International journal of Computer Science and its applications [ISSN 2250 - 3765]

Genetic Algorithm for Classification of Web Documents

Anuradha Purohit, Kirti Arora, Neela Pandit, Smita Sharma, Shalini Bansal

Computer Technology and Applications Department, SGSITS, Indore (M.P)

anuradhapurohit@rediffmail.com
arorakirti.8@gmail.com
soniyapandit12@gmail.com
smitamca.gsits@gmail.com
shalinibansal94@yahoo.com

***Abstract*— This paper presents a Genetic Algorithm (GA) based approach for selecting small number of representative features from a large dataset thereby leaving irrelevant features. Fitness function is used to decide the fitness of each individual using k-NN classifier and genetic algorithm. Real dataset such as Bank search dataset is used to test the proposed approach. Experimental results show that the number of features have been reduced significantly by using genetic algorithm.**

***Keywords*— Genetic Algorithm, k-NN classifier, crossover, mutation, reproduction Text classification, Similarity metrics, dictionary, fitness function.**

1. INTRODUCTION

With the increasing popularity of the Internet and the large number of documents available in electronic form such as PowerPoint, Word, Text, Pdf, and Flash as well as HTML files it is increasingly difficult to find specific data. If web pages can be categorized without human intervention, searches can then be applied to more limited categories instead of the entire web.

Text categorization or text classification can be framed as a supervised learning task in which a classifier attempts to learn a relationship between a training set of documents and their categories[1]. Document classification is a problem of assigning an electronic document to one or more categories. In our paper, we perform supervise document classification by using Nearest Neighbour Classifier (NNC) and genetic algorithms. A NNC approaches the problem of text classification by computing a similarity metric between feature vector representation of an unknown document and a set of known prototype documents. A major problem with text classification is the high dimensionality of the feature space. This paper investigates how genetic algorithm and NNC can help select relevant features in text classification. The accuracy and speed of NNC are dependent upon the choices of features.

Genetic algorithm is inspired by the mechanism of natural selection where stronger individuals are likely the winners in a competing environment. Genetic algorithm presumes that the potential solution of any problem is an individual and can be represented by a set of parameters. These parameters are regarded as the genes of a chromosome and can be structured by a string of values in binary form. A positive value, generally known as fitness value, is used to reflect the degree of "goodness" of the chromosome for the problem which would be highly related with its objective value. Throughout a genetic evolution, the fitter chromosome has a tendency to yield good quality offspring which means a better solution to any problem [2].

2. GENETIC ALGORITHM

Genetic algorithms (GAs) are efficient, adaptive and robust search and optimization processes that are usually applied in very large, complex and multimodal search spaces. GAs are loosely modelled on the principles of natural genetic systems, where the genetic information of each individual or potential solution is encoded in structures called chromosomes[3]. They use some domain or problem-dependent knowledge to compute the fitness function for directing the search in more promising areas. Each individual or chromosome has an associated fitness value, which indicates its degree of goodness with respect to the individual it represents. GAs search from a set of points, called a population. Various biologically inspired operators like selection, reproduction, crossover and mutation are applied on the chromosomes in the population to yield potentially better solutions [3].

Basic Algorithm

Step 1 (Initialization):
Randomly generate strings of the length n .

Step 2 (Genetic Operations):
Iterate the following procedures number of times for generating strings.

i) Randomly select a pair of strings from the current population.

ii) Apply a crossover operation to the selected pair of strings for generating two offspring.

iii) Apply a mutation operation to each bit value of the two strings generated by the crossover operation. The mutation operation changes the bit value from 1 to 0 or from 0 to 1.

Step 3 (Generation Update):
New generation is created replacing the old generation. Select the best strings with the largest fitness values from the enlarge -ed population to form the next population.

Step 4 (Termination Test):
If a pre-specified stopping condition is not satisfied, return to Step 2. Otherwise end the algorithm.

2.1 Genetic Operators

Selection: Selection is a genetic operator that chooses a chromosome from the current generation's population for inclusion in the next generation's population. There are five types of selection:

1. Roulette wheel
2. Tournament
3. Top percent
4. Best
5. Random

Reproduction: Individual solutions are selected through a fitness-based process, where fitter solutions (as measured by a fitness function) are typically more likely to be selected.

Crossover: It is a genetic operator that combines (mates) two chromosomes (parents) to produce a new chromosome (offspring)[5]. Here we are using one single point crossover with tournament selection.

A crossover operator that randomly selects a crossover point within a chromosome then interchanges the two parent chromosomes at this point to produce two new offspring.

Consider the following 2 parents which have been selected for crossover. The "|" symbol indicates the randomly chosen crossover point.

Parent1: 11001|010
Parent2: 00100|111

After interchanging the parent chromosomes at the crossover point, the following offspring are produced:

Offspring1: 11001|111
Offspring2: 00100|010

Mutation: The mutation operator randomly transforms the value of an attribute into another value belonging to the same domain of the attribute.

Consider the following parent which have been selected for mutation. The "|" symbol indicates that the value of the randomly chosen attribute between it has been changed.

Parent: 1100|1|010
offspring: 0010|0|111

3. k-NN CLASSIFIER

In the k-nearest neighbor classification (Cover and Hart, 1967), each new instance is classified by its nearest neighbor in a reference set. Usually all the given instances are used as the reference set for classifying new instances [5].

The algorithm is as follows:

Step 1: For each row (case) in the target dataset (the set to be classified), locate the k closest members (the k nearest neighbors) of the training dataset.

Step 2: A Euclidean distance measure is used to calculate how close each member of the training set is to the target row that is being examined.

Step 3: Examine the k nearest neighbors – which classification (category) do most of them belong to. Assign this category to the row being examined.

Step 4: Repeat this procedure for the remaining rows (cases) in the target set

4. TEXT CLASSIFICATION

Text categorization is a key problem in the field of machine learning. The task is, given two or more

classes of 'training' documents, to find some formula that reflects statistical differences between the classes and can be used to classify new documents[6].

This section is divided into three parts: Subsection 4.1 discusses genetic representation, Subsection 4.2 discusses dictionary, Subsection 4.3 discusses Similarity metrics, Subsection 4.4 discusses fitness function, Subsection 4.5 discusses the algorithm used to implement our approach.

4.1 Genetic Representations

Basic terminologies of genetic algorithm used in our project are as follows:

- Individual - any possible solution.
- Population - group of all individuals.
- Search Space - all possible solutions to the problem.

Each individual in the population is represented in binary form, that is, it is a string of 1 and 0's, where 1 represents *presence* of a feature and 0 represents *absence* of a feature. The length of string is dependent upon the number of features in a dataset used.

Example: number of features in data set is 10 then the string generated will be 1010111010.

Feature number 1, 3, 5, 6, 7, 9 will be used to train classifier.

4.2 Dictionary

A dictionary contains set of words from all classes of the dataset. These words are further selected as features for classification. The number of words determine the length of string. Once the basic dictionary is created, it is used to evolve the best fit dictionary by using genetic algorithm.

4.3 Similarity Metrics

Similarity metrics refers to a method to determine the distance between two documents. Similarity of text can be measured using standard bags of words (BOW) or edit-distances measures. The TFIDF (Term Frequency, Inverse Document Frequency) weighing scheme is used to assign higher weights to distinguished terms in a document. TFIDF makes two assumptions about the importance of a term First, the more a term appears in the document, the more important it is (term frequency). Second, the more it appears through the entire collection of documents, the less important it is since it does not characterize the particular document well (inverse document frequency). In the TFIDF framework, the weight for term tj in a document d_i, w_{ij} is defined as follows:

$$w_{ij} = tf_{ij} * \log_2 N/n \quad (1)$$

where,

tf_{ij} is the frequency of term t_j in document d_i

N is the total number of documents in a collection, and

n is the number of documents in which term t_j occurs at least once.

4.4 Fitness Function

A fitness function is a particular type of objective function that prescribes the optimality of a solution (that is, a chromosome) in a genetic algorithm so that particular chromosome may be ranked against all the other chromosomes.

The fitness of an individual **"s"** is given by

$$\mathbf{s = w_a * a - w_r * r}$$

where,

$\mathbf{w_a}$ is the weight associated with accuracy.

$\mathbf{w_r}$ is the weight associated with reduction.

a is the accuracy.

r is the reduction.

The two objective functions "accuracy" and "reduction" mentioned in the fitness function are calculated as follows:

Accuracy = {number of correctly classified documents}/
{total number of documents}

Reduction = {number of reduced features}/
{total number of features}

4.5 Algorithm

(i) Initialize population for specified number of generations.

for *Number of generations* **do**

(ii) Perform crossover.

(iii) Perform mutation.

(iv) // Evaluation of individuals

for each *individual in the population* **do**

(v) Train the classifier.

(vi) Evaluate accuracy.

(vii) Calculate reduction.

(viii) Evaluate fitness by using the fitness function:

$$s = w_a * a - w_r * r \quad (2)$$

end

(ix) Select best individuals for the next generation.

end

5. Experimental Details

The experiment to test the proposed approach is performed on Bank Search dataset.

5.1 Description of the Bank Search Dataset

The BankSearch web page dataset was chosen as the document corpus. It is freely available online at http://www.pedal.reading.ac.uk/banksearchdataset. There are ten classes of documents, with 1000 documents per class (Table I). These documents are web pages that have been human-categorized as part of the Open Directory Project and Yahoo! Categories. This dataset supports classification tasks of varying levels of complexity. We tested our system on two groups of similar documents (C/C++ and JAVA).

TABLE I. Classes
Within the Bank Search web page dataset

Class	Specific Topic	General Topic
1	Commercial Bank	Banking and Finance
2	Building Societies	Banking and Finance
3	Insurance Agencies	Banking and Finance
4	Java	Programming Languages
5	C / C++	Programming Languages
6	Visual Basic	Programming Languages
7	Astronomy	Science
8	Biology	Science
9	Soccer	Sport
10	Motor Sport	Sport

5.2 Results

The average results obtained are displayed in the table below.

DataSet	No. of nearest neighbour	Reduction	Accuracy	Fitness
Bank Search (for two classes)	11	43	84	86.40

6. Conclusion

We have implemented a Genetic Algorithm (GA) based approach for performing classification of web documents. We have used Bank Search dataset for testing our approach.GA has enabled us to select relevant features thus increasing the accuracy of k-NN classifier and number of features has been reduced to almost fifty percent due to the reduction in the size of the dictionary. Thus, we have reduced the space and time complexity.

REFERENCES

[1] Michelle Cheatham, Mateen Rizki: Feature and Prototype Evolution for Nearest Neighbor Classification of Web Documents, Proc. of the Third International Conference on Information Technology: New Generation 2006.

[2] K.F Man, K.S Tang, S. Kwong “Genetic Algorithms”, Springer-Verlag Publication, 2001.

[3] S. Bandyopadhyay, Sankar K. Pal, “Classification and Learning Using Genetic Algorithm”, Springer-Verlag Publication, 2007.

[4] Gilbert Syswerda, ”Uniform Crossover in genetic algorithms”, Morgan Kaufmann publishers Inc. san Francisco, CA,USA, pp. 2-9.

[5] T. M. Cover and P. E. Hart, “Nearest neighbor pattern classification”, IEEE Trans. on Information Theory, 13: 21-27 , 1967.

[6] Shlomo Argamon, Kevin Burns and Shlomo Dubnov(Eds.)“The Structure of Style: Algorithmic Approaches to Understanding Manner and Meaning” Springer-Verlag Publication, 2010.