An Iterative Hybrid Filter-Wrapper Approach to Feature Selection for Document Clustering

Mohammad-Amin Jashki, Majid Makki, Ebrahim Bagheri, and Ali A. Ghorbani

Faculty of Computer Science
University of New Brunswick
{a.jashki,majid.makki,ebagheri,ghorbani}@unb.ca

Abstract. The manipulation of large-scale document data sets often involves the processing of a wealth of features that correspond with the available terms in the document space. The employment of all these features in the learning machine of interest is time consuming and at times reduces the performance of the learning machine. The feature space may consist of many redundant or non-discriminant features; therefore, feature selection techniques have been widely used. In this paper, we introduce a hybrid feature selection algorithm that selects features by applying both filter and wrapper methods in a hybrid manner, and iteratively selects the most competent set of features with an expectation maximization based algorithm. The proposed method employs a greedy algorithm for feature selection in each step. The method has been tested on various data sets whose results have been reported in this paper. The performance of the method both in terms of accuracy and Normalized Mutual Information is promising.

1 Introduction

Research on feature selection has recently gained considerable amount of attention due to the increasing complexity of objects within the target domains of interest whose data sets consist of hundreds of thousands of features. These target domains cover areas such as the analysis and understanding of corporate and publicly available documents, the detection of the most influential genes based on DNA micro-array experiments, and experiments in combinatorial chemistry, among others [8]. Document clustering has specially been a fertile area for the employment of feature selection techniques due to its wide variety of emerging applications including automatic email spam detection, News article categorization, document summarization, and others. In these domains, documents are usually represented by 'bag-of-words', which is a vector equal in dimension to the number of existing vocabulary in the domain of discourse [4]. Studies have revealed that document collections whose vocabulary domain size are between 5,000 and 800,000 are common, whose categorization would hence require scalable techniques that are able to operate over a feature size of this magnitude [13]. It is clear that this representation scheme possesses two main characteristics: 1) high dimensionality of the feature space; and 2) inherent sparsity of each

Y. Gao and N. Japkowicz (Eds.): Canadian AI 2009, LNAI 5549, pp. 74–85, 2009.

vector over all terms, which are detrimental to the achievement of the optimal performance for many learning techniques.

Despite these problems, the bag-of-words representation usually performs well with small enhancements on textual data sets due to some of their inherent traits: First, since the bag-of-words representation is oblivious to the word appearance sequence in a document, neighboring terms do not have to necessarily co-exist in the bag-of-words vector. Second, investigations have shown that the large feature space developed by the bag-of-words representation typically follows the Zipf-like distribution, i.e., there are a few very common frequently seen terms in the document set along with many very unfrequent terms [5]. Third, even among the very common frequent terms, most of them do not possess a high discrimination power (frequent-but-non-discriminant terms). This may be due to the fact that such terms have a similar occurrence pattern in all existing document classes. Stop words and auxiliary verbs are examples of such terms. This can also be analyzed within the context of relevant but redundant terms. Therefore, feature selection techniques can be applied on features of the bag-of-words representation in document categorization in order to select a highly discriminant subset such that 1) the prediction performance of the classifiers/clusterers are enhanced; 2) classifiers/clusterers are learnt faster and more cost effectively; and 3) the underlying concepts behind the available corpus of data are revealed and understood.

In this paper, we propose an iterative feature selection scheme, which greedily selects the best feature subset from the bag-of-words that best classify the document set in each step. The method employs an Expectation Maximization (EM) approach to feature selection and document clustering due to the restriction that supervised feature selection techniques cannot be directly applied to textual data because of the the unavailability of the required class labels. Briefly explained, our method initially labels the documents with random labels. Based on these labels, it then greedily chooses the best representative subset from the feature space. The selected features are then employed for clustering the documents using the k-Means algorithm. This iterative process of feature selection and document clustering is repeated until the satisfaction of a certain stopping criterion. It is important to mention that the proposed greedy algorithm evaluates the suitability of the features locally within the context of each individual cluster. This local computation allows the greedy algorithm to find the most discriminative features. Our approach in combining the EM algorithm with a greedy feature selection method shows improved performance with regards to *accuracy* and *Normalized Mutual Information (NMI)* compared with some existing techniques for document clustering.

The paper is organized as follows: the next section, reviews some of the existing techniques for feature selection and document clustering. Section 3 introduces our proposed iterative feature selection and document clustering technique. The results of the evaluation of the performance of the proposed technique is given in Section 4. The paper is then concluded in Section 5.

2 Related Work

The work on the manipulation of the feature space for text categorization has been three-fold: 1) feature generation; 2) feature selection; and 3) feature extraction [13]. In feature generation, researchers have focused on employing the base features available in the initial feature space to create suitable discriminant features that best reflect the nature of the document classes. For instance, some approaches consider different word forms originating from the same root as one term in the feature space and employ stemming techniques to extract such features. Here, all of the terms such as *fishing, fished, fishy,* and *fisher* would be represented by their root word, *fish*, in the feature space. Along the same lines, some other techniques have gone further in exploring similarity between words by using thesaurus and ontologies to create groups of features, which would be considered as a single feature in the feature space. In this approach, terms such as *office, bureau,* and *workplace* would be all grouped into one feature in the feature space [2]. Furthermore, as was mentioned earlier, the bag-of-words approach fundamentally neglects the sequence of word occurrences in the document. To alleviate this, some researchers have made the observation that the use of word *n-grams*[1] for creating word phrase features can provide more specificity. In this approach, n-grams are mined from the documents (2-word phrases are most popular) and are used as representative features in the feature space. It is important to notice that although the number of potential n-grams increases exponentially with the value of n, there are only a small fraction of phrases that receive considerable probability mass and are found to be predictive; therefore, the size of the feature space would not grow very large. Other methods such as using unsupervised techniques for clustering the terms in the feature space can be used to create classes of similar features. These classes can themselves be used as a complex features in the new feature space.

There are three main approaches within the realm of feature selection [10] that have been introduced in the following:

1. *Filter methods* assess each feature independently, and develop a ranking between the members of the feature space, from which a set of top-ranked features are selected. The evaluation of each feature is performed on their individual predictive power. This can be done for instance using a classifier built using that single feature, where the accuracy of the classifier can be considered as the fitness of the feature. Here, a limitation for such an evaluation is the absence of the required class labels in document categorization data sets; therefore, unsupervised methods need to be employed to evaluate the features. Some filter methods assess each feature according to a function of its *Document Frequency (DF)* [11], which is the number of documents in which a term occurs in the data set. Other unsupervised measures have been defined on this basis such as *Term Strength (TS)* [11], which is based on the conditional probability that a feature occurs in the second half of a pair of

[1] Or character n-grams for languages such as Chinese and Japanese that do not have a space character.

related documents given that it appeared earlier, *Entropy-based Ranking* [3] that is measured by the amount of entropy reduction as a result of feature elimination, and *Term Contribution (TC)*, which is a direct extension of DF that takes term weights in to account. In addition, Liu et al [11]. have shown how several supervised measures like *Information Gain* and χ^2 *statistic* can be used for the purpose of feature ranking in document clustering. However, the computation time of this method is still under question. Filtering methods are computationally and statistically scalable since they only require the computation of n scores for n features, and also they are robust to overfitting, since although they increase bias, they may have less variance; however at the same time, they may be exposed to the selection of redundant features.

2. *Wrapper methods* employ AI search techniques such as greedy hill climbing or simulated annealing in order to find the best subset of features from the feature space. Different feature subsets are evaluated repeatedly through cross-validation with a certain learning machine of interest. One can criticize wrapper methods for being brute-force methods that need great amount of computation in order to cover all of the search space. Conceptually this is true, but greedy search strategies have been devised that are computationally wise and robust against overfitting. For instance, two popular greedy search strategies are *forward selection*, which incrementally incorporates features into larger subsets, and *backward elimination* that starts with the set of all features and iteratively eliminates the least promising ones. Actual instantiations of these two strategies have been proposed in the related literature. For instance, the Gram-Schmidt orthogonolization procedure provides the basis for forward feature selection by allowing the addition of the feature that reduced the mean-squared error the most at each step [6].
3. *Embedded methods* try to build a prediction model that attempts to maximize the goodness of fit of the developed model and minimize the number of input features of the model. Such methods are reliant on the specifics of the utilized learning machine used in the prediction model. Embedded methods that incorporate feature selection as a part of their training process possess some interesting advantages such as reaching a solution faster by avoiding the retrain of the model for each feature subset and also making better use of the available data by not needing to split the data into training and validation subsets. Decision tree learning algorithms such as CART inherently include an embedded feature selection method [1].

Finally, feature extraction methods are a subclass of the general dimensionality reduction algorithms. These methods attempt to construct some form of combination of all or a subset of the initial features in order to develop a reduced-size feature space that represents the initial data with sufficient amount of accuracy. emphPrinciple Component Analysis (PCA), emphLatent Semantic Analysis (LSA), and non-linear dimensionality reduction methods are some of the representatives of these methods [9]. One of the drawbacks of this approach is that the developed features of the new feature space are not easily interpretable since they may not have an obvious or straightforward human understandable interpretation.

3 Proposed Method

In practice, wrapper methods employ the prediction performance of the learning machine to assess the relative usefulness of the selected features. Therefore, a wrapper method needs to provide three main decisions with regards to the following concerns: First, a search strategy, such as the greedy forward selection and backward elimination strategies introduced earlier, needs to be created that would guide the process of searching all of the possible feature space efficiently. Second, methods for the assessment of the prediction performance of the learning machine need to be defined in order to guide the search strategy, and Third, the choice for the appropriate learning machine needs to be made. As it can be seen wrapper methods rely on the prediction performance of the learning machine, which is not a viable strategy since they require the class labels that are not present in document clustering. However, filter methods use other unsupervised measures such as DF, TS and others to evaluate the usefulness of the selected features and hence rank the available features accordingly and select the most competent, which makes them suitable for the task of document clustering.

As was discussed earlier, filter methods have been criticized for selecting redundant or locally optimum features from the feature space [15]. This is due to the fact that they tend to select the top best features based on a given measure without considering the different possibilities of feature composition available in the feature space. It seems enticing to create a hybrid strategy based on filter methods and wrapper methods to overcome their limitations and reap their individual capabilities. An efficient hybrid strategy would provide two main benefits for document clustering: 1) it decreases the chance of being trapped by a local optimum feature sets through the use of an iterative greedy search strategy; 2) both supervised and unsupervised measures can be used to evaluate the discriminative ability of the selected features in each iteration of the process. Here, we propose such a hybrid strategy.

Our proposed method for feature selection and document clustering employs a greedy algorithm, which iteratively chooses the most suitable subset of features locally in each round. The chosen features are used to create corresponding document clusters using the k-Means algorithm. The resulting document clusters are then employed as local contexts for the greedy algorithm to choose a new subset of features from each cluster. The representative features of each cluster are chosen such that they could maximally discriminate the documents within that cluster. The union set of all local features of clusters is developed, which would serve as the newly selected feature subset. Formally said, our approach is a combination of the expectation maximization algorithm accompanied by a greedy search algorithm for traversing the feature space, and an unsupervised feature ranking technique. Expectation maximization algorithms are employed for maximum likelihood estimation in domains with incomplete information. A typical EM algorithm estimates the expectation of the missing information based on the current observable features in its *E-step*. In the *M-step*, the missing information are replaced by the expected value estimates computed in the *E-step* in order to develop a new estimation that can maximize the complete

data likelihood function. These steps are iterated until a certain stopping criterion is satisfied.

Liu et al. [11] have proposed a general framework for the employment of EM for text clustering and feature selection. We employ their formulation of the problem statement within their framework and provide our own instantiation of the provided skeleton. Here, the basic assumption is that a document is created by a finite mixture model between whose components and the clusters there exists a one-to-one correspondence. Therefore, the probability of all documents given the model parameters can be formulated as follows:

$$p(D|\theta) = \prod_{i=1}^{N} \sum_{j=1}^{|C|} p(c_j|\theta) p(d_i|c_j, \theta) \tag{1}$$

where D denotes the document set, N the number of documents in the data set, c_j the j^{th} cluster, $|C|$ the number of clusters, $p(c_j|\theta)$ the prior distribution of cluster c_j, and $p(d_i|c_j, \theta)$ the distribution of document d_i in cluster c_j. Further, since we use the bag-of-words representation, we can assume that document features (terms) are independent of each other given the document class label. Hence, the likelihood function developed in the above equation can be re-written as

$$p(D|\theta) = \prod_{i=1}^{N} \sum_{j=1}^{|C|} p(c_j|\theta) \prod_{t \in d_i} p(t|c_j, \theta) \tag{2}$$

where t represents the terms in d_i, and $p(t|c_j, \theta)$ the distribution of term t in cluster c_j. Since not all of the terms in a document are equally relevant to the main concept of that document, $p(t|c_j, \theta)$ can be regarded as treated as the sum of relevant and irrelevant distributions:

$$p(t|c_j, \theta) = z(t)p(\text{t is relevant}|c_j, \theta) + (1 - z(t))p(\text{t is irrelevant}|\theta) \tag{3}$$

where $z(t) = p(\text{t is relevant})$, which is the probability that term t is relevant. Therefore, the likelihood function can be reformulated as below:

$$p(D|\theta) = \prod_{i=1}^{N} \sum_{j=1}^{|C|} p(c_j|\theta) \prod_{t \in d_i} \Big[z(t)p(\text{t is relevant}|c_j, \theta) + (1 - z(t))p(\text{t is irrelevant}|\theta) \Big] \tag{4}$$

Now, the expectation maximization algorithm can be used to maximize the likelihood function by iterating over the following two step:

$$\text{E-step: } \hat{z}^{(k+1)} = E(z|D, \hat{\theta}^{(k)}) \tag{5}$$

$$\text{M-step: } \hat{\theta}^{(k+1)} = argmax_\theta \; p(D|\theta, \hat{z}^{(k)}) \tag{6}$$

In the following, we provide details of each of the two steps of the customized EM algorithm for our hybrid feature selection and document clustering method.

It should be noted that in our proposed method we assume that the number of correct document classes is known *a priori*, denoted $\mathbf{k}$.

Lets assume that the vector $\mathbf{Y}$ represents the class labels for each of the N documents; therefore, $|\mathbf{Y}| = N$, and $\mathbf{Y}_i$ would denote the class label of the i^{th} document. In the first iteration of the process, the values for $\mathbf{Y}$ are unknown for which we assign randomly picked values to the class labels; hence, each document is randomly classified into one of the $\mathbf{k}$ clusters. Now, we would like to suppose that the clusters developed based on random label assignments are the best classification representatives of the available documents. Therefore, it is desirable to find the set of features locally within each cluster that provide the best approximation of the available data in that cluster. For this purpose, let us proceed with some definitions.

Definition 1. *Let D_i be the set of documents in cluster i, D_i^j be the j^{th} document in D_i, and t be a given term in the feature space. Local Document Frequency of t (in D_i), denoted $\mathcal{LDF}(D_i, t)$, is defined as follows:*

$$\mathcal{LDF}(D_i, t) = \sum_{j=1}^{|D_i|} (t \in D_i^j ? 1 : 0) \tag{7}$$

which is the number of documents in which the term t has appeared.

Definition 2. *Let C_i be the i^{th} cluster, and D_i be the set of documents in C_i. A feature such as t is assumed to be a competent feature of C_i iff:*

$$\forall j \neq i \in \mathbf{k}: \quad \mathcal{LDF}(D_i, t) > \mathcal{LDF}(D_j, t) \tag{8}$$

Informally stated, a feature is only a competent feature of a given cluster if it's local document frequency is highest in that given cluster compared to all other clusters.

Definition 3. *Let C_i be the i^{th} cluster. A competent set for C_i, denoted $\mathcal{COMP}(C_i)$, is defined as follows:*

$$\mathcal{COMP}(C_i) = \{t \mid t \text{ is a competent feature of } C_i\} \tag{9}$$

The competent set of features for each cluster possess the highest occurrence rate locally over all of the available clusters; therefore, they have a high chance of being a discriminant feature. This is because their local document frequency measure behavior is quite distinct from the same feature in the other clusters.

With the above definitions, we are able to locally identify those features that are competent. The competent set of features for cluster is hence identified. The union of all these features over all of the clusters is generated, which would represent the new feature space. Once the new feature space is developed, they are employed to cluster all of the documents once more using the k-Means algorithm. The k-Means algorithm would provide new values for $\mathbf{Y}$. The values of this vector are employed in order to identify the new competent set of features for each cluster,

which would consequently be used to re-cluster the documents. This process is iteratively repeated until a relatively stable cluster setting is reached.

The stopping criterion of the iterative process is based on the distance of the clusterings in the last two iterations. In other words, when the distance of these clustering is less than a threshold τ. The distance between two clusterings is computed by considering one of these clustering as the natural class labels and calculating the accuracy the other clustering.

4 Performance Evaluation

In the following, configurations of the running environment, data sets, and measuring methods for the experiments is explained. Results and corresponding analyzes are presented afterwards.

4.1 Experimental Settings

The proposed algorithm and the rivals have been implemented in Java. All experiments has been performed on an Intel Xeon 1.83GHz with 4GB of RAM.

Four data sets has been used to conduct the experiments. Table 1 shows the properties of the data sets. In this table, n_d, n_w, k, and $\hat{n}_c$ represent the total number of documents, the total number of terms, the number of natural classes, and the average number of documents per class respectively. Balance, in the last column, is the ratio of the number of documents in the smallest class to the number of documents in the largest one. The distance metric used for the k-means algorithm is Cosine similarity.

The *k1b* data set is prepared by the WebACE project [7]. In this data set, each document is a web page from the subject hierarchy of Yahoo!. *NG17-19* is a subset of a collection of messages obtained from 20 different newsgroups known as *NG20*. All three classes of the *NG17-19* data set are related to political subjects; hence, difficult to separate the documents by the clustering algorithms. The *hitech* and *reviews* data sets contain San Jose Mercury newspaper articles.

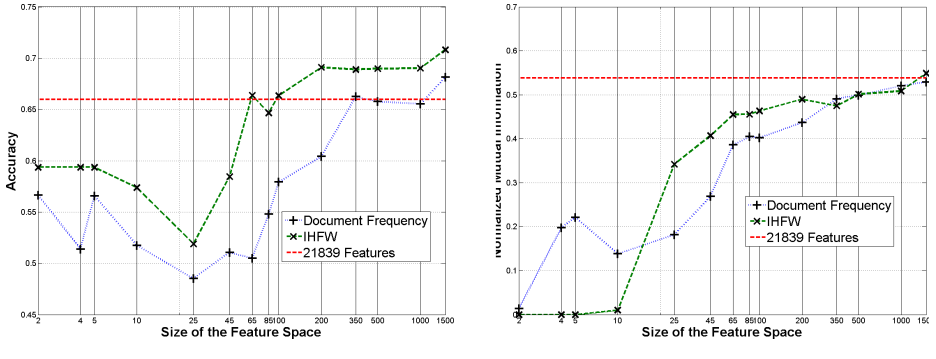

Fig. 1. Results for the k1b Data Set

Table 1. Data Sets

Data	Source	n_d	n_w	k	$\hat{n}_c$	Balance
k1b	WebACE	2340	21839	6	390	0.043
NG17-19	3 overlapping groups from NG20	2998	15810	3	999	0.998
hitech	San Jose Mercury(TREC)	2301	10080	6	384	0.192
reviews	San Jose Mercury(TREC)	4063	18483	5	814	0.098

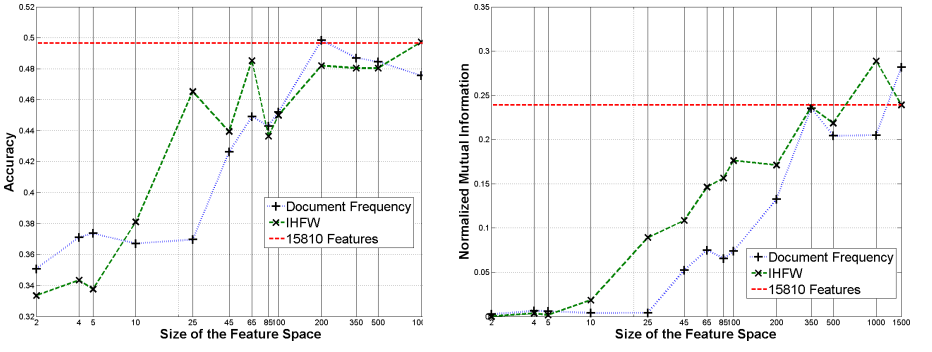

Fig. 2. Results for the NG17-19 Data Set

The former is about computers, electronics, health, medicine, research, and technology. Topics of the latter are food, movies, music, radio, and restaurants.

Two preprocessing steps including stemming and removal of the words appearing in less than three documents has been applied to all data sets.

In order to evaluate the efficiency of the proposed algorithm, the results has been compared to the DF method [14,11] which is a straightforward but efficient feature selection algorithm. In addition, the original k-Means algorithm has been applied to the data sets with all words in the feature space. The TS algorithm [14,11] is another possible rival method. However, the computing time of this algorithm is not comparable neither with IHFW[2] nor with DF. Since feature weighting schemes such as $tf - idf$ obscures/improves the efficiency of any dimensionality reduction method, comparison with TC or any other method which takes advantage of these schemes is unfair.

In addition to the accuracy of the clusterings (according to the class labels), the NMI [12] is reported due to its growing popularity.

NMI formula is presented here in Equation 10, where l is a cluster and h is a class of documents, n_h and n_l are the number of their corresponding documents, $n_{h,l}$ is the number of documents in class h as well as cluster l, and n is the size of the data set. Higher NMI values indicate high similarity between the clustering and the class labels.

[2] We refer to our proposed method as IHFW, hereafter.

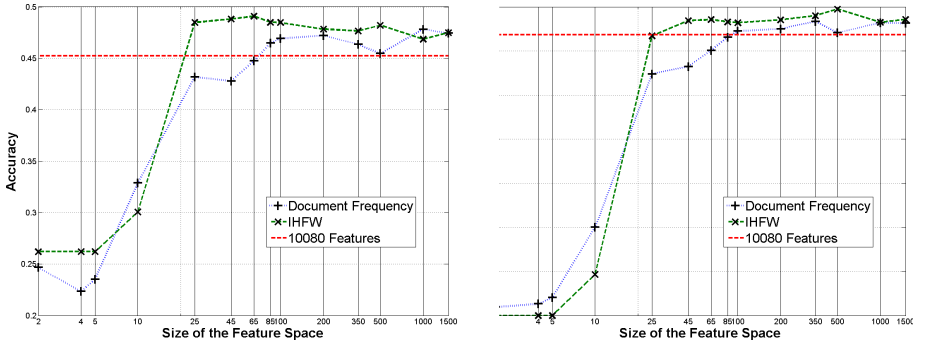

Fig. 3. Results for the hitech Data Set

$$NMI = \frac{\sum_{h,l} n_{h,l} \log \frac{n \cdot n_{h,l}}{n_h n_l}}{\sqrt{(\sum_h n_h \log \frac{n_h}{n})(\sum_l n_l \log \frac{n_l}{n})}} \tag{10}$$

Since k-Means is a randomized algorithm, the methods have been performed 10 times for each size of the feature space and the average is reported. The threshold used for the stopping criterion (maximum distance of the last two clusterings) is equal to 0.1. Smaller values may increase both the clustering performance and computation time.

4.2 Results and Discussions

Figures 1, 2, 3, and 4 illustrate the experimental results for all data sets. The x-axis is logarithmically scaled in all figures. The two line charts for each data set depict the evaluation metrics described in Section 4.1.

In almost all cases, the NMI value obtained by the proposed algorithm outperforms the DF method. There are only rare cases (feature sizes of 2, 4, 5, and 10 in Figure 1) that DF outperformed the proposed algorithm. The distance between DF and the proposed algorithm NMI values is higher when the number of features is reduced aggressively in particular. The difference between the NMI values is decreased as the number of selected features increases.

The NMI values for the proposed algorithm tend to reach that of k-Means as the number of selected features grows to 10% of the total number of words. For the *hitech* data set, the NMI value given by our algorithm outperforms the original k-Means even for very small feature sizes as illustrated in Figure 3.

Table 2. Comparing the Running Time (in Seconds) of IHFW with k-Means

a	**k1b**	**NG17-19**	**hitech**	**reviews**
$IHFW_{max}$	542.27	619.73	619.75	619.72
$IHFW_{average}$	**246.69**	383.16	288.87	**295.77**
k-Means	383.96	**173.93**	**233.0**	443.51

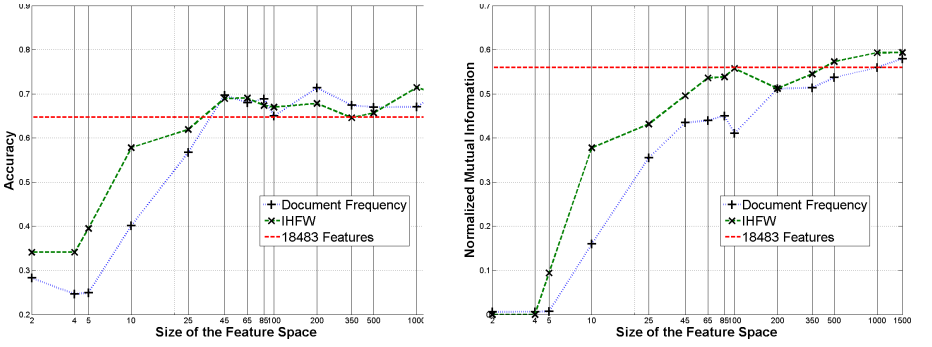

Fig. 4. Results for the reviews Data Set

The accuracy of the proposed algorithm also exceeds that of DF in most cases. For all feature sizes greater than 65 over all data sets except the *NG17-19*, the accuracy of the proposed algorithm is either higher than the accuracy of k-Means or there is at most 1% loss of accuracy. With regard to the *NG17-19* data set, at most 6% loss of accuracy for feature sizes greater than 65 is observed. It is notable that 65 features is 0.29%, 0.41%, 0.64%, 0.34% for the *k1b*, *NG17-19*, *hitech*, and *reviews* data sets respectively. Therefore, one can state that this algorithm is capable of aggressively reducing the size of the feature space with either negligible loss of accuracy or boosted accuracy.

The computation time of the DF algorithm is obviously dramatically lower than IHFW due to its simplicity. Table 2 shows the average and maximum running time for IHFW and k-Means. The average values have been computed over all feature sizes.

5 Concluding Remarks

In this paper, we have proposed a feature selection method that benefits from the advantages of both filter and wrapper methods. The method is conceptually based on the expectation maximization algorithm. It uses a greedy strategy for locally selecting the most competent set of features from the feature space. The method is advantageous over filter methods since it uses an iterative EM based feature selection strategy, which is *more likely* to reach the globally optimum feature set. In addition, it augments the capability of wrapper methods by allowing them to be used in the document clustering field where class labels are not available. The proposed method has been evaluated on various data sets whose results show promising improvement in terms of accuracy and normalized mutual information compared with several existing methods.

For the future, other clustering algorithms such as Bisecting k-Means and Spectral Clustering can be incorporated to the algorithm and experiments conducted. An interesting study regarding clustering algorithms is the study of dynamic clustering algorithms such as X-means in which the number of clusters

is not determined beforehand. Moreover, other feature ranking measures can be localized (as it was the case for DF in this paper) and applied to see how the algorithm can improve the performance of clustering.

References

1. Breiman, L.: Classification and Regression Trees. Chapman & Hall/CRC, Boca Raton (1998)
2. Chua, S., Kulathuramaiyer, N.: Semantic feature selection using wordnet. In: WI 2004: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, Washington, DC, USA, pp. 166–172. IEEE Computer Society, Los Alamitos (2004)
3. Dash, M., Choi, K., Scheuermann, P., Liu, H.: Feature selection for clustering - a filter solution. In: ICDM, pp. 115–122 (2002)
4. Dhillon, I., Kogan, J., Nicholas, C.: Feature Selection and Document Clustering, Survey of Text Mining: Clustering, Classification, and Retrieval (2004)
5. Forman, G.: An extensive empirical study of feature selection metrics for text classification. The Journal of Machine Learning Research 3, 1289–1305 (2003)
6. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. The Journal of Machine Learning Research 3, 1157–1182 (2003)
7. sam Han, E.h., Boley, D., Gini, M., Gross, R., Hastings, K., Karypis, G., Kumar, V., Mobasher, B., Moore, J.: Webace: a web agent for document categorization and exploration. In: Proc. of the 2nd International Conference on Autonomous Agents, pp. 408–415. ACM Press, New York (1998)
8. Jain, A., Zongker, D.: Feature selection: evaluation, application, and small sampleperformance. IEEE Transactions on Pattern Analysis and Machine Intelligence 19(2), 153–158 (1997)
9. Jolliffe, I.T.: Principal component analysis. Springer, New York (2002)
10. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artificial Intelligence 97(1-2), 273–324 (1997)
11. Liu, T., Liu, S., Chen, Z., Ma, W.-Y.: An evaluation on feature selection for text clustering. In: ICML, pp. 488–495 (2003)
12. Strehl, A., Ghosh, J.: Cluster Ensembles-A Knowledge Reuse Framework for Combining Partitionings. In: Proceedings of the National Conference on Artificial Intelligence, pp. 93–99. AAAI Press, MIT Press, Menlo Park, Cambridge (1999) (2002)
13. Wolf, L., Shashua, A.: Feature selection for unsupervised and supervised inference: The emergence of sparsity in a weight-based approach. J. Mach. Learn. Res. 6, 1855–1887 (2005)
14. Yang, Y.: Noise reduction in a statistical approach to text categorization. In: Proceedings of the 18th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1995), pp. 256–263. ACM Press, New York (1995)
15. Zhao, Z., Liu, H.: Spectral feature selection for supervised and unsupervised learning. In: ICML, pp. 1151–1157 (2007)