International Journal of Automation and Computing 15(3), June 2018, 290-298
DOI: 10.1007/s11633-015-0912-z

A Novel Active Learning Method Using SVM for Text Classification

Mohamed Goudjil[1] Mouloud Koudil[1] Mouldi Bedda[2] Noureddine Ghoggali[3]

[1]École nationale Supérieure d'Informatique (ESI), Oued Smar, Algiers, Algeria

[2]AL Jouf University, Sakaka, Kingdom of Saudi Arabia

[3]LAAAS laboratory, Faculté de Technologie, Université Batna 2, Fesdis, Algeria

Abstract: Support vector machines (SVMs) are a popular class of supervised learning algorithms, and are particularly applicable to large and high-dimensional classification problems. Like most machine learning methods for data classification and information retrieval, they require manually labeled data samples in the training stage. However, manual labeling is a time consuming and error-prone task. One possible solution to this issue is to exploit the large number of unlabeled samples that are easily accessible via the internet. This paper presents a novel active learning method for text categorization. The main objective of active learning is to reduce the labeling effort, without compromising the accuracy of classification, by intelligently selecting which samples should be labeled. The proposed method selects a batch of informative samples using the posterior probabilities provided by a set of multi-class SVM classifiers, and these samples are then manually labeled by an expert. Experimental results indicate that the proposed active learning method significantly reduces the labeling effort, while simultaneously enhancing the classification accuracy.

Keywords: Text categorization, active learning, support vector machine (SVM), pool-based active learning, pairwise coupling.

1 Introduction

Rapid technological advances in the speed and capacity of computers and networks have led to an enormous increase in the number and availability of text documents. Hence, there is a need to ensure that this textual information can be easily accessed and extracted by users. Traditional methods of manually classifying textual documents are time-consuming, costly, and increasingly impractical given the amount of data involved. In recent years, the machine learning paradigm has received widespread research interest in the field of text categorization, to the point that it is now possible to classify text documents automatically[1]. That is, if some documents are selected at random and classified by an expert, then this training set can be used to reproduce the labels for the whole collection via a supervised learning algorithm. However, we still need to manually classify a set of documents to output the model, and therefore hope this set of necessary documents is relatively small. This motivates us to develop an approach that, instead of blindly selecting documents at random, can guide the selection such that we need only to label a minimum number of documents before a particular level of classification accuracy is achieved. This is the real problem that active learning aims to solve[2].

Lewis and Gale introduced pool-based active learning scheme for classification[3]. In their scenario, the learner selects some samples from a set of unlabeled data (the "pool"), after which a human expert assigns the true labels of the selected samples. The labeled samples are used by the classifier to update itself, and the whole process is repeated iteratively. In this method, the main challenge is to find an appropriate strategy for selecting the most informative samples from the pool. This scenario has been widely used in domains such as remote sensing images classification[4], music annotation[5], and text categorization[6].

Several techniques based on statistical learning have been applied for the different steps of automatic text categorization, including the feature selection methods using naïve Bayes theorem[7], Ward's minimum variance measure[8], and classification methods such as support vector machines (SVMs)[9], k-nearest neighbor approaches[10], neural networks[11], generalized instance sets[12, 13] and Bayesian classifiers[14]. Empirical studies[15, 16] have shown that SVMs are one of the most effective of these methods. SVMs use a more efficient technique for model training, particularly when the training set is small and imbalanced. This motivates us to choose the SVM classifier for active learning in text categorization.

In the active learning scenario, a single sample with the highest classification uncertainty is selected for manual labeling in each iteration, and the classification model is retrained with this labeled sample until a reasonably good classification of the unlabeled data is achieved.

Some previous work in active learning has considered multiple-class SVMs. Because a sample that is informative for a binary SVM may be useless for a multi-class SVM, it has been argued[17] that a probabilistic model represents a

Research Article
Manuscript received October 14, 2014; accepted June 3, 2015; published online July 25, 2016
Recommended by Editor-in-Chief Huo-Sheng Hu

natural solution for multi-class problems (as long as there is a probability estimation for the output from a multi-class SVM). In this paper, we expand a previous active learning approach for multi-class SVMs[18], and propose an active learning method for a set of multi-class SVMs. Using a posterior probability model, we calculate the average probability given by a set of classifiers. We then label only those samples that have an average probability that is below some threshold. The classification accuracy of an SVM trained by our method on the selected samples is then compared with an SVM trained using all available data.

The rest of this paper is organized as follows. Section 2 discusses some related work, and Section 3 introduces the basic concepts of active learning and SVMs. In Section 4, we describe the proposed method, and present experimental results in Section 5. Finally, we conclude this paper in Section 6 by summarizing our findings.

2 Related work

The issue of reducing the amount of labeled samples required for classification has been the subject of various research. An algorithm combining linear kernel SVMs with the notion of a version space has been reported[6]. This approach queries the points that divide the current version space into two equal parts at each step, as these are more likely to be the actual support vectors. Three methods of approximating the above procedure were presented, with the simplest among them querying the point closest to the current hyperplane. Compared to random sampling, this approach reduces the number of points required in text categorization experiments. The above method was extended[19] to include a degree of confidence that measures the closeness of the current SVM to the optimal SVM, because the greedy search algorithm used in [6] was not thought to be powerful enough. Thus, the likelihood of a particular point being a support vector is estimated using a combination of two factors: the distance of the point from the current hyperplane and the degree of confidence. If the confidence factor is low, a random sampling factor is used. This method outperformed the "simple" method of [6] in a set of experiments. In [20], the expected error after adding a new example is estimated using class probabilities, with the new support vector defined by the sample that minimizes this error. The class probabilities are computed using logistic regression. However, as noted in [19], this method was developed for querying single points.

The approach described in [21] queries samples that are close to the current separating hyperplane and form large angles with previously selected candidates. This "combined" method was based on the "simple" method of [6], with the trade-off between the two controlled by a new parameter λ. Although the "combined" selection strategy is more robust than the "simple" method for several datasets, it is not easy to find the optimal value of λ[21]. The dynamic selection of one algorithm from a set of four was proposed[22] on the basis that, for several datasets, there was no single best active learning strategy.

Most SVM active learning algorithms select samples based on their proximity to the decision boundary. In contrast, Roy and McCallum[23] used a probabilistic model to select those samples that maximize the posterior entropy on the unlabeled dataset. Though this method was initially applied with naïve Bayes classifiers, it may still be applied using SVMs. Using a similar probabilistic approach, Mitra et al.[19] estimated a new confidence factor from local information using the k-nearest neighbor principle. This adaptive confidence factor was used with the current separating hyperplane to determine the candidate set of points to be queried. This makes the algorithm robust and ensures an efficient learning performance. Luo et al.[17] extended the active learning approach to multi-class SVMs, developing a suitable probabilistic model and querying the samples with the least classification confidence. Their system was used to recognize multiple types of plankton.

The most relevant research on text categorization is that reported in [3, 6, 24, 25]. SVMs were first applied to active learning using the notion of a version space in [6]. And in a recent study, Goudjil et al.[26, 27] presented a text categorization technique that selects a batch of documents in each learning iteration. The active learning approach proposed in that paper employs SVM to select a set of documents, the method has been applied on three datasets, two English datasets and one in Arabic. The proposed method in that study will be used as the baseline approach in our paper, and we refer to it by "SVM-AL".

3 Active learning approach with multi-class SVMs

Active learning[2] is a generic term describing an interactive, iterative process that builds high-performance classifiers with little labeled data. Unlike in passive learning, where the learning algorithm is presented with a static set of labeled samples that are then used to construct a model, the active learning paradigm requires the learning algorithm to choose the data from which it learns by selecting the samples which appear to be the most informative. Active learning is widely used in situations where vast amounts of unlabeled data are available.

3.1 Support vector machines

SVM classifiers are supervised learning models that attempt to find the optimal hyperplane separating two different classes of data (in our case, documents) that will generate the best model for future data. The SVM method was introduced by Vapnik[28], and has demonstrated very high accuracy for pattern recognition and text categorization[15].

For simplicity, let us assume that the training set consists of N vectors $x_i\,(i = 1, 2, \cdots, N)$ from the n-dimensional feature space $X \in \mathbf{R}^n$[29]. Each vector x_i has an associ-

ated target $y_i \in \{-1, +1\}$. The linear SVM classification approach searches for a boundary between the two classes in X by means of a hyperplane. In the nonlinear case, data are first mapped to a higher-dimensional feature space using a kernel function, i.e., $\Phi(X) \in \mathbf{R}^{n''}$. The membership decision is based on $\text{sgn}(f(x))$, where $f(x)$ represents the discriminant function associated with the hyperplane in the transformed space. This function is defined as

$$f(x) = w\Phi(x) + b. \tag{1}$$

There are many hyperplanes that can separate the classes, but only one (the optimal hyperplane) maximizes the distance between the hyperplane and the closest point. The optimal hyperplane defined by the weight vector $w = w^* \in \mathbf{R}^n$ and the bias $b = b^* \in \mathbf{R}$ is the one that minimizes a cost function that expresses a combination of two criteria: margin maximization and empirical risk minimization. When adopting a one-norm measure of the empirical errors, the SVM cost function is defined as

$$\Psi(w, \xi) = \frac{1}{2}w^2 + c\sum_{i=1}^{N}\xi_i. \tag{2}$$

3.2 Using probabilistic output

Due to their theoretical advantages and empirical success, SVMs is considered as an attractive method to use in active learning. To this end, we need to use a probabilistic output in the querying strategy to indicate which of the unlabeled samples will be most beneficial.

SVMs are mainly used to solve binary classification problems (those with only two known classes). However, we are considering a multi-class problem (i.e., more than two classes). Certain technical complexities conclude that using a single SVM to solve multi-class problems should be avoided. A better approach is to use a combination of multiple binary SVM classifiers.

The SVM approach can be extended to multi-class classification problems using three well-known methods:

1) One-against-all using a winner-takes-all strategy.

2) One-against-one implemented by max-wins voting.

3) Error-correcting codes.

Hastie and Tibshirani[30] used the binary SVM outputs to estimate the posterior probabilities

$$p_i = Prob(\omega_i|x);\ i = 1, \cdots, M$$

(as SVMs are discriminant classifiers, they do not naturally admit posterior probabilities). These probabilities were then used to implement a multi-class SVM classifier based on a pairwise coupling strategy. The pairwise coupling strategy assigns the sample under consideration to the class with the largest p_i[31]. Wu et al.[32] proposed two new pairwise coupling schemes for the estimation of class probabilities, and Duan and Keerthi[31] recommended the use of one of the pairwise coupling schemes in [30, 32] as the best kernel discriminant method for solving multi-class problems.

In the context of this work, we use the LIBSVM software[33] based on the pairwise coupling schemes in [32].

3.3 Active learning

In general, an active learner can be represented by the following parameters[34]:

1) C: a supervised classifier,

2) Q: a query function used to select the most informative unlabeled samples from a pool,

3) S: a supervisor who can assign the true class label to any unlabeled sample of U,

4) T: a labeled training set,

5) U: a pool of unlabeled samples.

The classifier C is first applied to the labeled training set T, and then it considers the pool of unlabeled samples U. Next, a query function Q is used to select the set of most informative samples from U, and a supervisor S is queried to assign their true class label. Active learning is an iterative process, so newly labeled samples are included in the training set T, and the classifier C is retrained using the updated training set. The querying and retraining operations are repeated for some predefined number of iterations, or until a stop criterion is satisfied[35].

Algorithm 1 describes the general active learning process.

Algorithm 1. Active learning procedure

1) Select a set of unlabeled samples from the pool (small set of random samples), and assign a class label to each sample. This set is the initial training set T.

2) Train the classifier C with the initial training set T constructed in the first step.

Repeat

3) Query a set of samples from the pool U using query function Q.

4) Supervisor S assigns a class label to each of the queried samples.

5) Add the newly labeled samples to the training set T.

6) Retrain the classifier.

Until stopping criteria is satisfied.

T should be as small as possible while still permitting the classifier to be well trained. The pool U should contain multiple samples, but must also represent all classes. A good active learning algorithm would be insensitive to the number of unlabeled samples[36].

4 Proposed active learning method

This section describes the different steps of SVM-based active learning methods. The main objective of the proposed method is to minimize the number of labeled samples without affecting the classification performance. This will produce the following advantages:

1) a reduction in the cost of sample labeling,

2) the acceleration of the classifier training process.

The main issue is to achieve an acceptable accuracy with consistent training and a tolerable labeling cost. If we use

too many labeled samples, we will achieve high training consistency, but an unacceptable cost, and vice versa.

4.1 Active learning using support vector machines

Goudjil et al.[26, 27] concentrated on the estimated probability for the active learning process.

The process of sample selection from the pool is performed sequentially using the labeled samples from the previous packet, and this procedure is repeated until all packets in the pool have been processed. This selection strategy is based on the SVM posterior probability. Goudjil et al.[26, 27] proposed a threshold to measure how informative each unlabeled sample in the pool is.

4.2 AL-SVM using multiple classifiers

AL-SVM provides a good balance between classification accuracy and the number of labeled samples[26], but cannot achieve optimum accuracy. Thus, we employ a set of SVM classifiers to select the most informative samples in each active learning iteration. This method improves the confidence of the multi-class classification.

In this method, the pool of unlabeled data is divided into packets of equal size and executed in sequence. The algorithm selects a number of samples from the packet using a predefined criterion. The selected samples are then labeled by an expert and added to the training set, while all the other unlabeled data are discarded. This technique enhances the accuracy of the classifier with each packet, and terminates the learning process with the last packet, which is a good stopping criterion. In every iteration of the learning process, the optimal kernel parameters of the SVM classifier are estimated using a cross-validation method.

The selection strategy in the proposed multi-classifier AL-SVM (AL-MSVM) is based on the average of the posterior probabilities estimated by a set of classifiers for each sample. Thus, the most informative samples are those with an average probability that is less than the threshold *tsh*. These samples are then labeled by an expert, and added to the training set of each classifier.

Algorithm 2. AL-MSVM

1) Start with a stream of packets of unlabeled data and an initial training set for each classifier.

Repeat

2) Estimate the best parameters for each classifier using a cross-validation method.

3) Apply each of the classifiers with their optimal parameters to the current packet. This will provide posterior probabilities for the packet samples for each classifier.

4) Calculate the average posterior probability for each sample.

5) Select samples with an average probability below the threshold tsh as informative samples to be labeled.

6) Present the selected samples to the expert for labeling.

7) Add the labeled samples to the training set of each classifier.

Until the last packet

5 Experiments and results

In this section, we evaluate the effectiveness of our proposed method of selecting training samples. Experiments were conducted using three text datasets, and the performance of the proposed method was compared to SVM method trained using whole dataset, we refer to this latter as naïve approach or simply "SVM". To stay compatible and comparable with previous works, we use the same experimental settings in [26], details are in next subsections.

5.1 Dataset description

The validation of the proposed method was conducted on the basis of three datasets in the field of text categorization (TC). These benchmarks[37] were downloaded from a publicly available repository of datasets for single-label text categorization[1]. Further details on these datasets are available on the website and in [38].

We used the following three class distributions for text categorization tasks:

R8: The documents in Reuters-21578 appeared on the Reuters newswire in 1987, and were manually classified by personnel from Reuters Ltd. For this dataset, we used the r8-train-stemmed and r8-test-stemmed files[38].

20ng: The 20ng dataset is a collection of approximately 20,000 newsgroup documents, partitioned (almost) evenly across 20 different newsgroups. For this dataset, we used the 20ng-train-stemmed and 20ng-test-stemmed files[38].

WebKB: The WebKB collection contains webpages from computer science departments collected by the World Wide Knowledge Base (WebKB) project of the CMU text learning group in 1997. For each of the different classes, the collection contains pages from four universities (Cornell, Texas, Washington, and Wisconsin), as well as miscellaneous pages collected from other universities. For this dataset, we used the WebKB-train-stemmed and WebKB-test-stemmed files[38].

To represent the articles, we adopted the term frequency–inverse document frequency (TFIDF) weighting method. This statistical measure is used to evaluate the importance of a word within an article in a dataset or corpus[39, 40].

5.2 Dataset preprocessing

The document representation is known to influence the quality of the classification results. The main aim of preprocessing the data is to reduce the problem of dimensionality by controlling the size of the system′s vocabulary.

In the proposed method, each class with fewer than 200 samples is omitted (details of the updated datasets are given in Table 1). The documents are then represented in a vector

[1] Available at http://web.ist.utl.pt/∼acardoso/datasets/

model using TFIDF.

In some situations, preprocessing the dataset can also unify the data in such a way as to improve the classification performance. With this in mind, we adopt a histogram feature extraction method by discarding every word that is not contained in at least 1% of the documents. Table 2 lists the original features as well as those remaining after preprocessing.

5.3 Experiments

In this initial phase of processing, the dataset is divided into an initial training set (Tr), a test samples set (Ts), and an unlabeled pool of samples (U) further divided into several packets of equal size.

A selection of samples to be labeled is taken from a packet, and their labels are determined by an expert. These samples are then added to the training set. This process continues until all the packets in a pool are exhausted. The selection strategy is based upon the SVM posterior probability, with the threshold tsh used to define the informativeness of each unlabeled sample in a pool. To determine a suitable value for tsh and the ideal size of Tr, we executed the active learning process for several thresholds with different training set sizes, and examined the resulting accuracy with the test set.

5.3.1 Preparing the training experiment

Before conducting the classification experiments, the datasets were randomly divided into six training sets of different sizes (10, 20, 25, 50, 75 and 100) for each class. The pool was divided into packets containing 200 samples. AL-SVM was applied to all training sets using different threshold values. The upper and lower accuracy levels were found by applying AL-SVM with $thr = 100\%$ and 0%, respectively.

Table 3 lists the classification accuracy obtained by the AL-SVM method using different training set sizes and threshold values.

Table 4 presents the number of samples labeled using the same initial training set size and different threshold values. The main objective of this experiment was to choose the best combination of threshold value and training set size. This minimizes the number of labeled samples and maximizes the accuracy. For this reason, the minimum size of the initial training set was chosen to be 20 samples per class with a 70% threshold. This provides fairly accurate results. These parameters were used in the other training experiments.

A threshold of 70% does not mean that we select 70% of the pool samples, rather it means that samples with a probability of less than 70% will be selected for labeling (which represents 10% of samples in our case).

5.3.2 Results from AL-SVM

The dataset was divided to give a training dataset with 20 samples per class. The poo1 was divided into packets containing 200 samples. AL-SVM was executed several times with different initial training sets and the same threshold (70%).

From Table 5, we can see that good results in terms of accuracy are obtained with low training set sizes. For example, AL-SVM achieves a classification accuracy of 95% with the R8 dataset.

Note that AL-SVM provides an accuracy that is close to the upper level using smaller training set sizes. Table 5 indicates that the number of labeled samples in the training set was reduced from 3 779 to 382 (reduction of 90%) with a loss of only 1.5% accuracy. Examining the other two datasets, we find that the number of labeled training samples can be reduced by 41% (20ng) and 56.54% (WebKB) with a loss of accuracy of only 1.5% and 5%, respectively.

Table 1 Preprocessed datasets

Dataset	Classes	Total number of documents	Smallest class	Largest class
R8	6	7 479	271	3 923
20ng	20	16 841	251	999
WebKB	4	4 199	504	1 641

Table 2 Original and remaining features of the datasets

Dataset	Original features	Remaining features	Gain
R8	4 982	2 031	59.23%
20ng	2 4040	7 971	66.84%
WebKB	4 856	2 280	53.05%

Table 3 Accuracy obtained by AL-SVM for R8 using different training set sizes and thresholds

		Number of samples per class					
		10	20	25	50	75	100
Threshold $< N\%$	10	83.92%	85.92%	86.16%	88.04%	91.92%	93.2%
	30	94.04%	95.16%	95.72%	95.92%	95.56%	95.24%
	50	94.56%	95.32%	95.64%	96.88%	97.36%	97.28%
	70	95.16%	95.52%	95.96%	97%	97.24%	97.8%
	90	96.08%	96.16%	96.28%	96.96%	97.4%	97.6%
	100	96.84%	96.88%	96.96%	97.52%	97.92%	97.88%

Table 4 Number of samples labeled by AL-SVM for R8 using different training set sizes and thresholds

		Number of samples per class					
		10	20	25	50	75	100
Threshold $< N\%$	10	0	0	0	0	0	0
	30	110	59	59	27	15	15
	50	279	193	189	147	118	109
	70	460	378	359	279	225	238
	90	782	699	685	555	500	489
	100	3 779	3 779	3 779	3 779	3 779	3 779

These results are very encouraging. However, we expect to obtain better results by applying the new AL-MSVM method.

5.3.3 Results from AL-MSVM

To apply AL-MSVM, the dataset was divided into five training sets with 20 samples per class, and the pool was divided into packets of 200 samples. We applied the AL-MSVM approach with a threshold of 70%.

Figs. 1 to 3 show the experimental results given by AL-SVM and AL-MSVM for each dataset. The results are compared to those of an SVM classifier trained on all available data.

Table 6 lists the improvements in accuracy obtained using AL-MSVM. The proposed method enhances the accuracy given by SVM for all datasets. For example, for the R8 dataset, AL-MSVM improves the classification accuracy by 0.36% using just 10.77% of the available samples, whereas for the WebKB dataset, the accuracy was enhanced by 1.27% with only 48.4% of the labeled pool samples. In the case of the 20ng dataset, AL-MSVM increased the accuracy by 2.4% using only 61.5% of the pool samples.

5.3.4 Discussion

In order to examine the detailed performance of methods, we evaluate the accuracy of each data set by varying the number of packets for each of the algorithms compared with other as shown in the results of Figs. 1 to 3, respectively.

The result shows that two curves in Figs. 1 to 3 are rising continuously from the first packet to the last one. It means that the selected samples are really informative and it gives more accuracy.

The performance of algorithms provides more accuracy if the number of packets increases and it becomes closer to SVM accuracy when using the latter packets. If we compare our proposed algorithm with other algorithms, we found that AL-MSVM performance is better than the baseline active learning algorithms AL-SVM. The result also shows that the last packets of AL-MSVM provides consistently better performance than SVM in all datasets.

Table 7 illustrates a comparison of SVM with the profit ratio by applying the algorithm of AL-SVM and AL-MSVM in terms of accuracy and labeled samples. It shows that results of both methods are very close to SVM accuracy but there is a difference between the two methods. The AL-SVM shows a better accuracy for dataset R8. It is closer to the "Upper" with a percentage ratio of -1.3 and this accuracy is achieved by using 10.11% samples. The accuracy of AL-SVM for other datasets of 20ng and WebKB has a minor difference from "Upper" with a ratio of 1.64% and 5.09% respectively. The same difference of AL-SVM for 20ng and WebKB has been observed in a number of labeled samples with a ratio of 59.19% and 43.45%, respectively.

The proposed method AL-MSVM provides a better result in accuracy than the "Upper" for all datasets. The accuracy of AL-MSVM for R8, exceeds the AL-SVM with a ratio of 1.7 % and this accuracy is gained by using 10.77% samples. The accuracy of AL-MSVM for 20ng is better than AL-SVM with a ratio of 4.13% . This accuracy is achieved by using 61.31% samples which means that the dataset 20ng provides a superior accuracy as compared to other datasets. The result shows that accuracy of AL-MSVM for dataset WebKB is better than AL-SVM with a ratio of 6.36%. This accuracy is achieved by using 48.4% samples.

These results might be explained by the nature of the used datasets, compared to the R8 dataset, the two datasets 20ng & WebKB are web related data collections. These datasets are using more features (unique words) as compared to R8. Tables 1 and 2 show that R8 contains 7 478 documents with 4 982 features which has average ratio of 1.5 while 20ng and WebKB has average ratio of 0.86 and 0.7 respectively. It shows that the feature number in 20ng and WebKB is about double as compared to R8. We do believe that this characteristic in web related dataset makes the classification more challenging than other text documents. In fact, it is known that to classify documents with more features we need more labeled samples in training. It is also this characteristic that makes active learning methods more beneficial for text classification for the web related datasets as compare to the other dataset. It permits additional features to SVM which help to find more hyperplanes separating the classes.

6 Conclusions

In this study, we developed a novel active learning method for text classification that selects a batch of informative samples for manual labeling by an expert. The proposed AL-MSVM is based on the posterior probability estimated by a set of SVM classifiers. Extensive experiments were performed on three well-known real text categorization datasets. To empirically assess the effectiveness of the proposed method, we compared it with the results given by an SVM classifier applied to the whole dataset. This comparison demonstrates that the proposed AL-MSVM method increases the classification accuracy and retains very good stability with all datasets.

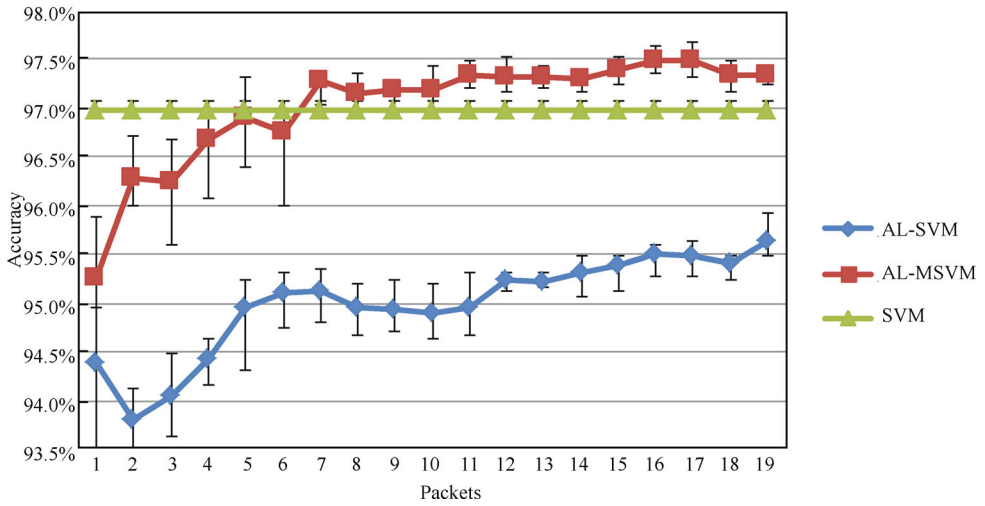


Fig. 1 Accuracy of AL-SVM, AL-MSVM, and naïve SVM classifier trained on all available data with R8 (20 initial training samples per class)

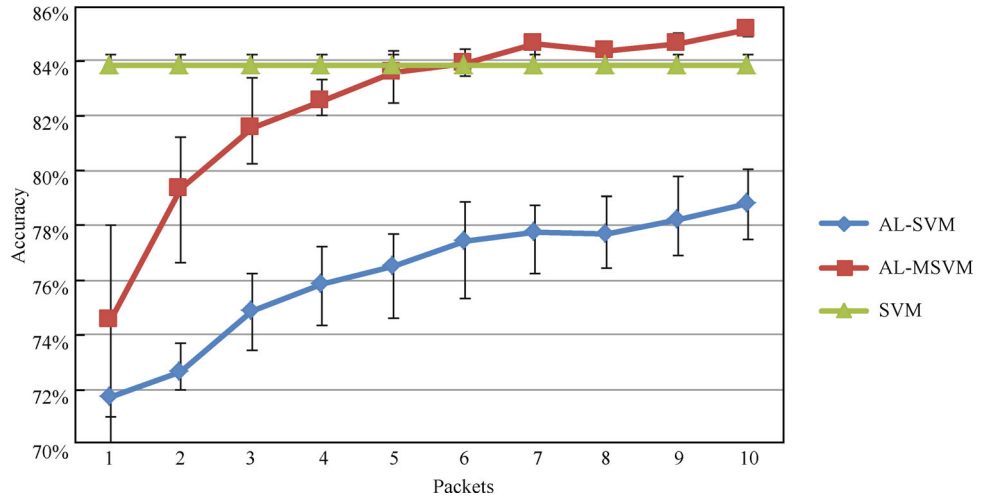


Fig. 2 Accuracy of AL-SVM, AL-MSVM, and nai SVM classifier trained on all available data with WebKB (20 initial training samples per class)

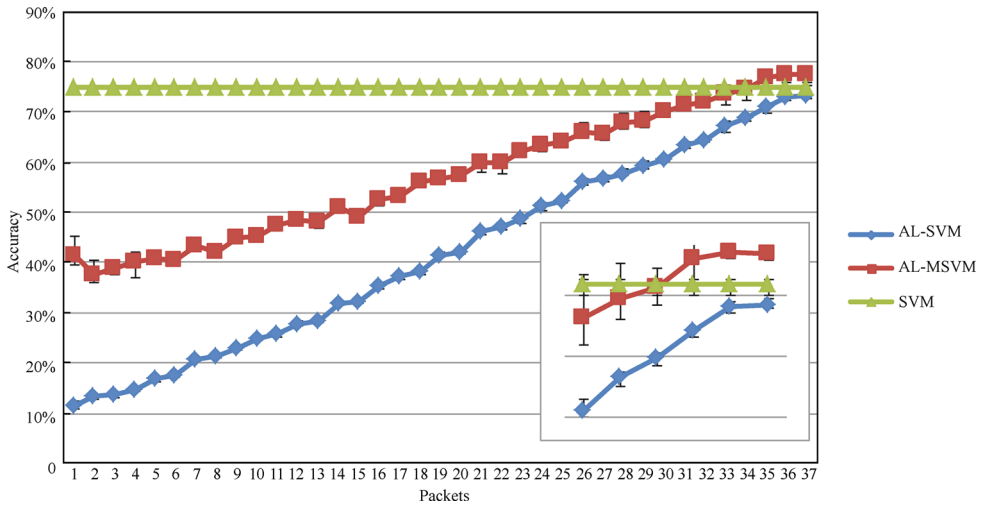


Fig. 3 Accuracy of AL-SVM, AL-MSVM, and naive SVM classifier trained on all available data with 20ng (20 initial training samples per class).

Table 5 Accuracy obtained by the AL-SVM approach for the different datasets

Dataset	Lower	Upper	AL-SVM	Labeled	Data size
R8	83.33%	96.98%	95.64%	382	3 779
20ng	43.79%	74.87%	73.23%	434 5	7 341
WebKB	53.07%	83.86%	78.77%	869	2 000

Table 6 Accuracy obtained by the AL-MSVM approach for the different datasets

Dataset	Lower	Upper	AL-MSVM	Labeled	Data size
R8	83.33%	96.98%	97.34%	407	3 779
20ng	43.79%	74.87%	77.36%	4 503	7 341
WebKB	53.07%	83.86%	85.13%	968	2 000

Table 7 Profit ratio by applying AL-SVM and AL-MSVM in terms of accuracy and labeled samples

Dataset	SVM	Profit in accuracy		Profit in number of labeled samples	
		AL-SVM	AL-MSVM	AL-SVM	AL-MSVM
R8	83.33%	−1.344	0.36%	10.11%	10.77%
20ng	43.79%	−1.64	2.49%	59.19%	61.34%
WebKB	53.07%	−5.089 36	1.27%	43.45%	48.40%

References

[1] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, vol. 34, no. 1, pp. 1–47, 2002.

[2] B. Settles. Active Learning Literature Survey. Computer Sciences Technical Report, 1648, University of Wisconsinadison, USA, 2010.

[3] D. D. Lewis, W. A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Springer-Verlag, New York, USA, 1994.

[4] C. Persello, L. Bruzzone. Active and semisupervised learning for the classification of remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 11, pp. 6937–6956, 2014.

[5] G. Chen, T. J. Wang, L. Y. Gong, P. Herrera. Multi-class support vector machine active learning for music annotation. *International Journal of Innovative Computing, Information and Control*, vol. 6, no. 3, pp. 921–930, 2010.

[6] S. Tong, D. Koller. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, vol. 2, pp. 45–66, 2002.

[7] S. A. A. Balamurugan, R. Rajaram. Effective and efficient feature selection for large-scale data using Bayestheorem. *International Journal of Automation and Computing*, vol. 6, no. 1, pp. 62–71, 2009.

[8] J. A. Mangai, V. S. Kumar, S. A. alias Balamurugan. A novel feature selection framework for automatic web page classification. *International Journal of Automation and Computing*, vol. 9, no. 4, pp. 442–448, 2012.

[9] I. Hmeidi, B. Hawashin, E. El-Qawasmeh. Performance of KNN and SVM classifiers on full word Arabic articles. *Advanced Engineering Informatics*, vol. 22, no. 1, pp. 106–111, 2008.

[10] B. Trstenjak, S. Mikac, D. Donko. KNN with TF-IDF based framework for text categorization. *Procedia Engineering*, vol. 69, pp. 1356–1364, 2014.

[11] S. Gazzah, N. E. B. Amara. Neural networks and support vector machines classifiers for writer identification using arabic script. *The International Arab Journal of Information Technology*, vol. 5, no. 1, pp. 92–101, 2008.

[12] W. Lam, Y. Q. Han. Automatic textual document categorization based on generalized instance sets and a metamodel. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 628–633, 2003.

[13] Q. Shen, R. Jensen. Rough sets, their extensions and applications. *International Journal of Automation and Computing*, vol. 4, no. 3, pp. 217–228, 2007.

[14] L. Messikh, M. Bedda, N. Doghmane. Binary phoneme classification using fixed and adaptive segment-based neural networkapproach. *The International Arab Journal of Information Technology*, vol. 8, no. 1, pp. 48–51, 2011.

[15] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning Chemnitz*, Springer, Chemnitz, Germany, pp. 137–142, 1998.

[16] Y. M. Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval*, vol. 1, no. 1–2, pp. 69–90, 1999.

[17] T. Luo, K. Kramer, S. Samson, A. Remsen, D. B. Goldgof, L. O. Hall, T. Hopkins. Active learning to recognize multiple types of plankton. In *Proceedings of the 17th International Conference on Pattern Recognition*, IEEE, Cambridge, USA, vol. 3, pp. 478–481, 2004.

[18] M. Goudjil, M. Koudil, N. Hammami, M. Bedda, M. Alruily. Arabic text categorization using SVM active learning technique: An overview. In *Proceedings of World Congress on Computer and Information Technology*, IEEE, Sousse, Tunisia, 2013.

[19] P. Mitra, C. A. Murthy, S. K. Pal. A probabilistic active support vector learning algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 3, pp. 413–418, 2004.

[20] G. Schohn, D. Cohn. Less is more: Active learning with support vector machines. In *Proceedings of the 17th International Conference on Machine Learning*, Morgan Kaufmann Publishers Inc., San Francisco, USA, pp. 839–846, 2000.

[21] K. Brinker. Incorporating diversity in active learning with support vector machines. In *Proceedings of the 20th International Conference on Machine Learning*, ACM, Washington, USA, pp. 59–66, 2003.

[22] Y. Baram, R. El-Yaniv, K. Luz. Online choice of active learning algorithms. *Journal of Machine Learning Research*, vol. 5, pp. 255–291, 2004.

[23] N. Roy, A. McCallum. Toward optimal active learning through monte carlo estimation of error reduction. In *Proceedings of the 18th International Conference on Machine Learning*, Bellevue, USA, pp. 441–448, 2001.

[24] A. K. McCallumzy, K. Nigamy. Employing EM and pool-based active learning for text classification. In *Proceedings of the 15th International Conference on Machine Learning*, Madison, USA, pp. 350–358, 1998.

[25] S. C. H. Hoi, R. Jin, M. R. Lyu. Large-scale text categorization by batch mode active learning. In *Proceedings of the 15th International Conference on World Wide Web*, ACM, New York, USA, pp. 633-642, 2006.

[26] M. Goudjil, M. Bedda, M. Koudil, N. Ghoggali. Using active learning in text classification of quranic sciences. In *Proceedings of International Conference on Advances in Information Technology for the Holy Quran and its Science*, Taibah University, Madinah, Saudi Arabia, pp. 209–213, 2013.

[27] M. Goudjil. Text Categorization using reduced training set. *Research Journal of Applied Sciences, Engineering and Technology*. vol. 10, no. 12, pp. 1363–1369, 2015.

[28] V. N. Vapnik. *Statistical Learning Theory*, New York, USA: Wiley, 1998.

[29] N. Ghoggali, F. Melgani, Y. Bazi. A multiobjective genetic SVM approach for classification problems with limited training samples. *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 6, pp. 1707–1718, 2009.

[30] T. Hastie, R. Tibshirani. Classification by pairwise coupling. *The Annals of Statistics*, vol. 26, no. 2, pp. 451–471, 1998.

[31] K. B. Duan, S. S. Keerthi. Which is the best multiclass SVM method? An empirical study. In *Proceedings of the 6th International Workshop, MCS 2005*, California, USA, pp. 278–285, 2005.

[32] T. F. Wu, C. J. Lin, R. C. Weng. Probability estimates for multi-class classification by pairwise coupling. *Journal of Machine Learning Research*, vol. 5, pp. 975–1005, 2003.

[33] C. C. Chang, C. J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, Article number 27, 2011.

[34] M. K. Li, I. K. Sethi. Confidence-based active learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 8, pp. 1251–1261, 2006.

[35] B. Demir, C. Persello, L. Bruzzone. Batch-mode active-learning methods for the interactive classification of remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 3, pp. 1014–1031, 2011.

[36] M. Sassano. An empirical study of active learning with support vector machines for Japanese word segmentation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, Association for Computational Linguistics, Stroudsburg, USA, pp.505-512, 2002.

[37] S. C. H, Hoi, R. Jin, M. R. Lyu. Batch mode active learning with applications to text categorization and image retrieval. *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1233–1248, 2009.

[38] A. Cardoso-Cachopo, A. L. Oliveira. Semi-supervised single-label text categorization using centroid-based classifiers. In *Proceedings of the ACM Symposium on Applied Computing*, ACM, Seoul, Korea, pp. 844-851, 2007.

[39] K. S. Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, vol. 60, no. 5, pp. 493–502, 2004.

[40] G. Salton, C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, vol. 24, no. 5, pp. 513–523, 1988.

Mohamed Goudjil received the M. Sc. degree in computer engineering from Boumerdes University, Algeria in 2008. He is currently a Ph. D. degree candidate in computer engineering at Ecole nationale Supérieure d'Informatique (ESI), Algeria. From 2005 to 2008, he was a researcher at Advanced Technologies & Resarchs Centre and a lecturer for seven years in different universities.

His research interests include text classification, arabic language processing and machine learning.

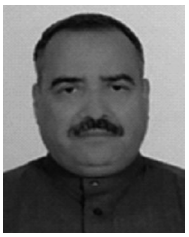
E-mail: m_goudjil@esi.dz (Corresponding author)

ORCID iD: 0000-0003-1712-7617

Mouloud Koudil received the Ph. D. degree in computer science from l'Ecole nationale Supérieure d'Informatique (ESI), Algeria in 2002. He is currently a full time professor and rector of the same institution.

His research interests include wireless sensor networks, networks on chips, and hardware/software codesign.

E-mail: m_koudil@esi.dz



Mouldi Bedda received the Ph. D. degree in electrical engineering from the University Nancy 2, France in 1985. From 1985 to 2006, he worked with the University Badji Mokhtar Annaba, Algeria. He was the director of Automatic and Signals Laboratory from 2001 to 2006. Since 2006, he is a full professor at the college of engineering of Al Jouf university KSA. He supervised several Ph. D. students in speech processing, biomedical signals, hand written recognition and image processing.

His research interests include speech processing, biomedical signals, hand written recognition and image processing.

E-mail: mouldi_bedda@yahoo.fr

Noureddine Ghoggail received the State Engineer degree in electronics from the University of Batna, Algeria in 2000, and the Ph. D. degree in information and communication technologies in Department of Information Engineering and Computer Science, University of Trento, Italy. He is currently an assistant professor at University of Batna in Algeria.

His research interests include pattern recognition and evolutionary computation methodologies for remote sensing image analysis.

E-mail: ghoggalinour@gmail.com