <li>. method of find features is in this case which features are extracted from positive training web pages. First, redundant symbols and terms on each page eliminate, and then Porter's stemming algorithm is applied on the remaining terms in a web page. Thus, the terms will be specified on each page, and then features (terms) tags in a Web page are extracted using feature extraction algorithm that is shown in Figure (1).

```
Algorithm : Feature Extraction
Input : Positive Web pages in the training dataset, Stopwords list.

Output : Feature list.
Titel={ } , Header={ } , Anchor = { } , Bold={ } , List_Item={ } , Paragragh = { }
for each positive Web page p in the training dataset do
    for   each word w   in p   do
        if   w   is not a stopword then
            if   w   belongs to <title> tag then
                Title=Title ∪ stem(w)
            else if   w   belongs to <h1> or <h2> or <h3> tag   then
                Header = Header ∪ stem(w)
            else if   w   belongs to <a>   tag then
                Anchor= Anchor ∪ stem(w)
            else if   w   belongs to <em> or <strong> or <b> or <i> tag then
                Bold= Bold ∪ stem(w)
            else if   w   belongs to <Li> tag then
                List_Item= List_Item ∪ stem(w)
            else if   w   belongs to <p> tag then
                Paragragh= Paragragh ∪ stem(w)
        end if
    end for
end for
```

Figure 1. Feature extraction algorithm

Feature extraction algorithm is applied on all positive training pages. Thus, we have a list of features that include terms tags in positive training pages. Terms tags in features list are listed in order of title, header, anchor, bold, list_item, paragraph. in order to select best features is used benchmark document frequency(df). df of a feature is defined as the number of positive pages in the training dataset that contains the feature. Higher df means that this feature has appeared in more number of positive training web pages. In the proposed algorithm have been chosen features with df = 41, and accordingly the list of features has made.

2.2. Construct vector web pages

To construct vector of a web page such as the Section 2.1, terms tags and frequency of occurrence of each terms in relevant tag from a web page is extracted. Then be divided The number of repetition of terms per tag on repetition maximum terms in the same tag. Thus justify the weight terms tags of a web page in the interval [0,1]. This action is named normalization. Formula (1) is representing vector of a web page:

$$D=(d_{11}, d_{12}, \ldots, d_{1N_1}, d_{21}, d_{22}, \ldots, d_{2N_2}, d_{m1}, d_{m2}, \ldots, d_{mN_m}) \qquad (1)$$

d_{ij} shows the weight of the feature (term) j in tag i of the page. Note that always the length of the vector web page is the same with length list of features.

2.3. Create initial population

The first step of Genetic algorithm is to create the initial population. In this algorithm, 30 initial chromosomes (classifier) are generated randomly(popsize=30) that the length of each chromosome is the same with the length of feature list. Random values of genes in a chromosome to represent the weight of the features included in the list of features. In fact, we are looking for the best combination of weight of features in the list of features, until to find the best classifier for classification. This feature selection method is a wrapper method. Formula (2) is representing a chromosome:

$$W=(w_{11}, w_{12}, \ldots, w_{1N_1}, w_{21}, w_{22}, \ldots, w_{2N_2}, w_{m1}, w_{m2}, \ldots, w_{mN_m}) \qquad (2)$$

w_{ij} is the weight of feature j in tag i of a sample chromosome which is generated randomly.

2.4. Evaluate initial population

To evaluate each chromosome(Individual), the cosine similarity of the vector positive and negative training pages with the chromosome is calculated using the formula (3):

$$sim(w,d)=\frac{\vec{w}*\vec{d}}{\left|\vec{w}\right|*\left|\vec{d}\right|}=\frac{\sum_{i=1}^{m}\sum_{j=1}^{N_i} w_{ij}*d_{ij}}{\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{N_i} w_{ij}^2}*\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{N_i} d_{ij}^2}} \qquad (3)$$

Threshold value to be calculated for each of the chromosomes. The cosine similarity of the positive pages with The sample chromosome are placed in a list, and in continues list, the cosine similarity negative pages with the same chromosome be placed (value list). The parallel to the value list, another list is considered that the values of its homes are for each cosine similarity positive pages 1 and for each cosine similarity negative pages -1 (key list). Then, the two lists are sorted Based on the amounts of value list . A pointer to the name of k is considered which began to move to the top of the key list. If key [i] <0 is, then k moves forward one unit (k = k +1), otherwise k does not move. Finally, k reaches a the position of the key list is no longer able to move and stops. In this situation, the threshold of chromosomes is calculated using the formula(4):

$$\frac{value[k]+value[k+1]+value[k-1]}{3}=\text{Threshold} \qquad (4)$$

Now, by the cosine similarity values of positive and negative pages with chromosome and the Threshold of the same chromosome, chromosome fitness value is calculated using the formula (5). In this formula, TP is the number of positive pages are correctly labeled, TN is the number of negative pages are correctly labeled, FP is the number of negative pages are incorrectly labeled, and FN is the number of positive pages that are incorrectly labeled. Cosine similarity value of each positive pages are compared with the Threshold chromosome, If it was higher than Threshold value, TP is increased one unit, otherwise FN increases one unit. Then, the cosine similarity value of each negative pages are compared with the Threshold chromosome, If it was less than Threshold value, TN is increased one unit, otherwise FP increases one unit. Thus, the fitness value of each chromosome of initial population is computed.

$$Fitness=\frac{(double)TN+TP}{(double)TP+FN+TN+FP} \qquad (5)$$

2.5. Reproduction

Select the best chromosomes is performed as follows: First, the selection probability of each chromosome is calculated using the formula (6):

$$P_X = \frac{F_X}{\sum_{i=1}^{popsize} F_i} \quad (6)$$

F_X is fitness value of Chromosome x which is calculated using the formula (5). Then an aggregate probability C_X for chromosome x is defined. the aggregate probability of x is the sum of the selection probability of the preceding chromosomes. Aggregate probability for chromosome x is calculated using the formula (7):

$$C_X = \sum_{i=1}^{X} P_i \quad (7)$$

Aggregate probability is calculated for all chromosomes in population. The following procedure is repeated popsize times. At any time, a random number r is generated in the range [0,1].The aggregate probability of each chromosome of population is compared with r, If the aggregate probability of chromosome x was greater than r or was equal to r, this chromosome will be selected as the parent ($c_{i-1} < r <= c_i$). The method explained above is the Roulette Wheel method.

2.6. Crossover

Crossover rate is equal to 0.8 in each generation. The alpha coefficient is defined using the formula (8):

$$alpha = 0.5 + \frac{generation}{gensize} \quad (8)$$

generation parameter represents the number of past generations of total generations (the total number of generations is equal to 400 that called gensize). The following steps popsize / 2 times are repeated on the parents was selected of the phase Reproduction. First, a random number r is generated in the interval[0,1], If r is smaller than the crossover rate, producing new child are done, otherwise their parents no changes will be considered as new child. If r is smaller than 0.8, a random number is generated between 0 and length of the chromosome, which represents the breakdown point is the parents (PointToCrossover). Then the gene values (new values are obtained using formulas 9 and 10) from the beginning to the PointToCrossover point will be transferred to the child. The fitness of the first parent (Population [i]) is in the firstfitness parameter, and the fitness of the second parent (Population[popsize-1-i]) is in the secondfitness parameter. If the fitness of the first parent was less than the fitness of the second parent, The alpha value will change to alpha = 1-alpha, otherwise it maintains the previous value itself. Because of we want consistently produced values in the range [0,1]. Gene value of the first parent is located in temp parameter, and the corresponding gene value in the second parent is located in temp1 parameter. Then according to the formulas (9) and (10), two new value is generated for these genes in two new child.

$$Population[i].genes[j] = alpha * temp + (1 - alpha) * temp1 \quad (9)$$

$$Population[popsize - 1 - i].genes[j] = (1 - alpha) * temp + alpha * temp1 \quad (10)$$

Use alpha and formulas 9 and 10 is for this purpose that we want the parent who is more fitness has more effective in producing new child. Thus, Characteristics of good parent are maintained, and are transmitted to next generations. This is the same exploit from found good examples. Progressively, the amount of alpha with the progresses of generations will be greater, And as a result the ability of exploit increases. This action will make the appropriate motion proposed algorithm in order to find better examples.

2.7. mutation

Mutation rate is different for different generations. This rate is calculated using the formula (11):

$$rate = 0.45 - \left(\frac{(double)0.5 * i}{gensize} \right)$$

$$\text{if (rate} < 0.05\text{) then rate=0.05} \quad (11)$$

According to formula 11, the value of changes of rate parameter is located in the interval [0.05, 0.45]. The maximum rate occurs in the first generation which is equal to 0.45, and the minimum rate occurs in the last generation (generation = 400) which is 0.05. The mutation rate is high at the beginning of the launch of Genetic algorithm. And Progressively with the progresses of the generations decreases. Because of we want at the beginning of the launch of Genetic algorithm explore value is high to search different parts of the problem space, And find good examples. Progressively, the explore value is reduced with the progresses of generations, because of we want to keep good examples that up to now have been found, and the search must continue around them. After identifying the mutation rate per generation, mutation operation is performed on chromosomes obtained from the phase crossover. Mutation operation is as follows. For all genes on each chromosomes is generated a random number r. If r <rate was, a random value is generated in the interval [0,1], And in the gene value of the desired chromosome is replaced, Otherwise the gene value will remain unchanged.

2.8. Selection

At this point, we are faced with the chromosomes that crossover and mutation operators are applied on them. Now, must select the best chromosomes from the new population (population obtained after the crossover and mutation operators in each generation) and the initial population. For this purpose, the fitness of each chromosomes in the new population is calculated in accordance with Section 2.4 and placed on the value list. The fitness of each chromosome in the initial population is calculated in accordance with Section 2.4 and are placed in continued value list. The Parallel to the value list is considered the key list that Home values of the key list is equal to home indexes of the value list. Then, the two lists are sorted based on amounts of the value list. Finally, 30 of the best chromosomes are selected using key list and will be considered as the initial population of the next generation.

phase 2.5 to 2.8 are repeated 400 times(the total number of generations). In the last generation, the fitness values of all chromosomes (population is obtained from the phase selection in the last generation) are compared to each other, and the most fittest chromosome is introduced to as the best classifier.

2.9. Classification

The best classifier was found in the training stage will be transferred to the testing stage. This classifier is a chromosome that represents the best combination of weight of features in the feature list. Now using this classifier, web pages that had been unseen to be classified. Such as training stage, At the first, the cosine similarity of the classifier with positive and negative testing pages are calculated. Then using Threshold classifier (Threshold classifier is computed during the training stage), the accuracy rate will be calculated by the formula (5).

3. Result

The proposed algorithm has been implemented under the c # programming language version 2010. This algorithm has been tested on a system with the characteristics CPU intel core i7 2.0GHz , RAM 8GB and Windows 7 operating system. Three datasets Conference, Course and Student are used to evaluate the performance of the proposed algorithm. The Conference dataset consists of the computer science related conference homepages. We labeled the conference web pages as positive pages in the dataset. To completed the dataset the short name of the conferences were queried using the google search engine, and the irrelevant pages in the result set were taken as negative pages. Conference smallest dataset used in this implementation. The Course and Student dataset are well known and freeware datasets that were obtained from WebKB project web site. The Course dataset contains computer science related course homepages and some irrelevant web pages. The Student dataset consists of graduate student' homepages from Cornell, Texas, Washington, and Wisconsin universities as well as some irrelevant pages from those four universities. The details about the datasets and the class names for each set are given in Table (1).

Table 1. Number of pages in each datasets

Dataset	Class	Number	Total
Conference	Conference Not Conference	235 57	292
Course	Course Not Course	230 821	1051
Student	Student Not Student	1641 3764	5405

To divide positive and negative pages of every one of datasets Between training and testing stages is used Ten-Fold Validation method. Performance of the proposed algorithm is compared with those basic classifiers such as Naive Bayes (NB), K nearest neighbor (KNN), and a basic Genetic algorithm (Ayse Ozel, 2010). Test results are presented in Table (2).

Table 2. Results of tests performed by improved Genetic algorithm, basic Genetic algorithm, NB classifiers, KNN classifiers

Student Dataset					Course Dataset					Conference Dataset				
df	Improved GA	GA	NB	KNN	df	Improved GA	GA	NB	KNN	df	Improved GA	GA	NB	KNN
90	**0.90**	0.82	**0.48**	0.86	20	0.94	0.93	0.62	0.88	15	0.85	0.73	0.62	0.83
180	0.89	**0.89**	0.26	**0.88**	30	0.95	0.92	0.71	0.89	20	0.86	0.80	0.62	0.85
285	0.86	0.87	0.24	**0.88**	**40**	**0.97(df=41)**	0.92	0.85	0.89	**25**	**0.87**	0.83	0.62	0.89
370	0.86	0.87	0.21	0.85	50	0.96	0.94	0.87	0.84	30	0.84	**0.86**	**0.62**	**0.90**
460	0.85	0.85	0.17	0.83	60	0.95	**0.95**	0.84	0.90	35	0.81	0.72	0.62	0.79
565	0.83	0.85	0.17	0.82	70	0.93	0.90	0.84	**0.91**	40	0.76	0.77	0.62	0.76
650	0.83	0.85	0.17	0.81	80	0.92	0.91	**0.89**	0.90	45	0.76	0.77	0.62	0.76

4. Discussions

Genetic algorithm has good performance at the beginning of the launch, But over the generations, its performance will be weak due to the crossover and mutation operators. It is possible crossover and mutation operators generate examples (chromosomes) which are inappropriate and led to the algorithm move in the wrong direction. In the proposed algorithm to avoid this problem particular configurations is done. The crossover operator by defining alpha coefficient and provide formulas for generating new instances, tries that the good examples are preserved and passed to future generations. This makes the proposed algorithm fewer number of false samples produces, and most continue to search around good examples.

In the mutation operator, mutation rate is variable. This means that at the beginning of the launch of the Genetic algorithm, the mutation rate is high, and Progressively and with the progresses of the generations decreases. This makes the explore property of Genetic algorithm be maintained in a meaningful manner, and algorithm be guided properly to achieve the optimal solution . Finally, an improved Genetic algorithm is achieved that has a good performance in comparison with classifiers such as NB, KNN, and basic Genetic algorithm.

Correspondence To
Solmaz Hedayati,
Department of Computer Engineering, College of Engineering and Technology, Kazerun Branch, Islamic Azad University, kazerun, Iran.
Current Address: Department of Computer Engineering, College of Engineering and Technology, Kazerun Branch, Islamic Azad University, Kazerun, Iran.
Email: hedayati_solmaz@yahoo.com
hedayati.solmaz62@gmail.com

7/13/2014

References

1. Chio B, Yao Z. web Page Classification. Springer 2005;Volume 180, 1434-9922, 221-274.
2. Qi X, Davison BD. Web Page Classification: Features and Algorithms. Department of Computer Science and Engineering Lehigh University 2007.
3. Zu Eissen SM, Stein B. Genre classification of web pages. Springer 2004; Volume 3238, 256-269.
4. Ribeiro A, Fresno V, Garcia-Alegre MC, Guinea D. Web page classification: A soft computing approach. Lecture Notes in Artificial Intelligence 2003; 103-112.
5. Qi D, Sun B. A genetic k-means approaches for automated Web page classification. In IEEE international conference on information reuse and integration 2004; 241-246.
6. Bai R, Wang X, Liao J. Combination of rough sets and genetic algorithms for text classification. Springer 2007; 256-268.
7. Pietramala A, Policicchio VL, Rullo P, Sidhu I. A genetic algorithm for text classification rule induction. Springer 2008; 188-203.
8. Ayse Ozel S. A Web page classification system based on a genetic algorithm using tagged-terms as features. Expert Systems with Applications (Elsevier) 2010.
9. Ayse Ozel S. A Genetic Algorithm Based Optimal Feature Selection for Web Page Classification. IEEE 2011; 282-286.
10. Chunping Y, Xiaoxia H. Chinese Web page classification based on CFS-GA feature selection algorithm**.** Journal of Shanghai Maritime University 2012.