WEMOTE - Word Embedding based Minority Oversampling Technique for Imbalanced Emotion and Sentiment Classification

Tao Chen[1], Ruifeng Xu[1], Bin Liu[1], Qin Lu[2],Jun Xu[1]
1 Key Laboratory of Network Oriented Intelligent Computation, Shenzhen Graduate School,
Harbin Institute of Technology, Shenzhen, China
{chentao1999, xuruifeng.hitsz, hit.xujun}@gmail.com,
bliu@insun.hit.edu.cn
2 Department of Computing, The Hong Kong Polytechnic University, Hong Kong
csluqin@comp.polyu.edu.hk

Abstract. Imbalanced training data always puzzles the supervised learning based emotion and sentiment classification. Several existing research showed that data sparseness and small disjuncts are the two major factors affecting the classification. Target to these two problems, this paper presents a word embedding based oversampling method. Firstly, a large-scale text corpus is used to train a continuous skip-gram model in order to form word embedding. A feature selection and linear combination algorithm is developed to construct text representation vector from word embedding. Based on this, the new minority class training samples are generated through calculating the mean vector of two text representation vectors in the same class until the training samples for each class are the same so that the classifiers can be trained on the fully balanced dataset. Evaluations on NLP&CC2013 Chinese micro blog emotion classification (multi-label) and English Multi-Domain Sentiment Dataset version 2.0 (single label) show that the proposed oversampling approach improves the imbalanced emotion/sentiment classification in Chinese (sentence level) and English (document level) obviously. Further analysis show that our approach can reduce the affection of data sparseness and small disjuncts in imbalanced emotion and sentiment classification.

Keywords: Imbalanced training, Oversampling, Emotion Classification, Sentiment Classification

1 Introduction

In the text emotion and sentiment classification tasks, the imbalanced training data are widely existed, where at least one class is under-represented relative to the others. The hitch with imbalanced datasets is that classification algorithms are often biased towards the majority class and therefore there is a higher misclassification rate for the minority class [1]. Such imbalanced training problem is common in many "real-world" machine learning systems from various domains. Thus, it is considered one of the key problems in categorization and data mining today [2].

Many solutions have been proposed to deal with this problem. Generally speaking, they can be categorized into three major groups [1]: *algorithmic modification* [3], *cost-sensitive learning* [4] and *data sampling* [5]. The algorithmic modification approach is oriented towards the adaptation of base learning methods to be more attuned to class imbalance issues [1]. The cost-sensitive learning approach considers the higher costs for the misclassification of the majority class with respect to the minority

class during the classifier training. The data sampling approach aims to produce a balanced class distribution through adjustment of the training data before classifier training. It attracts more research interest for its good generality. Typical data sampling methods include undersampling, oversampling and hybrid methods [1]. In which, undersampling method eliminates majority class instances while oversampling methods usually create new instances for minority class. Hybrids methods combine the two sampling methods above. The most renowned oversampling methods are the Synthetic Minority Oversampling TEchnique (SMOTE) [6] and its varieties: Borderline-SMOTE [7], Safe-level-SMOTE [8], DBSMOTE [9] etc.

Several investigations pointed out that some data intrinsic characteristics, such as data sparseness and small disjuncts [10], are the major sources where the difficulties for imbalanced classification emerge [1]. The performance for text classification degrades when the bag-of-word representation is used which generates thousands of uni-gram and/or bi-gram features. The high feature dimension leads to the data sparseness and small disjuncts. Small disjuncts make the over-sampling methods create more specific decision regions and leads to overfitting which affects the performance of classifiers [9]. Take three sentences "*I love you*", "*I like you*" and "*I hate you*" for example, using one-hot bag-of-word representation, the new sample generated from the first two sentences by SMOTE algorithm is more close to sentence "*I hate you*" then "*I like you*" which is not reasonable.

Unlike bag-of-word representation, word embedding (**WE**) is a distributed representation for words [11] through learning vector representations from a neural probabilistic language model. Since it is dense, low-dimensional and continuous, word embedding is expected to reduce data sparseness and small disjuncts in classification. Therefore, in this paper, we propose a general word embedding based oversampling method for unbalanced emotion and sentiment classification. To this end, firstly, we use a large-scale raw corpus to train a continuous skip-gram model [12] for constructing word embeddings. A feature selection method and a linear combination algorithm are developed to construct sentence/document level text representation vectors (**TRV**s) from training corpus using word embedding. These TRVs are dimension-controllable and float-valued as well as each dimension is normal distributed. To some degree, they capture the semantic information of the original text. The TRVs are employed as the input for oversampling. Through iteratively generate the mean vector of two randomly selected TRVs in the minority class as a new sample, the training dataset are fully balanced. We name this method as Word Embedding based Minority Oversampling TEchnique (WEMOTE). Based on this, we further develop a cluster-based oversampling method: CWEMOTE. In this method, clustering within each class is performed before applying WEMOTE on each cluster in order to improve the generality of WEMOTE.

The proposed oversampling techniques are evaluated on two dataset. Evaluation on NLP&CC2013 Chinese micro blog emotion classification dataset (sentence level with multi-labels) show that the proposed method achieves 48.4% average precision which is 11.9% higher than the state-of-art performance on this dataset (at 36.5%). Evaluations on English Multi-Domain Sentiment Dataset version 2.0 (document level with single label)[1] show that the proposed method achieves at most 83.5% geometric mean which is a 7.7% improvement over baseline system [13] (at 75.8%). The obtained obvious performance improvement show that the proposed word vector based oversampling method improves the performance of emotion classification and sentiment classification effectively.

[1] http://www.cs.jhu.edu/~mdredze/datasets/sentiment/

The rest of this paper is organized as follows. Section 2 briefly reviews related works. Section 3 presents the proposed oversampling method. Evaluations and discussions are given in Section 4. Finally, Section 5 gives the conclusion and future directions.

2 Related work

2.1 Oversampling method for imbalanced problem

Randomly duplicate of the minority class sample is an intrinsic oversampling method for imbalance problem. It is shown not very effective [6]. Chawla et al. designed a synthetic oversampling method called SMOTE, which generates synthetic instances along a line segment that join each instance to its selected nearest neighbor [6]. Borderline-SMOTE and Safe-Level-SMOTE are two varieties of SMOTE. The former one only generates the borderline instances of a minority class as new samples rather than all instances of the class like SMOTE does [7]. The latter one generates the safe instances which mainly near the centre of a minority class as new samples [8].

Based on synthetic oversampling method, DBSMOTE is a typical cluster-based sampling algorithms, which combines DBSCAN and SMOTE to generate a directly density-reachable graph before oversampling [9]. MWMOTE identifies and weights the hard-to-learn informative minority class samples according to their Euclidean distance from the nearest majority class samples before oversampling [14]. ADASYN (Adaptive Synthetic Sampling Approach for Imbalanced Learning) uses a weighted distribution for different minority class examples according to their learning difficulty in order to adaptively shifting the classification decision boundary toward the difficult examples [15].

2.2 Learning distributed word representations

Distributed representation is proposed by Hinton et al. [16] is a low-dimensional float vector for text representation. Distributed representation for words is usually called word representation or word embedding. It is dense, low-dimensional and continuous. It is shown effective to capture a large number of precise syntactic and semantic word relationships [12].

Word embeddings are typically induced by neural language models, which use neural networks as the underlying predictive model [17]. Bengio et al. propose a feedforward neural network with a linear projection layer and a non-linear hidden layer to construct neural language model [18]. This neural language model is applied to predict the current word when the previous n-1 words are given. Experiment results show that word embedding decreases the perplexity for 10-20% compared to the smoothed trigram. C&W model [19] introduced by Collobert and Weston is another neural language model for learning word embedding. It is also based on the syntactic contexts of words. It substitutes the center word of a sentence by a random word to generate a corrupted sentence as negative samples. The training objective is to minimize the loss function so that the original sentence can obtain a higher score than the corrupted sentence.

The main drawback of the neural probabilistic language models is that the training and testing are costly. The hierarchical log-bilinear model introduced by Mnih and Hinton is a fast hierarchical language model along with a simple feature-based algorithm for automatic construction of word trees from the data [20]. In feedforward networks, context of a word is limited to a window of n words. Mikolov et al. introduced recurrent neural networks based language model (RNNLM) [21] in which the

context of a word is represented by neurons with recurrent connections, so that the context length is unlimited.

2.3 Emotion and Sentiment Classification

Cambria et al. group the existing sentiment analysis approaches into four main categories: keyword spotting, lexical affinity, statistical methods and concept-based techniques [22]. Keyword spotting classifies text by affect categories based on the presence of unambiguous affect words. It is weak in two areas: it can't reliably recognize affect negated words, and it relies on surface features. Lexical affinity not only detects obvious affect words, it also assigns arbitrary words a probable "affinity" to particular emotions. Statistical methods use the classifiers such as Support Vector Machine, Hidden Markov Model and Naive Bayes for emotion classification. Generally, they are semantically weak, and don't work well on smaller text units such as sentences or clauses. Concept-based approaches use web ontologies or semantic networks such as SenticNet [23] to accomplish semantic text analysis [24-26]. They can analyze multi-word expressions that don't explicitly convey emotion, but are related to concepts that do.

3 Our approach

Target to the imbalance training problem, we propose a word embedding based minority oversampling technique for imbalanced emotion and sentiment classification. The framework of our approach is shown as Fig. 1. Firstly, large-scale raw text is used to train word embedding (Section 3.1). A feature selection algorithm is employed to select the words reflecting the semantics of text in imbalanced corpus (Section 3.2). Secondly, the sentence level and document level TRVs are learnt by a linear combination algorithm (Section 3.3). Finally, WEMOTE/CWEMOTE algorithms are developed to construct balanced corpus for classifier training, respectively (Section 3.4).

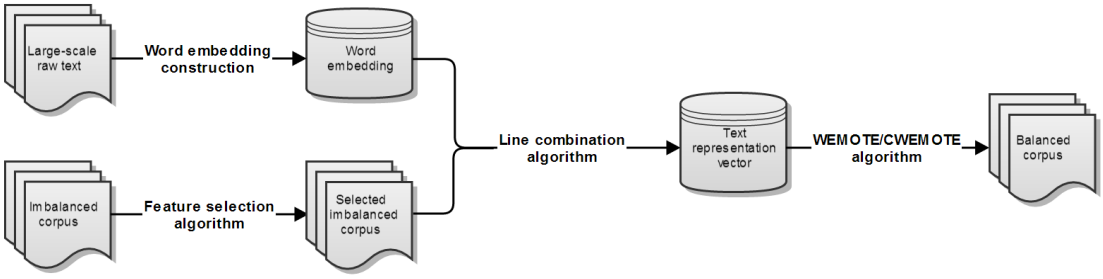

Fig. 1 An overview of our approach

3.1 Word embedding construction

Mikolov et al. introduced continuous Skip-gram model for learning high quality vector representations that capture a large number of syntactic and semantic word relationships from large unstructured text data [12]. The training objective is to find word representations for predicting the surrounding words in a sentence or a document. Given a sequence of training words $w_1, w_2, w_3, \ldots w_T$, the training objective is to maximize the average log probability

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\le i\le c, i\neq 0}\log p(w_{t+i}|w_t) \tag{1}$$

where c is the size of the training context, w_t is the center word. It uses the hierarchical softmax to reduce the computational complexity. Here, $p(w_{t+i}|w_t)$ is defined as below:

$$p(w_{t+i}|w_t) = \prod_{j=1}^{L(w_{t+i})-1} \sigma\left(\| n(w_{t+i}, j+1) = \mathrm{ch}\left(n(w_{t+i}, j)\right) \|\cdot {v'}_{n(w_{t+i},j)}{}^{\mathrm{T}} v_{w_t}\right) \quad (2)$$

where $(x) = \frac{1}{1+e^{-x}}$, $\| \mathrm{x} \| = \begin{cases} 1, & if\ x\ is\ true \\ -1, & else \end{cases}$. The hierarchical softmax uses a binary tree representation of the output layer with all the words as its leaves. $n(w, j)$ is the j-th node on the path from the root to w. $L(w)$ is the length of this path. ch(n) is an arbitrary fixed child of n. ${v'}_n$ is the representation of inner node n. v_w is the representation of word w.

Considering the fact that a word with different part-of-speech (POS) always leads to different lexical functions and representations, in this study, the word/POS pair is used as the basic lexical unit. We make use of a large micro blog corpus to learn the Chinese word embedding through using word2vec[2]. The dimension for each word vector is set by various applications ranging from 80 to 640 in other works [12]. In this study, the vector dimensions is empirically set to be 200 (denoted as 200-d to stand for 200 dimensions). As for English word embedding, several kinds of pre-trained entity vectors provided by word2vec web site are adopted.

3.2 Feature selection methods for TRV construction

Given word embeddings, we need select the words which reflect the core semantics of text and filter out the useless words. Many oversampling method researchers used corpus special features [6-9]. In this study, we propose two feature selection methods for word embedding based text categorization. One is POS-based filtering and another one is vector reversion for negation sentences as explained below:

- POS-based filtering: In this procedure, the punctuations, functional words and symbols are filtered out. Only the content words such as nouns, verbs, adjectives and negative adverb are reserved for TRV construction. It is because that word embedding only captures semantic word relationships such as *common capital city*, *all capital cities*, c*urrency*, *city-in-state*, *man-woman* [12]. They are all relationships between nouns. Thus, the sense representation ability of word embedding on nouns is much better than that of verbs and adjectives. In this procedure, verbs, adjectives and negative adverbs are remained in order to keep the text meaning complete. For example:

  Sentence 1: "*My/PRP$ dog/NN also/RB likes/VBZ eating/VBG sausage/NN ./.*"

  after lemmatization and POS-based filtering: "*dog like eat sausage*"

  word embedding used for TRV of sentence 1 can be: $(v_{\mathrm{dog}}, v_{\mathrm{like}}, v_{\mathrm{eat}}, v_{\mathrm{sausage}})$
- Negative expression processing: A non-obvious degree of language understanding can be obtained by using basic mathematical operations on the word embedding [12]. Vectors addition can produce composite meaning, while vectors reversion can produce negative meaning. In this step, the negation adverbs, such as "*no*, *not*, *never*, *rarely*, *scarcely*" in English and "不(*no/not*)", "没有(*without*)" , "未(*did not*)" , "别(*do not*)" , "不必(*need not*)" , "不曾(*never*)" in Chinese, are used as clues to identify negation sentences. The word vector of the first verb/adjective/noun following the negation adverb is marked as negative to reflect the reversion in polarity.

[2] https://code.google.com/p/word2vec/

Given the following example sentence:

Sentence 2: "*My/PRP$ dog/NN does/VBZ not/RB like/VB eating/JJ sausage/NN ./.*"

after lemmatization and POS filtering: "*dog not like eat sausage* "

the word embedding used for TRV of sentence 2 can be: $(v_{\text{dog}}, -v_{\text{like}}, v_{\text{eat}}, v_{\text{sausage}})$

3.3 Text representation vector construction

Given a dictionary of word embedding $v = (v_{w_1}, v_{w_2}, \ldots, v_{w_n})$, there are two ways to learn the text representation vector. One is linear combination and the other one is composition. Compared to composition approach, such as semi-supervised recursive autoencoder (RAE) network, linear combination approach is shown more efficient. Thus, in this paper, we use linear combination approach to construct TRV.

$$TRV = \sum_{i=1}^{n} v_{w_i} \tag{3}$$

For a given text, its TRV is defined as the sum of all the word embedding as given in Equation 3, where v_{w_i} represents the word embedding of the *i*-th content word in the text, *n* is the total number of content words in the text.

As an example, the TRV corresponding to sentence 1 is defined as:

$$TRV_{\text{sentence1}} = v_{dog} + v_{like} + v_{eat} + v_{sausage}$$

TRV corresponding to sentence 2 is defined as:

$$TRV_{\text{sentence2}} = v_{dog} - v_{like} + v_{eat} + v_{sausage}$$

3.4 WEMOTE/CWEMOTE algorithm

This section presents the over-sampling for TRV. After the TRVs for a corpus is constructed by using the above feature selection and corpus special feature such as emotion/topic words, TF-IDF etc. The observation on real data shows that compared to bag-of-word representation, TRVs are denser with smaller disjunctions within class.

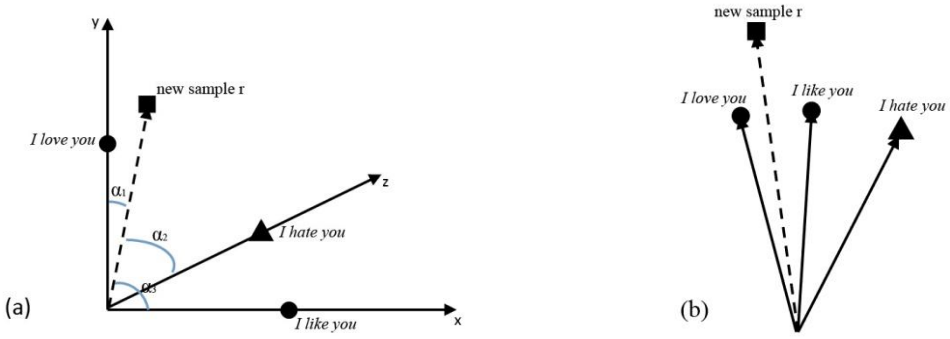

Fig. 2 An illustration of new sample generated by SMOTE algorithm using (a) bag-of-word representation, (b) word embedding

Take sentences "*I love you*", "*I like you*" and "*I hate you*" for example again. Define the first two sentences belongs to class *c* and the last one is belong to class *c'*. As illustrated in Fig. 2 (a) below, using one-hot bag-of-word representation, these three sentences have the same distance between each other. The neighborhoods of "*I hate you*" belong to different class. Thus, "*I hate you*" is regarded as a small disjunction in class *c*. The new sample generated from "*I love you*", "*I like you*" by SMOTE algorithm is more close to "*I hate you*" than "*I like you*", because $\alpha_2 < \alpha_3$. Fig. 2 (b) illustrates the TRVs of "*I love you*" and "*I like you*" are close to each other while both of them are far from the TRV

of "*I hate you*". Thus, the new sample created by SMOTE algorithm is also close to them but far from the TRV of "*I hate you*".

We also found the mean vector v_{mean} of two randomly selected TRVs v_1 and v_2 in the same minority class *c* has features listed below:

1. Using cosine distance as similarity metric, v_{mean} is always tends to be similar to v_1 and v_2, if v_1 is similar to v_2. As illustrated in Fig. 3, θ_1 and θ_2 is always smaller than θ. It indicates that v_{mean} is always similar with v_1 and v_2. The observation shows that TRVs achieves better clustering than bag-of-word representation. In some degree, v_{mean} is more similar to the TRVs in class *c* rather than the TRVs in other classes.

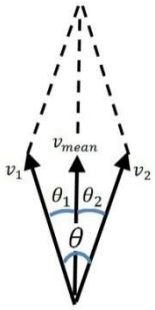

Fig.3 An illustration of mean vector and original vectors

2. The new generated samples have the same distribution with original samples. TRVs constructed by linear combination of word embedding proposed above obey the d-dimensional normal distribution. Where d is the dimensional of word embedding. Fig. 4 illustrates the top 4 dimensions of TRVs for NLP&CC2013 dataset. Each of them obeys a normal distribution.

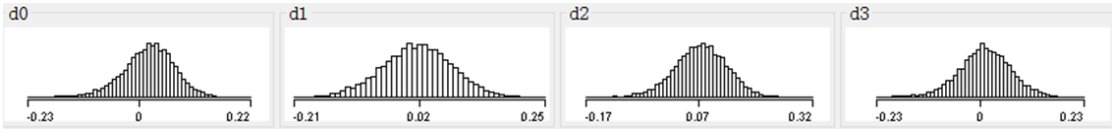

Fig. 4 An illustration of front 4 dimensions of TRVs for NLP&CC2013 dataset

Let $v_1 \sim N_d(\mu, \Sigma)$, $v_2 \sim N_d(\mu, \Sigma)$, then

$$v_{mean} = \frac{1}{2}(v_1 + v_2) \sim N\left(\frac{1}{2}(\mu + \mu), \frac{1}{2}(\Sigma + \Sigma)\right) = N_d(\mu, \Sigma) \quad (4)$$

Where $\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_d \end{pmatrix}$ is the mean vector of all the TRVs, μ_1 is the mean of the first dimensional of all the TRVs, μ_d is the mean of the d-th dimensional of all the TRVs, $\Sigma_1 = \begin{pmatrix} \sigma_{11}\ \sigma_{11} & \cdots & \sigma_{1d} \\ \sigma_{21}\ \sigma_{21} & \cdots & \sigma_{2d} \\ & \vdots & \\ \sigma_{d1}\ \sigma_{d1} & \cdots & \sigma_{dd} \end{pmatrix}$ is the covariance matrix of all the TRVs, $N_d(\mu, \Sigma)$ is the d-dimensional normal distribution which mean is μ and variance is Σ.

3. Many oversampling methods, such as SMOTE and its varieties need to determine the *k* nearest neighbors for one minority class sample. The time complexity of these methods is $O(n^2)$, where *n* is the number of training samples. The time complexity of the computation of mean vector for all the

TRVs in a class is $O(n_c)$, where n_c is the number of samples need to be generated in minority class c.

Based on these analysis, an oversampling method is designed which eventually balance each class. Assume the size of the maximum training samples of all classes is *M*, for any class *c* whose sample size is less than *M*, two TRVs in this class are randomly selected. Their mean vector is generated as a new sample for class *c*. This process repeats until the number of sample vectors in *c* reaches *M*. In this way, the training data of all different classes are fully balanced. We call the method WEMOTE because it is word embedding base oversampling method.

Considering the in-class imbalance problems (the TRVs of a minority class may cluster into more than one imbalanced sub-clusters), we cluster the training samples in the minority class before performing WEMOTE. We call this method CWEMOTE. In this study, *k*-means clustering is adopted. The number of samples generated for *i*-th sub-cluster of the minority class *c* is

$$n_i' = \lfloor \frac{n_i}{\sum_{j=1}^{m_c} n_j} n_c \rfloor \quad (5)$$

Where n_i is the size of *i*-th subcluster in *c*, m_c is the number of subclusters in *c*, n_c is the number of samples need to be generated in *c*. $\lfloor x \rfloor$ is the ceiling function which map a real number to the smallest following integer.

4 Evaluation and Discussions

In this section, the performance of WEMOTE/CWEMOTE are evaluated on a sentence level Chinese emotion classification dataset (with multi-labels) and a document level English sentiment classification dataset (with single label), respectively.

4.1 Chinese Emotion Classification on Micro Blog Text

4.1.1 Experiment Setup and Datasets.

The NLP&CC2013 Chinese micro blog emotion classification dataset (in short, NLP&CC2013 dataset) is adopted in the first group of experiment. It has seven emotion categories, namely *like*, *disgust*, *happiness*, *anger*, *sadness*, *surprise* and *fear*. The sentences are labeled with up to two emotion categories. The numbers and percentages of each emotion class in the training set and testing set are listed in Table 2, respectively. Note that in the training set, the size of the largest class, *like*, is about 4 times and 11 times of *surprise* and *fear*, respectively. Obviously, the training data is imbalanced.

Table 1. Emotional class distribution in NLP&CC2013 dataset

	Training Set				Testing Set			
Class	Primary Emotion		Secondary Emotion		Primary Emotion		Secondary Emotion	
Like	1226	24.8%	138	21.6%	2888	27.6%	204	26.1%
Disgust	1008	20.4%	187	29.2%	2073	19.8%	212	27.1%
Happiness	729	14.7%	95	14.8%	2145	20.5%	138	17.6%
Anger	716	14.5%	129	20.2%	1147	10.9%	82	10.5%
Sadness	847	17.1%	45	7.0%	1565	14.9%	84	10.7%
Surprise	309	6.2%	32	5.0%	473	4.5%	43	5.5%
Fear	114	2.3%	14	2.2%	186	1.8%	20	2.6%

A 3-billion-word Chinese micro blog corpus from weibo.com is used to train the continuous skip-gram model for generating word embedding while word/POS pair is used as the basic lexical unit. Empirically, the context window size is set to 10 words and the vector dimension is set to 200, respectively. Finally, 454,071 word embeddings are obtained.

Average precision, the metric adopted in NLP&CC2013 emotion classification evaluation, is used here as the performance measure. There are loose measures and strict measures in the evaluation according to different scoring criteria for the secondary emotion, respectively.

4.1.2 Experimental Results

A multi-label *k*-nearest neighboring (ML-*k*NN) classifier is trained using the orginial and balanced training data, respectively. In the nearest neighbor estimation, the similarity between two sentences is estimated by the cosine of the angle between their corresponding TRVs. Table 2 shows the achieved performance. Here, *k*=41 is used for the ML-*k*NN classifier.

Table 2. Emotion classification results with different training set and oversampling methods

Method	Average Precision(Bag-Of-Words unigram)		Average Precision (Word embedding)	
	Loose	Strict	Loose	Strict
Original training set	0.144	0.141	0.334	0.325
Duplicating instances	0.158	0.154	0.383	0.369
SMOTE	0.185	0.177	**0.501**	**0.478**
Borderline-SMOTE	0.175	0.167	0.416	0.397
Safe-Level-SMOTE	0.275	0.264	0.434	0.420
WEMOTE	-	-	0.484	0.468
CWEMOTE	-	-	**0.489**	**0.473**

For comparison, five different methods which use Bag-Of-Words and word embedding as features respectively are listed in Table 2. *Duplicating instances* indicates the oversampling method by randomly duplicate existing training samples as new samples for the minority classes. *SMOTE* oversamples all of the training samples for the minority classes while *Borderline-SMOTE* only oversamples the samples near borderline and *Safe-Level- SMOTE* only oversamples the safe samples. The definition of safe samples can be found in [9]. *WEMOTE* and *CWEMOTE* only use word embedding as features.

As shown in Table 3, Safe-Level-SMOTE with Bag-Of-Words features improves the performance significantly than SMOTE and Borderline-SMOTE, while Safe-Level-SMOTE with word embedding features not. It indicates that Bag-Of-Words representation has more small disjuncts than word embedding, because Safe-Level-SMOTE only oversamples the samples which have more positive neighbors than negative neighbors [9].

WEMOTE achieves 0.484 (loose) and 0.468 (strict) average precisions, respectively. Compared to the top performance in the 19 submitted systems in NLP&&CC2013, namely 0.365(loose) and 0.348(strict)[3]average precision, our approach improves 0.119 (loose) and 0.120 (strict) which means the 32.6% and 34.5% relative improvement, respectively. Meanwhile, SMOTE with word embedding features achieves 0.501(loose) and 0.478 (strict) average precisions which is much better than SMOTE with Bag-Of-Words unigram features. It shows that word embedding features are effective to reduce the affection of data sparseness and small disjuncts in imbalanced classification.

[3] http://tcci.ccf.org.cn/conference/2013/dldoc/evres02.pdf

It should be noticed that in this paper our purpose is not to compare WEMOTE with traditional SMOTE model. In fact, WEMOTE is 0.017(loose) and 0.01(strict) lower than SMOTE model with word embedding features. But WEMOTE is more efficient than SMOTE in time complexity as described in section 3.4.

4.2 English Sentiment Classification for Review Documents

4.2.1 Experiment Setup and Datasets.

An English sentiment corpus, namely Multi-Domain Sentiment Dataset (version 2.0), is adopted in the second group of experiments [13]. It contains product reviews taken from Amazon.com from 4 domains: Books, DVDs, Electronics and Kitchen. It consists of 2,000 negative samples from each domain and 14,580/12,160/7,140/7,560 positive samples from the four domains respectively. The imbalanced ratio is 7.29/6.08/3.57/3.78 for each domain respectively.

For English word embedding, we use 1.4M pre-trained entity vectors provided by word2vec web site[1] which is trained on 100B words from various news articles in Freebase.

Geometric mean, the metric adopted in reference [13], is used in the experiment as the evaluation metric.

4.2.2 Experimental Results

LibLINEAR classifier with parameter of "-S 1 -C 1.0 -E 0.01 -B 1.0" is trained using the balanced training data, respectively. For each domain, 2,000 negative samples are oversampled to the same number of positive samples by using WEMOTE method. 10-fold cross-validation is applied in the experiment. Baseline is the approach presented in reference [13]. It uses 7,000 selected samples for manually annotation and SVM as base classifier. Table 3 lists the achieved performance:

Table 3. Performance comparison of sentiment classification for English review documents

	Books	DVDs	Electronics	Kitchen
Baseline	0.758	0.764	0.781	0.815
Original training set (WE features)	0.618	0.604	0.625	0.631
Duplicating instances (WE features)	0.658	0.651	0.671	0.682
SMOTE (WE features)	**0.841**	**0.840**	**0.823**	**0.832**
WEMOTE	0.833	0.820	0.796	0.822
CWEMOTE	**0.835**	**0.825**	**0.803**	**0.829**

As shown in Table 3, SMOTE with word embedding features achieves best results, but it is not as efficient as WEMOTE. WEMOTE outperformed the baseline method by 0.075/0.056/0.015/0.007 in the four domains respectively. For CWEMOTE, it is 0.077/0.061/0.022/0.014. It is also inefficient in time complexity because it should do clustering before oversampling. These results show that our approach improves the performance of English sentiment classification at document level.

5 Conclusion

This paper presents a general oversampling method using text representation vectors learning from word embedding to improve imbalanced text categorization including text emotion/sentiment classification. Through iteratively generating new minority class training samples by using the mean vector of two existing text representation vectors, the training dataset are fully balanced. Evaluations on

two different dataset show that the proposed over-sampling method is effective to reduce the influence of data sparseness and small disjuncts in imbalanced emotion and sentiment classification. As the result, the approach improves imbalanced emotion/sentiment classification in English and Chinese, at sentence level (multi-labels) to document level (single label), respectively. Future directions of this research include the improvement of semantic representation ability of word embedding and the development of composition TRV construction approach for improving the semantic representation ability of TRVs.

6 Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 61300112, 61370165, 61203378), Natural Science Foundation of Guangdong Province (No. S2012040007390, S2013010014475), the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry, MOE Specialized Research Fund for the Doctoral Program of Higher Education 20122302120070, Open Projects Program of National Laboratory of Pattern Recognition, Shenzhen International Co-operation Research Funding GJHZ20120613110641217, Shenzhen Foundational Research Funding JCYJ20120613152557576 and JC201005260118A.

Reference

1. V. López, A. Fernández, S. García, V. Palade and F. Herrera. "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics." *Information Sciences*, 250：113-141, 2013.
2. Q. Yang and X.-D. Wu. "10 challenging problems in data mining research." *International Journal of Information Technology & Decision Making*, 5(4):597-604, 2006.
3. Y. Sun, M.-S. Kamel, A.-K.C. Wong, Y. Wang. "Cost-sensitive boosting for classification of imbalanced data." *Pattern Recognition*, 40(12):3358-3378, 2007.
4. N.-V. Chawla, N. Japkowicz and A. Kolcz. "Editorial: Special Issue on Learning from Im-balanced Data Sets." *SIGKDD Explorations*, 6(1):1-6, 2004.
5. Y. Tang, Y.-Q. Zhang, N.-V. Chawla and S. Krasser. "SVMs modeling for highly imbalanced classification." *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 39(1), 281-288, 2009.
6. *N.-V. Chawla, K.-W. Bo*wyer, L.-O. Hall, W.-P. Kegelmeyer. "SMOTE: synthetic minority over-sampling technique." *Journal of Artificial Intelligence Research*, 16:321-357, 2002.
7. H. Han, W.-Y Wang and B.-H. Mao. "Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning." *Advances in intelligent computing*, 878-887, 2005.
8. C. Bunkhumpornpat, K Sinapiromsaran and C Lursinsap. "Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem." *Advances in Knowledge Discovery and Data Mining*, 475-482, 2009.
9. C. Bunkhumpornpat, K Sinapiromsaran and C Lursinsap . "DBSMOTE: density-based synthetic minority over-sampling technique." *Applied Intelligence*, 36(3):664-684, 2012.
10. J. Taeho and N. Japkowicz. Class imbalances versus small disjuncts. *ACM SIGKDD Explorations Newsletter*, 6(1):40-49, 2004.
11. Y. Bengio, H. Schwenk, J.-S. Senécal, F. Morin and J.-L. Gauvain. Neural probabilistic language models. In *Innovations in Machine Learning*, 137-186, 2006.

12. T. Mikolov, K. Chen, G. Corrado and J. Dean. Efficient Estimation of Word Representations in Vector Space. *ICLR Workshop*, 2013.
13. S.-S. Li, S.-F. Ju, G.-D. Zhou and X.-J. Li. "Active learning for imbalanced sentiment classification." In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 139-148, 2012.
14. S. Barua, M.-M. Islam, X. Yao and K. Murase. "MWMOTE--Majority Weighted Minority Oversampling Technique for Imbalanced Data Set Learning." *IEEE Transactions on Knowledge and Data Engineering*, 26(2):405-425, 2014.
15. H.-B. He, B. Yang, E.- A. Garcia and S.-T Li. "ADASYN: Adaptive synthetic sampling approach for imbalanced learning." In *IEEE International Joint Conference on Neural Networks*, 1322-1328, 2008.
16. G.-E. Hinton. "Learning distributed representations of concepts." In *Proceedings of the eighth annual conference of the Cognitive Science Society*, 1-12, 1986.
17. Y. Bengio. "Neural net language models." *Scholarpedia*, 3(1):3881, 2008.
18. Y. Bengio, R. Ducharme, P. Vincent and C. Jauvin. "A neural probabilistic language model." *Journal of Machine Learning Research*, 3:1137-1155, 2003.
19. R. Collobert and J. Weston. "A unified architecture for natural language processing: Deep neural networks with multitask learning." In *Proceedings of the 25th International Conference on Machine Learning*, 160-167, 2008.
20. A. Mnih and G.-E. Hinton. "A Scalable Hierarchical Distributed Language Model." In *NIPS*, 1081-1088, 2008.
21. T. Mikolov, M. Karafi á t, L. Burget, J. Cernock ý and S. Khudanpur. "Recurrent neural network based language model." In *INTERSPEECH*, 1045-1048, 2010.
22. E. Cambria, B. Schuller, Y.-Q. Xia and C. Havasi. "New avenues in opinion mining and sentiment analysis." *IEEE Intelligent Systems*, 28(2), 15-21, 2013.
23. E. Cambria, D. Olsher, and D. Rajagopal. "SenticNet 3: a common and common-sense knowledge base for cognition-driven sentiment analysis. " *AAAI*, 2014.
24. A.-C.-R. Tsai, C.-E. Wu, R.-T.-H. Tsai and J.-Y.-J. Hsu. "Building a concept-level sentiment dictionary based on commonsense knowledge." *IEEE Intelligent Systems*, 28(2), 22-30, 2013.
25. A. Gangemi, V. Presutti and D.-R. Recupero. "Frame-based detection of opinion holders and topics: a model and a tool." *IEEE Computational Intelligence Magazine*, 9(1), 20-30, 2014.
26. S. Poria, A. Gelbukh, A. Hussain, D. Das and S. Bandyopadhyay. "Enhanced SenticNet with affective labels for concept-based opinion mining." *IEEE Intelligent Systems*, 28(2), 31-38, 2013.