Journal of Information Hiding and Multimedia Signal Processing ©2019 ISSN 2073-4212
Ubiquitous International Volume **10**, Number **1**, January **2019**

Deep Feature Weighting with A Novel Information Gain for Naive Bayes Text Classification

Wei He

School of Electronics and Optical Engineering
Nanjing University Of Posts And Telecommunications
No.9, WenYuan Road, Qixia District, Nanjing 210023, P.R.China
1027853246@qq.com

Yun Zhang, Shujuan Yu and Wenfeng Zhu

School of Electronics and Optical Engineering
Nanjing University Of Posts And Telecommunications
No.9, WenYuan Road, Nanjing 210023, P.R.China
Email: y021001@njupt.edu.cn; yusj@njupt.edu.cn; 923525984@qq.com

Received March 2018; revised June 2018

Abstract. *Naive Bayes (NB) has been widely used in text classification tasks for its simplicity and efficiency. However, its feature independence assumption limits its performance and the traditional TFIDF weighting method is not very ideal for its object is the entire corpus and the traditional weighting methods only associate weights with the final classification formula. For the above problems, we proposed a method named deep weighting with information gain of features category and document for Naive Bayes (IGDC-DWNB), which combines two-dimensional information gain of the features and incorporates the weights into conditional probability for deep feature weighting. The experiments on the Chinese and English corpus show that our IGDC-DWNB obtains a better performance than its competitors.*
Keywords: Naive Bayes, Text classification, Feature weighting, Two-dimensional information gains

1. **Introduction.** With the development of the Internet, the increase of text information and its diversification has token much attention on the task of text classification. Although there are many algorithms for text classification, such as SVM, KNN and neural network, Naive Bayes is ever better than other algorithms on simplicity and efficiency[1][2]. Naive Bayes algorithm proposed a feature independence assumption based on Bayes theorem, that is, assuming all the attributes are independent of each other and do not affect the classification results, so Naive Bayes can be used for text multi-category tasks effectively.

The Naive Bayes algorithm is based on the assumption of conditional independence, which represents all features obtaining identical weights. In fact, the importance of each feature is different, that is, the values of the weights are different. Thus, several methods were proposed to weaken the feature independence assumption. For example, [3] utilizes differential evolution algorithms to define the feature weights. Zhang and Sheng use gain ratio to compute the weight of feature, which mean feature with higher gain ratio deserves a higher weight [4]. In [5], Lee proposed an approach named value weighting method, which assign weights according to the value of feature. Li use chi-square score to weighting [6]. Hall built a decision tree to weighting features, which associated with

more other features be assigned lower weight [7]. There are also many feature selection methods to improve the classifier [8, 9, 10, 11]. However, almost all the approaches only incorporate the learned feature weights into the classification of the formula of Naive Bayes [2].

In order to improve the performance of Naive Bayesian classifier, this paper concentrates on the feature weighting methods and combines two-dimensional information gain of features, which are the information gain of category and the information gain of document, we call it IGDC, and then we incorporate the IGDC into conditional probability of Naive Bayes for deep feature weighting. The experimental results show that our method has good performance not only in Chinese text classification but English text classification.

2. **Naive Bayes Text Classification.**

2.1. **Naive Bayes classifier.** The text classification problem belongs to discrete data classification. There are usually two kinds of Bayesian models [12]: one is the Bernoulli Naive Bayes (BNB)[13], which only considers whether the features appeared in the documents. The other is the multinomial Naive Bayes (MNB)[14], which focuses on the number of frequencies of features in the documents. Through the experiment of [15], it was found that the classification effect of multinomial model is better than Bernoulli model. In this paper, the Bayesian model given in [14] is the multinomial model. The idea of the algorithm is: calculate the prior probability of each category, then use Bayes' theorem to calculate the posterior probabilities which are every feature belong to category. By selecting the category with the maximum a posteriori (MAP) to decide categorize.

Assume the document category collection $C = \{C_1, C_2, ..C_j\}$, $j = 1, 2, 3...V$ .In text classification, the document $D_i = \{t_1, t_2....t_m\}$, which is composed by m features.The category of the maximum probability is the category that the document Di belongs to. It can be described by the following equation:

$$P(C_j|D_i) = \frac{P(D_i|C_j)P(C_j)}{P(D_i)} \tag{1}$$

where $P(C_j)$ is the probability of documents which belong to the category C_j ; $P(D_i|C_j)$ is the probability of the documents D_i in the condition that this document belongs to the category C_j ; $P(D_i) = P(t_1, t_2...t_m)$ is the joint probability of all feature. It is obvious that $P(D_i)$ is a constant, the equation (1) can be converted into:

$$C_{map} = \max_{C_j \in C} P(C_j|D_i) = \max_{C_j \in C} P(C_j)P(D_i|C_j) \tag{2}$$

where C_map represents the final classification result.

According to Naive Bayes feature independence assumptionthe equation (2) can be simplified as:

$$C_{map} = \max_{C_j \in C} P(C_j|D_i) = \max_{C_j \in C} P(C_j)P(\{t_1, t_2...t_m\}|C_j) = \max_{C_j \in C} P(C_j) \prod_{k=1}^{m} P(t_k|C_j) \tag{3}$$

where m is the number of features,$t_k(k = 1, 2, 3..m)$is the kth feature word in the test document $D_i$the prior probability $P(C_j)$ and the conditional probability $P(t_k|C_j)$ can be estimated by (4) and (5)respectively:

$$P(C_j) = \frac{\sum_{i=1}^{n} \delta(C_i, C_j) + 1}{n + V} \tag{4}$$

$$P(t_k|C_j) = \frac{\sum_{i=1}^{n} TF_{it_k} \delta(C_i, C_j) + 1}{\sum_{k=1}^{m} \sum_{i=1}^{n} TF_{it_k} \delta(C_i, C_j) + m} \tag{5}$$

where n is the number of training documents, V is the number of classes, C_i is the class label of the ith training document, TF_{it_k} is the frequency count of word t_k in the ith training document, and $\delta(.)$is a binary function, which is define as:

$$\delta(x,y) = \begin{cases} 1 & , \quad x = y \\ 0 & , \quad x \neq y \end{cases} \tag{6}$$

2.2. **TFIDF Feature Weighting.** For naive Bayes algorithm does not consider the impact of different features in the classification, the feature is usually weighting by Term Frequency-Inverse Document Frequency (TF-IDF) algorithm [16] . Term Frequency (TF) is the frequency of a feature appears in a document, and Inverse Document Frequency (IDF) is the ratio of the total number of documents to the number of documents that a feature word appears. It means that the importance of a feature word proportion to its frequency appears in the document directly, but proportional to its frequency in the corpus inversely. The TF-IDF algorithm can be describe as :

$$IDF_{t_k} = lb(\frac{N}{n_{t_k}} + 0.01) \tag{7}$$

$$W_k = TF_{t_k} \times IDF_{t_k} = \frac{TF_{t_k} \times IDF_{t_k}}{\sqrt{\sum_{k=1}^{m} TF_{t_k} \times IDF_{t_k})^2}} \tag{8}$$

where IDF_{t_k} is the Inverse Document Frequency of t_k , TF_{t_k} is the frequency of feature word t_k , N is the total number of training documents, n_{t_k} is the number of documents that the feature word t_k appears in training documents. The ordinary feature weighting Naive Bayes model is described as follows:

$$C_{map} = \max_{Cj \in C} P(C_j|D_i) = \max_{Cj \in C} P(C_j) \prod_{k=1}^{m} P(t_k|C_j)^{W_k} \tag{9}$$

In order to avoid the case of underflow, the final feature weighting model is:

$$C_{map} = \max_{Cj \in C} P(C_j|D_i) = \max_{Cj \in C} [lnP(C_j) + \sum_{k=1}^{m} lnP(t_k|C_j) \times W_k \times TF_{t_k}] \tag{10}$$

2.3. **IGDC Feature Weighting.** Due to TFIDF focuses on the entire corpus, it neglects the influence of distribution of feature words in the category. Although TFIDF algorithm can improve the accuracy of classification, the effect is not obvious. It has verified that the information gain can be used to improve the classifier effectively [17][18]. This paper concentrates on information gain. We define a new weighting calculation function: IGDC function. As information gain is an index that describes the effect of a feature influence on classification, which means a feature with higher information gain deserves a higher weight. Therefore, this paper combines the information gain of the category and documents with deep feature weighting.

Giving a training set, the two-dimensional information gain of feature in the training set can be described as

$$IGDC(t_k) = IGD(t_k) \times IGC(t_k) \tag{11}$$

where $IGDC(t_k)$ represents the two-dimensional information gain of the feature word, $IGD(t_k)$ represents the information gain of t_k in the documents, $IGC(t_k)$ represents the information gain of t_k in the category. They can be computed by the following equations (12) (13):

$$IGC(t_k) = H(C) - H(C|t_k) \tag{12}$$

$$IGD(t_k) = H(C) - H(C|D_{t_k}) \tag{13}$$

where $H(C)$ is the entropy of category, $H(C|t_k)$ is the conditional entropy of feature t_k in the category C and C= $\{C_1, C_2...C_j\}$ $H(C|D_{t_k})$ is the conditional entropy of the document D_{t_k} in the category. D_{t_k} is the document that contain feature t_k .The calculation methods are defined as follows:

$$H(C) = -\sum_{j=1}^{V} P(C_j) \times \log_2 P(C_j) \tag{14}$$

$$H(C|t_k) = -\sum_{j=1}^{V} P(t_k, C_j) \times \log_2 P(t_k, C_j) \tag{15}$$

$$H(C|D_{t_k}) = -\sum_{j=1}^{V} P(D_{t_k}, C_j) \times \log_2 P(D_{t_k}, C_j) \tag{16}$$

where$P(C_j)$ is gave by equation (4)$P(t_k, C_j)$ and $P(D_{t_k}, C_j)$can be calculated by equations (17),(18) respectively

$$P(t_k, C_j) = \frac{tf(t_k, C_j) + L}{\sum_{j=1}^{V} tf(t_k, C_j) + V \times L} \tag{17}$$

$$P(D_{t_k}, C_j) = \frac{tf(D_{t_k}, C_j) + L}{\sum_{j=1}^{V} tf(D_{t_k}, C_j) + V \times L} \tag{18}$$

where $tf(t_k, C_j)$ is the frequency of the feature t_k in the C_j, L is a smoothing factor. In this paper, we take L=0.01, V is the number of classes. $tf(D_{t_k}, C_j)$ is the number of documents D_{t_k} in the class C_j . After obtaining the two-dimensional information gain of the feature $IGDC(t_k)$, we define the weight W_k as:

$$W_k = \frac{IGDC(t_k) - \min[IGDC(t_k)]}{\max[IGDC(t_k)] - \min[IGDC(t_k)]} \tag{19}$$

3. **Deep feature Weighting for Naive Bayes.** Although the traditional weighting methods can improve the Naive Bayesian algorithm, the effect is still not ideal. The main reason is that the traditional weighting methods only incorporate the weight into the final classification formula, that is to say, the features are weighted only once, while they ignore the impact of the weights on the conditional probability. Therefore, in this paper, weights are incorporated not only into the final classification formula of Naive Bayes but also into its conditional probability for ensuring being weighted again, we call this weighting method deep feature weighting (DFW).

Deep feature weighting model:

$$C_{map} = \max_{Cj \in C} P(C_j|D_i) = \max_{Cj \in C} [lnP(C_j) + \sum_{k=1}^{m} lnP(t_k|C_j, W_k) \times W_k \times TF_{t_k}] \tag{20}$$

where W_k is the weight of feature t_k, TF_{t_k} is the frequency of feature t_k in C_j, $P(t_k|C_j, W_k)$ is the conditional probability with deep weighting, which is different from the existing deep feature weighting methods in[1,2], this paper propose a new deep feature weighting formula (21) to improve the performance:

$$P(t_k|C_j, W_k) = \frac{\sum_{i=1}^{n} W_k TF_{it_k} \delta(C_i, C_j) + 1}{\sum_{k=1}^{m} \sum_{i=1}^{n} TF_{it_k} (W_k + 1) \delta(C_i, C_j) + m} \tag{21}$$

where m is the number of feature words.

The deep feature weighting for Naive Bayesian model is defined as DFWNB, and we apply IGDC to the deep feature weighting Naive Bayesian model to obtain the IGDC-DFWNB. The pseudocode of the classification process is as follows:

Algorithm: IGDC-DFWNB

Input: training set $D = \{D_1, D_2...D_i\}$, document $D_i = \{t_1, t_2...t_k\}$

Output: document label

(1) For each feature word t_k in training document D_i:
Calculates IGDC for each feature word t_k;

(2) For each feature word t_k in training document D_i:
Calculate the weight W_k of feature words;

(3) For document D_i from test set D:
Calculate the prior probability $P(C_j)$;
Calculate the conditional probability $P(t_k|C_j, W_k)$
Calculate posterior probability $P(C_j|d)$;

(4) Return the label of the document d according to $P(C_j|d)$.

4. **Experiment and Results.**

4.1. **Experimental dataset and evaluating indicators.**

Datasets used in our experiment are shown in Table1:

Table 1. Dataset.

Dataset	Documents	classes
20Newsgroup	1200	6
Reuters21578	900	6
Sougou Lab corpus	600	6
Fudan University corpus	600	6

Experiment selects six classes from those dataset and uses cross-validation method to verify the algorithms performance. Then we select 40% as training sets and 60% as test sets randomly. English text experimental data preprocess: remove punctuation, stop words, numbers and some special symbols. For Chinese text data, we should use jieba packet to do the word segmentation first and other steps are the same as English corpus.

The experiment uses F1 score, M_P, M_R and M_F1 score to evaluate the performance of classification algorithms and they are calculated as follows:

Precision:

$$P = \frac{TP}{TP + FP} \tag{22}$$

Macro Precision:

$$M\_P = \frac{1}{V} \sum\nolimits_{i=1}^{V} P \tag{23}$$

where V represents the number of categories.

Recall:

$$R = \frac{TP}{TP + FN} \tag{24}$$

Macro Recall:

$$M\_R = \frac{1}{V} \sum\nolimits_{i=1}^{V} R \tag{25}$$

F1 score:

$$F1 = \frac{2 \times P \times R}{P + R} \tag{26}$$

Macro F1 score:

$$M\_F1 = \frac{2 \times M\_P \times M\_R}{M\_P + M\_R} \tag{27}$$

where TP is the number of positive instances which are predicted correctly, the FP is the number of positive instances which are predicted incorrectly, the TN is the number of negative instances which are predicted correctly, and the FN the number of negative instances which are predicted incorrectly, the relationship can be shown in the Table 2:

Table 2. Parameters meaning.

True label	Predicted label	
	Pos	Neg
Pos	TP	FN
Neg	TP	TN

4.2. **Experimental results.** This paper uses four models for comparison, we choose IGDC-DWNB, DFWNB, Rw,cFW, OFWNB respectively:

OFWNB: MNB model employing TFIDF ordinary feature weighting approach [1].

DFWNB: MNB model employing TFIDF deep feature weights approach [1].

$R_{w,c}$FWNB: MNB model employing chi-square ordinary feature weighting approach [6].

IGDC-DFWNB: MNB model employing our IGDC deep feature weighting approach in this paper.

In order to reduce the complexity of space and the time of calculation, we use DF score to select feature words[1], and repeat the experiment ten times then calculate the average score to verify the performance of algorithms. The experimental results are shown in Table 3:

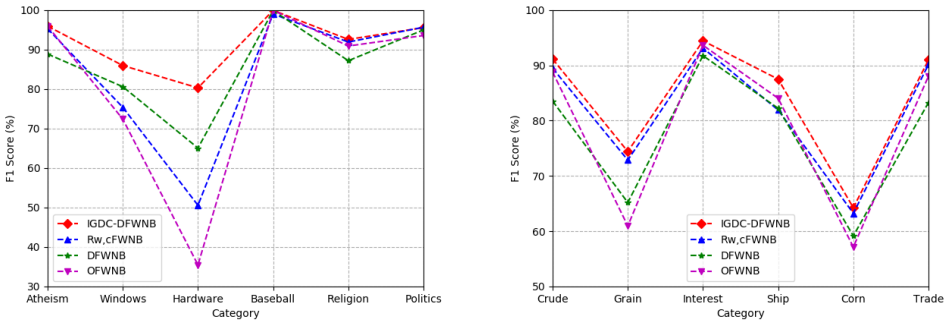
Table 3. Macro score on different datasets.

Dataset	IGDC-DFWNB			$R_{w,c}$FWNB			DFWNB			OFWNB		
	M_P	M_R	M_F1	M_P	M_R	M_F1	M_P	M_R	M_F1	M_P	M_R	M_F1
20_newsgroup	**0.933**	**0.918**	**0.926**	0.909	0.861	0.884	0.907	0.870	0.888	0.898	0.837	0.867
Reuters21578	**0.814**	**0.808**	**0.811**	0.805	0.797	0.800	0.772	0.758	0.764	0.787	0.778	0.782
Sougou corpus	**0.939**	**0.938**	**0.938**	0.908	0.905	0.906	0.885	0.884	0.885	0.908	0.904	0.906
Fudan corpus	**0.955**	**0.955**	**0.955**	0.913	0.900	0.906	0.908	0.899	0.904	0.913	0.900	0.906
Average score	**0.910**	**0.905**	**0.907**	0.883	0.868	0.874	0.868	0.853	0.860	0.876	0.854	0.865

It can be seen from Table 3 that the IGDC-DWNB obtains the best performance on the different datasets.The macro score of IGDC-DWMNB is much better than its competitors

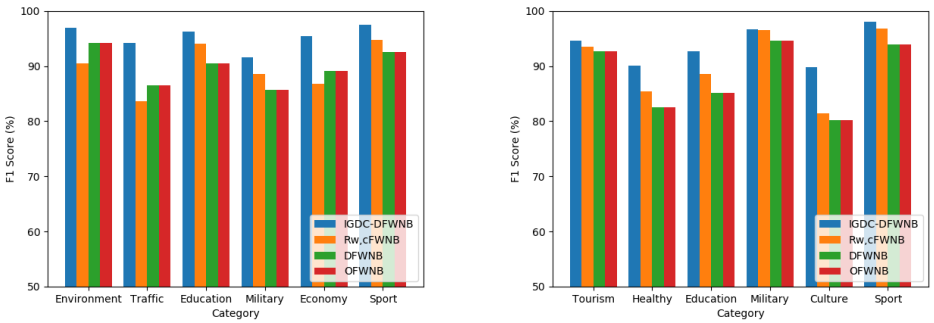
especially OFWNB.Compared to DFWNB and $R_{w,c}$FWNB, it can also increase 3% to 5% on average.

Figure 1,2,3,4 shows that the performance of each algorithm in every categories:



(a) Classification on 20_newsgroup. (b) Classification on Reuters21578.

FIGURE 1. Experimental results on English Text Classification.



(a) Classification on Fudan Corpus. (b) Classification on Sougou Lab Corpus.

FIGURE 2. Experimental results on Chinese Text Classification.

From Fig.1,the F1 score of IGDC-DWMNB is better than other algorithms for English text classification expecially on Windows and Hardware.DFWNB and OFWNB not only cannot classify hardware well but their results are uneven.In contrast, IGDC-DFWNB obtains a stable results which show that the algorithm is more robust in multi-classification tasks.In Fig.2, for Chinese text classification, IGDC-DWNB also obtains best F1 score, while there is no obvious gap between DFWNB and OFWNB.Compared with $R_{w,c}$FWNB, our algorithm is 5% averagely higher on Fudan corpus dataset and 3% averagely higher on Sougou corpus dataset.

5. **Conclusion.** In this paper, a method of deep feature weighting with information gain of features category and document for Naive Bayes (IGDC-DFWNB) is proposed, which combines the information gain of features documents and categories with deep feature weighting. Applying our approach to different dataset and comparing with the traditional TFIDF deep feature weighting and TFIDF ordinary feature weighting, the performance of IGDC-DFWNB is much superior to the traditional methods. For future work, we will use a more effective feature selection algorithm and feature weighting algorithm to improve the classification.

Acknowledgment. The authors would like to thank the support of this work by Grants from the National Natural Science Foundation of China (NSFC)(No. 61302155, No. 61274080)and Nanjing University of Posts and the introduction of talent Project(No. NY214052).

REFERENCES

[1] Q. Jiang, W. Wang, and X. Han, Deep feature weighting in Naive Bayes for Chinese text classification, *International Conference on Cloud Computing and Intelligence Systems*, pp.160-164, 2016.

[2] Jiang.L, Li.C and Wang.S(eds.), Deep feature weighting for naive Bayes and its application to text classification, *Engineering Applications of Artificial Intelligence*, vol. 52, pp.26-39, 2016.

[3] J. Wu, Z.Cai, Attribute Weighting via Differential Evolution Algorithm for Attribute Weighted Naive Bayes (WNB), *Journal of Computational Information Systems*, 2011.

[4] H. Zhang, S. Sheng, Learning weighted naive Bayes with accurate ranking, *International Conference on Data Mining*, pp. 567-570, 2005.

[5] C.H. Lee, A gradient approach for value weighted classification learning in naive Bayes, *Elsevier Science Publishers*, 2015.

[6] Y. J. LI, C. n. Luo, s. chung, weighted naive bayes for text classification using positive term-class dependency *International Journal on Artificial Intelligence Tools*, vol. 21, no. 01, pp.1250008-, 2012.

[7] M. Hall, A Decision Tree-Based Attribute Weighting Filter for Naive Bayes, *Knowledge-Based Systems*, vol. 20, no. 2, pp.120-126, 2007.

[8] P. Bermejo, Speeding up incremental wrapper feature subset selection with Naive Bayes classifier, *Knowledge-Based Systems*, vol. 55, pp.140-147, 2014.

[9] K. Javed, S. Maruf, H. A. Babri, A two-stage Markov blanket based feature selection algorithm for text classification, *Neurocomputing*, vol. 157, pp.91-104, 2015.

[10] M. L. Zhang, J. M. Pena, Robles.V, Feature selection for multi-label naive Bayes classification, *Information Sciences*, vol. 179, no. 19, pp.3218-3229, 2009.

[11] L. Zhang, L. Jiang, C. Li, A New Feature Selection Approach to Naive Bayes Text Classifiers, *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 30, no. 02, 2016.

[12] L. Zhang, L. Jiang, C. Li, Two feature weighting approaches for naive Bayes text classifiers, *Knowledge-Based Systems*, vol. 100, pp.137-144, 2014.

[13] J. M. Ponte, A language modeling approach to information retrieval, *University of Massachusetts*, 1998.

[14] A. Mccallum, K. Nigam, A Comparison of Event Models for Naive Bayes Text Classification, *Workshop on learning for text categorization*, vol. 62, no. 2, pp. 41–48, 1998.

[15] S. Wang, L. Jiang C. Li, A CFS-Based Feature Weighting Approach to Naive Bayes Text Classifiers, *International Conference on Artificial Neural Networks*, pp.555-562, 2014.

[16] G. Forman, BNS feature scaling: an improved representation over tf-idf for svm text classification, *Conference on Information and Knowledge Management*, pp. 263-270, 2008.

[17] K. Q. Li, X. Diao, J. Cao, Improved Algorithm of Text Feature Weighting Based on Information Gain, *Computer Engineering*, vol. 37, no. 1, pp. 16-15, 2011.

[18] L.I. Xue-Ming, L. Hai-Rui , L. Xue, TFIDF Algorithm Based on Information Gain and Information Entropy, *Computer Engineering*, vol. 38, no. 8, pp.37-40, 2012.