International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 - 5161

Available at http://inpressco.com/category/ijcet

Research Article

An Optimized Approach for Feature Selection using Membrane Computing to Classify Web Pages

Prabhjot Kaur[Å*] and Ravneet Kaur[Å]

[Å]Dept. of C.S.E., SGGSWU, Fatehgarh Sahib, Punjab, India

Accepted 10 Oct 2014, Available online 20 Oct 2014, **Vol.4, No.5 (Oct 2014)**

Abstract

As the information contained on the web is increasing from day to day, organizing this information could be a necessary requirement. Data mining process is to extract information from a data set and transform it into an understandable structure for further use. As each component in a Web page like HTML tags and terms is taken as a feature dimension of the classification problem becomes too high to be resolved by well-known classifiers decision trees and support vector machines etc. So we need efficient methods to select best features to reduce feature space of the Web page classification problem. In this study, a recent optimization technique namely the Membrane Computing (MC) is used, to select the best features. Membrane computing is an area within computer science, originate from natural computing. It is found that when features are selected by our membrane computing algorithm, J48 classifier is used to evaluate the fitness of selected features. WebKB datasets were classified without loss of accuracy. The experimental results of this study showed that, Membrane Computing algorithm is an acceptable optimization algorithm for Web Page feature selection.

Keywords: *Membrane Computing, P Systems, Optimization, Classification, Feature selection, Web page classification.*

1. Introduction

As the demand of the Web increases, the amount of information on the Web has been also increased. As a result of this it became difficult to manage the huge amount of online information and this caused the need for accurate and fast classification to increase the performance of search engines. [1]Vertical search engines (or focused crawlers) traverse a subset of the Web to only gather documents on a specific topic and to identify the promising links that lead to on-topic documents (Sumathi *et al*, 2006). During a focused crawling process of a vertical search engine, an automatic classification mechanism is required to determine whether the Web page being considered is "on the specific topic" or not.

Automatic Web page classification is a supervised learning problem in which a set of labeled Web documents is used for training a classifier, and then the classifier is employed to assign one or more predefined category labels to future Web pages . Automatic Web page classification is not only used for focused crawling, it is also essential to the development of Web directories, to topic-specific Web link analysis, to contextual advertising, to analysis of the topical structure of the Web, and to improve the quality of Web search (Sumathi *et al*, 2006).

Several classification methods such as decision trees, Bayesian classifier, support vector machines, k nearest neighbors have been developed (Xindong Wu *et al*, 2007) . Among these methods, decision trees, and support vector machines are suitable for classification problems in which number of features is small. Web page classification problem, on the other hand, is a high dimensional problem since each term in each HTML tag of each Web page can be taken as a feature.

In this study, we propose a membrane computing (MC) algorithm which finds the best features of Web pages, to make the classification fast and accurate. Membrane computing (MC) is a recent search and optimization technique, which was presented in computer science by Gheorghe Paun in 2000 (Alamelu Mangai J *et al*, 2010). Experimental results have shown that when the objects in a MC communicate very well in parallel mode, the time of feature selection decreases and best features are selected which have consequential effects on web page classification problem.

This paper is organized as follows: in the next section, it gives the more detail about Web page classification, and summarize related work on the MC applications. The third section describes the MC-based feature selection system. The data sets used in this study and the experimental results are presented in the fourth section. Finally, the fifth section concludes the study.

2. Related Work

Web page classification problem is defined as there is one or more predefined class labels and a classification model assigns Web pages to one or more predefined class labels. Web page classification assigns a label from a predefined set of labels to a Web page (Sumathi *et al*, 2006).. In this

*Corresponding author: **Prabhjot Kaur**

study, our aim is to determine the "role" of a Web page such as to decide whether the Web page is a "student home page", or a "course page", or a "department home page". While doing that, we give a single class label (e.g. "course page") to each Web page, and we make binary classification in which we categorize instances into exactly one of the two classes (e.g. "course page", or "not course page"). This kind of classification problem exists especially in focused crawling systems of vertical search engines. It is also possible to extend the solution technique developed in this study to other binary classification problems.

In (Liang Huang *et al*, 2006) MC a new evolutionary algorithm for solving optimization problems with large feasible solution space and large parameters is used. It is shown empirically that the optimization algorithm has good performance in solving benchmark functions as compared with some existing evolutionary algorithms. Due to its simplicity, converge fast, theoretical elegance, generality, and superiority, the optimization algorithm can be used to solve complex problems.

In (Rufai Kazeem Idowu *et al*, 2013) this paper presents a summary of part of the recent work on IDS's subset feature selection. Simulation results demonstrated that Membrane Computing paradigm is a better tool for enhancing Bee Algorithm based feature subset selection method in IDS. With the KDD-Cup datasets used in the experiments, we were able to establish that, MC has the potential of considerably increasing Classification Accuracy Rate and consequently decreasing the False Alarm Rate. Generally, our approach returned as high as 89.11% ADR. Quite remarkably however, when the results are compared to other previous approaches so far, it has the highest CAR with an average value of 95.60% and the lowest FAR of 0.004.

3. Feature Selection using Membrane Computing

Membrane computing emanated from natural computing (i.e computing which concerns itself with what is going on in nature and inspired by nature) (Ron Kohavi *et al*, 1997). Hence, membrane computing enriches the models of molecular computing by providing a spatial structure for molecular computations, inspired by the membrane structure and functioning of living cells. It is inherently and maximally a parallel computing model because communication between the multisets and objects within the regions and compartments of a membrane takes place concurrently. Usually, membranes which form hierarchical structure could be dissolved, divided, created and their permeability is modifiable. The communications between compartments and with the environment play an essential role in the processes.

Formally therefore, according to (Alamelu Mangai J *et al*, 2010) , a P system of degree n, $n \geq 1$, is a construct:
$\Pi = (O, \mu , w1,\ldots, wm , R1 , \ldots, Rm , \mathbf{i}o)$,
Where:
O is an alphabet, its elements are called objects;
μ is a membrane structure consisting of m membranes, with the membranes (and hence the regions)
injectively labelled with 1, 2, …, m; m is called the degree of Π;
wi, $1 \leq i \leq m$, are strings which represent multisets over **O** associated with the regions 1, 2, … , m of μ;
Ri , $1 \leq i \leq m$, are finite sets of rules over O; Ri is associated with the region i of μ;
io € {1, 2, …, m} is the label of an elementary membrane (the output membrane).

3.1 Membrane Structure

1. A typical membrane structure which could best be captured as in the Figure.1: has the following parts:
2. The skin (which is otherwise called 'container' membrane) is the external compartment that houses other membranes. A membrane is a discrete unit which can contain a set of objects (symbols/catalysts), a set of rules.
3. The environment (i.e area where P system is placed). While the environment can never hold rules, it may have objects passed into it during the computation. The objects found within the environment at the end of the computation constitute all or part of its "result."
4. Elementary membrane which is otherwise called an empty membrane.
5. A region is either a space delimited by an elementary/non-elementary membrane and all of its lower neighbors.
6. Other membranes which are non-empty and contain other compartments, rules and objects.

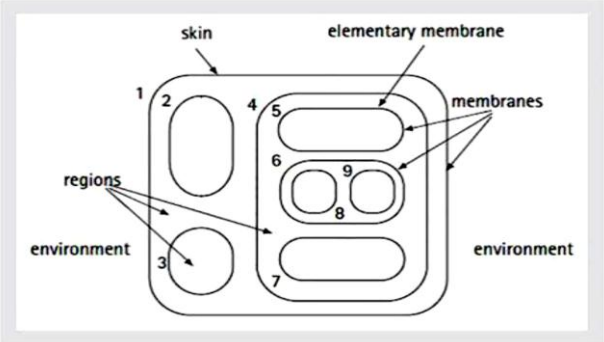

Fig.1 Membrane Structure

The MC is different from the previous studies in the following ways: (i) the number of features considered in MC is large, whereas the others consider only a few features, (ii) our system takes both terms and HTML tags together on a Web page as features, in other words each term in each HTML tag of the Web page is deemed as a different feature, however the other systems take only terms or only HTML tags, not both as features, (iii) we assign different weights to each feature and the weights are determined by the MC, in the other systems, they give either no weights to features (only consider inclusion or exclusion of a feature) or assign a constant weight to each HTML tag either manually or through some computation mechanism, (iv) our MC can be used for both classification and feature selection, the other systems on the other hand is designed for either classification or feature selection. In our previous study , we used our MC as a classifier; however in this study we use it as a feature

selector. Our classification system in this study consists of three main components: (i) feature extraction, (ii) MC based feature selector, and (iii) classification. In the following subsections we describe each component in more detail.

3.2 Feature Extraction

Feature extraction is a special form of dimensionality reduction. To extract features for web page classification problem, For each dataset, the features were extracted by taking stemmed terms that are not stopwords from the <title> tags and URLs of the positive (i.e., relevant) Web pages in the training set. Feature extraction is performed only once for all datasets as a pre-processing step. After extracting features, document vectors for the Web pages are created by counting the occurrences of each feature in each Web page. Then, document vectors are normalized. After that MC used for feature selection

3.3 MC-Based Feature Selector

With inspiration from biology, MC uses objects as transporting mechanisms through membranes. So, because a typical membrane structure consists of both internal and skin membranes, the membrane algorithm is also made up of subalgorithms which interact based on its communication rule. Membrane communications only occur in parallel between adjacent regions. So, during implementation, MC algorithm is designed to have two phases. The first phase deals with activities in the subalgorithms where initial solutions are generated. The second phase captures the proceedings within the skin membrane (otherwise called output membrane) which garners the initial solutions as its input to generate the final solution.

3.4 Classification

The MC described above returns the best features which is most similar to positive documents and least similar to negative documents in the training dataset. The returned feature consists of features and their weights. In the classification part of our system, we choose the weighted n features which are the most important terms for the training dataset, and we use these terms for the classification of the unseen data. For classification, J48 classifier (Esra Sarac *et al*, 2013) data mining tool is employed.

The steps of algorithm are described in Figure.2:

Step 1:	Distribute features as inner and outer layer of the membrane.
Step 2:	Pass or compute inner features of web as lipids.
Step 3:	Compute, if weight =< weight of any web data, then L = threshold
Step 4:	Outer layer passes the weight according to threshold value.
Step 5:	Compute features of higher accuracy and repeat step 2. to step 3. until all features complete with process.
Step 6:	Find and show the segment at layer which have higher weight.
Step 7:	Find and select the features of higher weight and pass it to outer layer.
Step 8:	Pass selected features to classifier.

Fig.2 MC Feature Selection Algorithm

4. Experimental Evaluation and Results

All the implementations for the experiments were made in MATLAB programming language. The proposed method was tested under Microsoft Windows 7 Ultimate operating system. The hardware used in the experiments had 1 GB of RAM and Intel Core2Duo 1.60 GHz processor.

4.1 Parameters

For experimentation various parameters are used which confirms the results. In terms of Precision, Recall, F-measure and accuracy it compares the results with previous work which clears that proposed approach produces the more refined results as compared to previous approach.

1. **F-measure**: A measure that combines precision and recall is the harmonic mean of precision and recall, the traditional F-measure or balanced F-score:

$$\text{F-measure} = \frac{(2 \times \text{Recall} \times \text{Precision})}{(\text{Recall} + \text{Precision})}$$

2. **Precision:** In simple language precision is the fraction of retrieved documents that are relevant to the search, for a text search on a set of documents precision is the number of correct results divided by the number of all returned results.

$$\text{Precision} = \frac{\text{Number of relevant documents retrieved}}{\text{Total number of documents retrieved}}$$

3. **Recall:** Recall is the fraction of the documents that are relevant to the query that are successfully retrieved, for text search on a set of documents recall is the number of correct results divided by the number of results that should have been returned.

$$\text{Recall} = \frac{\text{Number of relevant documents retrieved}}{\text{Total number of relevant documents}}$$

4. **Accuracy:** The accuracy is the proportion of true results (both true positives and true negatives) in the population. It means how close the result to standard value.

$$\text{Accuracy} = \frac{TP+TN}{TP + TN + FP + FN}$$

4.2 Datasets

The WebKB dataset is a well-known dataset that is obtained from the WebKB project (Esra Sarac *et al*, 2013). The WebKB dataset contains course, department, faculty, project, staff, and student Web pages gathered from the Computer Science departments of the Cornell, Texas, Washington, and Wisconsin universities as well as some irrelevant pages from those four universities. We used course, faculty, project and student classes in our

experiments since these classes have more instances than others. The dataset contains 7648 Web pages in total such that it has 883 course, 1028 faculty, 493 project, and 1480 student homepages, and 3764 negative (belongs to other classes) Web pages. The train and the test datasets were constructed as described in the WebKB project Web site (Esra Sarac *et al*, 2013). For this study we used pages from Cornell, Texas, and Washington universities in the training, and pages from Wisconsin university in the test phase. We used the WebKB dataset as a binary class classification dataset. For example the Course dataset contains 883 course and 3764 negative pages, the Faculty dataset has 1028 faculty and 3764 negative pages, and so on.

4.3 Using MC as a Feature Selector

In this it selected the top weighted n features by the MC as feature list, after that J48 (decision tree), classifiers of the Weka tool is applied (Esra Sarac *et al*, 2013). As the total numbers of features are too high it is not possible to directly apply the classifiers of Weka to the original dataset. The accuracy of the classification process with respect to selected number of features for each class is presented in Figures 1-5. As it can be seen from the figures, when the number of features used in the classification process decreases the accuracy of the classification increases. The best classification accuracy for the WebKB dataset belongs to Student class which is 95% when the J48 is employed. As the number of features used in the classification decreases, the running time of the algorithms also decreasesWhen 10 features are used, the running time becomes too small to be measured.

4.4 Results

In this study, it chooses a predefined number of features. Performance of the proposed MC based method with respect to precision, recall and F-measure can be seen below.

Performance on the basis of Precision

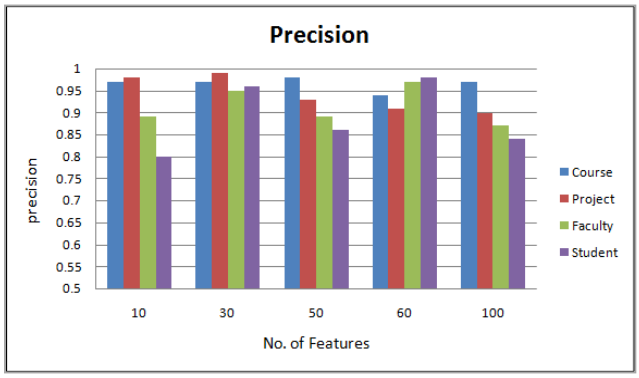

Fig.3 Representation of Precision

The figure 3 and table 1 shows the performance on the basis of precision of the proposed Membrane Computing based Feature Selection algorithm for # number of features from Course, Project, Faculty, Student Classes.

Table 1 Precision of Membrane Computing

No. of Features	Course	Project	Faculty	Student
10	0.97	0.98	0.89	0.80
30	0.97	0.99	0.95	0.96
50	0.98	0.93	0.89	0.86
60	0.94	0.91	0.97	0.98
100	0.97	0.90	0.87	0.84

Performance on the basis of Recall

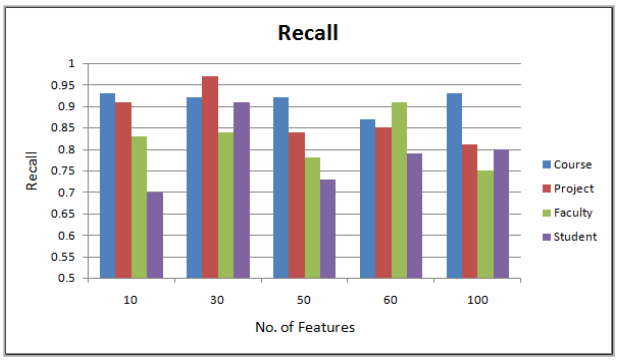

Fig.4 Representation of Precision

Table 2 Recall of Membrane Computing

No. of Features	Course	Project	Faculty	Student
10	0.93	0.91	0.83	0.70
30	0.92	0.97	0.84	0.91
50	0.92	0.84	0.78	0.73
60	0.87	0.85	0.91	0.79
100	0.93	0.81	0.75	0.80

The above figure 4 and table 2 shows the performance on the basis of recall of the proposed Membrane Computing based Feature Selection algorithm for # number of features from Course, Project, Faculty, Student Classes.

Performance on the basis of F - measure

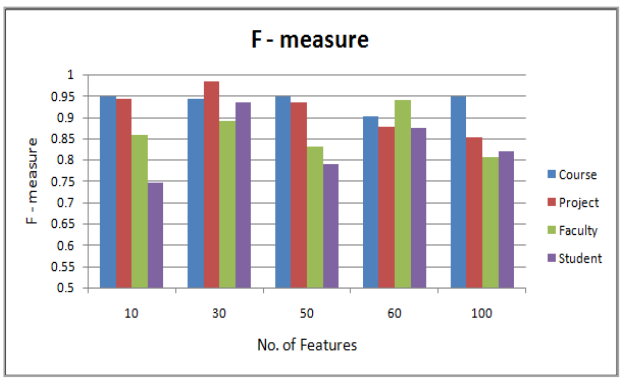

Fig.5 Representation of F – measure

The figure 5 and table 3 shows the performance on the basis of F- measure of the proposed Membrane Computing based Feature Selection algorithm for # number of features from Course, Project, Faculty, Student Classes.

Table 3 F-measure of Membrane Computing

No. of Features	Course	Project	Faculty	Student
10	0.949	0.943	0.858	0.746
30	0.944	0.984	0.891	0.934
50	0.949	0.935	0.831	0.789
60	0.903	0.878	0.939	0.874
100	0.949	0.852	0.805	0.819

Table 4 Comparison Table of MC, FA, ACO

Optimization	**Membrane Computing**			**FireFly Algorithm**			**Ant Colony Optimization**		
Classes	Precision	Recall	F measure	Precision	Recall	F measure	Precision	Recall	F measure
Course	0.97	0.92	0.944	0.95	0.92	0.934	0.92	0.84	0.880
Project	0.99	0.94	0.984	0.99	0.97	0.981	0.99	0.97	0.983
Faculty	0.95	0.87	0.891	0.80	0.72	0.757	0.95	0.89	0.917
Students	0.96	0.91	0.934	0.86	0.79	0.824	0.96	0.89	0.927

Comparison Analysis among MC, FA, ACO

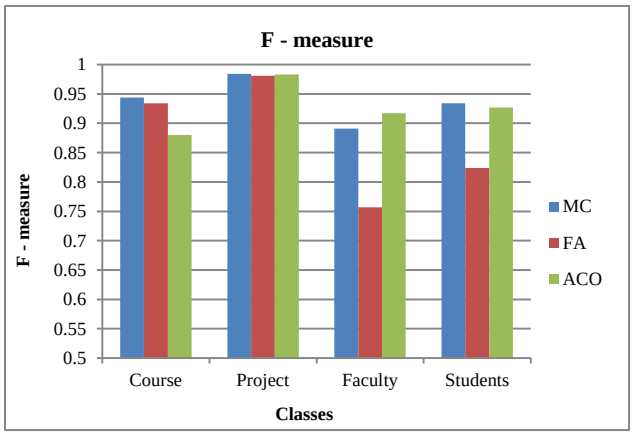

Fig.6 F – measure of MC, FA, ACO

The above Figure shows the comparison of f – measure for 30 number of features with previous results. Table 4 shows the comparison of the proposed Membrane Computing algorithm for WEBKB dataset with Firefly Algorithm and Ant Colony Optimization (30 Features).

5. Results based on Accuracy

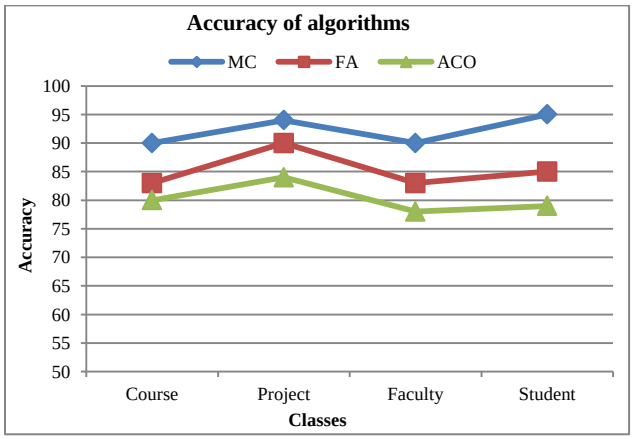

Fig.7 Accuracy of MC, FA, ACO

Table 5 Comparison Table for Accuracy

Dataset	**MC**	**FA**	**ACO**
Course	90	83	80
Project	94	90	84
Faculty	90	83	78
Student	95	85	79

The above Figure 7 depicts the overall accuracy of the algorithms. The accuracy of FA and ACO or Course class is 83 and 80 respectively for the data set. But the results get improved when applied a proposed algorithm Membrane Computing. Now, the accuracy becomes 90 which show that approximately 13% improvement came into the results.

Conclusion

In this study, Membrane Computing algorithm is used which takes each term in each HTML tag as a different features. Membrane Computing tries to find the best features that are important for the classification process by assigning higher weights to features. It is observed that by using selected features, Web pages can be classified faster and 95% accuracy is attained. Even more in some cases, the f-measure of the classification is improved by making feature selection; since it allows removing unnecessary features that reduces the classification accuracy, In future, cross validation of the experiments can be performed, and experiments can be repeated for other datasets and for other HTML tag sets and URLs. Another artificial intelligence techniques that can be used for optimizing features.

References

Sumathi, S.N. Sivanandam (2006), Introduction to Data Mining and its Applications, Studies in Computational Intelligence, Volume 29, Springer.

Xindong Wu, Vipin Kumar • J. Ross Quinlan (2007), Top 10 algorithms in data mining , December 2007 © Springer-Verlag London Limited , pp. 1-37.

B. Choi and Z. Yao (2005), Web Page Classification © Springer-Verlag Berlin Heidelberg , pp. 221-274.

Rufai Kazeem Idowu, Ali Maroosi (2013), An Application of Membrane Computing to Anomaly-Based Intrusion Detection System , The 4th International Conference on Electrical Engineering and Informatics (ICEEI), pp. 523-529 .

Rudolf Freund Gheorghe Paun (2005), Membrane Computing , 6th International Workshop, WMC 2005 Vienna, Austria, July 18-21.

Liang Huang and Ning Wang (2006), An Optimization Algorithm Inspired by Membrane Computing , ICNC 2006, Part II, LNCS 4222, pp. 49–52.

http://en.wikipedia.org/wiki/Artificial_intelligence.

Alamelu Mangai J & Santhosh Kumar V (2010), Recent Research in Web Page Classification-A Review, International Journal of Computer Engineering and Technology(IJCET),Volume 1, Number .

Ron Kohavi , George H. John (1997), Wrappers for feature subset selection , 0004-3702/97/@ Elsevier Science B.V, pp. 273-324.

M.F. Porter, An algorithm for suffix stripping, Program,14(3), pp 130−137.

Monika Yadav, Mr. Pradeep Mittal (2013), Web Mining: An Introduction , International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 3, pp. 683-687.

Abdelhakim Herrouz, Chabane Khentout, Mahieddine Djoudi (2013), Overview of Web Content Mining Tools, The International Journal of Engineering And Science (IJES), Volume 2, Issue 6.

Esra Sarac, Selma Ayse Ozel (2013), Web Page Classification Using Firefly Optimization, IEEE International Symposium on Innovations in Intelligent Systems and Applications (INISTA), 19-21, pp 1-5.

Lim Wern Han and Saadat M. Alhashmi (2010), Joint Web-Feature (JFEAT): A Novel Web Page Classification Framework, IBIMA Publishing, Vol. 2010 , Article ID 73408, 8 pages.

Pikakshi Manchanda, Sonali Gupta, Komal Kumar Bhatia (2012), On The Automated Classification of Web Pages Using Artificial Neural Network, IOSR Journal of Computer Engineering (IOSRJCE), Volume 4, Issue 1, pp 20-25.

Ajay S. Patil, B.V. Pawar (2012), Automated Classification of Web Sites using Naive Bayesian Algorithm, proceedings of the International Multi Conference of Engineers , Vol. 1, p466.

Aixin Sun, Ee-Peng Lim, Wee-Keong Ng (2002), Web Classification Using Support Vector Machine WIDM'02, November 4.9, McLean, Virginia, USA.

Xiaoguang Qi, Brian D. Davison (2009), Web Page Classification: Features and Algorithms, ACM Computing Surveys, Vol. 41, No. 2, Article 12.

Basem O. Alijla, Lim Chee Peng, Ahamad Tajudin Khader and Mohammed AzmiAl- Betar (2013), Intelligent Water Drops Algorithm for Rough Set Feature Selection, Springer-Verlag Berlin Heidelberg, pp. 356–365.

Selma Ayse Ozel, A Genetic Algorithm Based Optimal Feature Selection for Web Page Classification , Department of Computer Engineering, Cukurova University,01330 Balcal, Sarcam, Adana, Turkiye saozel@cu.edu.tr.