Information Processing and Management 47 (2011) 202–214

Contents lists available at ScienceDirect

Information Processing and Management

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

Exploiting probabilistic topic models to improve text categorization under class imbalance

Enhong Chen [a,*], Yanggang Lin [a], Hui Xiong [b], Qiming Luo [a], Haiping Ma [a]

[a] *School of Computer Science and Technology, P.O. Box 4, Hefei, Anhui 230027, PR China*
[b] *Department of Management Science and Information Systems, Rutgers, The State University of New Jersey, New Brunswick, NJ 08901-8554, USA*

ARTICLE INFO

Article history:
Received 6 October 2009
Received in revised form 25 July 2010
Accepted 29 July 2010
Available online 1 September 2010

Keywords:
Class imbalance
Rare class analysis
Text categorization
Probabilistic topic model
Noisy data

ABSTRACT

In text categorization, it is quite often that the numbers of documents in different categories are different, i.e., the class distribution is imbalanced. We propose a unique approach to improve text categorization under class imbalance by exploiting the semantic context in text documents. Specifically, we generate new samples of rare classes (categories with relatively small amount of training data) by using global semantic information of classes represented by probabilistic topic models. In this way, the numbers of samples in different categories can become more balanced and the performance of text categorization can be improved using this transformed data set. Indeed, the proposed method is different from traditional re-sampling methods, which try to balance the number of documents in different classes by re-sampling the documents in rare classes. Such re-sampling methods can cause overfitting. Another benefit of our approach is the effective handling of noisy samples. Since all the new samples are generated by topic models, the impact of noisy samples is dramatically reduced. Finally, as demonstrated by the experimental results, the proposed methods can achieve better performance under class imbalance and is more tolerant to noisy samples.

1. Introduction

Text categorization/classification (Sebastiani, 2002) is a common technique for automatically organizing documents into predefined categories. While existing text categorization techniques have shown promising results in many application scenarios (e.g. Ko & Seo, 2009; Paradis & Nie, 2007; Sebastiani, 2002), text categorization techniques for handling data sets with imbalanced class distributions remain a challenging research issue.

In the case of class imbalance, the classifiers tend to ignore rare classes in favor of larger classes due to the size effect. In fact Yang and Liu (1999) compared the robustness and classification performances of several text categorization methods such as Support Vector Machine (SVM), Naive Bayes classifier, and K-Nearest Neighbor (KNN) classifier on data sets with various class distributions, and the experimental results show that all these classifiers achieve relatively low performance for rare classes. A promising direction to handle the class imbalance problem is applying re-sampling techniques. Specifically, over-sampling techniques (Japkowicz, 2000) can be used to increase the number of data instances in rare classes and under-sampling techniques can be used to reduce the number of data instances in large classes (Japkowicz & Stephen, 2002). The ultimate goal is to adjust the sizes of classes to a relatively balanced level.

Although both over sampling and under sampling can alleviate the class imbalance problem, there are some side effects. For instance, replicating samples by over sampling could result in overfitting. Also, some useful samples in large classes may

[*] Corresponding author.
E-mail address: cheneh@ustc.edu.cn (E. Chen).

0306-4573/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ipm.2010.07.003

be missing due to under sampling. This, in turn, will hinder the classification performance. Therefore, many modified re-sampling methods were developed to overcome the disadvantages described above. For example, the overfitting problem in random over-sampling methods can be avoided to a certain extent by bringing in random Gaussian noise to samples in rare classes or synthetically generating samples for rare classes (Chawla, Bowyer, Hall, & Kegelmeyer, 2002). Also, a better performance than random under-sampling methods can be achieved by only eliminating samples that are further away from class boundaries (Batista, Prati, & Monard, 2004).

As a matter of fact, most re-sampling techniques were developed for general types of data rather than for text. Our approach intends to exploit the semantic characteristics uniquely existing in text documents to improve text categorization under class imbalance. New samples of rare classes are generated by using global semantic information of classes represented by probabilistic topic models instead of replicating samples in rare classes. Probabilistic topic models (Blei, Ng, & Jordan, 2003; Griffiths & Steyvers, 2004) are effective tools to capture the semantic topics (more details Section 2.1). By exploiting probabilistic topic models of rare classes to generate new samples, overfitting is less likely to occur. In addition, if the original training samples are replaced by new samples generated by probabilistic topic models, the training data can be smoothed and the impact on the classification performance by noisy samples will also be alleviated.

In this paper, we propose two re-sampling methods based on probabilistic topic models: DECOM (data re-sampling with probabilistic topic models) and DECODER (data re-sampling with probabilistic topic models after smoothing). DECOM deals with class imbalance by generating new samples of rare classes using probabilistic topic models. In addition, for data sets with a high proportion of noisy samples, DECODER first smoothes data by regenerating all samples in data sets and then generates more samples for rare classes by probabilistic topic models. Experimental results on various real-world data sets show that both DECOM and DECODER can achieve better classification performances on rare classes. Also, DECODER is more robust in handling very noisy data.

Overview: The remainder of this paper is organized as follows. Section 2 introduces probabilistic topic models as well as two re-sampling methods based on probabilistic topic models. In Section 3, we present experimental results on a number of real-world document data sets. Section 4 describes some related work. Finally, we draw conclusions in Section 5.

2. Re-sampling with probabilistic topic models

In this section, we first present a probabilistic topic model based on Latent Dirichlet Allocation (LDA) (Blei et al., 2003). Then we propose two re-sampling methods: DECOM and DECODER.

2.1. Probabilistic topic models

Probabilistic topic models are generative models that specify a probabilistic process for generating documents. Table 1 shows an example of probabilistic topic models. To generate a new document, a distribution over topics is first selected. Then for each word token in this document, a topic is selected randomly by sampling from the distribution over topics and next a word is selected randomly by sampling from the word distribution of this topic.

Fig. 1 illustrates the generation of documents by probabilistic topic models. There are only two topics containing a certain number of words in this simple example. The distribution of topics is different in different documents. For example, topic 2 does not occur in DOC1 while the probabilities of both topic 1 and topic 2 are 0.5 in DOC2. For the first word in DOC1, topic 1 is selected from the topic distribution and then the word "money" is selected randomly according to the word distribution of topic 1. While the first word in DOC2 is to be created, topic 1 is selected randomly from the topic distribution and then the word "money" is selected. For the second word, topic 1 is selected again randomly and the word we get this time is "bank". Then, for the third word, topic 2 is selected and "bank" is selected according the word distribution of topic 2. By this way, all words in DOC2 can be decided.

We adopt the probabilistic topic model proposed in Steyvers and Griffiths (2007) as the generative model for training samples. We use $P(w)$ to denote the probability of generating the current word w

Table 1
An example of probabilistic topic models.

Topic 1	0.01116	Topic 2	0.00608	Topic 3	0.01207	. . .	TOPIC *n*	0.00756
Figure	0.16022	Module	0.04493	Single	0.01947		Rules	0.07821
Curve	0.02541	Modules	0.04191	Individual	0.01920		Rule	0.06315
Size	0.02134	Attention	0.03090	Based	0.01815		Category	0.02234
Curves	0.02110	Gain	0.02859	Presented	0.01746		Set	0.02070
Top	0.01505	Context	0.01518	Produce	0.01430		Categories	0.01905
Show	0.01471	Control	0.01287	Multiple	0.01372		Knowledge	0.01541
Shape	0.01437	Figure	0.01287	Order	0.01372		Similarity	0.01185
Plotted	0.01287	System	0.01208	Techniques	0.01271		Examples	0.01056
Bottom	0.01036	Selection	0.01163	Technique	0.01198		Extracted	0.00992
Area	0.01026	Selected	0.01128	Effects	0.01087		Form	0.00992
. . .		. . .		. . .			. . .	
Fig. 1b	0.00494	Hybrid	0.00364	Parts	0.00464		Group	0.00328

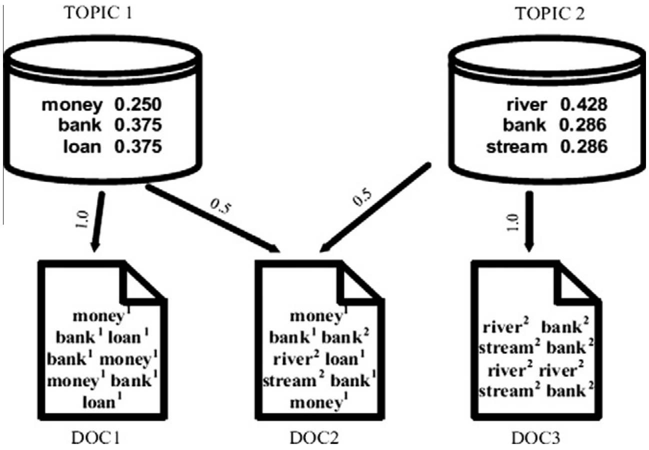


Fig. 1. Probabilistic generation of documents.

$$P(w) = \sum_{i=1}^{T} P(w|z=i)P(z=i) \tag{1}$$

where T is the number of topics.

From Eq. (1), new documents can be generated if $P(w|z=i)$ and $P(w|z=i)$ are both known. $P(w|z=i)$ denotes the probability of sampling the word token w from the word distribution of the ith topic. $P(z=i)$ gives the probability of sampling the ith topic from the topic distribution of the current document.

Given a collection of D documents containing T topics expressed over W unique words, we represent the word index with w_i and document index with d_i for each word token i. Here "word token" refers to an instance of a word at a certain position in a document. For each word token in the text collection, the Gibbs sampling method (Casella & George, 1992) estimates the conditional distribution of assigning this word to each topic conditioned on the topic assignments to all other word tokens. A topic is sampled from this conditional distribution and then stored as the new topic assignment for the word token. We represent this conditional distribution as $P(z_i=j|z_{-i}, w_i, d_i,\cdot)$, where $z_i=j$ means the topic assignment of token i to topic j, z_{-i} refers to the topic assignments of all other word tokens, and "$\cdot$" refers to all other known or observed information such as all other word and document indices w_{-i} and d_{-i}, and hyperparameters α, and β. Griffiths and Steyvers (2004) shows that this probability $P(z_i=j|z_{-i}, w_i, d_i,\cdot)$ can be computed by the following equation:

$$P(z_i=j|z_{-i}, w_i, d_i, .) \propto \frac{C^{WT}_{w_i j}+\beta}{\sum_{w=1}^{W} C^{WT}_{wj}+W\beta}\,\frac{C^{DT}_{d_i j}+\alpha}{\sum_{t=1}^{T} C^{DT}_{d_i t}+T\alpha} \tag{2}$$

where C^{WT} and C^{DT} are matrices with dimensions $W \times T$ and $D \times T$ respectively. $C^{WT}_{w_i j}$ represents the number of times word w is sampled from topic j, not including the current token i, and $C^{DT}_{d_i j}$ refers to the number of times topic j is assigned to some word tokens in document d_i, not including the current word instance i. The left part of the right side of the equation is the probability of sampling word w_i given topic j whereas the right part is the probability of sampling topic j given the current topic distribution for document d_i. The Gibbs sampling algorithm begins with the assignment of each word token to a random topic in $[1 \ldots T]$ to determine the initial state of the Markov chain. This chain then runs for a number of iterations and finds a new state by sampling each z_i from the distribution specified by Eq. (2) in each iteration. After enough number of iterations for the chain to approach the target distribution, the current values of the z_i variables are recorded. By Gibbs sampling, each word token in the document set is assigned to a topic and according to the final assignment $P(w|z=i)$ and $P(z=i)$ can be estimated by:

$$\begin{aligned} P(w|z=i) &= \frac{C^{WT}_{wi}+\beta}{\sum_{k=1}^{W} C^{WT}_{ki}+W\beta} \\ P(z=i) &= \frac{\sum_{d=1}^{D} C^{DT}_{di}+\alpha}{\sum_{k=1}^{T}\sum_{d=1}^{D} C^{DT}_{dk}+T\alpha} \end{aligned} \tag{3}$$

2.2. Re-sampling with probabilistic topic models

In this study on text categorization, we assume that documents belonging to the same class have the same topic distribution. That is, documents of the same class are assumed to be generated by the topic model of this class, as shown in Fig. 2.

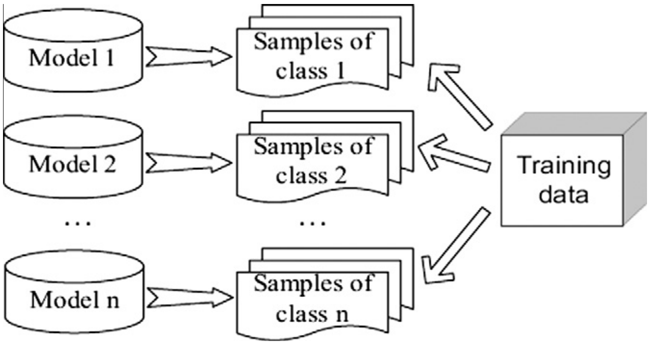


Fig. 2. An illustration of training data and probabilistic topic models.

Hence if we generate a new document using the topic model of a class, the new document should still belong to this class. Based on this assumption, we propose two re-sampling methods DECOM and DECODER based on topic models for different situations. Assuming that the training samples belong to n classes $C = \{c_1, c_2, \ldots, c_n\}$, the probabilistic topic models $M = \{m_1, m_2, \ldots, m_n\}$ for each class can be extracted from the documents belonging to each class by the Gibbs sampling algorithm, respectively.

Algorithm 1: GenerateSample(M, N)

```
input: M is the topic model used to generate a new sample
N is the number of words in the new sample
output: a new sample generated from M by probabilistic procedure
wordlist = $\Phi$ //wordlist is the set of words in the new sample
while $N > 0$ do
    TopicPro is array of probability of topics in M
    Ti = Roulette(TopicPro)
    //select a topic according the probability of topics in M
    WordPro is array of probability of words in topic $T_i$
    word = Roulette(WordPro)
    //select a word according the probability of words in topic $T_i$
    Append(wordlist, word)
return wordlist //end of GenerateSample
```

Algorithm 2: Roulette(Pro)

```
//Function to select a element in array Pro by Roulette Wheel Selection
sum = 0
for $i = 1$ to size(Pro) do
    sum+ = Pro($i$)
$p$ = random number between 0 and sum
index = 0
$i = 1$
while $index < p$ do
    index+ = Pro($i$)
    $i$++
return $i - 1$ //end of Roulette
```

2.2.1. DECOM

The number of samples in the largest class is denoted as *MAX*. For any other class c_i which contains $|c_i|$ samples, *MAX*-$|c_i|$ new samples are generated by the corresponding topic model. Then the new samples of each class are used together with the original samples to train the classifier. Different from other synthetic over-sampling methods such as SMOTE (Chawla et al., 2002), the generation of new samples by the topic model is based on the whole sample space, not just several local samples. The process of generating new samples by topic models is presented in Algorithm 1 and Algorithm 2. Specifically, for each

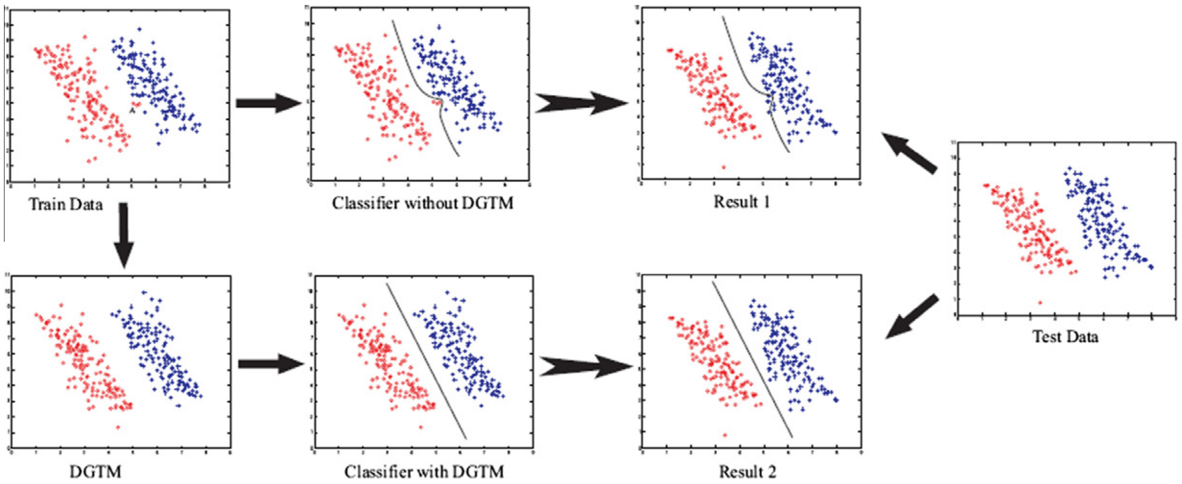


Fig. 3. An illustration of DECODER.

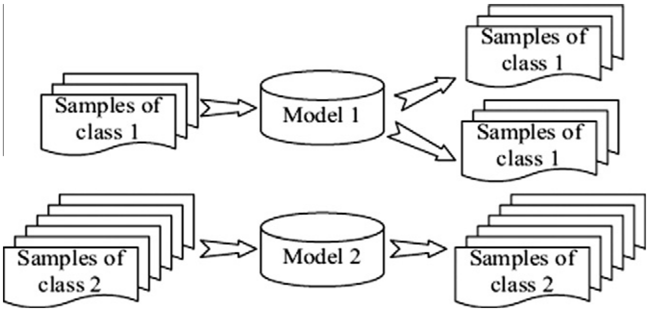


Fig. 4. An illustration of data smoothing by probabilistic topic models.

word token in a new sample containing N words, a topic T_i is selected by the roulette algorithm (Algorithm 2) according to the topic distribution in topic model M. Then, a word is selected as the current word token according to the word distribution of topic T_i. By this way, all words are selected and they compose the new sample document. For each class, the size of the new sample N (in terms of words) is randomly picked from the distribution of the sizes of documents in this class.

2.2.2. Data smoothing

All samples in the training data are assumed to be labeled without any errors. However, in reality it is labor-intensive to prepare the training data with labels, particularly when its scale is large. So it is possible for samples to be mislabeled, and noisy samples are unavoidable in training data (Zhu & Wu, 2004). The tolerance to noisy samples is very important to the performance of classification because it is difficult to eliminate all noisy samples without eliminating some correctly labeled samples. For this reason, topic models can be used to smooth the available training samples and to reduce the impact of noisy samples on the performance of classification. Before training the classifier, $|c_i|$ new samples are generated by topic model m_i for each class $c_i \in C$. Then new samples are used for training instead of original samples. An illustration of smoothing is shown in Fig. 3,[1] where three noisy samples belonging to the blue class are mislabeled as the red class. The classifier is affected by the noisy samples. As shown in “result 1” some samples in the blue class are mislabeled as the red class when test data is classified by this classifier. If the training data is smoothed in advance, the new training data can prevent the noisy samples from affecting the classifier and the correct result would be achieved for the test data as shown in “result 2”.

2.2.3. DECODER

For data sets with a high proportion of noisy samples, the DECODER method, a modified version of DECOM, is proposed. In this case, the original data is replaced by the same amount of new samples generated by topic models. After this phase, the training data is balanced by generating more new samples for rare classes until the size of each class is approximately equal. Fig. 4 provides an illustration of DECODER. As shown in Fig. 3, the size of class 2 is two times that of class 1 and this is not

[1] For interpretation of color in Fig. 3, the reader is referred to the web version of this article.

changed after data smoothing. Then model 1 continues generating more samples of class 1 until the size of class 1 is the same as that of class 2.

3. Experimental evaluations

In this section, we evaluate the effectiveness of the proposed DECOM and DECODER re-sampling methods in two stages. In the first stage (Sections 3.2.1 and 3.2.2), we use several unbalanced real-world data sets to show the effectiveness of DECOM and DECODER in improving the classification performances on rare classes. In the second stage, we evaluate the performance of DECODER on very noisy data.

3.1. The experimental setup

SVM is adopted as the base classifier in our experiments because of its good performance in most cases (Cristianini & Shawe-Taylor, 2000). The implementation of SVM used in this study is LIBSVM (Chang & Lin, 2001) and the parameters of SVM are set as follows: linear kernel, the parameter gamma in kernel function and the parameter C of C-SVC is obtained automatically by grid-search using cross-validation for each data set as shown in Table 2, and other parameters are set as default. The IG (Information Gain) feature selection algorithm is used to select the top 1000 words as features. The parameters of probabilistic topic models extraction algorithm are set as follows: the number of topics $T = 30$, the number of iterations $N = 300$, $\alpha = 50/T$, $\beta = 0.01$. For comparison, besides pure SVM without any balancing, we also consider other balancing methods (Maimon & Rokach, 2005; Tan, Steinbach, & Kumar, 2005): random over sampling (ROS), random under sampling (RUS) and SMOTE (Chawla et al., 2002). Note that we run all methods 10 times and return the average results in our experiments. The data sets with various values of CV (Coefficient of Variation (DeGroot & Schervish, 2001)) used in the experiments are from three sources as shown in Table 2. CV indicates the dispersion of the class distribution for each data set. In general the larger the CV value is, the more unbalanced the data is.

The 20 Newsgroups data set is a collection of approximately 20,000 newsgroup documents, partitioned nearly evenly across 20 different topics. For the purpose of studying class imbalance, we reduce the size of some classes artificially in the experiments. The Reuters-21578 data set is currently the most widely used test collection for text classification research. The documents in the Reuters-21578 collection appeared on the Reuters newswire in 1987. The data was originally collected and labeled by Carnegie Group, Inc., and Reuters, Ltd. in the course of developing the CONSTRUE text classification system. Besides, another data set k1b is from the WebACE project (Han et al., 1998). Each document corresponds to a web page listed in the subject hierarchy of Yahoo!. The RCV1-v2 data set is a large corpus of newswire stories provided by Reuters Ltd. and later corrected by Lewis, Yang, Rose, and Li (2004). We use a sample of 20402 documents belonging to five categories and refer to it as RCV1-v2_t5.

3.2. Experimental results

In this subsection, the experimental results show how the classification performance can be improved by DECOM. Since two-class data set is intuitive and simple, it is most suitable to demonstrate the effectiveness of each method on imbalanced data set without other sources of interference. For comparing the balancing methods more accurately, we conduct experi-

Table 2
Characteristics of data sets.

Dataset	#Samples	#Classes	MaxClassSize	MinClassSize	Training set size	Test set size	*CV*	Best C	Best gamma
20 Newsgroups data sets									
20news_b1	1200	2	1000	200	600	600	0.943	128.0	0.0078125
20news_b2	2000	2	1000	1000	1000	1000	0	32.0	0.0078125
20news_m1	5400	8	1000	300	2700	2700	0.410	2048.0	0.0078125
20news_m2	2100	3	1000	300	1050	1050	0.515	32.0	0.0078125
Reuters-21578 data sets									
reuters1	2655	2	2369	286	1847	808	1.113	8.0	0.0078125
reuters2	610	2	486	124	463	147	0.839	32.0	0.0078125
reuters3	868	2	582	286	630	238	0.530	8.0	0.0078125
reuters4	2929	14	717	30	2176	753	1.002	8.0	0.0078125
reuters5	2586	10	717	30	1914	672	0.893	8.0	0.0078125
reuters6	4301	5	2369	286	3038	1263	0.970	2.0	0.0078125
WebACE data sets									
k1b	2340	6	1389	60	1169	1171	1.317	32.0	0.0078125
RCV1 data set									
RCV1-v2_t5	20402	5	7406	680	4079	16323	1.36	512.0	0.0078125

ments on two-class data sets at first. In practice multi-class data sets are often encountered, so we also experiment on multi-class data sets. The results on multi-class data sets are consistent with those on two-class data sets.

3.2.1. Results on two-class data sets

Table 3 presents the performance comparison based on the F-measure with other balancing methods on two-class data sets. DECOM shows a good performance on two-class data sets on various data sets. When the *CV* value is low, random under sampling has a better performance than random over sampling. For example, on the reuters3 data set with *CV* value of 0.530, under sampling has a competitive performance close to the best result achieved by DECOM. However, when the *CV* becomes high, over sampling beats under sampling significantly. The SMOTE method has limited improvement because new samples are related to a few local samples as we discussed above. Note that on reuters2 data set, DECODER has better performance than DECOM. An alternative explanation is that there exits some noisy samples in this data set. Compared to pure SVM, the classification result by DECOM favors the rare class. The recall value of rare class (denoted as class 1) is increased and the precision value is decreased while it is contrary for the large class (denoted as class 2), as shown in Table 4. Although DECOM may sometimes cause decrease in recall value or precision value of each class, the F-measures of both the rare classes and the large classes are improved by DECOM. Large improvement of the recall value or precision value can be achieved with a little cost of decreasing the other one. For example, the recall value of class 1 in 20news_b1 data set is improved from 0.6000 to 0.9920 while the precision value is decreased just from 0.9677 to 0.9094.

3.2.2. Results on multi-class data sets

An overview of macro F-measure comparison of these methods is shown in Table 5. In this case, random under sampling does not perform well on multi-class data sets. DECOM still performs better than other balancing methods although the improvement by DECOM is not as large as on two-class data sets in some cases. Besides DECOM, random over-sampling method also has stable performance on all data sets while random under sampling and SMOTE are not so stable. The results in the table show that DECOM is more robust on multi-class data sets than others.

Another observation is that the increase of the F-measure value of the rare classes is not at high cost of the decrease of others. As shown in Fig. 5, different from other traditional balancing methods such as random over sampling, DECOM can improve the classification accuracy for most classes. For example, the performance of about 83% classes is improved by DECOM on reuters6 data set while it is only 60% for random over sampling. Fig. 6 shows how DECOM improves performance for each class on the data sets described in Table 6. On the reuters5 data set, the performance in rare classes such as class 9 with only 59 samples is greatly increased by DECOM while only the performance in class 6 is decreased slightly. On k1b data set, the decrease of accuracy in class 6 can be ignored compared to the great increase of accuracy in class 2. Notice that the overall performance of DECODER is a little weaker than DECOM because some information useful for classification may be lost in the data smoothing phase.

Table 3
A performance comparison on two-class data sets by macro/micro F-measure.

Data set	*CV*	Balancing methods					
		Pure SVM	ROS	RUS	SMOTE	DECOM	DECODER
20news_b1	0.943	0.8660/0.9300	0.9302/0.9623	0.9012/0.9352	0.8896/0.9418	0.9697/0.9822	0.9360/0.9582
		Gain % over SVM	7.41/3.47	4.06/0.56	2.73/1.27	11.97/5.61	8.08/3.03
reuters1	1.113	0.9296/0.9735	0.9759/0.9904	0.9665/0.9859	0.9534/0.9823	0.9825/0.9929	0.9663/0.9856
		Gain % over SVM	4.98/1.74	3.97/1.27	2.56/0.90	5.69/1.99	3.95/1.24
reuters2	0.839	0.8925/0.9320	0.9895/0.9932	0.9752/0.9837	0.9296/0.9551	0.9895/0.9932	0.9916/0.9946
		Gain % over SVM	10.87/6.57	9.27/5.55	4.16/2.48	10.87/6.57	11.10/6.72
reuters3	0.530	0.8833/0.8908	0.9274/0.9311	0.9397/0.9416	0.9034/0.9101	0.9514/0.9521	0.9276/0.9311
		Gain % over SVM	4.99/4.52	6.39/5.70	2.28/2.17	7.71/6.88	5.02/4.52

Table 4
A performance comparison on two-class data sets in detail.

Dataset	Method	Class 1			Class 2		
		Recall	Precision	F-measure	Recall	Precision	F-measure
20news_b1	Pure SVM	0.6000	0.9677	0.7407	0.9960	0.9257	0.9595
	DECOM	0.9920	0.9094	0.9489	0.9802	0.9984	0.9892
reuters1	Pure SVM	0.7595	1.0000	0.8633	1.0000	0.9711	0.9853
	DECOM	0.9888	0.9494	0.9686	0.9934	0.9986	0.9960
reuters2	Pure SVM	0.6667	1.0000	0.8000	1.0000	0.9213	0.9590
	DECOM	0.9667	1.0000	0.9831	1.0000	0.9915	0.9957
reuters3	Pure SVM	0.7640	0.9315	0.8395	0.9664	0.8727	0.9172
	DECOM	0.9877	0.8951	0.9391	0.9309	0.9922	0.9605

Table 5
An overview of macro/micro F-measure comparison on multi-class data sets.

Dataset	*CV*	Balancing methods					
		Pure SVM	ROS	RUS	SMOTE	DECOM	DECODER
20news_m1	0.410	0.7969/0.8070	0.8041/0.8150	0.7731/0.7733	0.7850/0.7992	0.9260/0.9229	0.9021/0.9097
		Gain % over SVM	0.90/0.99	−2.99/−4.18	−1.49/−0.97	16.20/14.36	13.20/12.73
reuters4	1.002	0.7785/0.7570	0.7946/0.7946	0.6653/0.5995	0.7859/0.7744	0.8381/0.8125	0.7987/0.7448
		Gain % over SVM	2.07/5.20	−14.54/−20.18	0.95/2.30	7.66/7.33	2.59/−1.61
reuters5	0.893	0.7731/0.7768	0.8016/0.8037	0.7074/0.6399	0.8227/0.8054	0.8638/0.8320	0.8497/0.8116
		Gain % over SVM	3.69/3.46	−8.50/−17.62	6.42/3.68	11.73/7.11	9.91/4.48
reuters6	0.970	0.8472/0.9153	0.8531/0.9125	0.8563/0.9089	0.8490/0.9074	0.8596/0.9159	0.8605/0.9118
		Gain % over SVM	0.70/−0.31	1.07/−0.70	0.21/−0.86	1.46/0.07	1.57/−0.38
k1b	1.317	0.7898/0.9240	0.8242/0.9340	0.8221/0.8972	0.8318/0.932	0.8869/0.9495	0.8679/0.9247
		Gain % over SVM	4.36/1.08	4.09/−2.90	5.32/0.87	12.29/2.76	9.89/0.08
RCV1-v2_t5	1.36	0.8922/0.9082	0.8972/0.9099	0.8756/0.8984	0.8880/0.9065	0.8972/0.9105	0.8871/0.9099
		Gain % over SVM	0.56/0.19	−1.86/−1.08	−0.47/−0.19	0.56/0.25	−0.57/0.19

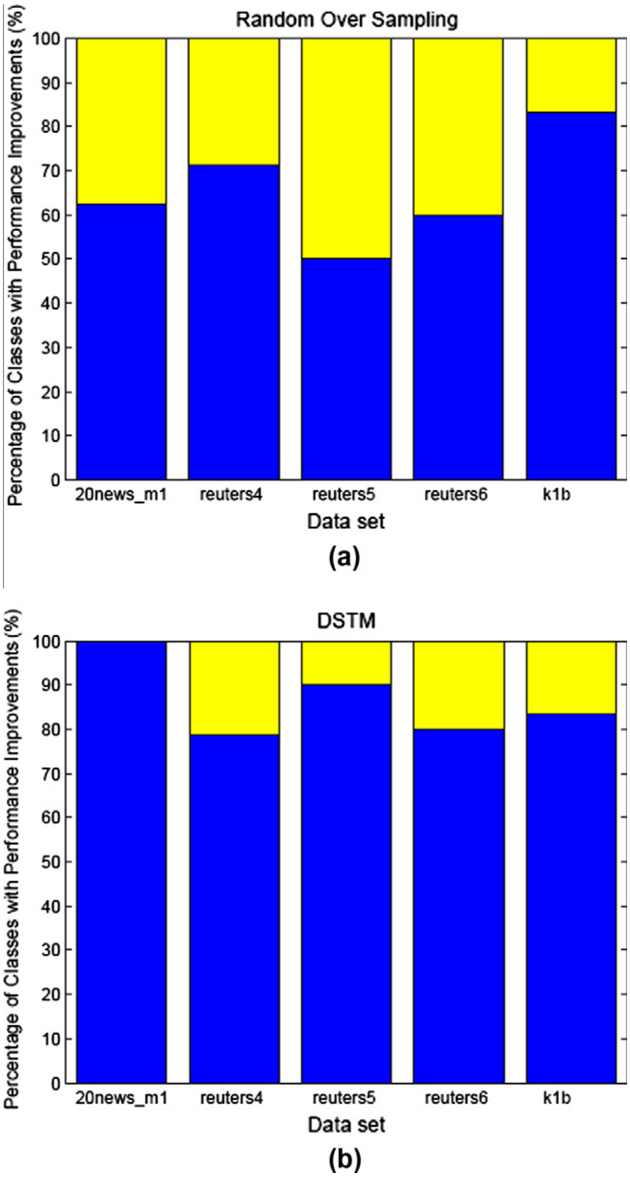
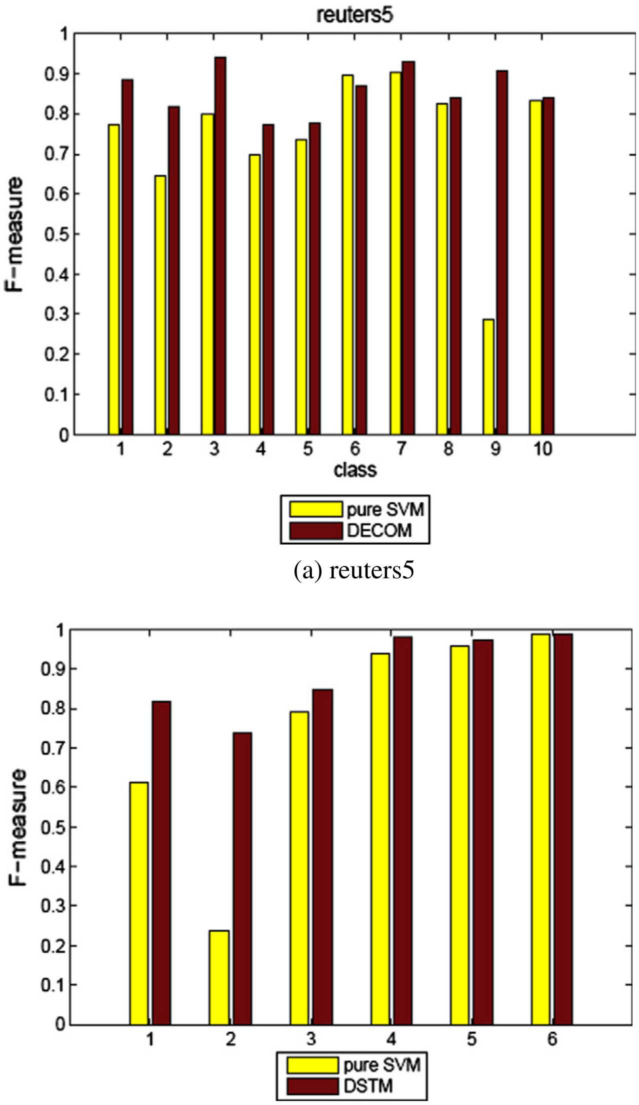


Fig. 5. Percentage of classes with performance improvements by random over sampling (a) and DECOM (b).



(a) reuters5

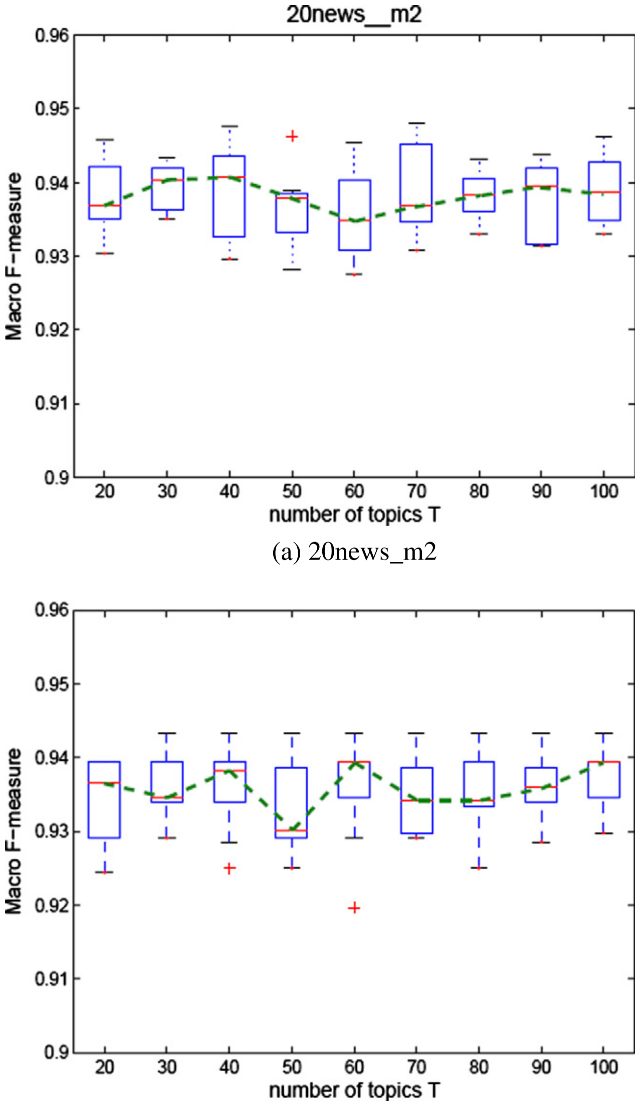
(b) k1b

Fig. 6. Classification performances of DECOM and pure SVMs on reuters5 and k1b data sets.

Table 6
Number of samples in each class for reuters5 and k1b data sets.

Data set	#Samples of class										*CV*
	1	2	3	4	5	6	7	8	9	10	
reuters5	486	30	162	286	105	717	478	124	59	139	0.893
k1b	142	60	114	141	1389	494	–	–	–	–	1.317

The number of topics in topic models (T) is an important parameter that may affect the performance of classification. Griffiths and Steyvers (2004) has investigated the relation between the likelihood $P(w|T)$ and T, and has proved that the likelihood is affected by T greatly. It is difficult to find the best value of T in DECOM since the value with optimal likelihood $P(w|T)$ may not always result in the best classification performance. In fact, the best value of T for DECOM is distinct on different data sets. Fortunately, the impact that T has on the classification performance is limited. Fig. 7 shows the macro F-measure with different values of T on 20news_m2 and reuters3 data sets. Obviously, the impact of T is small and acceptable.



(a) 20news_m2

(b) reuters3

Fig. 7. The impact of different number of topics.

Table 7
Results on the 20news b2 data set.

Method	Class 1			Class 2			Macro F-measure	Micro F-measure
	Recall	Precision	F-measure	Recall	Precision	F-measure		
Pure SVM	0.9860	0.9591	0.9724	0.9580	0.9856	0.9716	0.9722	0.9720
DECODER	0.9704	0.9953	0.9827	0.9954	0.9712	0.9831	0.9830	0.9829

3.3. Tolerance to noisy samples

As we discussed in Section 3.2, the data sets may have noisy samples. Even if all training samples are labeled correctly, some samples are important for classification while some samples are unimportant. Table 7 shows an interesting result that DECODER can improve the performance on balanced data set 20news_b2. In this case, DECODER does the same thing as pure SVM except for data smoothing since the *CV* value of the data set is exactly 0. An alternative explanation is that the classifier is affected by unimportant or noisy samples in the data set and this effect can be reduced by just smoothing the data samples with DECODER.

Although there may be some noisy samples in the data sets used in this paper, the number of noisy samples could be relatively small since these data sets are considered as the benchmarks for text classification research. As a result, not every data set will be similar to the one in Table 7, which has enough noisy samples to affect the classification results. For this reason, we have preprocessed the data sets so that they can be suitable for noise tolerance experiments. In order to investigate the tolerance of each method to noisy samples, we get three two-class data sets from 20 Newsgroups data sets and Reuters-21578 data sets intentionally, as shown in Table 8. Some samples are removed from one class and are used to randomly replace samples in the other class as noisy samples. The proportion of noisy samples in the other class (noisy class) is referred to as "noise ratio". Note that the noisy class is the rare class for imbalanced data sets in our experiment. The results achieved by the balancing methods are shown in Fig. 8. On 20news_b21 data set, DECODER has slower performance decrease with increasing ratio of noisy samples than pure SVM. Even when the noise ratio is 50%, a macro F-measure of about 87% can be achieved. On reuters8 data set, when the noise ratio is under 5%, DECOM can achieve a better performance than DECODER

Table 8
Data sets with a high proportion of noisy samples.

Data set	Source	#Samples	*CV*	#Noisy samples
20news_b21	20 Newsgroups	1200	0	400
reuters7	Reuters-21578	1600	0.707	700
reuters8	Reutgrs-21578	1300	0.197	700

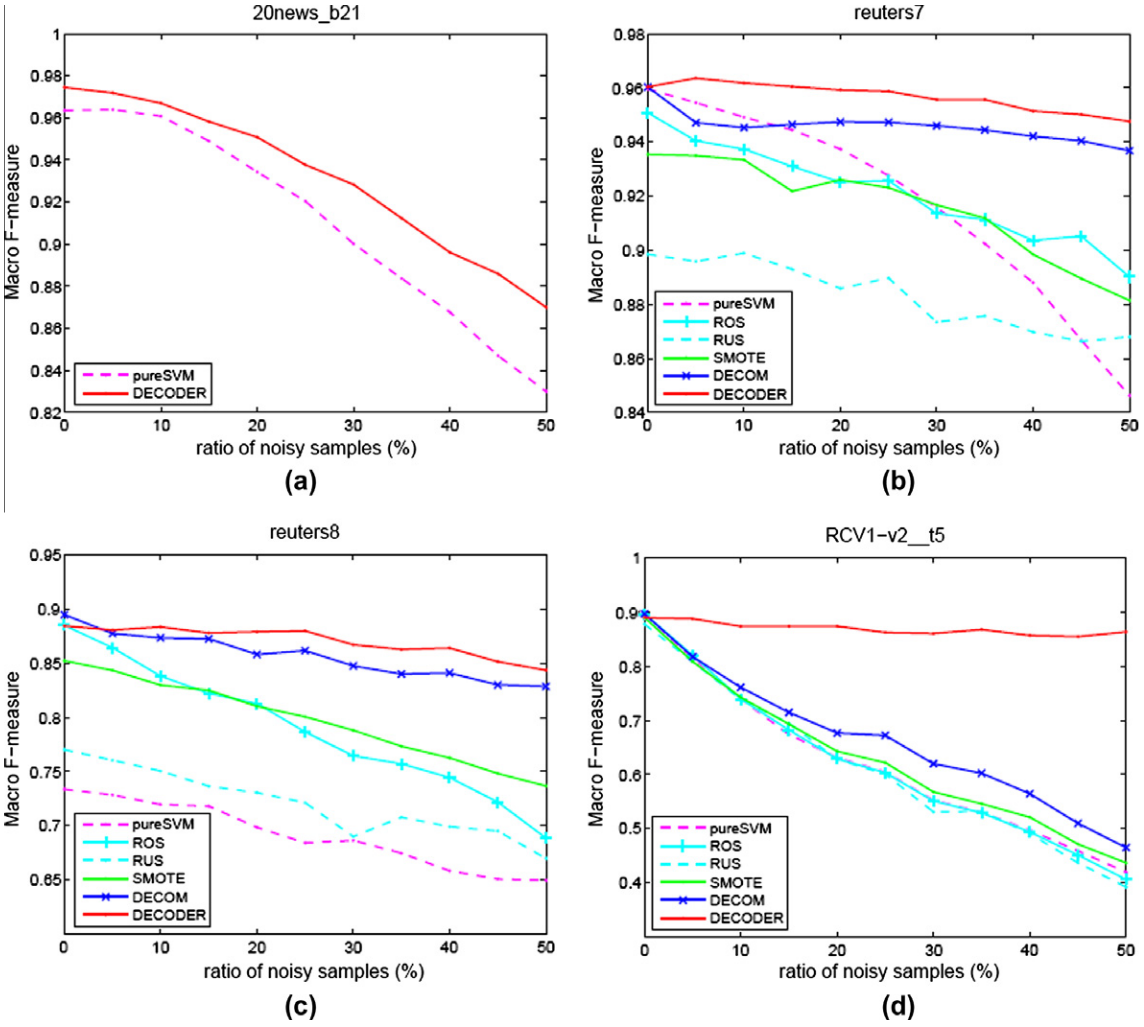


Fig. 8. A performance comparison on very noisy data.

Table 9
A performance comparison on noisy multi-class data set in term of macro F-measure.

Unlabeled ratio	Unlabeled F-measure	Pure SVM	ROS	RUS	SMOTE	DECOM	DECODER
0.5	0.8700/0.8741	0.7719/0.7744	0.7798/0.7810	0.8275/0.8252	0.7524/0.7667	0.7958/0.7952	0.8821/0.8881
		Gain % over SVM	1.02/0.97	7.20/6.56	−2.53/−0.99	3.10/2.69	14.28/14.68
0.6	0.8525/0.8611	0.7962/0.7967	0.8109/0.8089	0.7989/0.7833	0.7760/0.7819	0.7918/0.7881	0.8754/0.8837
		Gain % over SVM	1.85/1.53	0.34/−1.68	−2.54/−1.86	−0.55/−1.08	9.95/10.92
0.7	0.8256/0.8296	0.7760/0.7789	0.7921/0.7919	0.8066/0.8030	0.7761/0.7844	0.7850/0.7800	0.8773/0.8867
		Gain % over SVM	2.07/1.67	3.94/3.09	0.01/0.71	1.16/0.14	13.05/13.84
0.8	0.7767/0.7894	0.7221/0.7241	0.6797/0.6941	0.7731/0.7670	0.7286/0.7330	0.7381/0 7210	0.8665/0.8726
		Gain % over SVM	−5.87/−4.14	7.06/5.92	0.90/1.23	2.22/−0.30	20.00/20.51
0.9	0.6998/0.7263	0.6425/0.6619	0.6595/0.6859	0.7187/0.6919	0.6260/0.6596	0.6720/0.6841	0.8308/0.8522
		Gain % over SVM	2.65/3.63	11.86/4.53	−2.57/−0.35	4.59/3.35	29.31/28.75

since replacing the original samples may lose the information which can be helpful for classification. However, with the noise ratio increasing, the performance of DECOM becomes worse and only DECODER still has a stable performance. On RCV1-v2_t5 when the noise ratio increases from 0% to 50%, the performance decrease of DECODER is only 3% as measured by F-measure, in comparison to more than 40% for all the other approaches. In order to make more natural multi-class data sets with noisy samples, a novel experiment approach is used. First, the original training data set is divided into two parts with a ratio: new training set and unlabeled set. The new training set is used to train a pure SVM classifier and then the samples in the unlabeled set are labeled by this classifier. After this phase, the unlabeled set can be considered as the training set with noisy samples and the macro F-measure reflects the amount of noise in the training set. Now this noisy training set is used as the training set for our experiment and the test set is the original test set. The comparison of each balancing method on 20news_m1 data set is shown in Table 9. Due to data smoothing, the performance decrease of DECODER is much slower than others when the macro F-measure of unlabeled data set is decreasing.

4. Related works

The class imbalance problem is a major research issue, and researchers have addressed this problem from various perspectives.

First, there are research works focusing on understanding the class imbalanced problem. For example, it has been shown (Japkowicz & Stephen, 2002; Wu, Xiong, Wu, & Chen, 2007) that the imbalance problem is not only related to the degree of class imbalance in the data sets, but also related to the overall size of the training data as well as the complexity of the concepts in the data sets with imbalanced classes. The higher the degree of class imbalance, the higher the complexity of the concepts, and the smaller the overall size of the training set, the greater the effect of class imbalance on the classification performance.

Second, researchers have addressed the class imbalance problem from an algorithmic perspective by examining how different algorithms or algorithm design decisions can make an impact on this problem. For example Brank and Grobelnik (2003) studied the training of SVMs with few positive samples, while Tan (2005) presents the NKNN method which improves the performance of k-NN on imbalanced data sets. Also, Zheng, Wu, and Srihari (2004) addressed the class imbalance problem by feature selection and suggested that negative features are quite valuable on imbalanced data. Moreover, the COG method (Wu et al., 2007) has an elegant way to decompose the complex concepts in large classes, which hinder the performances of linear classifiers. In the COG method, each large class is divided into several small sub-classes with relatively balanced sizes. This, in turn, can greatly alleviate the class imbalance problem and helps to improve the performance.

The third direction is to manipulate the training data using sampling techniques, such as random over sampling, focused over sampling, random under sampling, and focused under sampling. For example, researchers have developed methods (Tomek, 1976; Hart, 1968; Kubat & Matwin, 1997; Laurikkala, 2001; Wilson, 1972) of under and over sampling to balance the class distribution of training data. According to experimental evaluation of these methods and their combination, Batista et al. (2004) found that, in general, over-sampling methods provide more accurate results than under-sampling methods considering the area under the ROC curve. Furthermore, in order to avoid overfitting due to just replicating samples of rare classes in simple over sampling, Chawla et al. (2002) proposed SMOTE method (Synthetic Minority Over-sampling TEchnique). When a new sample of rare class is to be created, an original sample A of this class is selected and then one of the K-nearest neighbors of A is selected randomly as B. The value of new sample is the value of A plus a Gaussian random value of the difference between A and B. The SMOTE method overcomes the overfitting of over sampling to a certain extent. However, the generation of new sample is just dependent on a few samples so that noisy samples may have more impact on the classification performance.

However, the above-mentioned approaches do not make use of semantic information, which is essential to text documents. Instead, in this paper, our focus is on exploiting global semantic information of classes, which are represented by probabilistic topic models. The scope of our methods is limited to text categorization.

5. Conclusion

In this paper, we have proposed a semantic re-sampling method to handle the class imbalance problem in text categorization. Specifically, two re-sampling techniques DECOM and DECODER were developed based on probabilistic topic models. DECOM was proposed to deal with class imbalance by generating new samples of rare classes. For data sets with noisy samples and rare classes, DECODER was developed to smooth the data by regenerating all samples in each class using probabilistic topic models and then expanding the sizes of rare classes.

A key idea of our re-sampling techniques is the exploitation of global semantic information captured by probabilistic topic models. Experimental results on various real-world data sets show that DECOM and DECODER can achieve better classification performances on rare classes. In addition, DECODER is more tolerant to noisy samples.

Acknowledgements

The authors wish to thank the reviewers for their invaluable comments. The work described in this paper was supported by grants from Natural Science Foundation of China (Grant No. 60775037), the Key Program of National Natural Science Foundation of China (Grant No. 60933013), the National High Technology Research and Development Program of China (No. 2009AA01Z123) and Research Fund for the Doctoral Program of Higher Education of China (20093402110017).

References

Batista, G. E. A. P. A., Prati, R. C., & Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explorations Newsletter, 6*(1), 20–29.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *The Journal of Machine Learning Research, 3*, 993–1022.

Brank, J., Grobelnik, M. (2003). Training text classifiers with SVM on very few positive examples. Tech. Rep. MSR-TR-2003-34, Microsoft Research, Redmond.

Casella, G., & George, E. I. (1992). Explaining the Gibbs sampler. *The American Statistician, 46*(3), 167–174.

Chang, C.-C., Lin, C.-J. (2001). LIBSVM: A library for support vector machines. <http://www.csie.ntu.edu.tw/cjlin/libsvm>.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, 16*, 321–357.

Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machines and other Kernel-based learning methods*. Cambridge University Press.

DeGroot, M., & Schervish, M. (2001). *Probability and statistics* (3rd ed.). Addison Wesley.

Griffiths, T. L., & Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America, 101*(Suppl. 1), 5228–5235.

Han, E.-H., Boley, D., Gini, M., Gross, R., Hastings, K., Karypis, G., et al. (1998). Webace: A web agent for document categorization and exploration. In *AGENTS '98* (pp. 408–415).

Hart, P. E. (1968). The condensed nearest neighbor rule. *IEEE Transactions on Information Theory, IT-14*, 515–516.

Japkowicz, N. (2000). The class imbalance problem: Significance and strategies. In *IC-AI'2000* (Vol. 1, pp. 111–117).

Japkowicz, N., & Stephen, S. (2002). The class imbalance problem: A systematic study. *Intelligent Data Analysis, 6*(5), 429–449.

Maimon, O., & Rokach, L. (Eds.). (2005). *The data mining and knowledge discovery handbook*. Springer.

Ko, Y., & Seo, J. (2009). Text classification from unlabeled documents with bootstrapping and feature projection techniques. *Information Processing and Management, 45*(1), 70–83.

Kubat, M., Matwin, S. (1997). Addressing the curse of imbalanced training sets: One-sided selection. In *ICML 1997* (pp. 179–186).

Laurikkala, J. (2001). Improving identification of difficult small classes by balancing class distribution. Tech. Rep. A-2001-2, University of Tampere.

Lewis, D. D., Yang, Y., Rose, T. G., & Li, F. (2004). Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research, 5*, 361–397.

Paradis, F., & Nie, J.-Y. (2007). Contextual feature selection for text classification. *Information Processing and Management, 43*(2), 344–352 (special issue on AIRS2005: Information Retrieval Research in Asia).

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys, 34*(1), 1–47.

Steyvers, M., Griffiths, T. (2007). Probabilistic topic models. In *Handbook of latent semantic analysis*.

Tan, S. (2005). Neighbor-weighted k-nearest neighbor for unbalanced text corpus. *Expert Systems with Applications, 28*(4), 667–671.

Tan, P.-N., Steinbach, M., & Kumar, V. (2005). *Introduction to data mining*. Addison Wesley.

Tomek, I. (1976). Two modifications of cnn. *IEEE Transactions on Systems Man and Cybernetics, 769*, 772.

Wilson, D. L. (1972). Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man and Cybernetics, 2*(3), 408–421.

Wu, J., Xiong, H., Wu, P., Chen, J. (2007). Local decomposition for rare class analysis. In *KDD '07* (pp. 814–823).

Yang, Y., Liu, X. (1999). A re-examination of text categorization methods. In *SIGIR '99* (pp. 42–49).

Zheng, Z., Wu, X., & Srihari, R. (2004). Feature selection for text categorization on imbalanced data. *SIGKDD Explorations Newsletter, 6*(1), 80–89.

Zhu, X., & Wu, X. (2004). Class noise vs. attribute noise: A quantitative study. *Artificial Intelligence Review, 22*(3), 177–210.