Knowledge-Based Systems 166 (2019) 198–206

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

An effective feature selection method for web spam detection

Faeze Asdaghi [a,*], Ali Soleimani [b]

[a] Faculty of Computer Engineering, Shahrood University of Technology, Shahrood, Iran
[b] Faculty of Electrical Engineering and Robotic, Shahrood University of Technology, Shahrood, Iran

ARTICLE INFO

Article history:
Received 18 May 2018
Received in revised form 20 December 2018
Accepted 21 December 2018
Available online 31 December 2018

Keywords:
Web spam
Feature selection
Content-based features
Link-based features
Unbalanced data
Index of balanced accuracy (IBA)

ABSTRACT

Web spam is an illegal and immoral way to increase the ranking of web pages by deceiving search engine algorithms. Therefore, different methods have been proposed to detect and improve the quality of results. Since a web page can be viewed from two aspects of the content and the link, the number of extracting features is high. Thus, selection of features with high separating ability can be considered as a preprocessing step in order to decrease computational time and cost. In this study, a new backward elimination approach is proposed for feature selection. The main idea of this method is measuring the impact of eliminating a set of features on the performance of a classifier instead of a single feature which is similar to the sequential backward selection. This method seeks for the largest feature subset that their omission from whole set features not only reduces the efficiency of the classifier but also improves it. Implementations on WEBSPAM-UK2007 dataset with Naïve Bayes classifier show that the proposed method selects fewer features in comparison with other methods and improves the performance of the classifier in the IBA index about 7%.

1. Introduction

Nowadays, with the growth of information on the web, search engines are considered as a key tool to enter the websites. Research shows that roughly 60% of users visit only five initial results of the first page [1]. As a result, a page presence in the top results of search engines means more visitors and more revenue. Web spam is an illegal and unethical way to increase the ranking of web pages by deceiving search engine algorithms. Web spam reduces the quality of search results and, as a result, they waste the time of users. Thus, many papers have been published to detect web spam [2] .

There are two main approaches for detecting web spam. The first approach is based on the web graph in which web pages are the nodes and links between these pages are edges. Considering this graph and estimating the amount of trust in a page in terms of the validity of the pages, web spam can be identified. For example, in [3], a new method based on web page differentiation (DPR) is proposed in order to improve classic PageRank algorithm's disadvantage of assigning link weights evenly and ignoring the authority of the web pages. Also in [4], an asynchronous anti-trust algorithm is developed to significantly reduce the number of arithmetic operations compared to the traditional synchronous Anti-TrustRank algorithm without degrading the performance in detecting Web spam.

The second approach is based on the content and components of a web page, which is a matter of classifying and supervised learning [5–7]. These articles are divided into two broad categories according to their aspects. The first category is a set of methods that emphasize the features extracted from web pages and the goal is finding powerful features which are able to discriminate between spam and normal pages in order to increase the detection rate of web spam. For example, in [8], some features due to the entropy of outliers are introduced. In [9–12] features based on language model and in [13,14] features based on qualified links of a web page are proposed. Also in [15–17] topic modeling and in [18,19], Lexical Items are applied to extract the features.

The second category consists of some methods that introduce classification algorithms with better performance in which data is distributed unevenly and small data is available in spam class. Recently, in [20] a systematic framework based on the CHAID algorithm and in [21] a fuzzy logic based framework has been proposed to detect web spam pages. Moreover, in [22,23] a framework is presented to detect web spam using incremental learning. In [24] a classification method based on Minimum Description Length Principle is introduced. Artificial neural network, deep belief network and dual margin SVM are the other algorithms for spam detection [13,25,26].

Despite a variety of web spam detection methods, there are many challenges in the field of web spam detection. For example, web spam classification problem is usually faced with the problem of the high dimensionality of the vector space and the lack of training samples. Therefore, in real-world applications, these methods

* Corresponding author.
E-mail addresses: asdaghi@shahroodut.ac.ir (F. Asdaghi), solimani_ali@shahroodut.ac.ir (A. Soleimani).

https://doi.org/10.1016/j.knosys.2018.12.026
0950-7051/

commonly encounter with problems such as large amounts of data, unbalanced classes, high computational costs, and memory consumption. In this study, we intend to consider the number of features and their impact on the accuracy of classifiers. Also, we try to improve the detection rate of the classifier by dimension reduction by applying a new feature selection method.

Contribution of this study: The proposed method, which is a backward elimination method, tends to choose the smallest subset of attributes that have the greatest impact on classifier performance by considering the role of each attribute alone and along with other features in order to increase the classifier's efficiency with the following contributions:

- Introducing a classifier performance measure which is suitable for unbalanced data and comparing it with other performance measures
- Comparing common feature selection methods with the typical performance measure and choosing the best one as a preprocessor of the proposed algorithm
- Presenting a new feature selection method which is suitable for nighing local optimal to the global optimal feature set
- Customizing the method for the unbalanced dataset, especially for web spam detection

Organization of this study: The rest of this article is organized as follows. Section 2 reviews the related literature. Section 3 presents the proposed method. Section 4 represents the implementation process and the obtained results. Finally, Section 5 provides the concluding remarks and future research.

2. Preliminaries

In this section, we discuss the problem of feature selection and its classical methods in order to compare between them and the proposed algorithm. Also, since the dataset is unbalanced, the performance measures in these issues will be examined.

2.1. Feature selection

Feature selection is one of the important steps in pattern recognition, machine learning, and data mining. Its purpose is eliminating irrelevant and redundant variables in order to understand data, reducing computation requirement, decreasing the effect of the curse of dimensionality and improving the performance of the predictor. The focus of feature selection is to select an optimal subset of the features from input data which still provide good prediction results or in the other words to optimize the value of an evaluation function. Since the optimal solution is obtained by creating all possible subsets and evaluating them, finding the optimal solution is difficult and very costly for a large number of features. Therefore, many methods such as Exhaustive, Best First, Genetic Algorithm, Greedy, and Forward Selection have proposed that use explicit or random search instead of exhausted search to increase computational time against the decline of performance [27,28].

There are three major approaches to select a subset of features that are referred to as Wrapper, Filter, and Embedded. Wrapper methods such as Sequential Forward Selection, Sequential Backward Selection, Bidirectional Search, and Relevance in Context use a predictive model to score a subset of features, which can be computationally complicated but often choose the best subset. Filter methods use an approximate scale to score a subset of features instead of using the error rate to reduce computational load. The most important criteria used in this category are Mutual Information, Correlation, Consistency, Gain Ratio, Information Gain, Symmetrical uncertainty, and Chi-Squared. Embedded methods are a group of techniques that feature selection is a part of the process of making the model. In these methods, searching for an optimal feature set is within the structure of the classifier and usually, it is difficult to control the proper number of features. But, the advantage of these methods is the low computational cost compared to filter methods. Decision trees and Grafting are among the most important algorithms in this field [25].

2.2. Unbalanced data

One of the challenges of classification is "unbalanced datasets". This is especially true in bi-classes applications that a class has more features compared to the other class while more important features are in the minority class. Abnormalities detection problems and web spam are subcategories of this issue. Dealing with unbalanced data is always known as a challenging issue in data mining because in most classification algorithms, the tendency towards a class that has the largest number of samples. Thus, they show less ability to quantitatively predict the accuracy of the minority class. Therefore, selecting a suitable benchmark for the proper evaluation of their performance is highly necessary. Often, in order to evaluate binary classification problems, measures in Table 1 are used.

Where *TP* is the number of positive samples categorized correctly by the algorithm, *TN* is the number of negative samples categorized by the algorithm, *FP* is the number of positive samples that are not correctly categorized by the algorithm, and *FN* is the number of negative samples that are not correctly categorized by the algorithm.

In addition to these two criteria, Receiver Operating Characteristic (ROC) is a common criterion for evaluating the classification accuracy in which the range of changes in this criterion is between zero and one. If changes are closer to one, then the accuracy is better.

All of these criteria are a combination of error rates and accuracy, which are individually measured for each class. This leads to a reduction in the bias of the classification performance. Nonetheless, the point not being taken into account is the role of a class's dominance over other classes. For this reason, the results do not reflect the role of each class in overall performance. While, in some cases, knowing whether the precision of the classes is balanced or which class is predominant is necessary. Thus, in order to better evaluate these features, Index of Balanced Accuracy (IBA) is used [29].

The main purpose of the IBA is to weight a measure suitable for those results with better classification rates in the minority classes. For this purpose, a concept called Dominance Factor is used. This criterion refers to the relationship between classes in terms of the degree of dominance and it is a number in the range [-1, +1]. IBA calculation is as follows:

$$IBA_\alpha = (1 + \alpha \cdot Dominance)M \tag{1}$$

$$Dominance = TPR_. - TNR_. \tag{2}$$

$$M = TPR \times (1 - FPR) \tag{3}$$

where M is the under the curve of a two-dimensional graph in which one axis is the geometric mean square of the precision of the classes and the other axis is the marked difference between the precision of the classes. The factor α is also used to weight the dominance criterion. It is shown in [30] that 0.1 is a suitable value for α factor. This criterion takes into account the results that have a relatively better classification rate in the minority class.

3. The proposed algorithm

In this section, the inspiration and mathematical modeling of the proposed algorithm are described in detail.

Table 1
Measures for binary classification.

	Formula	Evaluation Focus
Accuracy	$\frac{TP+TN}{(TP+TN+FP+FN)}$	Overall effectiveness of a classifier
Precision	$\frac{TP}{(TP+FP)}$	Class agreement of the data labels with the positive labels given by the classifier
Recall	$\frac{TP}{(TP+FN)}$	Effectiveness of a classifier to identify positive labels
F-score	$\frac{2TP}{(2TP+FP+FN)}$	Relations between data's positive labels and those given by a classifier
Specificity	$\frac{TN}{(TN+FP)}$	How effectively a classifier identifies negative labels
AUC	$\frac{1}{2}\left(\frac{TP}{TP+FN}+\frac{TN}{TN+FP}\right)$	Classifier's ability to avoid false classification

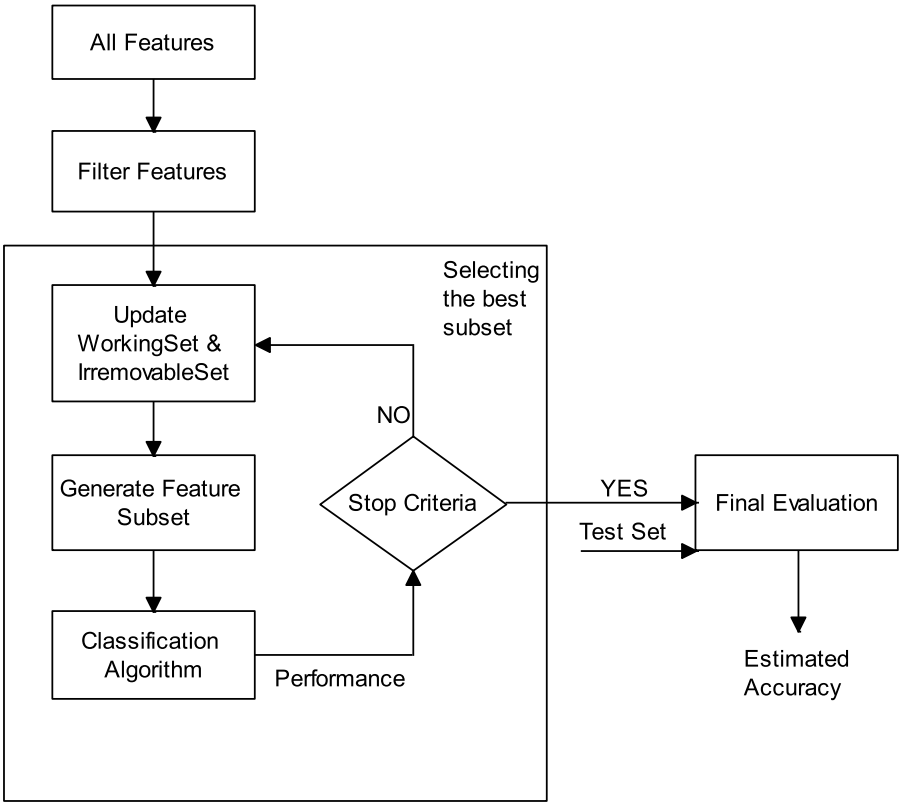

Fig. 1. Diagram of the Smart-BT algorithm.

3.1. Inspiration

As mentioned in the previous section, the sequential backward selection is one of wrapper suboptimal feature selection techniques. This method by starting from the set of all features sequentially removes the feature x^- that least reduces the value of the objective function $(Y - x^-)$. It will be continued until $(Y_k - x^-)$ is not less than (Y_k). Although SBS has a high accuracy rate over using the classifier error rate as the evaluation method, it has a main limitation due to the irreversible decision of removing features. It causes the inability of reevaluating the usefulness of a feature and therefore occurring nesting problem that means the best subset of size k need not contain the best subsets of size $k-1, k-2, \ldots, 1$.

To overcome this problem, we introduced a new backward elimination feature subset selection that is called smart-BT. The main idea of this method is measuring the effect of eliminating a set of features on the performance of a classifier instead of a single feature like what is done in SBS. In order to prevent imposing an extra computational cost in each round of evaluation, feature sets are divided into two sets of *Irremovable* and *LowInfo* sets. The *Irremovable* set consists of feature combinations that never should be removed and *LowInfo* set contains feature sets that eliminating them will improve classifier performance. The goal of this method is finding the largest set of features so that the omission of its member can cause maximizing classifier performance. Diagram of this algorithm is shown in Fig. 1.

3.2. Naïve Bayes classifier

As a wrapper feature selection method, classification is an important part of the Smart-BT and it should have general and specific properties. Low time complexity is the general property because it is used as an evaluator. The specific properties refer to the type of dataset. In this problem because of Non-Gaussian distribution of some features, the chosen classifier must be non-sensitive to the distribution of variables. In addition, unbalancing dataset and the importance of small class, makes us select a specific classifier like Naïve Bayes classifier.

Naive Bayes classifier has been built upon the famous Bayes' theorem with the (not so) "naive" assumption of independence between each pair of features. The theorem says that if $P(C_i|E)$ is the probability that example E is of class C_i, misclassification rate is minimized if and only if E is assigned to the class C_k for which

Table 2
Comparison of different classification algorithms performance.

	Accuracy	Precision	Recall	F-score	Specificity	AUC	CPU Time (ms)
Naïve Bayes	69.68%	0.115	0.636	0.195	0.64	0.67	390+780
SVM	94.26%	0.667	0.019	0.036	0.02	0.51	57650+6410
Random Tree	92.75%	0.333	0.252	0.287	0.25	0.61	80+300
Decision Tree	91.61%	0.250	0.224	0.236	0.22	0.59	2780+230
Random Forest	94.86%	0.833	0.140	0.240	0.14	0.57	4760+320
KNN	91.72%	0.189	0.131	0.155	0.13	0.55	20+18040
MLP	94.31%	1.000	0.019	0.037	0.02	0.51	194080+1350

$P(C_k|E)$ is maximum (Eq. (4)).

$$P(C_i|E) = \frac{P(C_i)\,P(E|C_i)}{P(E)} \tag{4}$$

In the above Equation, $P(E)$ can be ignored, since it is the same for all classes, and does not affect the relative values of their probabilities. If the attributes are independent given the class, $P(E|C_i)$ can be decomposed into the product of probabilities leading to $P(C_i|E) = f_i(E)$ while

$$f_i(E) = P(C_i)\prod_{j=1}^{\alpha} P(A_j = v_{jk}|C_i) \tag{5}$$

In the other words, using $P(C_i|E)$ as the discriminant functions $f_i(E)$ is the optimal classification procedure. Although in practice, attributes are seldom independent and that is why this assumption is "naive". However, Domingos and Pazzani in [31] have shown this method can be optimal even when the assumption of attribute independence does not hold and therefore $P(C_i|E) \neq f_i(E)$.

The Naïve Bayes method has very low time complexity (linear in the size of the training and test set) and low storage requirements. Moreover, its assumption usually works quite well in some real-world situations such as spam filtering and document classification. As a consequence of the decoupling of the conditional probability distributions of different features, the probability distribution of each feature can be independently estimated as one-dimensional distribution, which in turn helps alleviate problems stemming from the curse of dimensionality. It is more robust to irrelevant features than some more complex learning methods and when we have many equally important features is better than methods like decision trees. For better comparison, different classification method performance is shown in Table 2.

These algorithms are run on the WEBSPAM-UK2007 dataset, which is described in detail in the next section and consists of 275 features. The training set contains 3637 positive and 208 negative samples while the test set contains 107 negative and 1740 positive instances. As it is seen, Naïve Bayes classifier after the random tree has the best time complexity (1170 ms) and significant performance in detecting instances of the small class (64% of negative samples).

3.3. Index of balanced accuracy

When data is unbalanced and the goal is detecting samples of the small class, none of the common performance measures, including accuracy, precision, recall, F-score, specificity, and AUC show a good perspective of classifier performance. Hence, as mentioned in the previous section, we use the Index of Balanced Accuracy as a performance measure. IBA has α parameter that authors in [30] are shown that 0.1 is an appropriate amount of tuning it. For more certainty, the performance of the above classifiers is measured using a different amount of α parameter. The results are shown in Table 3.

Comparing performance measure of Naïve Bayes and random tree shows $\alpha = 1$ is not a good choice because IBA_1does not show properly differential ability to detect small class (specificity measure). Also, according to accuracy and specificity of random forest and KNN, although random forest seems to have a better efficiency than KNN, when $\alpha = 0.1$ it is not shown appropriately. Hence, in this case, it seems that $\alpha = 0.5$ is a reasonable amount for calculating IBA.

Table 3
Comparing the different amount of α parameter in IBA.

	Accuracy	Specificity	AUC	IBA_1	$IBA_{0.5}$	$IBA_{0.1}$
Naïve Bayes	69.68%	0.64	0.67	0.47	0.46	0.45
SVM	94.26%	0.02	0.51	0.04	0.03	0.02
Random Tree	92.75%	0.25	0.61	0.42	0.33	0.26
Decision Tree	91.61%	0.22	0.59	0.37	0.29	0.23
Random Forest	94.86%	0.14	0.57	0.26	0.20	0.15
KNN	91.72%	0.13	0.55	0.23	0.18	0.14
MLP	94.31%	0.02	0.51	0.04	0.03	0.02

3.4. Chi-square

Since backward methods are suitable when the optimal feature subset is large, then a preprocess step is added to the proposed method to decrease features. In order to prevent time wasting, a filter method is used. Some common metrics for evaluating the usefulness of features are Chi-squared, Information Gain and Symmetrical uncertainty. Information Gain is symmetrical measure, which says that the information gained about Y after observing X is equal to the information gained about X after observing Y. (Eq. (6))

$$IG = H(Y) - H(Y|X) = H(X) - H(X|Y) \tag{6}$$

A weakness of the IG criterion is that it is biased in favor of features with more values even when they are not more informative. The symmetrical uncertainty criterion compensates for the inherent bias of IG by dividing it by the sum of the entropies of X and Y. It is given by

$$SU = 2\frac{IG}{H(Y) + H(X)} \tag{7}$$

A value of $SU = 1$ means that the knowledge of one feature completely predicts, and the other $SU = 0$ indicates, that X and Y are uncorrelated. In opposition to the IG, the SU favors variables with fewer values. Chi-squared attribute evaluation evaluates worthy of a feature by computing the value of the chi-squared statistic (X^2) with respect to the class. The initial hypothesis H_0 is the assumption that the two features are unrelated, and it is tested by the chi-squared formula:

$$X^2 = \sum_{i=1}^{r}\sum_{j=1}^{c}\frac{(O_{ij} - E_{ij})^2}{E_{ij}} \tag{8}$$

where O_{ij} is the observed frequency and egg is the expected (theoretical) frequency asserted by the null hypothesis. The greater the value of X^2, the greater evidence to reject the hypothesis H_0. As the features of extracted from web pages do not have a normal distribution, then it seems that the Chi-square test can be the best evaluator.

3.5. Steps of proposed algorithm

As mentioned in the previous chapter, feature selection methods consist of two main components of the search strategy and evaluator function. In the proposed method, the searching strategy is backward elimination and the amount of IBA criterion obtained via Naïve Bayes classifier is used as the evaluator. The reason for using IBA instead of accuracy is unbalancing of the dataset. The pseudo-code of this algorithm is shown in Fig. 2.

In this algorithm, we suppose that most of the current feature set is near to optimal set and separation ability of a few features is low. Therefore the goal is to find the largest subset of the features so that by removing its members from the dataset, the greatest improvement in the classification accuracy emerges. Steps of the algorithm are as follows:

Step 1. Start with the full set of $X = x_1, \ldots, x_n$ where n is the number of features.

Explanation: Create X set so that its members are the name of dataset features.

Step 2. Classify dataset using all features and calculate the index of balanced accuracy (IBA_n).

Explanation: It is not important which classification method is used because the target is measuring the impact of each feature on classification accuracy. In this step, the performance of dataset classification is calculated by using all of the extracted features and it is used as a base state in order to be used in comparison in the next steps.

Step 3. $k = 1$

Step 4. Create all k-element subsets of X and put it in *working* set.

Step 5. Remove $w_i \in Workingset \mid \exists\, Ir_i \in Irremovable \,\&\, Ir_i \subset w_i$

Explanation: *Irremovable* is a set that contains features so that removing them decreases the performance of classification. *Workingset* is the set that contains sets of candidate features to be removed from all features set (X) and its impact on classification performance is investigated. In this step, before checking the elimination of *Workingset* member's effect, if any of features in *Irremovable* set exists in the candidate sets, it drops. This step helps to reduce computational cost. To clarify, suppose that primary dataset is covering 5 features. Then, we have $X = 1, 2, 3, 4, 5$. Now, if $k = 2$, then $Workingset = \{1, 2\}, \{1, 3\}, \{1, 4\}, \{1, 5\}, \{2, 3\}, \{2, 4\}, \{2, 5\}, \{3, 4\}, \{3, 5\}, \{4, 5\}$.

On the other hand, if $Irremovable = 2, 5$ then *Workingset* changes to $Workingset = \{1, 3\}, \{1, 4\}, \{3, 4\}$ and just 3 candidates test will be instead of 10 primitive candidates.

Step 6. Calculate the value of w_i by taking out its member from X and computing the amount of IBA for the remaining features. $V(w_i) = IBA_n - IBA_{n-w_i}$

Explanation: In this step, the value of each *Workingset* remaining members is calculated. The value of a *Workingset* member is the amount of its elimination effect on classifier performance. To calculate this item, the dataset is classified using features that are not in intended member. Then, the obtained results are compared with the base state in order to determine the positive or negative impact of this omission on classification performance. For example, according to the previous step example, we need to calculate each of the 3 members of *Workingset*. The value of $\{1, 3\}$ is measured by classifying dataset using $X - \{1, 3\} = 2, 4, 5$ features. The classifier's IBA in this state is 0.65 and IBA of the base state is 0.53. Thus, the value of $\{1, 3\} = -0.12$.

Step 7. If $(w_i) > 0$, it means information involved with members of w_i is important for the classifier. So, it transfers to *Irremovable* set. Else, w_i is a candidate for dropping out from X and transfers to *LowInfo* set.

Step 8. Store the best result and corresponding w_i subset.

Step 9. $k \leftarrow k + 1$

Step 10. Create all k-element subsets of *LowInfo* set and put it in *working* set.

Step 11. Go to step 4

This algorithm continues until the elimination of the members of this set improves the base IBA. As it is seen, the result of the proposed algorithm is very near to the global optimal solution and it is appropriate for issues like real-time problems in which data dimension is very important. However, because of considering all feature subsets, it would be time-consuming when the primitive feature set is large. For this reason, in a large-scale problem, it is a good idea to do a preprocessing for dimension reduction. According to the problem and the nature of the dataset, a quick and efficient method like Ranker and Chi-Square is a good choice.

3.6. Computational complexity

In order to calculate the time complexity of the proposed algorithm, we first compute it for each function. There are four functions in the proposed method as follows:

- *Evaluate()*
  This function classifies dataset using input features and calculates the index of balanced accuracy. Rebuilding the dataset requires $O(n \times k)$ times where n is the number of data instance and k is the number of input features. Naïve Bayes classifier also requires $O(n \times k)$ time. Then the overall time complexity of this function is $O(n \times k)$.
- *CreateSubSet()*
  This function creates all k-element subset of features set. It is equal to the combination of k item from d item. Thus the time complexity $= O(C(k, d)) = O(\frac{d!}{k! \times (d-k)!})$ where d is the dataset dimension and k is the number of elements in the subsets. Since, each time when we get a combination, we should copy it to the output variable that is $O(k)$, the total time complexity is $O(C(k, d) \times k)$.
- *Eliminate()*
  Eliminates the input set from *WorkingSet*. Finding a set with k element in a set of k-element sets is $O(k \times n)$ where n is the number of subsets with k element which leads to the total time complexity of $O(C(k, d) \times k)$.
- *Remove()*
  It removes source set from *WorkingSet* and copies it to destination set. As the main operation of this function is finding source set, then like *Eliminate()* function, the time complexity is $O(C(k, d) \times k)$.

Since while and for loops are implemented to check all possible subsets of features, they are repeated for 2^d times. According to condition checking at the beginning of 'for' loop, for each candidate subset one of the functions *Eliminate()* or *Evaluate()* plus *Remove()* is run. In the worst case, the time complexity of the algorithm is $O\left(2^{2d} \times n\right)$ where n represents the number of instances in the dataset and d is the dimension of feature space. This state occurs if only one feature is useful and other remove. As we do a preprocess and are sure that features are good enough and not more than a third of them would remove, then the algorithm is $\Theta(2^d \times n)$. In the best case that all features are useful and none of them can

```
Algorithm finding unvalued features
input: feature set of dataset
output: feature set Result such that removing it from dataset leads achieving BestIBA
Initialize the Irremovable and LowInfoFeatures set to Null, i to 1 and BestAnswer to 0
    BaseIBA = Evaluate (AllFeatures)
    /* calculate IBA of dataset using input features */
    CreateSubSet (WorkingSet, i)
    /* Create all i-member subset of features set and put it in WorkingSet */
    while ( WorkingSet(i) is not null) do
        for each ws ∈ WorkingSet do
            if (exist ir ∈ Ivvremovable that ir is a subset of ws) then
                Eliminate(ws)
                /* eliminates ws from WorkingSet */
            else
                e := Evaluate(AllFeatures - ws)
                diff := BaseIBA - e
                if (e > 0)
                    Remove (ws, Ivvremovable)
                    /* removes ws to Ivvremovable set */
                else
                    Remove (ws, LowInfoFeatures)
                    if (e >= BaseIBA) then
                        BestIBA := e
                        Result := ws
                    end if
                end if
            end if
        end for
        i := i + 1
        CreateSubSet (WorkingSet, i)
    end while
```

Fig. 2. The pseudo code of Smart-BT.

remove the time complexity is $\Omega(n \times d)$. As the order of this algorithm is exponential and dependent on the dimension of the feature space, so reduction of the number of features in preprocess is indispensable.

The space complexity of the Smart-BT algorithm is the maximum amount of space used by three main variables of this algorithm (*WorkingSet*, *Irremovable*set and *LowInfo* set). As 2 last sets are first empty and filled by moving working set members to them, then, the total space complexity is depending on the *WorkingSet* size that is O(k!).

4. Experimental results

To validate the performance of the proposed algorithm, we use the WEBSPAM-UK2007 database, which is introduced in this section. Then, by executing step-by-step the algorithm, we compare the results with other methods.

4.1. Dataset

WEBSPAM-UK2007 database [32] is one of the most reliable datasets in identifying web spam. The dataset contains a set of pages that are derived from the results of a crawler in the .uk domain, which is derived from 105.9 million pages out of 114529 hosts. The training package contains features of 3845 non-spam hosts and 208 spam hosts. The data of the test set are also extracted from 1740 non-spam hosts and 107 spam hosts. The ratio of the data of the two classes indicates that this dataset is unbalanced (Table 4). Among these hosts, 275 features have been extracted in the form of three sets named Content-based features, Link-based features, and Transformed link-based features. These features are calculated for both the home page and the most trusted host page (the page with the highest Page Rank).

Content-based features, which focus on the content of web pages, include features such as the number of words on the page, the number of words in the title, the average word length, the compression rate, and the entropy of the page. In this dataset, 96 content-based features have been extracted. Link-based features, which focus on links in web pages, include features such as the number of page output links, the number of page entry links, and the ratio of the number of output links to internal pages to the total output links. In this database, 41 link-based features have been extracted. Transformed link-based features include simple numeric conversions and a combination of link-based features. Among these features, one can point out the logarithm of the link-based features and the ratio between them. In this database, 138 features have been extracted.

4.2. Results

Due to the exponential computational cost of the proposed algorithm and its dependency to the number of features, and the high dimension of the web spam dataset, a preprocessing is done for dimension reduction. Therefore, by applying the classic algorithms

Table 4
WEBSPAM-UK2007.

	WEBSPAM-UK2007	
	Train	Test
Spam	208	107
Non-Spam	3845	1740
Ratio	5%–95%	6%–94%

Table 5
Comparison of IBA values obtained from different selection algorithms.

Attribute evaluator	Search method	No. of Selected features	IBA
Correlation	Forward selection	44	0.337
	Best first	66	0.362
	LForward selection	12	0.232
	Greedy stepwise	66	0.362
	Genetic search	63	0.273
Consistency	Forward selection	17	0.109
	Best first	8	0.245
	LForward selection	21	0.183
	Greedy stepwise	5	0.162
	Genetic search	15	0.208
Chi Square	Ranker	**32**	**0.369**
Information Gain		29	0.363
Gain Ratio		39	0.35
Symmetric Uncertainty		30	0.325
Relief		51	0.204
OneR		19	0.027
Naïve Bayes Classifier	Genetic search	30	0.21
Naïve Bayes Classifier	PSO	19	0.307
Naïve Bayes Classifier	ABC	110	0.307
Naïve Bayes Classifier	ACO	184	0.307

of feature selection like Weka software [33], we chose the best method according to the index of balanced accuracy of the Naïve Bayes classifier. The reason of using Naïve Bayes classifier is the results mentioned in [34] that has compared the performance of different classifiers on web spam dataset and indicated that the Naïve Bayes classifier is the best classifier in this case. The results are shown in Table 5.

As it is seen in Table 5, eight filter methods and one wrapper method along with 6 search modes are used to select the features. We applied these methods to 275 features in the dataset. The basic IBA (before the feature selection) is 0.337. But, using the ranking search method and the Chi-Square evaluation function leads to the selection of a 32-member subset of features that increases the IBA index to 0.369. It is worth noting that although the use of the information gain assessment function results in the least number of features, the result is not as effective as the Chi-Square evaluator.

After using the Chi-Square evaluation function, which reduced the number of features from 275 to 32, we examine the impact of the removal of each feature on the classification accuracy by applying the proposed algorithm. According to the results, we find the largest non-removable subset of the features and reduce the features vector dimensions. The results obtained during the implementation of the algorithm are shown in Fig. 3. The horizontal axis of this graph shows the number of selected features and the vertical axis shows the gained IBA. As can be seen, the highest IBA was obtained by selecting 27 features in which the IBA is increased from 0.369 to 0.411.

In order to better showing this method performance, we run the proposed method on each of the three features categories of the dataset which mentioned in the previous section. The results are shown in Table 6.

As it is seen, decreasing the dimension of the dataset using the proposed method cause achieving similar and sometimes better

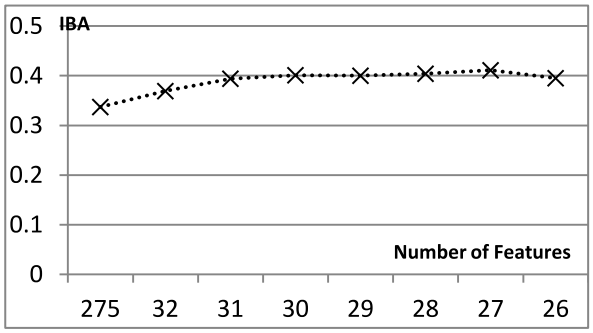

Fig. 3. Features reduction and performance improvements.

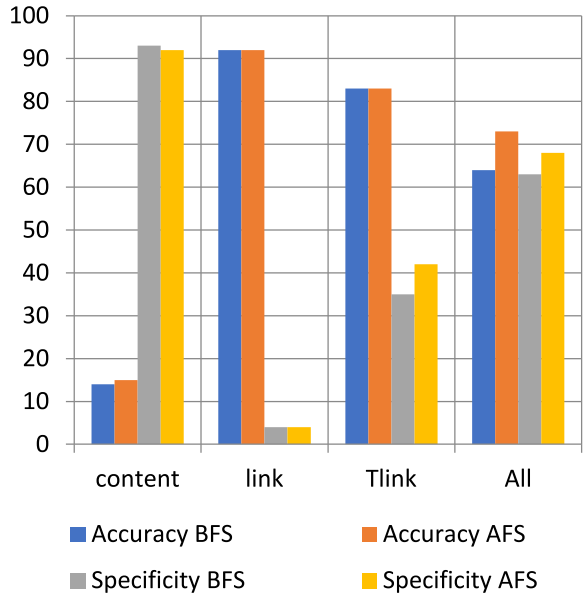

Fig. 4. Comparing the performance of classifier before and after feature selection.

results by fewer features compared with using the whole feature set. Also, we calculated Laplacian score (LS) of each feature and show that omitted features are those which have a fewer score (Table 7).

LS is fundamentally based on Laplacian Eigenmaps and Locality Preserving Projection [35]. Its basic idea is to evaluate the features according to their locality preserving power. The above table indicates the proper functionality of the proposed method from the mathematics aspect. The importance of this dimension reduction and its impact on the detection rate is illustrated in Fig. 4.

According to Fig. 4, it is obvious that the accuracy of the classifier after feature selection is unchanged or even increased. Also, the spam detection rate except for an inconsiderable decrease in content-based feature set is unchanged or even increased after feature selection. It shows that the proposed method is able to reduce the data dimension without losing efficient information involved with features for web spam detection. We also compare obtained results with other research works in Table 8. It can be observed that amounts of IBA in all three cases are close to each other, but the number of features is completely different.

Moreover, the examination of selected features shows that the text of a web page contains more information about whether the web page is spam or not (See Table 9). In this table, each row indicates feature type distribution of first column feature sets.

As it is seen, in the initial feature set, most of the features are link-based. But after using the Ranker feature selection method,

Table 6
Comparing obtained results on different feature sets of WEBSPAM-UK2007.

Feature Set	FS Method	No. Features	Accuracy	Specificity	IBA
Content-Based	–	96	14%	0.90	0.002
	Chi squared + ranker	76	14%	0.91	0.006
	Proposed Method	**57**	**15%**	**0.92**	**0.020**
Link-Based	–	**41**	**92%**	**0.05**	**0.029**
	Chi squared + ranker	22	92%	0.04	0.014
	Proposed Method	10	92%	0.04	0.017
Transformed Link-Based	–	138	82%	0.34	0.192
	Chi squared + ranker	47	82%	0.43	0.248
	Proposed Method	40	83%	0.42	0.243
All Features	–	275	70%	0.64	0.337
	Chi squared + ranker	32	72%	0.63	0.359
	Proposed Method	**27**	**73%**	**0.68**	**0.411**

Table 7
Laplacian score of features and result of Smart-BT feature selection method.

	Feature number										
	1	2	3	4	5	6	**7**	8	9	10	
LS	0.63	0.63	0.39	0.67	0.70	0.65	**0.18**	0.71	0.85	0.41	
Selected by Smart-BT	Yes	Yes	Yes	Yes	Yes	Yes	**No**	Yes	Yes	Yes	
	11	**12**	13	**14**	15	16	17	18	19	20	21
LS	0.56	**0.11**	0.90	**0.19**	0.52	0.80	0.52	0.20	0.19	0.20	0.29
Selected by Smart-BT	Yes	**No**	Yes	**No**	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	22	**23**	24	25	26	27	**28**	29	30	31	32
LS	0.93	**0.15**	0.74	0.66	0.59	0.81	**0.23**	0.49	0.34	0.76	0.74
Selected by Smart-BT	Yes	**No**	Yes	Yes	Yes	Yes	**No**	Yes	Yes	Yes	Yes

Table 8
Comparison of the results.

	Number of features	Classifier	IBA	Recall	F-Score	AUC
Smart-BT	28	Naïve Bayes	0.41	**0.636**	0.226	0.718
Silva [24]	137	MDLclassifier	0.40		0.225	–
Patils [10]	296	SVM	**0.43**	–	**0.44**	**0.80**
Fdez-Glez [36]	275	C5.0 + SVM + REGEX	–	0.442	0.41	0.673

Table 9
Specifications of selected features by the proposed algorithm.

Feature set	Content-based	Link-based	Transformed-link-based
Initial (275)	96	41	138
Ranker + Chi square (32)	18	7	7
Proposed algorithm (28)	**14**	7	7

content-based features gain more portion than link-based. This recurs even running the proposed algorithm. Note that transformed link-based features are the link-based features that have only been transformed and are not new ones. For this reason, it seems that content-based features are more precious than link-based one.

5. Conclusion

The main objective of this study is considering the impact of dimension reduction on increasing the performance of classification, especially on unbalanced datasets like web spam detection problem. This paper presents a feature selection method called Smart-BT. Due to the importance of each feature, the proposed method tries to find the largest subset of features which eliminating them from the initial set not only does not decrease the efficiency of classification but also increases the efficiency of classification. The fundamental concept behind this algorithm is attention to the different behavior of features, individually and together. The proposed method has been tested on WEBSPAM-UK2007 dataset. The results reveal that Smart-BT provides very competitive results in comparison with other well-known feature selection methods such as Forward selection, Ranker, genetic algorithm, and PSO. Also, regarding its nature, it is an efficient method in low dimension datasets and selects the near optimal feature set. Hence, the proposed model is appropriate in the real-time problems, where the slightest changes in the data size are very important due to the impressing the running time. In the future work, we plan to increase the detection rate of web spam by considering the concept drift topic and extracting the proper features for this.

References

[1] D. Loiz, Advanced Web Ranking, Caphyon, 5 June 2018, (online). https://www.advancedwebranking.com/ctrstudy/. (Accessed 5 July 2018).

[2] C. Castillo, D. Donato, A. Gionis, V. Murdock, F. Silvestri, Know your neighbors: Web spam detection using the web topology, in: 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, The Netherlands, 2007.

[3] M. Yu, J. Zhang, J. Wang, J. Gao, T. Xu, R. Yu, The research of spam web page detection method based on web page differentiation and concrete clusters centers, in: International Conference on Wireless Algorithms, Systems, and Applications, Tianjin, China, 2018.

[4] J.J. Whang, Y.S. Jeong, I. Dhillon, S. Kang, J. Lee, Fast asynchronous anti-trustrank for web spam detection, in: WSDM workshop on Misinformation and Misbehavior Mining on the Web, Los Angeles, California, USA, 2018.

[5] M. Danandeh Oskuie, S.N. Razavi, A survey of web spam detection techniques, Intl. J. Comput. Appl. Technol. Res. 3 (3) (2014) 180–185.

[6] K.L. Goh, A.K. Singh, Comprehensive literature review on machine learning structures for web spam classification, Proc. Comput. Sci. 70 (2015) 434–441.

[7] T. Lingala, G. Saritha, Towards evaluating web spam threats and countermeasures, Intl. J. Innov. Adv. Comput. Sci. 7 (3) (2018) 71–80.

[8] S. Wei, Y. Zhu, Cleaning out web spam by entropy-based cascade outlier detection, in: International Conference on Database and Expert Systems Applications, Lyon, France, 2017.

[9] L. Araujo, J. Martinez-Romo, Web spam detection: new classification features based on qualified link analysis and language models, IEEE Trans. Inform. Forensics Secur. 5 (3) (2010) 581–590.

[10] R.C. Patil, D.R. Patil, Web spam detection using svm classifier, in: 9th International Conference on Intelligent Systems and Control, Coimbatore, India, 2015.

[11] A. Alarifi, M. Alsaleh, Web spam: a study of the page language effect on the spam detection features, in: 11th International Conference on Machine Learning and Applications, Boca Raton, FL, USA, 2012.

[12] P.K. Karunakaran, S. Kolkur, Language model issues in web spam detection, Enhanc. Res. Manag. Comput. Appl. 3 (2) (2014) 39–43.

[13] S. Kumar, X. Gao, I. Welch, Novel features for web spam detection, in: 28th International Conference on Tools with Artificial Intelligence, San Jose, CA, USA, 2016.
[14] K. Hunagund, S. Kumar, Spam web page detection based on content and link structure of the site, Adv. Res. Comput. Commun. Eng. 4 (8) (2015) 348–351.
[15] C. Dong, B. Zhou, Effectively detecting content spam on the web using topical diversity measures, in: International Conferences on Web Intelligence and Intelligent Agent Technology, Macau, China, 2012.
[16] J. Wan, M. Liu, J. Yi, Detecting spam web pages through topic and semantics analysis, in: Global Summit on Computer & Information Technology, Sousse, Tunisia, 2015.
[17] Y. Suhara, H. Toda, S. Nishioka, S. Susaki, Automatically generated spam detection based on sentence-level topic information, in: 22nd International Conference on World Wide Web, Rio de Janeiro, Brazil, 2013.
[18] M. Luckner, M. Gad, P. Sobkowiak, Stable web spam detection using features based on lexical items, Comput. Secur. 46 (2014) 79–93.
[19] M.S.I. Mamun, M.A. Rathore, A. Habibi Lashkari, N. Stakhanova, A.A. Ghorbani, Detecting malicious urls using lexical analysis, in: Network and System Security, 2016.
[20] H. Jelodari, Y. Wang, C. Yuan, X. Jiang, A systematic framework to discover pattern for web spam classification, in: The 8th IEEE Annual Information Technology, Electronics and Mobile Communication Conference, Vancouver, BC, Canada, 2017.
[21] T. Singh, M. Kumari, S. Mahajan, Feature oriented fuzzy logic based web spam detection, Inform. Optim. Sci. 38 (6) (2017) 999–1015.
[22] J. Fdez-Glez, D. Ruano-Ordas, J.R. Méndez, F. Fdez-Riverola, R. Laza, R. Pavón, A dynamic model for integrating simple web spam classification techniques, Expert Syst. Appl. 42 (21) (2015) 7969–7978.
[23] H. Li, Web spam detection based on improved tri-training, in: International Conference on Progress in Informatics and Computing, Shanghai, China, 2014.
[24] R.M. Silva, T.A. Almeida, A. Yamakami, Towards web spam filtering using a classifier based on the minimum description length principle, in: 15th International Conference on Machine Learning and Applications, Anaheim, CA, USA, 2016.
[25] A. Chandra, M. Suaib, R. Beg, Low cost page quality factors to detect web spam, Inform. Eng. 2 (3) (2014) 1–7.
[26] Y. Li, X. Nie, R. Huang, Web spam classification method based on deep belief networks, Expert Syst. Appl. 96 (1) (2018) 261–270.
[27] G. Chandrashekar, F. Sahin, A survey on feature selection methods, Comput. Electr. Eng. 40 (2014) 16–28.
[28] A.S. Abdullah, C. Ramya, V. Priyadharsini, C.S. Reshma, S. S., A survey on evolutionary techniques for feature selection, in: Conference on Emerging Devices and Smart Systems, Tiruchengode, India, 2017.
[29] V. García, R.A. Mollineda, J.S. Sánchez, Index of balanced accuracy: A performance measure for skewed class distributions, in: 4th Iberian Conference on Pattern Recognition and Image Analysis, Portugal, 2009.
[30] V. García, R.A. Mollineda, J.S. Sánchez, Theoretical analysis of a performance measure for imbalanced data, in: International Conference on Pattern Recognition, Istanbul, Turkey, 2010.
[31] P. Domingos, M. Pazzani, On the optimality of the simple bayesian classifier under zero-one loss, Mach. Learn. 29 (2–3) (1997) 103–130.
[32] C. Castillo, D. Donato, L. Becchetti, P. Boldi, S. a. S. M. Leonardi, S. Vigna, A reference collection for web spam, SIGIR Forum 40 (2) (2006) 11–24.
[33] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The weka data mining software: an update, ACM SIGKDD Explor. Newslett. 11 (1) (2009) 10–18.
[34] f. Asdaghi, A. Soleimani, Study the effect of classification methods on detection rate of web spams, in: 3rd International Conference on Applied Reserach in Computer and IT, Tehran, 2016.
[35] G. Dhiman, V. Kumar, Astrophysics inspired multi-objective approach for automatic clustering and feature selection in real-life environment, Modern Phys. Lett. B 32 (31) (2018) 1–23.
[36] J. Fdez-Glez, D. Ruano-Ordás, R. Laza, J.R. Méndez, R. Pavón, F. Fdez-Riverola, WSF2: a novel framework for filtering web spam, Sci. Program. 2016 (2016) 1–18.