Neural Comput & Applic (2013) 22 (Suppl 1):S83–S100
DOI 10.1007/s00521-012-1056-5

ORIGINAL ARTICLE

Improving the precision-recall trade-off in undersampling-based binary text categorization using unanimity rule

Zafer Erenel · Hakan Altınçay

Received: 30 May 2011 / Accepted: 22 June 2012 / Published online: 11 July 2012

Abstract The distribution of documents over two classes in binary text categorization problem is generally uneven where resampling approaches are shown to improve F_1 scores. The improvement achieved is mainly due to the gain in recall where precision may deteriorate. Since precision is the primary concern in some applications, achieving higher F_1 scores with a desired level of trade-off between precision and recall is important. In this study, we present an analytical comparison between unanimity and majority voting rules. It is shown that unanimity rule can provide better F_1 scores compared to majority voting when an ensemble of high recall but low precision classifiers is considered. Then, category-based undersampling is proposed to generate high recall members. The experiments conducted on three datasets have shown that superior F_1 scores can be realized compared to the support vector machines(SVM)-based baseline system and voting over a random undersampling-based ensemble.

Keywords Class imbalance · Resampling · Classifier ensemble · Unanimity rule · Binary text categorization

Z. Erenel
Department of Computer Engineering, European University of Lefke, Lefke, Northern Cyprus, Turkey
e-mail: zerenel@eul.edu.tr

H. Altınçay (✉)
Department of Computer Engineering, Eastern Mediterranean University, Famagusta, Northern Cyprus, Turkey
e-mail: hakan.altincay@emu.edu.tr

1 Introduction

The volume of information on the web has grown drastically in the last decade mainly due to the increased demand for sharing knowledge and cheaper storage mediums. Most of the existing hypertext documents such as academic and business web pages, news articles, and forums are in the form of inter-linked natural language files. Organization of such files into predefined categories for fast and effective retrieval and spam filtering for electronic mails are two primary application areas of automatic text categorization which aims to save time and efforts. The basic components of an automatic text categorization system are document representation and classifier design. The bag of words approach is generally used for document representation where each word corresponds to a different feature [1]. In this approach, the order, meaning, structure, and grammar of words are not considered, and each document is represented by a high-dimensional feature vector where the number of entries is equal to the number of words selected from the vocabulary of the textual documents under concern [2]. A variety of pattern classification techniques such as neural networks [3], Rocchio method [4], naive Bayes [5, 6], k-nearest neighbors [7], and support vector machines (SVM) [8] are widely studied for text categorization. The robustness of SVM in very high dimensional feature space sets it as the state-of-the-art classifier for text categorization since documents are generally represented as feature vectors consisting of thousands of entries [9]. In its simplest implementation, a linear SVM computes a hyperplane which separates the samples belonging to different categories with the largest margin. Recent studies clearly demonstrate that SVM achieves better scores compared to its competitors such as k-NN, naive Bayes, and neural networks [10, 11].

The text categorization problem is generally tackled as the solution of several independent binary classification sub-problems. For a particular category, the positive class includes all documents belonging to the target (minority) category, whereas the negative (majority) class consists of all documents from other categories. As a matter of fact, the distribution of training documents in the classes is generally uneven which leads to the well-known class imbalance problem. Since the negative class has much more documents compared to the positive, learning algorithms are overwhelmed by the negative class, and hence, they generally tend to classify the test samples as negative, producing many false negatives [12–14]. In other words, the performance of the categorization systems is generally poor on the positive class. Since the performance on the positive class is of primary concern, precision which is defined as the percentage of documents which are correctly labeled as positive and recall which is the percentage of correctly classified positive documents are generally used to compare different systems. However, since a categorization system can be tuned to maximize either precision or recall at the expense of the other, their harmonic mean named as F_1 score is considered as more significant [15].

In text categorization, it is desirable to have higher F_1 scores by boosting both precision and recall. Nevertheless, with imbalanced training examples, SVM-based categorization systems often provide high precision but low recall [16, 17]. Although it is a young field of pattern classification, studies in class imbalance are rapidly growing and various different techniques are proposed [18]. Resampling technique which is also known as dataset balancing is the most widely used approach to address the imbalance problem [19]. In this approach, *undersampling* the majority or *oversampling* the minority class before classifier construction is applied [20]. In random undersampling approach, a randomly selected subset of negative samples is used in training the categorization system. Alternatively, informed undersampling can be used. For instance, in NearMiss-2 method [21, 22], the distance to the three farthest minority samples is considered in selecting the majority class samples. In general, better F_1 scores are achieved using undersampling techniques by improving recall where precision generally deteriorates [23]. This is due to the fact that undersampling mitigates the bias toward the majority class and hence increases the number of false positives. On the other hand, oversampling techniques such as SMOTE (Synthetic Minority Over-sampling TEchnique) [24] rarely produce higher precision in text categorization compared to the case where resampling is not applied and the gain in recall is generally smaller compared to undersampling [23]. It is recently shown by Sun et al. [16] that random undersampling provides better F_1 scores compared to SMOTE on three benchmark text categorization datasets. Li et al. [17] have also recently shown that, despite the gain in recall, oversampling the minority class brings down the precision value achieved by SVM in text categorization problem. It is generally argued that undersampling techniques are more promising compared to oversampling in various domains including text categorization [22, 23, 25] although the opposite is observed in some cases [12].

Since large number of negative samples are ignored, the main drawback of undersampling is the loss of potentially useful information. In order to avoid this, the use of an ensemble of classifiers is considered and plenty of schemes are proposed [13, 15, 18, 26]. Although the developed schemes differ in various aspects, the major difference is in the way the samples are selected. The most popular approaches are random partitioning, clustering, bagging, and boosting [26–33]. In bagging- and boosting-based approaches, the ensembles are made up of weak learners, each of which focuses on a different set of samples. SMOTEBoost and DataBoost-IM are examples of such schemes. In the former approach, samples from the minority class are synthetically generated by focusing on the difficult samples [34]. In the latter approach, synthetic samples are generated from the hard samples of both classes which is followed by re-balancing the total weights of different classes to alleviate the bias toward the majority class [35]. RUSBoost which is based on random undersampling of the majority until desired number of samples are obtained is another AdaBoost-based iterative scheme where, it is recently shown that, RUSBoost surpasses SMOTEBoost about twice the cases when SMOTEBoost outperforms RUSBoost [36]. In these schemes, C4.5, naive Bayes and RIPPER are generally considered as the weak learners [37]. The joint decision is formed by averaging the outputs of the member classifiers or weighted voting on the individual decisions [32, 38]. It is generally argued that an ensemble of undersampling-based classifiers outperforms a single classifier [15, 38]. However, due to its impressive performance on sparse and very high dimensional feature vectors, SVM is more widely used compared to boosting weak learners in the field of text categorization.

Improved F_1 scores are achieved by balancing techniques mainly due to the gain in recall compared to the case when resampling is not used. However, precision may be the major consideration in some text categorization applications. For instance, it may be required to have small number of false positives at the top results of web search applications which correspond to high precision [39, 40]. Similarly, a spam filter that can detect all spam messages which form the positive class (i.e., perfect recall) but classifies many legitimate mails as spam (i.e., low precision) cannot be accepted [39, 41]. Improved F_1 score with a poor precision value is not advantageous for such implementations. Hence, for SVMs which generally provide high precision, improving F_1 score without dropping precision is an important problem. In other words, achieving a good trade-off between precision and recall is important [42].

Although majority voting is more popular in the pattern classification literature, unanimity rule which requires the agreement of all members on the positive class for a positive decision is known to provide better precision values compared to majority voting but worse recall in general [42–44]. In this study, we mainly focused on clarifying the characteristics of unanimity and majority voting rules. An analytical investigation of these rules is firstly presented. To the best of our knowledge, this is the first study to prove that unanimity rule can provide better F_1 scores compared to majority voting when an ensemble of high recall but low precision classifiers is considered. In other words, the relative performance of unanimity and majority voting rules is shown to depend on the ensemble under concern. In order to generate high recall members for text categorization, category-based undersampling is proposed where the number of subsets from the negative class is selected as the number of categories it includes. More specifically, each undersampled set consists of documents from a different category. This type of undersampling is shown to provide members having higher individual recall values compared to random partitioning. Then, fusion of classifiers generated using category-based undersampling by unanimity rule is studied. Experiments conducted on three datasets have shown that better trade-off between precision and recall can be achieved compared to the SVM-based baseline system and voting using a random undersampling-based ensemble.

Section 2 presents a review about differences between unanimity and majority voting rules which are explained using an artificial example. In Sect. 3, the F_1 scores provided by unanimity and majority voting rules are studied by expressing them as functions of precision and recall of the ensemble members. The category-based undersampling scheme for text categorization is also presented in that section. In order to evaluate the proposed approach, experiments are conducted on Reuters-21578 ModApte Top10, WebKB and 7-Sectors datasets that are presented in Sect. 4. The last part, Sect. 5 summarizes the conclusions drawn from this study.

2 Undersampling-based ensembles: a brief review

The superiority of SVM in text categorization compared to the other well known machine learning schemes is the trade-off it provides in precision and recall which is generally expressed in terms of F_1 score. However, it is known that recall has room for improvement. This can simply be achieved using a resampled subset of the majority class. The cost of this improvement might be the deterioration of precision since the decision boundary moves to the majority class due to the reduced bias on that class [23, 33]. For further clarification of this well known fact, consider Fig. 1 where the decision boundaries computed when all or a randomly undersampled set of majority samples shown by ‘•’s is used in an artificial dataset. As seen in the figure, the decision boundary is moved toward the majority class when an undersampled training set is used. The new boundary corresponds to a reduction in the number of false negatives and hence an improvement in recall. However, this also corresponds to an increase in the number of false positives which causes a decrease in precision. Consequently, if precision is of main interest, an improved F_1 score due to a major gain in recall may not be considered as valuable.

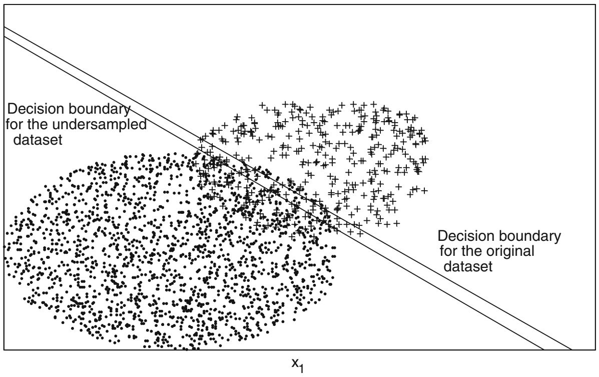

Fig. 1 The decision boundaries computed using SVM when an undersampled set or all samples from the majority class are used

Using an ensemble of classifiers for the solution of a binary classification problem, each of which is trained using an undersampled set of the majority class, generally produces a low precision system as mentioned in Sect. 1. For a better understanding of this fact, assume that M different subsets, $r_1, r_2, \ldots, r_M$ from the majority class denoted by c_2 are formed and a binary classifier is trained for each subset where all samples from the positive class, c_1 are used in each member. This corresponds to M binary classifiers, one for each of the sample set pairs $\{c_1, r_1\}$, $\{c_1, r_2\}, \ldots, \{c_1, r_M\}$. It is plausible that there are similarities between some r_m's and c_1. In other words, r_i may be more similar to c_1 compared to r_j. In such a case, a negative sample may be classified as positive by the corresponding classifiers. Since each wrong vote is assigned to the same (positive) class, they can reach the majority, leading to a false positive joint decision when majority voting is used. The practical consequences of this observation can be easily understood when the members trained on subsets of the majority samples are investigated using an artificial

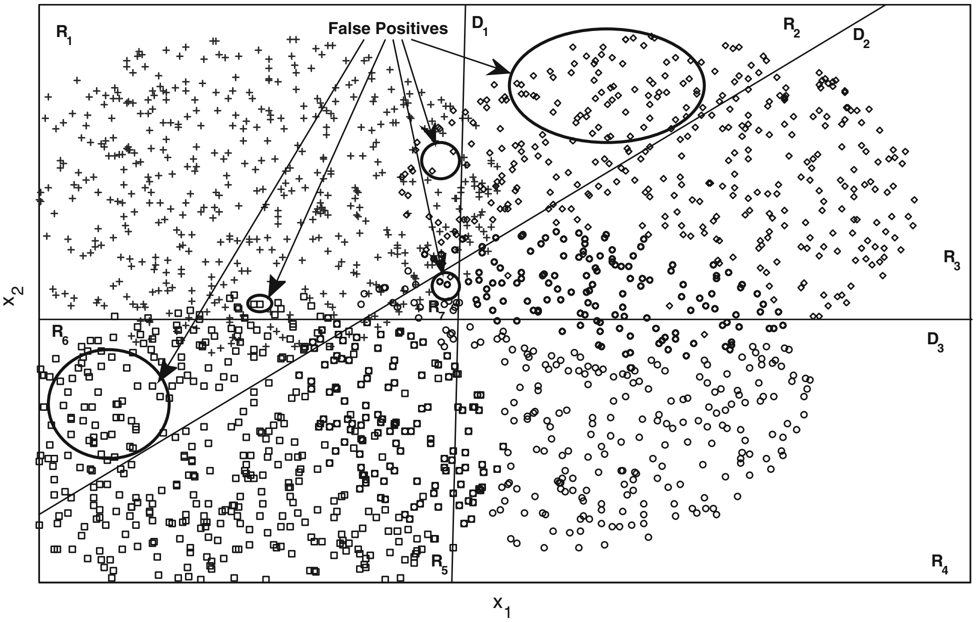

Fig. 2 Three decision boundaries computed using SVM classifier. For each binary classifier, all positive samples are considered but a subset of the negatives marked by either '◇', '^' or '□' are used

dataset as depicted in Fig. 2. The samples in the negative class are partitioned into three sets r_1, r_2 and r_3 which are shown using the markers '◇', '^' and '□', respectively. The samples of the positive class (c_1) are marked using the symbol '+'. D_1, D_2, and D_3, respectively, denote the boundaries computed using an SVM classifier for $\{c_1, r_1\}$, $\{c_1,r_2\}$ and $\{c_1,r_3\}$. It can be easily seen in the figure that these decision boundaries correspond to classifiers which individually have poor precision values due to large number of false positives. Assume that voting is applied on the outputs of these binary classifiers. For this combination scheme, the samples in the regions labeled as R_1, R_2, R_6 and R_7 that are bounded by the nearest decision boundaries and axes are classified as positive. The reason for this is that some samples in r_1 (lying in R_2) are more similar to those in c_1 compared to r_2 and they are on the same side of the decision boundary, D_2. Similarly, some samples in r_3 (lying in R_6) are more similar to those in c_1 than r_2 and they are again on the same side of D_2. These similarities lead to false positives in R_2 and R_6. In the figure, five sets of samples belonging to the negative class that are wrongly classified as positive (i.e., false positive) are shown using circles. This explains the reason for poor precision generally achieved when voting is applied on undersampling-based classifiers. The gain in recall can be explained by the reduced number of false negatives due to these decision regions. On the other hand, it can be easily seen that only the samples that are in R_1 are labeled as positive with unanimity rule where samples in R_2, R_6 and R_7 are labeled as negative. As a matter of fact, the number of false positives is reduced at the cost of increased number of false negatives which are located in R_2, R_6 and R_7. This corresponds to a smaller recall value compared to the voting-based scheme which is tolerable in applications where high precision is essential, provided that a better F_1 score is achieved.

The review presented above can be summarized as follows:

- Undersampling the majority class increases recall at the expense of precision.
- Combination of undersampled classifiers using unanimity rule provides better precision than majority voting at the expense of recall.
- Combination of undersampled classifiers using majority voting rule provides better recall than unanimity at the expense of precision.

Although the relative performance of unanimity and majority voting rules in terms of precision and recall is well established, their relative performance in terms of F_1 scores depends on the selected ensemble. In order to clarify this relationship, an analytical investigation is presented in the following section.

3 Analytical investigation of unanimity and majority voting rules

In the proposed analysis, the F_1 score of the combined system is firstly formulated in terms of the classification

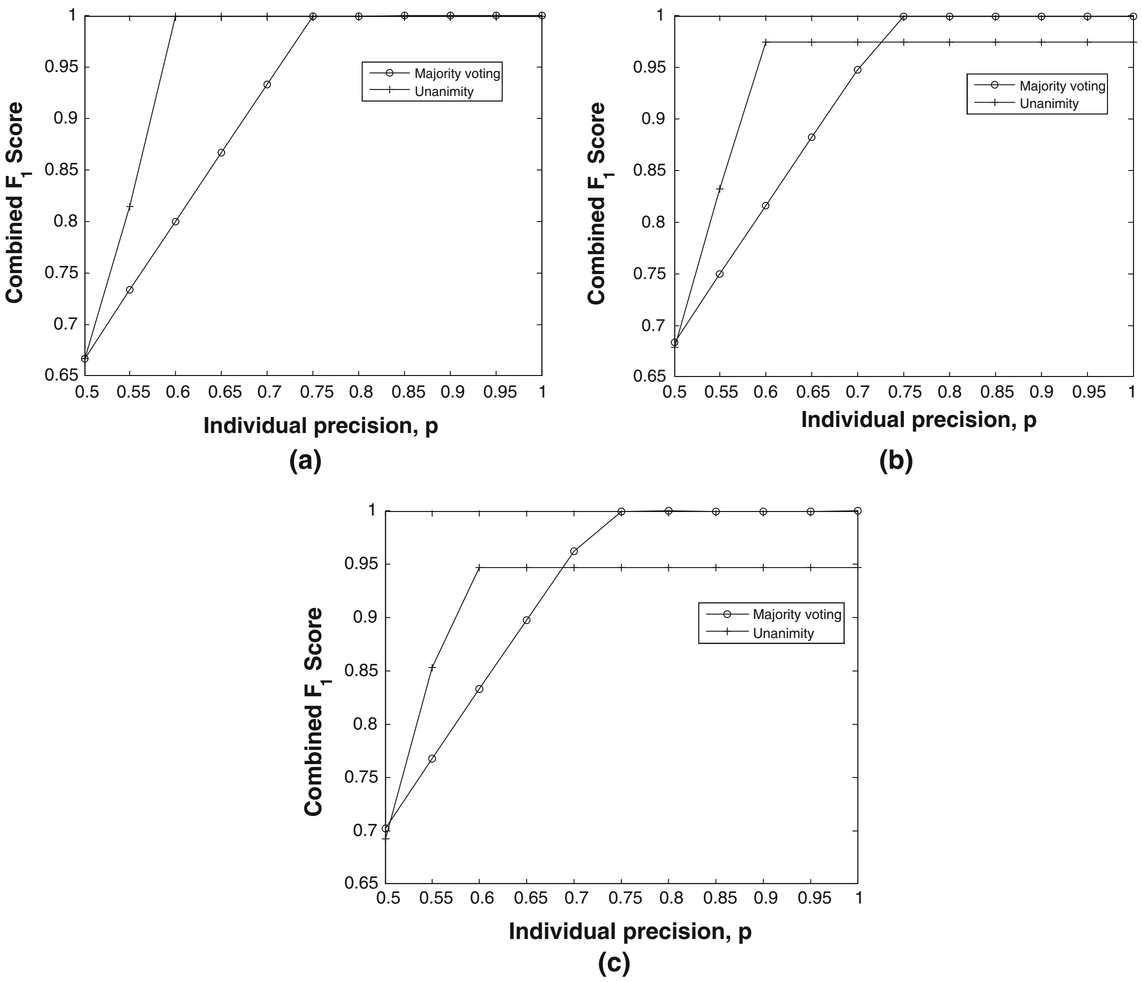

Fig. 3 The best combined F_1 scores that can be computed using three ensemble members as a function of individual precision, p where **a** $r = 1.0$, **b** $r = 0.95$, **c** $r = 0.90$

behaviors of the individual members for both majority voting and unanimity rules. For each rule, the best combined score that can be achieved is then defined as an optimization problem. The formulation of the optimization problem is given in "Formulation of the optimization problem" of the "Appendix". In the analysis, it is assumed that there are M ensemble members, each providing the same precision (p) and recall (r) values. The findings of this analysis is presented in the following propositions.

Proposition 1 *The F_1 score of the unanimity rule-based combined system will be* 1, *if* $r = 1$ *and* $p \geq \frac{2M}{3M+1}$.

Proof The proof is given in "Proof of proposition 1" of the "Appendix". □

Proposition 2 *The F_1 score of the majority voting rule-based combined system will be* 1, *if* $r = 1$ *and* $p \geq \frac{2M}{3M-1}$.

Proof The proof is given in "Proof of proposition 2" of the "Appendix". □

The special case considered in Propositions 1 and 2 clearly show that, for high values of individual recall, unanimity rule is able to achieve $F_1 = 1$ (perfect combined system) for smaller precision values compared to majority voting. In other words, for small precision values, the performance of the unanimity rule can be better provided that the individual recall is large. In fact, this is the case when ensemble members are generated using undersampling. The members generally have small precision but high recall values as mentioned in Sect. 2.

The solutions presented in the "Appendix" for proving Propositions 1 and 2 put restrictions to the joint distributions of the member outputs. However, the solutions for the optimization problems (defined in Eqs. 11, 13) are not unique. Because of this, other forms of solutions might provide different limits. Although different solutions can be computed, the solutions found in the proofs clearly verify the fact that better combined scores can be obtained using unanimity rule. In order to compare these rules by considering all feasible solutions and more general cases than the special one considered in the propositions, the optimization toolbox of Matlab is used to study the best combined F_1 scores that can be achieved for $p \in [0.5, 1.0]$. Fig. 3 presents the results for three ensemble members and three recall values, 1.0, 0.95, and 0.90, respectively, in parts (a), (b) and (c). The superiority of unanimity for low precision values is evident in the figures. In particular, when $r = 1.0$, it can be seen in the figure that perfect combined system can be achieved using classifiers having $p \geq \frac{2M}{3M+1} = \frac{3}{5} = 0.6$ whereas p should be greater than $\frac{2M}{3M-1} = \frac{3}{4} = 0.75$ as found in the proofs of Propositions 1 and 2, respectively. It should be noted that, as seen in Fig. 3, majority voting rule provides superior performance than unanimity when both precision and recall values are large. The experimental results presented in [45] are consistent with this observation.

For a better visualization of the relative performance of the two rules, Fig. 4 presents the contour lines where each line represents the values of p and r which provide the same difference between the best F_1 scores that can be computed using unanimity and majority voting rules, i.e ($F_1^{\mathrm{una}} - F_1^{\mathrm{mv}}$). As seen in the figure, for small values of p and large values of r, unanimity performs better.

The method of undersampling is an important factor that may affect the performance of unanimity-based systems since, as shown above, high recall is an important parameter. It should also be noted that, we agree with the argument of Chang et al. [46] that using random undersampling for generating negative sample subsets will generate geometrically inconsistent sample sets. Clustering-based undersampling may be a better approach. However, the effectiveness of a clustering algorithm in partitioning the given data into homogenous clusters depends on various parameters such as the clustering algorithm, number of clusters and distance measure. In text categorization where thousands of features are used, it is difficult to justify a particular choice of these design parameters by any technique different from empirical evaluation. Moreover, it is argued that the geometric relations between majority and minority sub-populations may be lost if resampling is not done appropriately [29]. Because of these, we followed a different path for the determination of clusters for the negative documents.

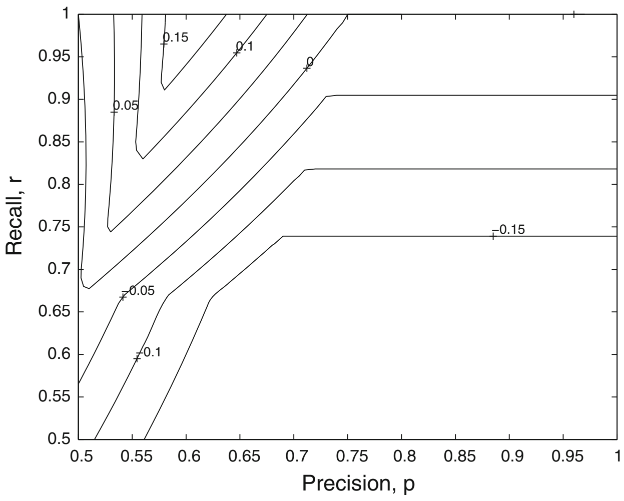

Fig. 4 The *contour plot* of the difference of the best F_1 scores ($F_1^{una} - F_1^{mv}$) for $p, r \in [0.5, 1.0]$

In text categorization, the documents in the negative class belong to one or more (multi-labeled case) pre-defined categories. These categories naturally constitute sub-populations in the negative class. Instead of applying a clustering algorithm to estimate the clusters, we propose to define the set of documents that belong to a different category as a separate cluster. The implementation of this rule is illustrated in Fig. 5. If the documents are multi-labeled, then they will exist in more than one cluster. The present form of this approach does not require any design parameter to be tuned. The number of clusters considered and correspondingly the number of ensemble members depend on the number of categories involved in the negative class. This is reasonable since, as the number of categories increases, the negative class becomes more heterogenous and it is advantageous to use increased number of clusters. As it will be shown in the following section, this form of undersampling leads to ensemble members having higher recall when compared to random partitioning which is advantageous for unanimity rule of combination.

4 Experiments

In this study, three widely used datasets are considered. The ModApte split of top ten classes of Reuters-21578 is the first where the negative classes are defined to include documents which belong to one or more of the remaining nine categories [47]. The highly imbalanced category distribution of Reuters-21578 ModApte Top10 makes it significant among other datasets. WebKB is a collection of web pages which belong to either of seven categories [48]. They were collected by the Carnegie Mellon University Text Learning Group from several universities in 1997. Four of the categories namely, “Student”, “Faculty”, “Course” and “Project” which contain totally 4,199 documents are generally used in text categorization experiments [49]. The 7-Sectors dataset contains 4,581 web pages. Each page belongs to one parent category and one subcategory in hierarchical order. Following the previous experiments in the literature [50], seven parent categories are considered in the simulations. The training and test sets of Reuters-21578 dataset are defined. However, this is not

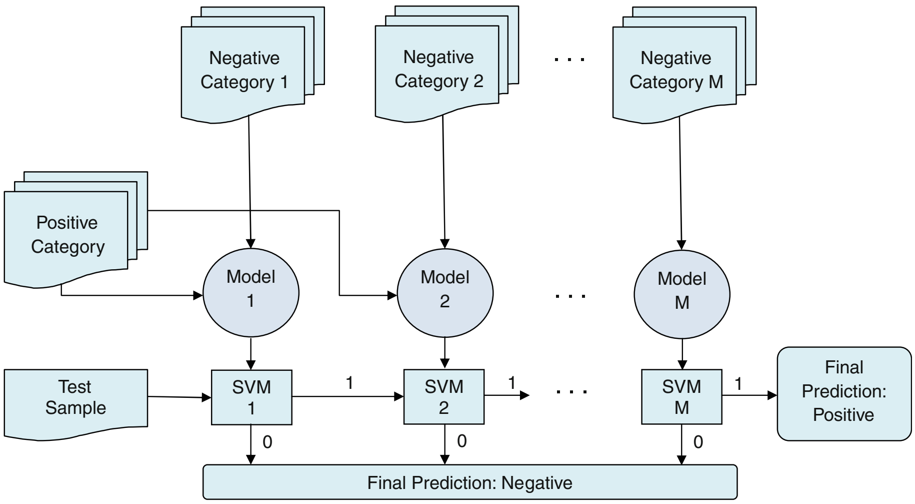

Fig. 5 *Block diagram* for the implementation of the unanimity rule in an ensemble of classifiers that are trained using category-based undersampling of the negative class

the case for WebKB and 7-Sectors. Because of this, following the work of Bekkerman et al. [51] and Xue et al. [49], four-fold cross-validation is performed on these datasets. For this purpose, the available data is initially partitioned into four folds. Four experiments are then performed where, in each experiment, one fold's data are used for testing while data in the remaining folds are used for training. The average scores are reported.

Before training the classifiers, preprocessing is applied on the datasets. Firstly, stopwords such as prepositions (in, on, down, etc.), conjunctions (and, but, while, etc.) and articles (a, an, the, etc.) are removed using SMART stoplist [52]. Subsequently, Porter stemming algorithm is applied to group the words with the same stems together [53]. Porter stem produced does not need to be meaningful or identical to the original root of the words. For instance, the words "comparing", "compared" and "comparable" are stemmed as "compar". Term frequencies in each document are cosine-normalized to eliminate the effect of varying document lengths. By exploiting different kernel functions, it is possible to generate linear and nonlinear SVMs [54]. Previous experiments have shown that linear SVMs perform better than nonlinear ones in text categorization [55]. Hence, linear SVM is adopted in our experiments. The SVM-based classification toolbox, SVM$^{\text{light}}$ with default parameters ($C = 1/\text{avg}(\bar{x}^T\bar{x})$ which is the inverse of the average of the inner product values of the training data) and linear kernel is employed as the classification scheme [8, 56]. The overall performances of the schemes considered are firstly evaluated using precision, recall and F_1 scores where both macro and micro scores are presented [1]. More specifically, macro-precision, macro-recall and macro-F_1 are computed as the averages of the corresponding scores obtained for each individual category. Micro-F_1 scores obtained by assigning equal weights to all documents are also reported. Although F_1 score is the most commonly used performance measure in text categorization studies, the area under the precision-recall curve (AUP) is also considered as a powerful metric, especially for evaluating the decision surface of the generated classifier [16]. AUP is also employed as an alternative metric for the evaluation of the proposed scheme.

The bag of words approach used for document representation generally leads to a very high-dimensional feature space consisting of tens of thousands of words even for medium-sized datasets. Although the computational power is elevating rapidly in today's world, there is a need for a decrease in the number of original features due to the fact that all terms may not be useful for discriminating different categories and the curse of dimensionality problem is encountered in many classification algorithms. In a recent study, it is observed that the F_1 scores of most weighting schemes plateau after 5000 features for SVM [55]. Because of this, top 5,000 features ranked by the term selection measure, chi-square (χ^2) are considered in the experiments [57].

After the feature set is specified, term weighting is generally applied as the following step [58]. The main idea is to quantify the relative importance of the selected terms where discriminative terms are assigned larger weights [55]. Term weighting has been traditionally formulated as the product of the term frequency and inverse document frequency, tf × idf [59]. In this study, relevance frequency (RF)-based weighting (tf × RF) which is recently proposed and shown to deliver the best results on several benchmark datasets is used [55]. RF is defined as [55]

$$\text{RF} = \log\left(2 + \frac{A}{C}\right) \tag{1}$$

where A and C, respectively, represent the number of documents which contain the term under concern in the positive and negative classes.

Table 1 The macro-F_1 and micro-F_1 scores obtained on three datasets when the number of negative samples is selected as the number of samples in the positive class and for $s = 2, 3, 4$ and 5

Score	Dataset	$s = 2$	$s = 3$	$s = 4$	$s = 5$	Equal no. of documents
Macro-F_1	Reuters-21578	90.64	90.60	90.34	89.90	82.08
	WebKB	85.97	84.92	83.73	82.22	70.21
	7-Sectors	86.95	87.16	86.37	84.88	73.13
Micro-F_1	Reuters-21578	94.83	94.62	94.41	94.12	90.60
	WebKB	87.51	86.01	84.48	82.69	68.70
	7-Sectors	87.02	87.24	85.96	83.30	75.35

The proposed approach referred as CATEGORY-UNANIMITY in the following context is compared with other well known approaches. Firstly, we studied the performance of a single random undersampling-based categorization system. This system which is referred as UNDERSAMPLING in the following context is trained and tested for ten times and the average scores are reported. The scheme is tested for eight undersampling rates, s in the interval [2, 9]. For instance, for $s = 3$, all training samples from the positive category and a random set made of one-third of the negative class training documents are used where the rest are not considered during training.

Secondly, random partitioning of the negative class is implemented where the negative class is split into s partitions. For instance, for $s = 3$, the negative class is randomly partitioned into three non-overlapping subsets. Each subset is used separately together with all positive documents to construct $M = 3$ different classifiers. For small categories, if s is small, the data may still be imbalanced. It may be argued that the number of samples in the negative class should depend on the number of samples in the minority (positive) class. For instance, the number of majority (negative) samples can be equal to that of the minority class. In fact, it is shown that this is not a trivial task since the best-fitting number of the majority class samples depends on the classification scheme and the problem under concern [60]. In order to investigate this, UNDERSAMPLING system is tested for equal number of positive and negative samples. The macro-F_1 and micro-F_1 scores obtained are presented in the last column of Table 1. The scores obtained for $s = 2, 3, 4$ and 5 are also presented. It can be seen in the table that choosing equal number of negative samples as the positive samples provides remarkably worse scores compared to using larger number of negatives.

In text categorization, it is shown that weighted voting performs better compared to plain voting [61]. In our simulations, we used the SVM scores as weights where a higher positive score is considered as a more confident positive decision and similarly a lower negative score is considered as a more confident negative decision. In binary text categorization, this weighted combination rule can be implemented simply by averaging the SVM scores. Hence, weighted voting (referred as PARTITION-W_VOTING) and unanimity rule (referred as PARTITION-UNANIMITY) are used for the aggregation of the outputs provided by the individual classifiers trained on disjoint partitions. In weighted voting-based fusion, the joint decision is positive if the average score is greater than zero.

In the implementation of the proposed system, M is dataset dependent since the number of categories included in the negative class varies in different datasets. In the case of Reuters-21578, since there are ten categories three of which are “Earn”, “Acquisition” and “Money-fx”, $M = 9$. For instance, in studying the binary problem of categorizing the test documents as belonging to “Earn” or not, nine classifiers are generated using the training data. The positive training samples of the first classifier are those having “Earn” as one of their labels whereas the negative class involves the samples which have “Acquisition” as one of their labels but not “Earn”. Similarly, the positive class is comprised of samples that have “Earn” as one of their labels for the second member classifier whereas the negative class involves the samples which have “Money-fx” as one of their labels. Other member classifiers are built in the same manner. The values of M are three and six, respectively, for WebKB and 7-Sectors datasets.

The macro-F_1 and micro-F_1 scores achieved on three different datasets are presented in Figs. 6 and 7, respectively, as functions of the sampling rate, s. The system that exploits all negative documents in model training denoted by SVM is also provided as a reference. The results clearly show that the highest macro-F_1 and micro-F_1 scores are achieved by the proposed approach on all three datasets. The results achieved should be evaluated for both the type of resampling and output aggregation method used. When CATEGORY-UNANIMITY and PARTITION-UNANIMITY are compared, it can easily be seen that the proposed approach provides higher macro-F_1 and micro-F_1 scores for all values of s. On WebKB dataset, PARTITION-UNANIMITY cannot provide a higher macro-F_1 or micro-F_1 score than the baseline system (SVM) for any value of s. On 7-Sectors, its macro-F_1 and micro-F_1 scores drop below that of SVM for large values of s. It can be concluded that, when unanimity rule is used, the proposed resampling approach provides better scores. This is in fact reasonable since, for deciding on the positive class, unanimity rule necessitates the use of individual classifiers having high recall. In other words, the number of misclassified positive documents (false negatives) should be

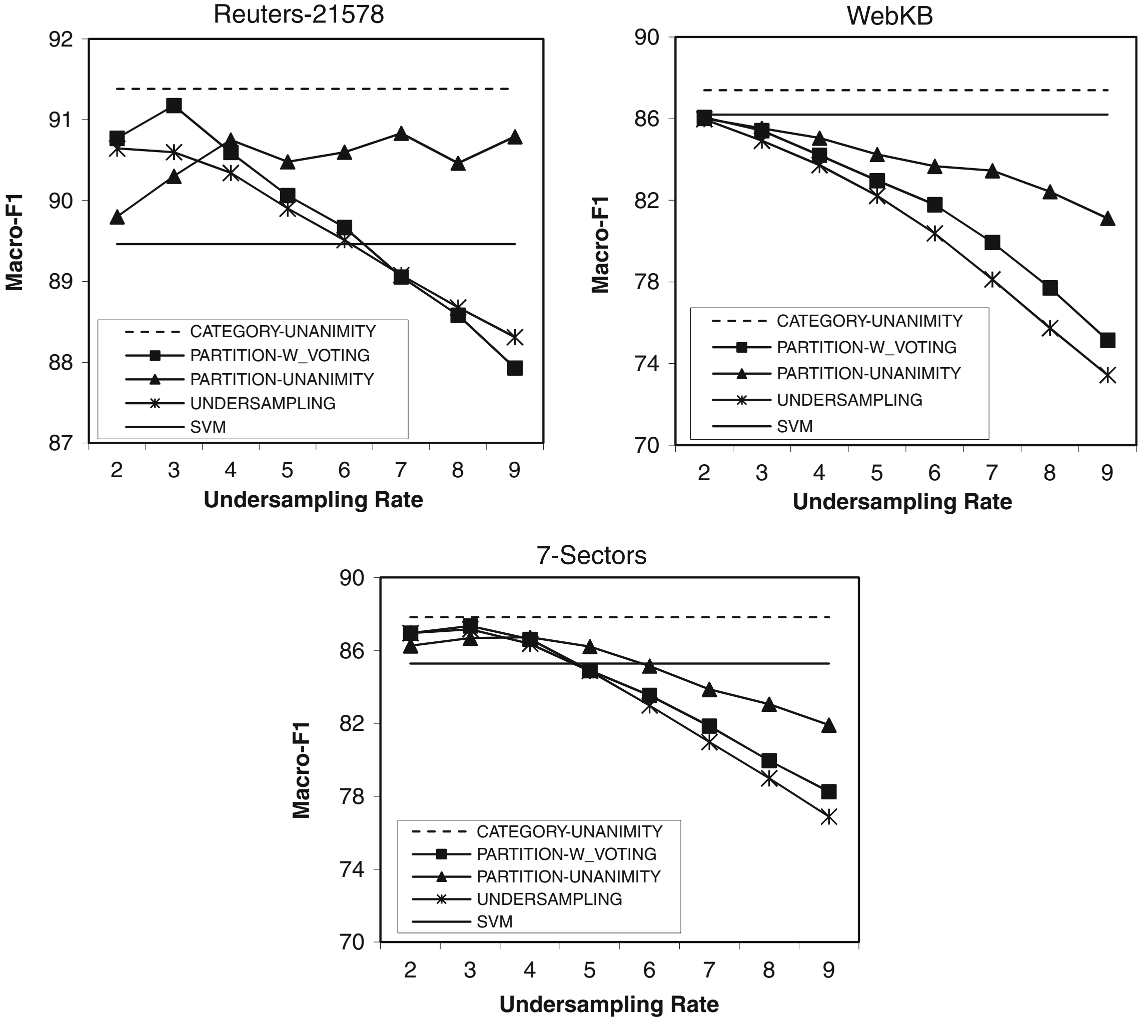

Fig. 6 The macro-F_1 scores computed on three datasets for eight different sampling ratios

very small. In the ideal case, unanimity rule requires all positive documents to be correctly classified by all member classifiers. On the other hand, it has very high tolerance to large number of false positives since it is adequate for one member classifier to provide correct decision for avoiding misclassification of a negative document as positive.

In order to investigate the recall values of the individual members, consider Fig. 8 which presents the average precision and recall values of nine individual members of category-based and random partitioning-based resampling schemes on Reuters-21578 dataset. As seen in the figure, the proposed category-based partitioning provides higher recall but lower precision values on all categories. Lower precision values of the individual members mean that the number of false positives is larger which is a consequence of having larger recall. This is mainly due to the fact that the data used to train the ensemble members of the category-based partitioning approach do not cover the whole feature space. However, as stated above, unanimity rule has high tolerance to larger number of false positives since a decision on the positive class is made only if all members agree on that class.

As it can be seen in Fig. 8, the recall values obtained using category-based partitioning are smaller on categories where the number of training samples is small such as “Wheat”, “Ship” and “Corn”. For the corresponding binary categorization problems, if the training data from the negative documents belong to a category involving large number of training samples, there may still be a bias toward the negative samples which explains the corresponding comparatively smaller recall values. On the other hand, if the tested category is large, the comparatively smaller number of samples in the negative class corresponding to a small category cannot form a bias.

As it can be seen in Fig. 6, the macro-F_1 scores provided by PARTITION-W_VOTING and PARTITION-UNANIMITY generally degrade when four or more members are considered. However, PARTITION-UNANIMITY is

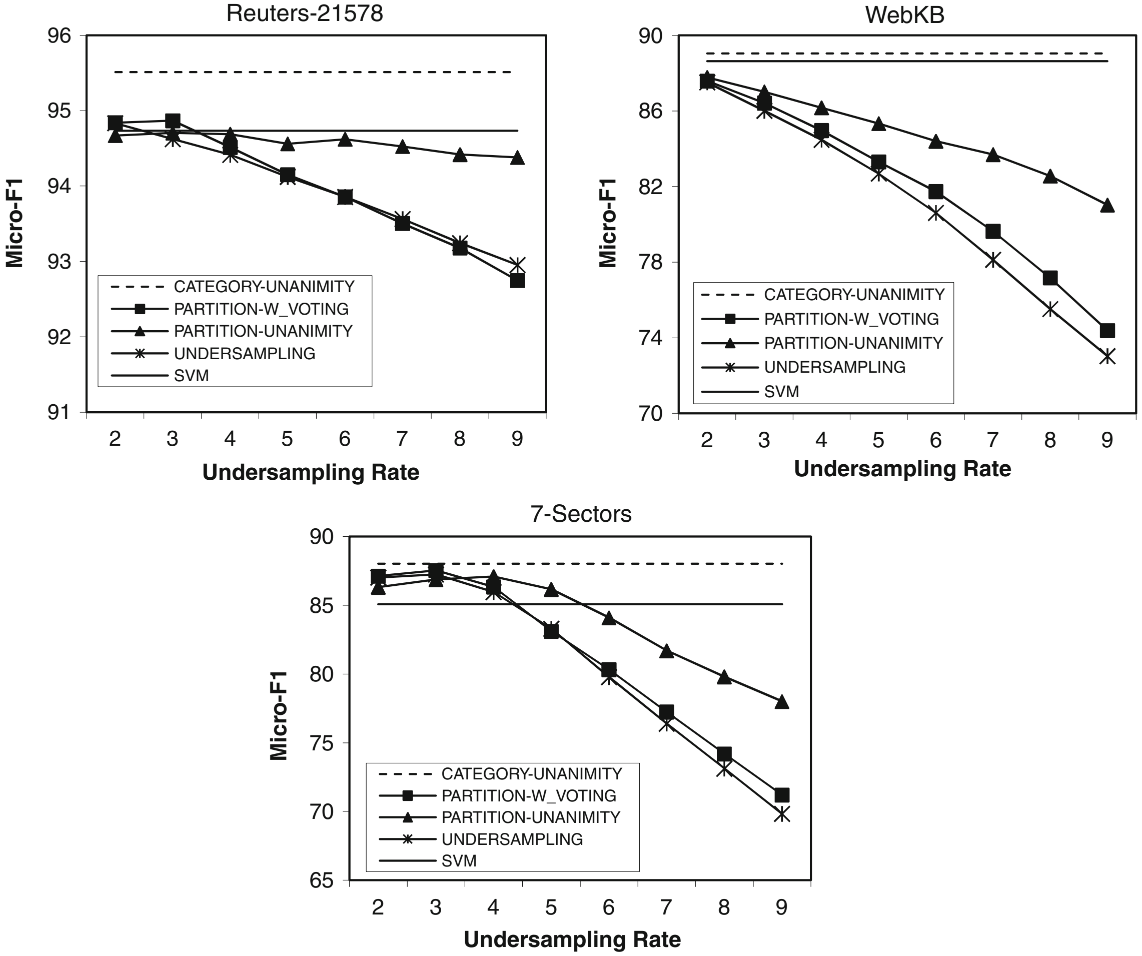

Fig. 7 The micro-F_1 scores computed on three datasets for eight different sampling ratios

more robust compared to PARTITION-W_VOTING. For a better understanding of this behavior, consider the macro-precision and macro-recall values presented in Fig. 9 which are computed as the averages of the corresponding scores obtained from the independent binary categorization tasks of each dataset. As it can be seen in the figure, the recall values obtained by using PARTITION-W_VOTING are higher than those of PARTITION-UNANIMITY for all values of s. The robustness of F_1 is achieved as a consequence of the robustness of precision to the increasing values of s which is the main characteristic of unanimity rule.

In order to get further insights about the differences among the schemes, consider the F_1 scores computed for each category as given in Table 2. The categories at the top ten rows are from Reuters-21578 dataset. The following four categories are from WebKB dataset. On fourteen categories out of twenty one, unanimity-based systems (PARTITION-UNANIMITY or CATEGORY-UNANIMITY) provide the best F_1 scores where, on eleven of these, the proposed approach (CATEGORY-UNANIMITY) provides the best F_1. On eighteen categories, PARTITION-W_VOTING performs better than UNDERSAMPLING. Nevertheless, the performance improvements are not significant.

PARTITION-W_VOTING achieves its best F_1 scores when s is equal to 2 or 3 on seventeen categories. As the number of partitions is increased, the performance generally degrades. However, PARTITION-UNANIMITY provides its best F_1 score using a larger number of splits compared to PARTITION-W_VOTING on eleven categories. This can be explained by considering the fact that, as mentioned above, unanimity rule requires classifiers having high recall which can be generated by using a smaller number of negative documents in their training.

For a further comparison of unanimity- and voting-based schemes, we investigated their performances on category-based undersampling. The conventionally used maximum-voting scheme is employed where the total number of the positive and negative predictions are considered in making the decision. This scheme is named as CATEGORY-MAX_VOTING. It should be noted that, as illustrated in Fig. 4, maximum-voting performs better compared to unanimity rule when the member classifiers

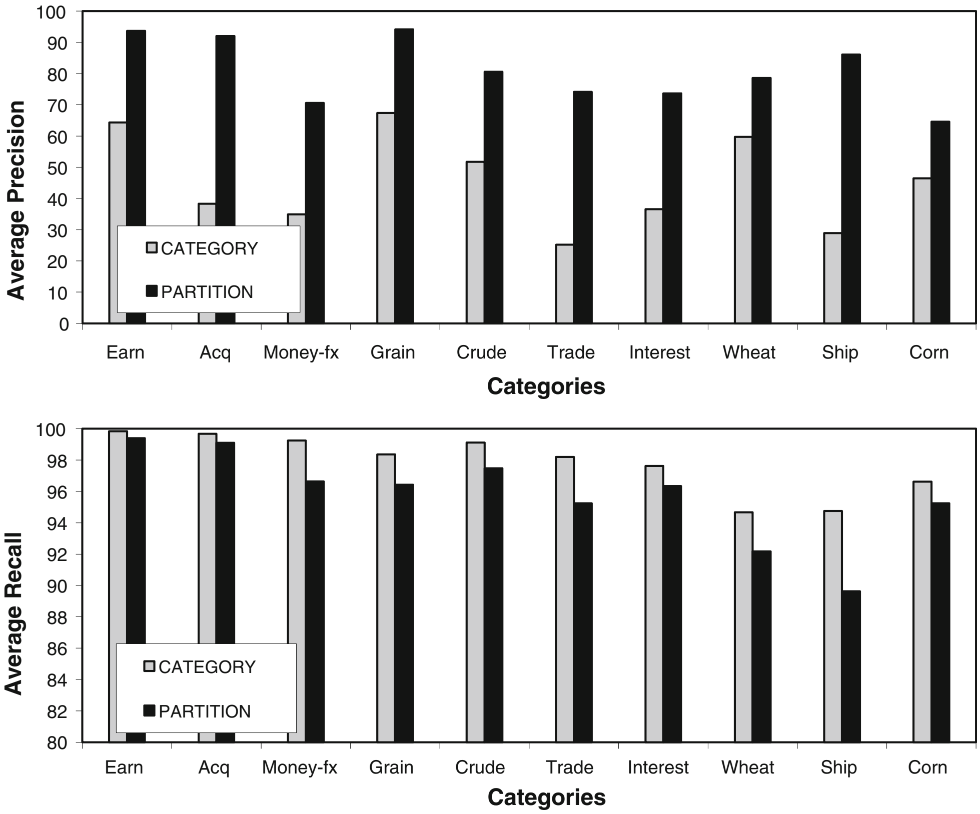

Fig. 8 The average precision and recall values of nine individual members of category-based and random partitioning-based ($s = 9$) resampling schemes

achieve high precision values. Since category-based undersampling provides smaller precision values compared to random partitioning as shown in Fig. 8, the unanimity rule is expected to surpass maximum-voting rule. The macro-F_1 and micro-F_1 scores achieved on three datasets are presented in Table 3. As expected, the scores achieved by the unanimity rule are significantly superior to those provided by the maximum-voting-based scheme.

In order to assess the statistical significance of the improvements in the F_1 scores provided by the proposed approach, hypothesis tests are performed using the t-test approach. The null hypothesis is defined as "$H_0 =$ mean of the improvement is equal to zero" and the alternative hypothesis is defined as "$H_1 =$ mean of the improvement is greater than zero". The tests are performed between CATEGORY-UNANIMITY and baseline SVM-based system. The null hypothesis is rejected at significance level of $\alpha = 0.05$, with p values 0.0118 and 0.0057 for Reuters-21578 and 7-Sectors, respectively. On the other hand, PARTITION-W_VOTING provides statistically significant improvements only on Reuters-21578 only for $s = 2$ whereas UNDERSAMPLING does not achieve significant improvements on any of the datasets under concern for any s value. Significance tests are similarly performed between CATEGORY-UNANIMITY and PARTITION-UNANIMITY for which the s values providing the highest macro-F_1 scores are considered for each dataset. More specifically, the values of s are 7, 2, and 4, respectively, for Reuters-21578, WebKB and 7-Sectors for PARTITION-UNANIMITY system as seen in Fig. 6. The null hypothesis is rejected at significance level of $\alpha = 0.05$, with p values 0.0291 and 0.0273 for WebKB and 7-Sectors, respectively.

In a recent study, it is observed that the relative performances of different schemes may not be consistent when different performance measures are used [16]. In particular, when AUP is considered, SVM trained by the whole training set which is considered as the baseline provided the best scores on two of the three datasets when compared to various undersampling, oversampling or instance weighting-based schemes. On the other hand, its performance came out to be the worst on all three datasets in terms of macro-F_1 which is also consistent with the results of our work. This observation was explained by the fact that the precision-recall curve characterizes the performance of the classifier for different thresholds on the prediction values whereas the F_1 score employs the precision and recall obtained at the default threshold that is zero. By tuning the threshold on the *test data*, they have shown that the baseline SVM becomes the best in terms of the macro-F_1 score, concluding that threshold setting is still an open problem. In this study, the proposed scheme is also evaluated in terms of AUP score and the scores achieved are compared with the baseline SVM which is recently found to provide the best scores by Sun et al. [16]. Table 4

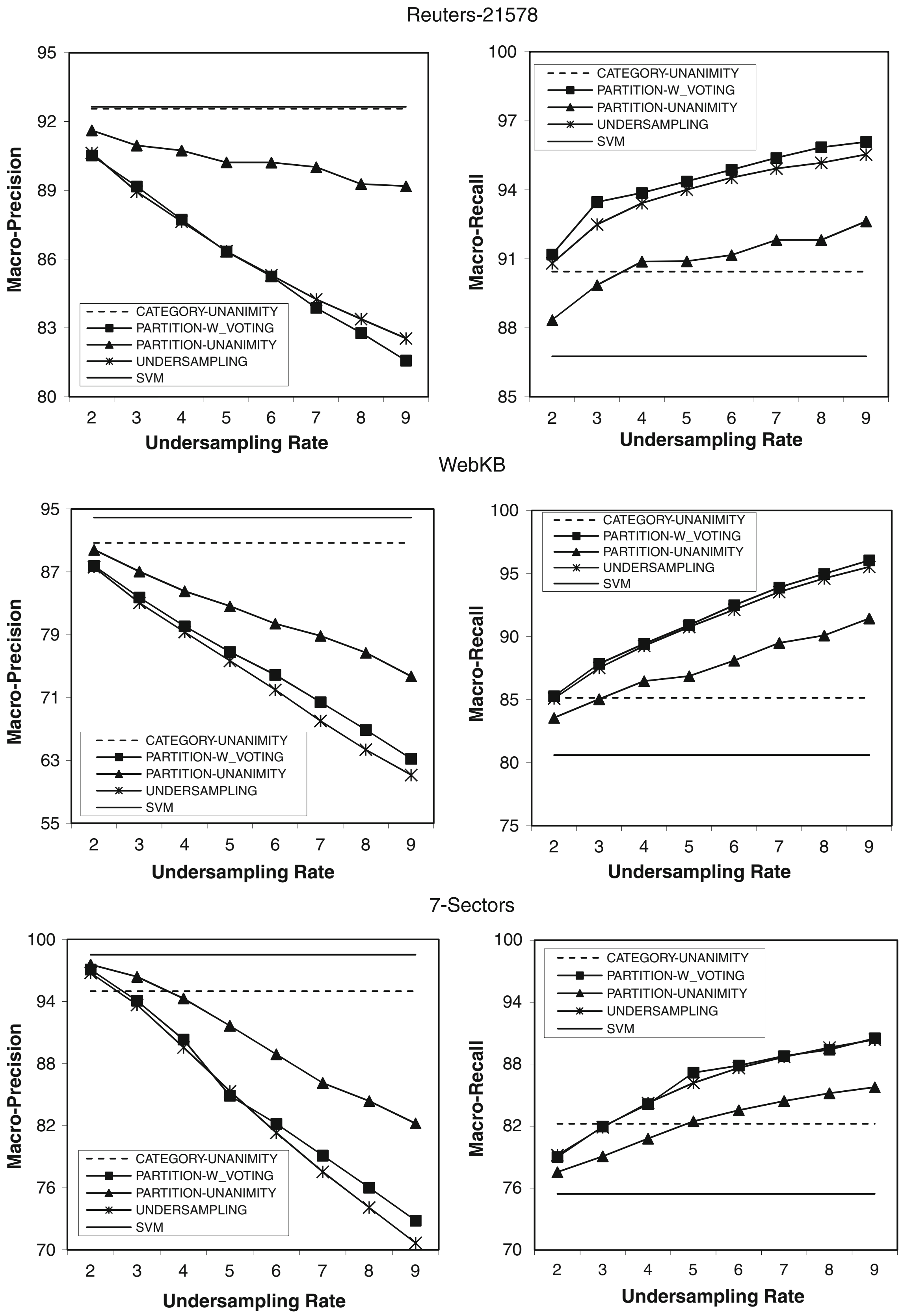

Fig. 9 The macro-precision and macro-recall values computed on three datasets for eight different undersampling rates

Table 2 The F_1 scores obtained for the individual categories of Reuters-21578, WebKB and 7-Sectors datasets

Category	SVM	UNDER-SAMPLING	PARTITION-W_VOTING	PARTITION-UNANIMITY	CATEGORY-UNANIMITY
Earn	98.58	$98.07^{(s=2)}$	$98.00^{(s=2)}$	$98.17^{(s=2)}$	**98.72**
Acquisition	97.47	$97.68^{(s=2)}$	$97.70^{(s=2)}$	$97.69^{(s=2,3)}$	**97.76**
Money-fx	86.81	$87.01^{(s=2)}$	$87.86^{(s=3)}$	$86.02^{(s=8)}$	**89.89**
Grain	96.58	$96.05^{(s=2)}$	$96.62^{(s=2,3)}$	**96.95**$^{(s=2)}$	96.22
Crude	**92.35**	$90.73^{(s=2)}$	$90.59^{(s=3)}$	$91.79^{(s=3)}$	**92.35**
Trade	87.11	$86.83^{(s=3)}$	$87.14^{(s=3)}$	$86.67^{(s=9)}$	**88.51**
Interest	81.97	$86.98^{(s=3)}$	**87.82**$^{(s=3)}$	$87.55^{(s=6)}$	86.51
Wheat	84.67	$87.73^{(s=4)}$	**88.44**$^{(s=3)}$	$88.11^{(s=5)}$	85.51
Ship	82.28	$88.36^{(s=4)}$	$89.02^{(s=4)}$	**89.41**$^{(s=7,9)}$	88.62
Corn	86.79	$89.24^{(s=2)}$	$89.66^{(s=3)}$	**91.89**$^{(s=7)}$	89.72
Student	**91.24**	$88.15^{(s=2)}$	$88.13^{(s=2)}$	$88.62^{(s=2)}$	89.75
Faculty	84.52	$84.52^{(s=2)}$	$84.56^{(s=2)}$	$84.63^{(s=2)}$	**86.93**
Course	95.45	$95.81^{(s=2)}$	$95.91^{(s=2)}$	$95.87^{(s=2)}$	**96.07**
Project	73.58	$75.41^{(s=2)}$	$75.90^{(s=3)}$	$75.54^{(s=7)}$	**76.83**
Technology	83.76	$86.13^{(s=3)}$	$86.54^{(s=3)}$	$86.15^{(s=3)}$	**87.43**
Financial	86.37	$88.00^{(s=3)}$	$88.29^{(s=2,3)}$	$88.70^{(s=4)}$	**90.57**
Basic materials	81.14	$86.07^{(s=3)}$	**86.45**$^{(s=3)}$	$85.97^{(s=4)}$	85.48
Transportation	91.50	$92.12^{(s=4)}$	**92.31**$^{(s=5)}$	$92.02^{(s=8)}$	91.72
Healthcare	87.35	$87.77^{(s=2)}$	**87.83**$^{(s=2)}$	$87.52^{(s=2)}$	87.08
Energy	86.23	$88.49^{(s=3)}$	**89.20**$^{(s=7)}$	$88.43^{(s=9)}$	88.69
Utilities	80.59	$82.22^{(s=6)}$	$83.15^{(s=9)}$	$82.31^{(s=8)}$	**83.78**

The best score of each category is presented in boldface

Table 3 The macro-F_1 and micro-F_1 scores obtained on three datasets using unanimity and maximum-voting-based schemes to combine classifiers obtained using category-based undersampled sets

Score	Dataset	CATEGORY-MAX_VOTING	CATEGORY-UNANIMITY
Macro-F_1	Reuters-21578	62.74	91.38
	WebKB	71.99	87.39
	7-Sectors	73.03	87.82
Micro-F_1	Reuters-21578	62.54	95.51
	WebKB	70.33	89.04
	7-Sectors	66.70	88.02

Table 4 The average AUP scores obtained on three datasets

Dataset	SVM	CATEGORY-UNANIMITY
Reuters-21578	0.9563	**0.9626**
WebKB	0.9436	**0.9464**
7-Sectors	**0.9673**	0.9601

The best score of each category is presented in boldface

presents the average AUP scores for each dataset where the best scores are typed in boldface. As it can be seen in the table, the proposed scheme provided the best scores on two of the three datasets utilized in this study.

It should be noted that Kumar and Gopal have recently shown that one-versus-rest approach provides better scores compared to one-versus-one approach in multi-class text categorization [62]. The one-versus-rest approach solves the multi-class problem by defining multiple binary classification problems where the negative class of each problem is defined as the union of the documents from all other categories. Instead of using the union of the documents from all other categories as the negative class, the proposed scheme generates a different binary classifier for each negative category whose outputs are then combined. As a matter of fact, the proposed scheme is also a potential candidate to solve each binary classification problem in multi-class text categorization.

5 Conclusions and future work

In this study, an analytical investigation of unanimity and majority voting rules is presented. It is shown that unanimity rule can provide better F_1 scores compared to majority voting when an ensemble of high recall but low precision classifiers is considered. On the other hand, it is shown that majority voting provides better F_1 score if both precision and recall are high. Then, in order to generate high recall members, category-based undersampling is proposed where the number of subsets from the negative class is selected as the number of categories it includes. Combination of the classifiers generated using category-based undersampling by unanimity rule is studied next.

Experiments conducted on three datasets have shown that random undersampling method provides individual members having higher precision than members obtained using category-based undersampling whereas the latter approach provides members having higher recall. When applied on the members trained using category-based subsets of the negative documents, unanimity rule is shown to attain a better precision-recall trade-off compared to the baseline SVM-based system, random undersampling approach and weighted voting on the outputs of the ensemble members generated using randomly partitioned negative data and, improves F_1 score at satisfactory levels.

Hereafter, there are several tasks awaiting us. Firstly, in order to achieve better F_1 scores, the proposed undersampling scheme can be modified for small target categories. The category selected from the negative class might include many more documents compared to the target category. Splitting the category-based negative sample sets into smaller groups may be considered as a solution for such cases. Therefore, studying the effect of the number of samples in the subsets of the negative class will be considered for future work. The class imbalance problem is extensively studied in the literature where numerous resampling schemes are proposed. Secondly, the relative performance of unanimity and majority voting should be studied for these alternative schemes to obtain a comprehensive experimental evaluation.

Acknowledgments The numerical calculations reported in this paper were partly performed at the ULAKBIM High Performance Computing Center of the Turkish Scientific and Technical Research Council (TUBITAK). This work was supported by the research grant MEKB-09-02 provided by the Ministry of Education and Culture of Northern Cyprus.

Appendix

Formulation of the optimization problem

The notation used for the analytical evaluation in similar to the one that was used in one of the co-author's previous studies [63]. Let the positive and negative classes be labeled as '1' and '0', respectively. Assume that each classifier is trained using a different subset of the negative documents. Given a test document, the output of a classifier is 1 when the decision is on the positive class and 0 when the decision is on the negative class. Assume that $\mathbf{X}$ is the ordered set of M-dimensional joint output vectors, x_i where the elements of the vectors are the decisions of the individual classifiers where $|\mathbf{X}| = 2^M$, $|.|$ denoting the cardinality of the set. As an example, for the case of three classifiers, $\mathbf{X} = \{[0\,0\,0], [0\,0\,1], \ldots, [1\,1\,1]\}$. The second element of $\mathbf{X}$, $x_2 = [0 \;\; 0 \;\; 1]$ corresponds the case when the first two classifiers decide on the negative class and the third classifier decides on the positive class. The conventional way of computing the joint decision is to apply majority voting on the outputs of the individual classifiers.

Consider a simple example of three classifiers as given in Table 5 which summarizes the probabilities and the decisions corresponding to all possible classifier outputs for a 3-classifier system for both majority voting and unanimity rule. As seen in the table, the joint decision of unanimity rule is 1 only when the individual decisions of all classifiers are 1. On the other hand, majority voting provides 1 for four different combinations of the classifier outputs. p_i denotes the probability that x_i is the joint classifier output when the tested document is from the positive class. Similarly, q_i denotes the probability that x_i is the joint classifier output when the tested document is from the negative class.

Assume that three classifiers, each having the precision value, p and the recall value, r are considered where, using various resampling schemes, different joint distributions can be obtained. In order to identify the major characteristics and relative performances of the two combination schemes under concern, let us compute the highest F_1 scores that can be achieved by each scheme when the classifiers used have the same precision and recall values. This is equivalent to computing the set of p_i and q_i values that provide the highest combined F_1 score. This can be formulated as an optimization problem with some constraints. The first constraint is the recall values of the individual classifiers. It should be noted that the recall of a

Table 5 All possible outcomes of a 3-classifier ensemble and its probability distribution

	[0 0 0]	[0 0 1]	[0 1 0]	[0 1 1]	[1 0 0]	[1 0 1]	[1 1 0]	[1 1 1]
Majority voting	0	0	0	1	0	1	1	1
Unanimity	0	0	0	0	0	0	0	1
Class 1	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8
Class 0	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8

classifier is defined as the percentage of correctly classified positive documents *TP*/(*TP* + *FN*) where *TP* denotes the number of true positives and *FN* denotes the number of false negatives. Assuming the same individual recall values for all classifiers, we obtain

$$\begin{aligned}
&p_5 + p_6 + p_7 + p_8 = r \\
&p_3 + p_4 + p_7 + p_8 = r \\
&p_2 + p_4 + p_6 + p_8 = r
\end{aligned} \qquad (2)$$

respectively, for the first, second and the third classifier. The individual precision values are assumed to be p. The precision which is defined as the percentage of documents that are correctly labeled as positive can be computed as TP/(TP + FP), where FP denotes the number of false positives. Without any loss of generality, assume that the number of test documents in the positive class, N_{pos} and in the negative class, N_{neg} are the same. Then, for the first classifier, $\text{TP} = N_{\text{pos}} \times (p_5 + p_6 + p_7 + p_8)$ and $\text{FP} = N_{\text{neg}} \times (q_5 + q_6 + q_7 + q_8)$. Hence,

$$\frac{N_{\text{pos}} \times (p_5 + p_6 + p_7 + p_8)}{N_{\text{pos}} \times (p_5 + p_6 + p_7 + p_8) + N_{\text{neg}} \times (q_5 + q_6 + q_7 + q_8)} \\ = p \qquad (3)$$

Since $N_{\text{pos}} = N_{\text{neg}}$, the following three constraints are obtained for three classifiers.

$$\begin{aligned}
&\frac{(p_5 + p_6 + p_7 + p_8)}{(p_5 + p_6 + p_7 + p_8) + (q_5 + q_6 + q_7 + q_8)} = p \\
&\frac{(p_3 + p_4 + p_7 + p_8)}{(p_3 + p_4 + p_7 + p_8) + (q_3 + q_4 + q_7 + q_8)} = p \\
&\frac{(p_2 + p_4 + p_6 + p_8)}{(p_2 + p_4 + p_6 + p_8) + (q_2 + q_4 + q_6 + q_8)} = p
\end{aligned} \qquad (4)$$

Since the numerators of the three equations given above in Eq. 4 are equal to r as given in Eq. 2, they can be re-written as,

$$\begin{aligned}
&q_5 + q_6 + q_7 + q_8 = \frac{r(1-p)}{p} \\
&q_3 + q_4 + q_7 + q_8 = \frac{r(1-p)}{p} \\
&q_2 + q_4 + q_6 + q_8 = \frac{r(1-p)}{p}
\end{aligned} \qquad (5)$$

In the text categorization problem, F_1 score which is the harmonic mean of precision and recall is generally used as the objective function since either precision or recall can be maximized at the expense of the other. The F_1 score is defined as

$$F_1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + r\text{recall}}. \qquad (6)$$

For the system presented in Table 5, the precision, recall and the F_1 score of the *combined system* using unanimity rule can be computed as

$$\begin{aligned}
\text{precision} &= \frac{p_8}{p_8 + q_8} \\
\text{recall} &= p_8 \\
F_1^{\text{una}} &= \frac{2 \times p_8}{p_8 + q_8 + 1}
\end{aligned} \qquad (7)$$

This can be generalized to M classifiers case as follows:

$$\begin{aligned}
\text{precision} &= \frac{p_{(2^M)}}{p_{(2^M)} + q_{(2^M)}} \\
\text{recall} &= p_{(2^M)} \\
F_1^{\text{una}} &= \frac{2 \times p_{(2^M)}}{p_{(2^M)} + q_{(2^M)} + 1}
\end{aligned} \qquad (8)$$

The precision, recall and the F_1 score of the *combined system* using majority voting rule can be computed as

$$\begin{aligned}
\text{precision} &= \frac{p_4 + p_6 + p_7 + p_8}{p_4 + p_6 + p_7 + p_8 + q_4 + q_6 + q_7 + q_8} \\
\text{recall} &= p_4 + p_6 + p_7 + p_8 \\
F_1^{\text{mv}} &= \frac{2 \times (p_4 + p_6 + p_7 + p_8)}{p_4 + p_6 + p_7 + p_8 + q_4 + q_6 + q_7 + q_8 + 1}
\end{aligned} \qquad (9)$$

The maximum F_1^{una} which can be obtained using the member having individual precision and recall constraints can be obtained as the solution of the following optimization problem:

$$\begin{aligned}
&\text{Max}\, F_1^{\text{una}} = \frac{2 \times p_8}{p_8 + q_8 + 1} \\
&\text{Subject to,} \\
&\quad p_5 + p_6 + p_7 + p_8 = r \quad q_5 + q_6 + q_7 + q_8 = \tfrac{r(1-p)}{p} \\
&\quad p_3 + p_4 + p_7 + p_8 = r \quad q_3 + q_4 + q_7 + q_8 = \tfrac{r(1-p)}{p} \\
&\quad p_2 + p_4 + p_6 + p_8 = r \quad q_2 + q_4 + q_6 + q_8 = \tfrac{r(1-p)}{p} \\
&\quad \sum_{i=1}^{8} p_i = 1 \qquad\qquad\quad \sum_{i=1}^{8} q_i = 1 \\
&\quad p_i, q_i \geq 0, i = 1, \ldots, 8.
\end{aligned} \qquad (10)$$

The problem can be expressed in a more compact form as follows:

$$\begin{aligned}
&\text{Max}\, F_1^{\text{una}} = \frac{2 \times p_8}{p_8 + q_8 + 1} \\
&\text{Subject to,} \\
&A\mathbf{s} = \mathbf{b} \\
&\mathbf{s} \geq 0,
\end{aligned} \qquad (11)$$

where $\mathbf{s} = [p_1 p_2 \ldots p_8 q_1 q_2 \ldots q_8]^T$ is the solution vector and

$$A = \begin{bmatrix} 0&0&0&0&1&1&1&1&0&0&0&0&0&0&0&0 \\ 0&0&1&1&0&0&1&1&0&0&0&0&0&0&0&0 \\ 0&1&0&1&0&1&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1 \\ 0&0&0&0&0&0&0&0&0&0&1&1&0&0&1&1 \\ 0&0&0&0&0&0&0&0&0&1&0&1&0&1&0&1 \\ 1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1 \end{bmatrix}$$

$$\mathbf{b} = \begin{bmatrix} r \\ r \\ r \\ \frac{r(1-p)}{p} \\ \frac{r(1-p)}{p} \\ \frac{r(1-p)}{p} \\ 1 \\ 1 \end{bmatrix} \tag{12}$$

The last two rows of the matrix A correspond to the constraints $\sum_{i=1}^{8} p_i = 1$ and $\sum_{i=1}^{8} q_i = 1$, respectively.

The maximum F_1^{mv} which can be obtained using the member having individual precision and recall constraints can be similarly obtained as the solution of the following optimization problem:

$$\begin{aligned} &\mathrm{Max}\, F_1^{\mathrm{mv}} = \frac{2 \times (p_4 + p_6 + p_7 + p_8)}{p_4 + p_6 + p_7 + p_8 + q_4 + q_6 + q_7 + q_8 + 1} \\ &\text{Subject to,} \\ &A\mathbf{s} = \mathbf{b} \\ &\mathbf{s} \geq 0, \end{aligned} \tag{13}$$

where A and $\mathbf{b}$ are as defined in Eq. 12.

Proof of Proposition 1

The matrix A can be written as

$$A = \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \\ [1\,1 \ldots 1] & [0\,0 \ldots 0] \\ [0\,0 \ldots 0] & [1\,1 \ldots 1] \end{bmatrix}. \tag{14}$$

A_1 and A_4 are $M \times 2^M$ matrices where ith columns are x_i^T. On the other hand, A_2 and A_3 are also $M \times 2^M$ matrices where all entries are zero. Since $r = 1$, a feasible solution should be in the form $[0, \ldots, 0, 1, q_1, q_2, \ldots, q_{(2^M)}]^T$. The solution should satisfy the constraint $A_4 \times [q_1, q_2, \ldots, q_{(2^M)}]^T = [\frac{(1-p)}{p}, \ldots, \frac{(1-p)}{p}]$. For $F_1^{\mathrm{una}} = 1$, a feasible solution should have $q_{(2^M)} = 0$ and $q_i \geq 0 \;\forall\; i \neq 2^M$ as it can be seen in Eq. 7. For simplicity, the feasible solution is constructed by following the steps applied in proving Theorem 1 in [63] where only some of q_i's are assumed to have a nonzero value. More specifically, let k denote the number of columns in A_4 which have exactly $(M + 1)/2$ number of ones which can be computed as

$$k = \frac{M!}{(\frac{M+1}{2})!(\frac{M-1}{2})!}. \tag{15}$$

Assume that these columns are numbered as $i_1, i_2, \ldots, i_k$. In the case of $M = 3$, $k = 3$ and $i_1 = 4$, $i_2 = 6$ and $i_3 = 7$. Let us put these columns into a matrix named as $\tilde{A_4}$. The total number of ones in the matrix $\tilde{A_4}$ is $l = k \times (M + 1)/2$. Due to its uniform structure, the number of ones in each row of $\tilde{A_4}$ that is the same for all rows can be computed as

$$m = \frac{l}{M} = \frac{k \times (M+1)}{2M} \tag{16}$$

since there are M rows. Using Eq. 15 we can compute m as

$$m = \frac{(M-1)!}{(\frac{M-1}{2})!(\frac{M-1}{2})!}. \tag{17}$$

A feasible solution can be obtained as $q_1 = \beta$ and $q_{i_1}, q_{i_2}, \ldots, q_{i_k} = \alpha$ where α and β should satisfy the equations

$$\begin{aligned} m\alpha &= \frac{1-p}{p} \\ k\alpha + \beta &= 1 \end{aligned} \tag{18}$$

where the second equation is obtained by considering the fact that the sum of all probabilities should be equal to one. Simultaneous solution of these equations for β gives

$$\beta = \frac{p(m+k) - k}{mp} \tag{19}$$

Since $\beta \geq 0$, we should have $p \geq \frac{k}{m+k}$ to achieve perfect combined system (i.e., $F_1 = 1$). Using Eq. 16 to replace m, $p \geq \frac{2M}{3M+1}$ is obtained.

Proof of Proposition 2

Consider the decomposition of matrix A into sub-matrices in Eq. 14. Since $r = 1$, a feasible solution is of the form $[0, \ldots, 0, 1, q_1, q_2, \ldots, q_{(2^M)}]^T$. This means that $p_4 = p_6 = p_7 = 0$. The solution should satisfy the constraint $A_4 \times [q_1, q_2, \ldots, q_{(2^M)}]^T = [\frac{(1-p)}{p}, \ldots, \frac{(1-p)}{p}]$. Note that, for $F_1^{\mathrm{mv}} = 1$, we need to have $q_4 = q_6 = q_7 = q_8 = 0$ as seen in Eq. 13. Let k denote the number of columns in A_4 which have exactly $(M - 1)/2$ number of ones which can be computed as

$$k = \frac{M!}{(\frac{M+1}{2})!(\frac{M-1}{2})!}. \tag{20}$$

Assume that these columns are numbered as $i_1, i_2, \ldots, i_k$. In the case of $M = 3$, $k = 3$ and $i_1 = 2$, $i_2 = 3$ and $i_3 = 5$. Let us put these columns into a matrix named as $\tilde{A_4}$. The total number of ones in the matrix $\tilde{A_4}$ is $l = k \times (M - 1)/2$. Due to its uniform structure, the number of ones in each row of $\tilde{A_4}$ that is same for all rows can be computed as

$$m = \frac{l}{M} = \frac{k \times (M-1)}{2M} \tag{21}$$

or, similarly

$$m = \frac{(M-1)!}{(\frac{M+1}{2})!(\frac{M-3}{2})!}. \tag{22}$$

A feasible solution can be obtained as $q_1 = \beta$ and $q_{i_1}, q_{i_2}, \ldots, q_{i_k} = \alpha$ where α and β should satisfy the equations

$$\begin{aligned} m\alpha &= \frac{1-p}{p} \\ k\alpha + \beta &= 1 \end{aligned} \tag{23}$$

where the second equation is obtained by considering the fact that the sum of all probabilities should be equal to one. Simultaneous solution of these equations for β gives

$$\beta = \frac{p(m+k) - k}{mp} \tag{24}$$

Since $\beta \geq 0$, we should have $p \geq \frac{k}{m+k}$ to achieve perfect combined system (i.e., $F_1 = 1$). Using Eq. 21 to replace m, $p \geq \frac{2M}{3M-1}$ is obtained.

References

1. Sebastiani F (2002) Machine learning in automated text categorization. ACM Comput Surv 34(1):1–47
2. Jiang C, Coenen F, Sanderson R, Zito M (2010) Text classification using graph mining-based feature extraction. Knowl-Based Syst 23(4):302–308
3. Selamat A, Omatu S (2004) Web page feature selection and classification using neural networks. Inf Sci 158:69–88
4. Joachims T (1997) A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. In: Proceedings of 14th international conference on machine learning, pp 143–151
5. Chen J, Huang H, Tian S, Qu Y (2009) Feature selection for text classification with naive Bayes. Expert Syst Appl 36:5432–5435
6. Lu SH, Chiang DA, Keh HC, Huang HH (2010) Chinese text classification by the naïve Bayes classifier and the associative classifier with multiple confidence threshold values. Knowl-Based Syst 23(6):598–604
7. Tan S (2005) Neighbor-weighted k-nearest neighbor for unbalanced text corpus. Expert Syst Appl 28:667–671
8. Joachims T (1998) Text categorization with support vector machines : Learning with many relevant features. In: Proceedings of 10th European conference of machine learning, pp 137–142
9. Leopold E, Kindermann J (2002) Text categorization with support vector machines. how to represent texts in input space? Mach Learn 46(1–3):423–444
10. Liu Z, Lv X, Liu K, Shi S (2010) Study on SVM compared with the other text classification methods. In: Proceedings of the 2010 second international workshop on education technology and computer science, March 2010
11. Basu A, Watters C, Shepherd M (2003) Support vector machines for text categorization. In: Proceedings of the 36th Hawaii international conference on system sciences, January 2003
12. Estabrooks A, Japkowicz N (2001) A mixture-of-experts framework for text classification. In: ConLL ’01: proceedings of the 2001 workshop on computational natural language learning. Association for Computational Linguistics, Morristown, pp 1–8
13. Estabrooks A, Jo T, Japkowicz N (2004) A multiple resampling method for learning from imbalanced data sets. Comput Intell 20(1):18–36
14. Imam T, Ting KM, Kamruzzaman J (2006) z-SVM: an SVM for improved classification of imbalanced data. In: Australian conference on artificial intelligence, pp 264–273
15. Liu X, Wu J, Zhou Z (2009) Exploratory undersampling for class-imbalance learning. IEEE Trans Syst Man Cyber B 39(2):539–550
16. Sun A, Lim E, Liu Y (2009) On strategies for imbalanced text classification using svm: a comparative study. Decis Support Syst 48(1):191–201
17. Li X, Yan Y, Peng Y (2009) The method of text categorization on imbalanced datasets. In: Proceedings of the 2009 international conference on communication software and networks, February 2009
18. He H, Garcia EA (2009) Learning from imbalanced data. IEEE Trans Knowl Data Eng 21(9):1263–1284
19. Visa S and Ralescu A (2005) Issues in mining imbalanced data sets—a review paper. In: Proceedings of the sixteen midwest artificial intelligence and cognitive science conference, pp 67–73
20. Tian J, Gu H, Liu W (2011) Imbalanced classification using support vector machine ensemble. Neural Comput Appl 20:203–209
21. Zhang J and Mani I (2003) KNN approach to unbalanced data distributions: a case study involving information extraction. In: Proceedings of the international conference on machine learning (ICML-2003)
22. Yen SJ, Lee YS (2009) Cluster-based under-sampling approaches for imbalanced data distributions. Expert Syst Appl 36(3):5718–5727
23. Liu AY (2004) The effect of oversampling and undersampling on classifying imbalanced text datasets. Master’s thesis, Graduate School of The University of Texas at Austin
24. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) Smote: synthetic minority over-sampling technique. J Artif Intell Res 16:321–357
25. Drummond C, Holte RC (2003) C4.5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In: Proceedings of the ICML’03 workshop on learning from imbalanced datasets, August 2003
26. Li C (2007) Classifying imbalanced data using a bagging ensemble variation (BEV). In: ACM-SE 45: proceedings of the 45th annual southeast regional conference. ACM, New York, pp 203–208
27. Dong YS, Han KS (2004) A comparison of several ensemble methods for text categorization. In IEEE international conference on services computing. IEEE Computer Society, Los Alamitos, pp 419–422

28. Dong YS, Han KS (2005) Boosting svm classifiers by ensemble. In: WWW '05: special interest tracks and posters of the 14th international conference on World Wide Web. ACM, New York, pp 1072–1073
29. Lin SC, Chang YI, Yang WN (2009) Meta-learning for imbalanced data and classification ensemble in binary classification. Neurocomputing 73(1–3):484–494
30. Hulse JV, Khoshgoftaar TM, Napolitano A (2009) An empirical comparison of repetitive undersampling techniques. In: IRI'09: Proceedings of the 10th IEEE international conference on information reuse & integration. IEEE Press, Piscataway, pp 29–34
31. Yan R, Liu Y, Jin R, Hauptmann A (2003) On predicting rare classes with svm ensembles in scene classification. In: ICASSP, pp 21–24
32. Ricamoto MT, Marrocco C, Tortorella F (2008) MCS-based balancing techniques for skewed classes: an empirical comparison. In: Proceedings of the 19th international conference on pattern recognition, (ICPR2008), December 2008
33. Yoon K, Kwek S (2007) A data reduction approach for resolving the imbalanced data issue in functional genomics. Neural Comput Appl 16:295–306
34. Chawla NV, Lazarevic A, Hall LO, Bowyer KW (2003) SMOTEBoost: improving prediction of the minority class in boosting. In: Seventh European conference on principles and practice of knowledge discovery in databases, pp 107–119
35. Guo H, Viktor HL (2004) Learning from imbalanced data sets with boosting and data generation: the databoost-im approach. SIGKDD Explor Newslett 6(1):30–39
36. Seiffert C, Khoshgoftaar TM, Hulse JV, Napolitano A (2010) Rusboost: a hybrid approach to alleviating class imbalance. IEEE Trans Syst Man Cybern A Syst Hum 40(1):185–197
37. Duda RO, Hart PE, Stork DG (2001) Pattern classification. Wiley, New Jersey
38. Kang P, Cho S (2006) EUS SVMs: ensemble of undersampled SVMs for data imbalance problems. In: 13th international conference on neural information processing, ICONIP 2006, Hong Kong, vol 4232, pp 837–846
39. Kolcz A, Yih W (2007) Raising the baseline for high-precision text classifiers. In: Proceedings of the 13th ACM SIGKDD international conference on knowledge discovery and data mining, San Jose, pp 400–409
40. Masuyama T, Nakagawa H (2004) Two step POS selection for SVM based text categorization. IEICE Trans Inf Syst E87-D(2):1–7
41. Wu S-H, Lin K-P, Chen C-M, Chen M-S (2008) Asymmetric support vector machines: low false-positive learning under the user tolerance. In: Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining, August 2008
42. Gasparini F, Corchs S, Schettini R (2005) A recall or precision oriented skin classifier using binary combining strategies. Pattern Recognit 38:2204–2207
43. Bryan K and Cunningham P (2007) Balboa: extending bicluster analysis to classify orfs using expression data. In: BIBE'07, pp 995–1002
44. Delany SJ, Cunningham P, Tsymbal A, Coyle L (2005) A case-based technique for tracking concept drift in spam filtering. Knowl-Based Syst 18:187–195
45. Yang H, Nenadic G, Keane JA (2008) Identification of transcription factor contexts in literature using machine learning approaches. BMC Bioinform 9(Suppl 3):S11
46. Chang YI (2003) Boosting SVM classifiers with logistic regression. Technical report, Academia Sinica, (http://www.stat.sinica.edu.tw/library/c_tec_rep/2003-03.pdf)
47. Chen D, Müller HM, Sternberg PW (2006) Automatic document classification of biological literature. BMC Bioinform 7
48. Craven M, DiPasquo D, Freitag D, McCallum A, Mitchell T, Nigam K, Slattery S (1998) Learning to extract symbolic knowledge from the world wide web. In: Proceedings of the fifteenth national conference on artificial intelligence, Madison, pp 509–516
49. Xue XB, Zhou ZH (2009) Distributional features for text categorization. IEEE Trans Knowl Data Eng 21:428–442
50. Nunzio GMD (2009) Using scatterplots to understand and improve probabilistic models for text categorization and retrieval. Int J Approx Reason 50(7):945–956
51. Bekkerman R, El-Yaniv R, Tishby N, Winter Y (2003) Distributional word clusters versus words for text categorization. J Mach Learn Res 3:1183–1208
52. Buckley C (1985) Implementation of the smart information retrieval system. Technical report, Cornell University, Ithaca, USA
53. Porter MF (1980) An algorithm for suffix stripping. Program 14(3):130–137
54. Burges CJC (1998) A tutorial on support vector machines for pattern recognition. Data Min Knowl Discov 2(2):121–167
55. Lan M, Tan CL, Su J, Lu Y (2009) Supervised and traditional term weighting methods for automatic text categorization. IEEE Trans Pattern Anal Mach Intell 31(4):721–735
56. Joachims T (1999) Making large-scale SVM learning practical. In: Schölkoph B, Burges CJC, Smola AJ (eds) Advances in Kernel methods—support vector learning, MIT Press, Cambridge, pp 169–184
57. Yang Y, Pedersen JO (1997) A comparative study on feature selection in text categorization. In: Proceedings of ICML'97, 14th international conference on machine learning, Morgan Kaufmann Publishers, San Francisco, pp 412–420
58. Altınçay H, Erenel Z (2010) Analytical evaluation of term weighting schemes for text categorization. Pattern Recognit Lett 31:1310–1323
59. Debole F, Sebastiani F (2004) An analysis of the relative hardness of Reuters-21578 subsets. J Am Soc Inform Sci Technol 56(6):584–596
60. Estabrooks A, Japkowicz N (2001) A mixture-of-experts framework for text classification. In: Proceedings of the intelligent data analysis conference, IDA
61. Sarinnapakorn K, Kubat M (2007) Combining subclassifiers in text categorization: a DST-based solution and a case study. IEEE Trans Knowl Data Eng 19:1638–1651
62. Kumar MA, Gopal M (2010) A comparison study on multiple binary-class svm methods for unilabel text categorization. Pattern Recognit Lett 31(11):1437–1444
63. Demirekler M, Altınçay H (2002) Plurality voting based multiple classifier systems: statistically independent with respect to dependent classifier sets. Pattern Recognit 35(11):2365–2379