Research Journal of Applied Sciences, Engineering and Technology 10(12): 1363-1369, 2015
DOI: 10.19026/rjaset.10.1835
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: February 26, 2015 Accepted: March 25, 2015 Published: August 25, 2015

Research Article

Text Categorization using Reduced Training Set

[1]Mohamed Goudjil, [1]Mouloud Koudil, [2]Mouldi Bedda and [3]Noureddine Ghoggali
[1]Ecole Nationale Supérieure d'Informatique (ESI), Oued Smar, Algiers, Algeria
[2]AL JOUF University, Sakaka, Kingdom of Saudi Arabia
[3]University of Batna, Batna, Algeria

Abstract: The machine learning approaches to text categorization proceed by teaching the system how to classify through labeled samples. In real application scenarios, the collection of training (labeled) samples to design a classifier is not always trivial due to the complexity and the cost which characterize the process. A possible solution to this issue can be found in the exploitation of the large number of unlabeled samples which are accessible at zero cost from the web. Active learning strives to reduce the required labeling effort while retaining the accuracy by intelligently selecting the samples to be labeled. This Study presents a novel active learning method for text classification that selects a batch of informative samples for manual labeling by an expert. The proposed method uses the posterior probability output of a multi-class SVM method. The experiments are performed with two well-known datasets and the presented experimental results show that employing our active learning method can significantly reduce the need for labeled training data.

Keywords: Active learning, pairwise coupling, pool-based active learning, support vector machine, text classification

INTRODUCTION

The Automated Categorization of Text documents (ATC) into topical categories has a long history. In the past, the most effective approach to the problem seemed to be that of having human experts manually building automatic classifiers, which is known as knowledge-engineering techniques (Sebastiani, 2002). In such techniques, the expert knowledge is encoded in a set of manually defined rules which is usually considered a time-consuming task.

With the booming production and availability of online documents, there is a growing need for tools that automate the text management task, because the traditional techniques have become too expensive, or simply not feasible given the number of documents involved.

In order to solve this problem, one can resort to machine learning approaches, which are mainly divided into three well-consolidated categories: supervised, unsupervised and semi-supervised approaches. Based on these approaches, several techniques have been developed for the different ATC steps, such as the naïve Bayes classifier (Balamurugan and Rajaram, 2009), Maximum Entropy classification model (Fragos *et al*., 2014), support vector machines (Li and Chen, 2014), minimum variance measure (Mangai *et al*., 2012), k-nearest neighbors (Basu and Murthy, 2014), neural networks (Wang and Wang, 2014) and generalized instance sets (Lam and Han, 2003; Shen and Jensen, 2007).

The machine learning approaches to text categorization are done by teaching the system how to classify through labeled samples, which is called supervised learning, while unsupervised learning uses unlabeled samples. In real application scenarios, collecting training samples to design a supervised classifier is not always trivial. Unfortunately, manually generating collections of labeled examples is typically time-consuming and expensive since it usually involves experts. Accordingly, sometimes this constrains the supervised learning to be carried out with small numbers of training samples. This leads to weak estimates of the classifier parameters and high classification error rates, particularly if the class distributions are overlapped. A possible solution to this issue can be found in the exploitation of the large number of unlabeled samples which are accessible at zero cost from the web. Indeed, statistical intuition suggests that it is reasonable to expect to get stronger classifier parameter estimates and thus to improve the classifier accuracy by combining labeled (training) and unlabeled samples. However, the question is how such combination can be performed? Methods dealing with this issue are divided into two categories: semi-supervised and active learning methods. The main difference between the two methods is that semi-supervised learning uses a small amount of labeled

Corresponding Author: Mohamed Goudjil, Ecole Nationale Supérieure d'Informatique (ESI), Oued Smar, Algiers, Algeria

samples to classify the unlabeled ones, which will be added with their predicted labels to the training set and the procedure of training is repeated again. By contrast, in active learning the learner is able to interactively query an "expert" to obtain the labels of some unlabeled samples.

Active learning is well-motivated in many modern machine learning problems where data may be abundant but labels are rare or expensive to obtain. Among these problems we can find text categorization (Tong and Koller, 2002; Goudjil *et al*., 2013; Cai *et al*., 2014), remote sensing image classification (Ding *et al*., 2014; Persello and Bruzzone, 2014) and recommender systems (Elahi *et al*., 2014). Most active learning algorithms are conducted in an iterative fashion. In each iteration, the sample with the highest classification uncertainty is chosen for manual labeling and the classification model is retrained with the additional labeled samples. The two steps of training a classification model and soliciting a label are iterated many times.

Most active learning approaches, however, have focused on selecting only one unlabeled instance at a time, while retraining the classifier on each iteration. When the training process is hard or time-consuming, this repeated retraining is inefficient. Furthermore, if a parallel labeling system is available, a single instance selection system can make wasteful use of the resource. Thus, a batch mode active learning strategy that selects multiple instances each time is more appropriate under these circumstances. In fact, a new batch mode active learning approach has been proposed. The problem with such an approach is that the selected samples should be informative to the classification model and at the same time it should be diverse enough such that information provided by different samples does not overlap. In general, the key in batch mode active learning is to ensure little redundancy among the selected samples such that each one provides unique information for model updating.

In this Study, we propose a new active learning approach based upon SVM to classify text documents. Indeed, our novel approach uses a multi-class SVM method to make a decision about suitable samples to be labeled.

LITERATURE REVIEW ON ACTIVE LEARNING

Active learning (Settles, 2010) is a generic term describing a special, interactive and iterative learning process that can be used to build high performance classifiers with small amounts of labeled data. Unlike 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 means that the learning algorithm has the possibility 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 there are vast amounts of unlabeled data available.

The AL process: In general, an active learner can be represented by the following parameters (C, Q, S, T, U) (Li and Sethi, 2006), where:

- C is a supervised classifier.
- Q is a query function used to select the most informative unlabeled samples from a pool.
- S is a supervisor who can assign the true class label to any unlabeled sample of U.
- T is a labeled training set.
- U is a pool of unlabeled samples.

The first stage starts by training the classifier C on the labeled training set T and applies the classifier on the pool of unlabeled samples U. After that, a query function Q is used to select a set of samples-the most informative-from the pool U and a supervisor S is used to assign them the true class label. The Active Learner (AL) process is an iterative process, so the new labeled samples are included into the training set T and the classifier C is retrained using the updated training set. These operations of querying and retraining are repeated for some predefined iterations or until a stop criterion is satisfied (Demir *et al*., 2011).

Algorithm 1 gives a description of a general AL process.

Algorithm 1: AL procedure

1. Select a set of unlabeled samples from the pool (small set of random samples), assign a class label to each sample. This set is 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. Assign a class label to each of the queried samples by the supervisor S.
5. Add the new labeled samples to the training set T.
6. Retrain the classifier.

Until: The stopping criterion is satisfied.

In general, there are some parameters that should be defined in an active learning process. For example, as we have seen in the initial stage, a small number of labeled samples and a large number of unlabeled ones are used. The problem here is about the size of each set.

How many labeled samples should we have in an initial training set T? And how many unlabeled samples should we have in an initial pool P?

It's known that T should be as small as possible, but not less than what the classifier needs to perform a good training. The pool U, as well, should contain as many samples as possible, but it should also represent all the classes. A good active learning algorithm should be insensitive to the number of unlabeled samples (Sassano, 2002); it should always achieve good performance without regard to the number of unlabeled samples.

SUPPORT VECTOR MACHINE

For simplicity, let us first consider a supervised binary classification problem (Ghoggali *et al*., 2009). Let us assume that the training set consists of N vectors xi $\in\Re^n$ (i = 1, 2, . . ., N) from the n-dimensional feature space X. For each vector xi, there is an associated target yi$\in$ {-1, +1}. The linear SVM classification approach consists in looking for a separation between the two classes in X by means of an opportune hyper-plane. In the nonlinear case, data are first mapped with a kernel method in a higher dimensional feature space, i.e.: Φ (X) $\in\Re^{n'}$ (n'>n). The membership decision rule is based on the function sign [f (x)], where f (x) represents the discriminant function associated with the hyper-plane in the transformed space and is defined as:

$$F(x) = w.\Phi(x) + b \quad (1)$$

The optimal hyper-plane defined by the weight vector $w = w^* \in \Re^{n'}$ and the bias $b = b^* \in \Re$ 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:

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

And is subject to the following functional margin constraints:

$$y_i(w.\Phi(x_i) + b) \geq 1 - \xi_i, \quad i = 1, 2, \ldots, N \quad (3)$$

With,

$$\xi_i \geq 0, \quad i = 1, 2, \ldots, N \quad (4)$$

The ξi's are the so-called slack variables introduced to account for non-separable data. The constant c represents a regularization parameter that allows control of the tradeoff between model complexity and empirical risk. Large values of c favor the empirical risk minimization, thus leading to complex decision boundaries and over-fitting problems. Conversely, small values push toward model simplicity and hence lead to under-fitting issues.

The dual formulation of the aforementioned optimization problem is given by:

$$\max_\alpha \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j K(x_i, x_j) \quad (5)$$

Under the constraints:

$$\alpha_i \geq 0 \ for \ i = 1, 2, \ldots, N \quad (6)$$

$$\sum_{i=1}^{N}\alpha_i y_i = 0 \quad (7)$$

where, $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_N]$ is a vector of Lagrange multipliers. The final result is a discriminant function conveniently expressed as a function of the data in the original (lower) dimensional feature space X:

$$f(x) = \sum_{i\epsilon S}\alpha_i^* y_i \ K(x_i, x) + b* \quad (8)$$

where K(·,·) is a kernel function. The set S is a subset of the indices {1, 2, ..., N} corresponding to the nonzero Lagrange multipliers αi's which define the so-called support vectors. The kernel K(·,·) must satisfy the condition stated in Mercer's theorem so as to correspond to some type of inner product in the transformed (higher) dimensional feature space Φ (X). A typical example of such kernels is represented by the following Gaussian function:

$$K(x_i, x) = exp(-\gamma\|x_i - x\|^2) \quad (9)$$

where, γ represents a parameter inversely proportional to the width of the Gaussian kernel.

Using probability output: The theoretical advantages and the empirical success of support vector machines make them an attractive choice as a learning method to use with active learning. To this end, we need to use a probabilistic output for the classifier in the querying strategy to indicate which of the unlabeled samples are more suitable for labeling.

Support vector machines are mainly used to solve binary classification problems (a classification problem with only two known classes). However, our problem is a multi-class problem (a classification problem with more than two classes). As binary problems are much easier to solve and due to some other complexities, using a single SVM to solve multi-class problems is usually avoided; a better approach consists of using a combination of multiple binary SVM classifiers for a multi-class classification problem.

The extension of the SVM approach to multi-class classification problems can be done with different strategies, for which there are three well-known methods:

- One-against-all method using winner-takes-all strategy
- One-against-one method implemented by max-wins voting
- Error-correcting codes

Hastie and Tibshirani (1998) proposed to use the binary SVM outputs to estimate the posterior probabilities pi = Prob (ωi|x); i = 1, ..., M by a particular method (because SVMs are discriminant classifiers and do not give out posterior probabilities naturally). Then they used these probabilities to implement a multi-class SVM classifier based on a good general strategy called pairwise coupling. The pairwise coupling strategy assigns the sample under consideration to the class with the largest pi (Duan and Keerthi, 2005). Ting-Fan *et al*. (2004) proposed two new pairwise coupling schemes for the estimation of class probabilities. Duan and Keerthi (2005) in their empirical study entitled "Which Is the Best Multiclass SVM Method?" recommended using one of the pairwise coupling schemes in Hastie and Tibshirani (1998) and Ting-Fan *et al*. (2004) as the best kernel discriminant method for solving multi-class problems (Duan and Keerthi, 2005).

In the context of this Study, we have used LIBSVM supplied by Chang and Lin (2011) as SVM software based on the pairwise coupling schemes in Ting-Fan *et al*. (2004).

PROPOSED ACTIVE LEARNING METHOD

Support Vector Machines (SVMs) have become a popular learning algorithm, in particular for large and high-dimensional classification problems. SVMs give the most accurate classification results in a variety of applications (especially for text classification), which is why we focus on the SVM classifier.

Instead of predicting the label, the classification probability can be used to develop an SVM-based active learning strategy.

The proposed method benefits from this probability to define the most informative samples to be labeled. In fact, at the beginning, the SVM classifier is trained using an initial training set T of size 'N1' and this classifier is applied to a pool set U of size 'N2' to get probabilistic output for each unlabeled sample. The most informative samples to be selected are defined as the samples with probabilities that are lower than a threshold "tsh".

A supervisor S will be requested to assign a true class to these samples. Then, the newly labeled samples are added into the set T and the SVM classifier is retrained using the updated training set. This closed loop of selecting and retraining continues for each part of the pool.

As we have seen before, each active learning process has some important parameters to be defined. In the proposed method, there are two parameters: the threshold 'tsh' and the initial training size 'N1'. These two parameters are fixed in the initial phase of this Study and are used in the next phases.

Initial phase: In this initial phase, the dataset is divided into three main parts defined as:

- An initial training set Tr
- Test samples Ts
- A pool U of unlabeled samples subdivided into several parts (packets) with equal sizes

The process of selecting samples from the pool is done by ranking each packet one time using the labeled samples selected from the previous packets and continues until all the packets of the pool are processed. The selection strategy is based upon the SVM posterior probability and a threshold tsh is used to define the informativeness measure for each unlabeled sample in the pool. To define a suitable threshold and the ideal size for the initial training set, the active learning process is iterated for several thresholds with different training set sizes and tested each time on a test set to evaluate the attained accuracy. It is clear here that the choice of the dataset used in this phase is very important.

SVM active learning method: In this section the different steps of the SVM active learning method "SVM-AL" are described. As a simple pool-based active learning method, the "SVM-AL" uses the same configuration as the initial phase explained above, except that it uses a fixed threshold tsh and a predefined size for the training set. The algorithm concentrates on the estimated probability of the active learning process and selects a number of samples from the packet using a predefined criterion (a threshold to measure how informative is each unlabeled sample in the pool). 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.

The main objective of the "SVM-AL" is to minimize the labeled samples needed to train the classifier without affecting the performance of this latter. Minimizing the labeled samples means minimizing the cost of labeling these samples and accelerating the process of training. So the question is: how many samples need to be labeled to get good training? Too few samples may cause bad training and too many samples cause a high cost. So the number of samples is a tradeoff between cost and training consistency.

To evaluate the effectiveness of the method, the system accuracy based on the newly labeled samples

should be compared to the initial system based on all of the samples in the pool.

The Active Learning SVM (SVM-AL) algorithm is as follows.

Algorithm 2: SVM-AL:

0. Start with a stream of packets of unlabeled data and an initial training set.

Repeat:

1. Estimate the best parameters for the classifier using a cross-validation method.
2. Apply the classifier with the best parameters to the current packet. This will provide a posterior probability for each sample in the packet. The sample is assigned to the class with the highest probability.
3. Select the samples with probabilities below some threshold tsh as informative samples to be labeled.
4. Present the selected samples to the supervisor (expert) for labeling.
5. Added the labeled samples to the training set of the classifier.

Until the last packet in the stream.

EXPERIMENTAL RESULTS

Dataset description: The experimental validation of the proposed method was conducted on the basis of three different known datasets in the field of text classification. The TC benchmarks used (Cardoso-Cachopo and Oliveira, 2007) have been downloaded from a publicly available repository of datasets for single-label text categorization[1]. In this website, there is also a description of the datasets, their standard train/test splits, how they were processed to become single-labeled and the pre-processing techniques that were applied to each dataset, namely character clean-up, removal of short words, removal of stop words and stemming.

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 files r8-train-stemmed and r8-test-stemmed, available from that website (Cardoso-Cachopo and Oliveira, 2007).

20ng: The 20ng dataset is a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups. For this dataset, we used the files 20ng-train-stemmed and 20ng-test-stemmed, available from that website (Cardoso-Cachopo and Oliveira, 2007).

Table 1: Preprocessed data sets

Dataset	Classes	Total docs	Smallest class	Largest class
R8	6	7479	271	3923
20ng	20	16841	251	999

Table 2: Original and remaining features for the datasets

Dataset	Original features	New features	Gain (%)
R8	4982	2031	59.23
20ng	24040	7971	66.84

Dataset pre-processing: It is widely accepted that the way that documents and queries are represented influences the quality of the results that can be achieved. The main aim of preprocessing the data is to reduce the problem's dimensionality by controlling the size of the system's vocabulary.

In this Study first each class with a size of fewer than 200 samples in each dataset is omitted. After that, the documents are represented in a vector model using the TFIDF (Sparck Jones, 1972; Salton and Buckley, 1988; Spärck Jones, 2004) technique.

The preprocessed datasets are represented in Table 1.

In some situations, aside from reducing the complexity of the problem, the preprocessing of the dataset also unifies the data in a way that improves performance. For this end, we have adopted the histogram feature extraction method by discarding every word that doesn't repeat in at least 1% of the documents of the dataset. Table 2 shows the original and remaining features.

First experiment (initial phase): The objective of this experiment is to define the initial size of the training set T as well as the threshold value to be used in our method. This experiment was conducted using Reuter's dataset. Reuter's is one of the most used datasets in text classification.

Preparing the experiment: Before carrying out the experiment, the dataset was randomly divided into 6 training sets with different numbers of samples (10, 20, 25, 50, 75 and 100, respectively) per class. The poo1 is divided into packets with 200 samples each. We got 19 packets. After that, the naïve approach is applied for all the training sets, each using different threshold values. The lower and upper accuracies were found by applying the naïve approach with 0 and 100 as the thresholds.

Table 3 shows the accuracies obtained by the naïve method using different sizes for the training set and different thresholds. Table 4 shows the size of samples labeled using the same size of initial training set and different values for the threshold. The aim of this experiment is to choose the best combination between the threshold and training set in order to minimize the number of labeled samples and maximize the accuracy obtained. So it is a compromise between the initial training set size, the threshold, the obtained accuracy and the number of labeled samples. This is why the initial training set has been chosen with 20 samples per

Table 3: The accuracy of SVM for R8, N` and thsh

		Number of samples per class					
		10	20	25	50	75	100
Threshold <N%	10	83.92	85.92	86.16	88.04	91.92	93.20
	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	97.00	97.24	97.80
	90	96.08	96.16	96.28	96.96	97.40	97.60
	100	96.84	96.88	96.96	97.52	97.92	97.88

Table 4: The datasets sizes

		Number of samples per class					
		10	20	25	50	75	100
Number of samples	10	0	0	0	0	0	0
added <N%	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	3779	3779	3779	3779	3779	3779

Table 5: The accuracies obtained and data labeled by the "SVM-AL" approach for the different datasets

Dataset	Lower	Upper	SVM-AL	Labeled	Data size
R8	83.32	96.98	95.64	382	3779
20ng	43.78	74.86	73.22	4345	7341

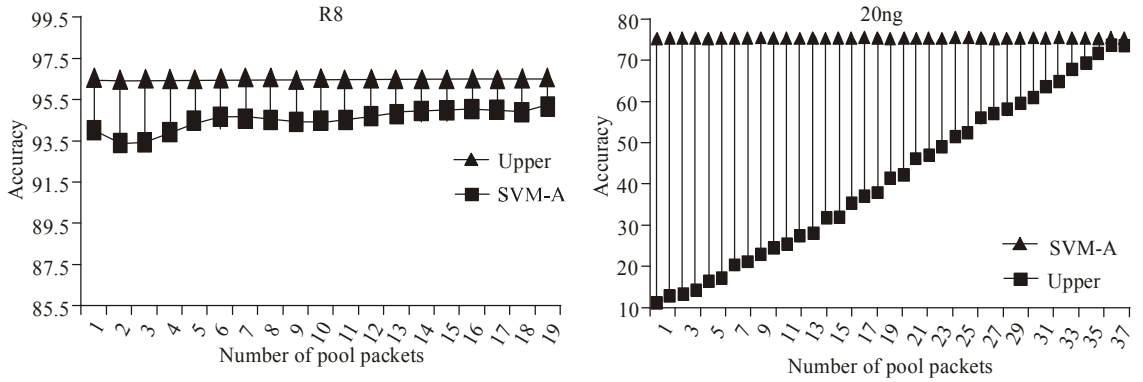

Fig. 1: Comparison of "SVM-AL" and SVM in terms of accuracy

class as the minimum size which can give good accuracies and the threshold at 70%. Note that the threshold at 70% does not mean that we will select 70% of the pool samples, but that sample with a probability less than 70% will be selected (which represents 10% of samples in our case).

After that, the chosen threshold and training set size will be considered in this experiment as a reference to be used in subsequent experiments.

Second experiment (SVM-AL): In this experiment, the dataset was divided into 5 different training sets, with the same size of 20 samples/class. The poo1 is divided into packets of 200 samples. The naïve approach is applied with the threshold at 70%. The lower and upper accuracies were found by applying the naïve approach with 0 and 100 respectively as thresholds.

Figure 1 shows the accuracies obtained by the proposed method for the different datasets.

Table 5 shows that good accuracies have been obtained with low training sizes. For R8, for example, we got a result higher than 95%. From this table, note that we were able to get accuracies that are very close to the upper accuracies, using lower training sizes. In R8, for example, we were able to reduce the difference of the accuracy to less than 1.5%, while labeling only about 10% of the pool. In 20ng, the difference of accuracies was also about 1.5% and the labeled samples represent about 59%.

CONCLUSION

In this Study, multi-class SVM active learning has been applied for text classification. A novel active learning method (SVM-AL) for text classification was presented. It selects a batch of informative samples for manual labeling by an expert. The posterior probability output of a multi-class SVM method is used. Extended experiments have been performed on two well-known

real text classification datasets. To empirically assess the effectiveness of the proposed method, we have compared it with the results of an SVM classifier applied to the whole dataset. In this comparison, we observed that the proposed method significantly reduces the need for labeled training data and provides the best tradeoff between classification accuracy and the number of labeled samples that are needed.

REFERENCES

Balamurugan, S.A.A. and R. Rajaram, 2009. Effective and efficient feature selection for large-scale data using Bayes' theorem. Int. J. Autom. Comput., 6(1): 62-71.

Basu, T. and C. Murthy, 2014. Towards enriching the quality of k-nearest neighbor rule for document classification. Int. J. Mach. Learn. Cybern., 5(6): 897-905.

Cai, F., H. Chen and Z. Shu, 2014. Web document ranking via active learning and kernel principal component analysis. Int. J. Mod. Phys. C, 26(4): 18.

Cardoso-Cachopo, A. and A.L. Oliveira, 2007. Semi-supervised single-label text categorization using centroid-based classifiers. Proceeding of the ACM Symposium on Applied Computing. ACM, New York, pp: 844-851.

Chang, C.C. and C.J. Lin, 2011. LIBSVM: A library for support vector machines. ACM T. Intell. Syst. Technol., 2(3): 27.

Demir, B., C. Persello and L. Bruzzone, 2011. Batch-mode active-learning methods for the interactive classification of remote sensing images. IEEE T. Geosci. Remote, 49(3): 1014-1031.

Ding, S., B. Li and X. Fu, 2014. Active learning methods for classification of hyperspectral remote sensing image. In: Huang, D.S. *et al*. (Eds.), ICIC, 2014. LNAI 8589, Springer International Publishing, Switzerland, pp: 484-491.

Duan, K.B. and S.S. Keerthi, 2005. Which is the best multiclass SVM method? An empirical study. In: Oza, N.C. (Eds.), MCS, 2005. LNCS 3541, Springer-Verlag, Berlin, Heidelberg, pp: 278-285.

Elahi, M., F. Ricci and N. Rubens, 2014. Active learning in collaborative filtering recommender systems. In: Hepp, M. and Y. Hoffner (Eds.), EC-Web 2014. LNBIP 188, Springer International Publishing, Switzerland, pp: 113-124.

Fragos, K., P. Belsis and C. Skourlas, 2014. Combining probabilistic classifiers for text classification. Proc. Soc. Behav. Sci., 147: 307-312.

Ghoggali, N., F. Melgani and Y. Bazi, 2009. A multiobjective genetic SVM approach for classification problems with limited training samples. IEEE T. Geosci. Remote, 47(6): 1707-1718.

Goudjil, M., M. Koudil, N. Hammami and M. Bedda, 2013. Arabic text categorization using SVM active learning technique: An overview. Proceeding of the World Congress on Computer and Information Technology (WCCIT), pp: 1-2.

Hastie, T. and R. Tibshirani, 1998. Classification by pairwise coupling. Ann. Stat., 26(2): 451-471.

Lam, W. and Y. Han 2003. Automatic textual document categorization based on generalized instance sets and a metamodel. IEEE T. Pattern Anal., 25(5): 628-633.

Li, M. and I.K. Sethi, 2006. Confidence-based active learning. IEEE T. Pattern Anal., 28(8): 1251-1261.

Li, Q. and L. Chen, 2014. Study on Multi-class Text Classification Based on Improved SVM. In: Wen, Z. and T. Li (Eds.), Practical Applications of Intelligent Systems. Advances in Intelligent Systems and Computing 279, Springer-Verlag, Berlin, Heidelberg, pp: 519-526.

Mangai, J.A., V.S. Kumar, S.A. alias Balamurugan, 2012. A novel feature selection framework for automatic web page classification. Int. J. Autom. Comput., 9(4): 442-448.

Persello, C. and L. Bruzzone, 2014. Active and semisupervised learning for the classification of remote sensing images. IEEE T. Geosci. Remote, 52(11): 6937-6956.

Salton, G. and C. Buckley, 1988. Term-weighting approaches in automatic text retrieval. Inform. Process. Manag., 24(5): 513-523.

Sassano, M., 2002. An empirical study of active learning with support vector machines for Japanese word segmentation. Proceeding of the 40th Annual Meeting on Association for Computational Linguistics. USA, pp: 505-512.

Sebastiani, F., 2002. Machine learning in automated text categorization. ACM Comput. Surv. (CSUR), 34(1): 1-47.

Settles, B., 2010. Active learning literature survey. Uni. Wisconsin, Madison, 52(55-66): 11.

Shen, Q. and R. Jensen, 2007. Rough sets, their extensions and applications. Int. J. Autom. Comput., 4(3): 217-228.

Sparck Jones, K., 1972. A statistical interpretation of term specificity and its application in retrieval. J. Doc., 28(1): 11-21.

Spärck Jones, K., 2004. A statistical interpretation of term specificity and its application in retrieval. J. Doc., 60(5): 493-502.

Ting-Fan, W., L. Chih-Jen and C.W. Ruby, 2004. Probability estimates for multi-class classification by pairwise coupling. J. Mach. Learn. Res., 5: 975-1005.

Tong, S. and D. Koller, 2002. Support vector machine active learning with applications to text classification. J. Mach. Learn. Res., 2: 45-66.

Wang, J.H. and H.Y. Wang, 2014. Incremental neural network construction for text classification. Proceeding of the International Symposium on Computer, Consumer and Control (IS3C). Taichung, pp: 970-973.

End note:

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