PAPER

Improved Gini-Index Algorithm to Correct Feature-Selection Bias in Text Classification

Heum PARK[†a)], *Nonmember and* **Hyuk-Chul KWON**[†b)], *Member*

SUMMARY This paper presents an improved Gini-Index algorithm to correct feature-selection bias in text classification. Gini-Index has been used as a split measure for choosing the most appropriate splitting attribute in decision tree. Recently, an improved Gini-Index algorithm for feature selection, designed for text categorization and based on Gini-Index theory, was introduced, and it has proved to be better than the other methods. However, we found that the Gini-Index still shows a feature selection bias in text classification, specifically for unbalanced datasets having a huge number of features. The feature selection bias of the Gini-Index in feature selection is shown in three ways: 1) the Gini values of low-frequency features are low (on purity measure) overall, irrespective of the distribution of features among classes, 2) for high-frequency features, the Gini values are always relatively high and 3) for specific features belonging to large classes, the Gini values are relatively lower than those belonging to small classes. Therefore, to correct that bias and improve feature selection in text classification using Gini-Index, we propose an improved Gini-Index (I-GI) algorithm with three reformulated Gini-Index expressions. In the present study, we used global dimensionality reduction (DR) and local DR to measure the goodness of features in feature selections. In experimental results for the I-GI algorithm, we obtained unbiased feature values and eliminated many irrelevant general features while retaining many specific features. Furthermore, we could improve the overall classification performances when we used the local DR method. The total averages of the classification performance were increased by 19.4 %, 15.9 %, 3.3 %, 2.8 % and 2.9 % (kNN) in Micro-F1, 14 %, 9.8 %, 9.2 %, 3.5 % and 4.3 % (SVM) in Micro-F1, 20 %, 16.9 %, 2.8 %, 3.6 % and 3.1 % (kNN) in Macro-F1, 16.3 %, 14 %, 7.1 %, 4.4 %, 6.3 % (SVM) in Macro-F1, compared with tf*idf, χ^2, Information Gain, Odds Ratio and the existing Gini-Index methods according to each classifier.

key words: *feature selection, Gini-Index, text classification, dimensionality reduction, feature selection bias*

1. Introduction

In early work, Gini-Index was used as a split measure for splitting attributes in choosing the most appropriate splitting attribute at each node in a decision tree, and in achieving enhanced categorization precision. The recent typical studies on the Gini-Index have concerned a formal methodology for comparing multiple split criteria and a formal description of how to theoretically select between split criteria (Laura Elena and Raileanu 2004), feature construction using Gini-Index for genetic programming with decision classification (Mohammed et al. 2004), varieties of decision tree induction algorithms using splitting methods based on Gini-Index (Pang-Ning Tan et al. 2006), a discretization algorithm based on the Gini criterion for transforming continuous features into a finite number of intervals (Xiao-Hang Zhang et al 2007), and a fuzzy decision tree algorithm Gini-Index (B. Chandra et al. 2009), among still others [1], [6], [8], [10], [15].

And several researchers have indicated that feature selection was biased towards attributes with a large number of possible values, having more values, a larger number of categories, multiple-valued attributes, a large number of missing values, etc, and many studies on unbiased split selection have been introduced [6]. Recently, Carolin Strobl et al. (2007) introduced unbiased split selection for classification trees based on the Gini-Index and a new split selection criterion that avoids variable selection bias on standard impurity measures, and Marco Sandri (2008) presented a simple and effective method for bias correction focused on the easily generalizable case of the Gini-Index [3], [7]. However, those were mostly concerning split selections, not feature selection in text classification.

W. Shang et al. (2006) presented an adaptive Fuzzy kNN classifier based on the Gini-Index for feature selection and introduced a novel feature selection algorithm using Gini-Index for text categorization (2007). They proved that the improvement of classification results using Gini-Index was better than those of other feature selection methods [12], [13]. Additionally Hiroshi Ogura et al. (2008) proposed feature selection with a measure of deviation from the Poisson in text categorization. In their experiments, they compared feature selection performance of the Gini-Index algorithm of W. Shang et al. (2007), in text classification, with their proposed method, and proved that to be better than Information Gain (IG) and χ^2-statistic [5]. Sanasam Ranbir Singh (2010) proposed a feature selection method using 'within class popularity (WCP)' based on the concept of the Gini coefficient of inequality [11].

In using the Gini-Index, if the high-frequency features and all of the members of a feature belong to the same class, the Gini value is 0 (on impurity measure), indicating useful information. And if all of the members of a feature are distributed evenly to all of the classes, it has a high or the maximum value, and is not useful information. Thus, in a decision tree, an attribute can be split easily with the Gini value and have a good performance.

However, feature selection in text classification using the Gini-Index is still biased with regard to unbalanced datasets containing a huge number of features and a large number of documents. When we used the existing

Manuscript received September 16, 2010.
Manuscript revised November 4, 2010.
[†]The authors are with Pusan National University, Korea.
a) E-mail: parkheum2@empal.com
b) E-mail: hckwon@pusan.ac.kr. Corresponding author
DOI: 10.1587/transinf.E94.D.855

Gini-Index algorithm for feature selection with unbalanced datasets, for some of the general and specific features, we found biased Gini-Index values. The specific reasons for that Gini-Index bias in feature selection are 1) the Gini values of low-frequency features are low (on purity measure) overall irrespective of the distribution of features among classes, 2) for high-frequency features, the Gini values are always relatively high and 3) for specific features belonging to large classes, the Gini values are relatively lower than those belonging to small classes. Gini values are more affected by $P(t)$ and $P(t|c_i)^2$ than the distribution of feature frequencies among classes in unbalanced datasets; thus, higher frequency features have higher Gini values than those of low frequency features, where t is a term and c_i is the i-th class among classes.

For ideal estimation of feature subsets in unbalanced datasets, it is necessary to eliminate irrelevant general features, to retain specific features, and to clearly select representative features having the specific characteristics of a document. In feature selection for text classification, the specific low-frequency features are also useful and meaningful, and thus kept as representative features, especially in small documents. For general high-frequency features, if they are distributed to several classes, even though their Gini values are over the threshold, they must be eliminated from feature subsets. Additionally, for specific low-frequency features, it must assess the representative features irrespective of the size of classes they belong to.

Thus, we undertook to reformulate the Gini-Index expressions to avoid those biases and the unbiased Gini-Index algorithm for feature selection in text classification in order to find the best measures for the goodness of features using Gini-Index. To measure the goodness of a feature in feature selection, generally, the average or maximum feature values of classes are used: $f_{avg}(t) = \sum_{i=1}^{|C|} P(c_i)f(t, c_i)$ or $F_{\max}(t) = \max_{i=1}^{|C|} P(c_i)f(t, c_i)$, where t is a term and c_i is the i-th class among classes [4].

Therefore, in this paper, we propose as a means of removing the Gini-Index bias in unbalanced datasets, a new Improved Gini-Index (I-GI) algorithm containing three reformulated Gini-Index expressions for feature selection. We experimented not only with the proposed algorithm (I-GI) but also with Shang's Gini-Index (2007) and the typical feature selection methods: χ^2, Information Gain (IG) and Odds Ratio (OR), using the kNN and SVM classifiers for text classification, and compared their results.

In Sect. 2, we discuss the Gini-Index theory and the existing Gini-Index algorithms presented by W. Shang. In Sect. 3, we introduce the I-GI algorithm using three new reformulated Gini-Index expressions. In Sect. 4, by means of experimental results, we compare and discuss the classification performances for the various feature selection methods. In Sect. 5, we draw conclusions and consider future work.

2. Feature Selection Bias of Gini-Index

2.1 Gini-Index Theory for Feature Selection

The main idea behind Gini-Index theory is as follows. Suppose S is a set of s samples, and that these samples have k different classes ($C_i, i = 1, \ldots, k$). According to the differences between classes, we can divide S into k subsets ($S_i, i = 1, \ldots, k$). Suppose S_i is a sample set that belongs to class C_i, and that s_i is the sample number of sets S_i. Then the Gini-Index of set S is:

$$Gini(S) = 1 - \sum_{i=1}^{k} p_i^2 \tag{1}$$

where P_i is the probability, estimated with s_i/s, that any sample belongs to C_i. $Gini(S)$'s minimum is 0, all of the members in the set belong to the same class, indicating that the maximum useful information can be obtained. When all of the samples in the set distribute equally for each class, $Gini(S)$ is at its maximum, indicating that the minimum useful information can be obtained [6], [12], [13]. However, most studies of Gini-Index have been used only for splitting attributes in a decision tree.

Recently, for feature selection in text classification, W. Shang et al. (2007) presented a novel Gini-Index algorithm based on Gini-Index theory for text feature selection with a new measure function of the Gini-Index. The original form of the Gini-Index algorithm was used to measure the impurity of attributes towards categorization. The smaller the impurity is, the better the attribute is. They adopted the measure of purity, whereby the larger the value of the purity is, the better the attribute is. Their new Gini-Index algorithm has shown better performance in text classification than other feature selection methods [13]. The original form of Gini-Index expression is as follows:

$$\begin{aligned} Gini(W) = {} & P(W)\left(1 - \sum_{i=1}^{m} P(C_i|W)^2\right) \\ & + P(\overline{W})\left(1 - \sum_{i=1}^{m} P(C_i|\overline{W})^2\right) \end{aligned} \tag{2}$$

where W is a feature and C_i is the i-th class among classes. When expression (2) is used, some words that do not appear still contribute to the judging of the text class. However, this contribution is far less significant than that of words that do appear, particularly when the distribution of the class and the feature frequencies are highly unbalanced. Therefore, they eliminated the affection factor expressing words that do not appear, and adopted a measure of purity instead of impurity to emphasize the $P(W)$ factor, namely *Gini-A*, as in expression (3).

$$Gini(W) = P(W) \sum_{i=1}^{m} P(C_i|W)^2 \tag{3}$$

In addition, they adopted, in considering the unbalanced class distribution, the posterior probability when feature W appears $\sum_i P(W|C_i)^2$, to replace $P(W)$, namely *Gini-B*, as shown in expression (4).

$$Gini(W) = \sum_i^m P(W|C_i)^2 P(C_i|W)^2 \quad (4)$$

In this formula, if feature W appears in every document of class C_i, the maximum value, Gini value=1, can be obtained. When the documents distribute evenly where W appears, the minimum Gini value is obtained. The feature W's conditional probability, combining the posterior probability and the conditional probability to depress the affection when the class is unbalanced, was considered [13].

2.2 Feature Selection Bias of Existing Gini-Index

We experimented with those Gini-Index expressions (3) and (4) for feature selection using unbalanced datasets. If all of the members of a feature belonged to the same class, it had a high Gini value (close to the maximum value). And when all of members of that feature were distributed equally for each class, it had a low value. However, for the high-frequency features, it was not always valid. Because the unbalanced datasets contained a large number of features and unbalanced classes, their Gini values are relatively high irrespective of the distribution of features among classes. For the low-frequency features, because the $P(W)$ and $P(W|C_i)^2$ were low, thus the Gini values were relatively low (on purity measure) close to zero, irrespective of the distribution of feature frequencies among classes. As a result, for the high-frequency general features belonging to several classes and the low-frequency specific features belonging to one or two classes, they have similar Gini values and concentrated near one point.

Figure 1 shows the number of features for which the Gini-Index values are between the minimum and maximum values using *Gini-A* and *Gini-B*, respectively, for the Reuters-21578 and Web datasets. The X-axis marks the intervals of the percentages of the Gini values between the minimum and the maximum values divided into 20, and the Y-axis shows the feature counts belonging to each interval. Most of the Gini values are concentrated near the minimum value, because most features are specific low-frequency features or high-frequency features distributed to several classes. Thus, the Gini-Index values in unbalanced datasets are biased.

Table 1 shows the numbers of features for which the Gini values were below 1 % between the minimum and maximum values, and their ratio to the total number of training features, using *Gini-A* and *Gini-B*, for the Reuters-21578 and Web datasets. In the case *Gini-A*, the ratios of the number of features that were below 1 % for the Gini values were 57.8~90 %. In the case of *Gini-B*, the ratios were 91.5~99.6 %. We can see that most of the Gini values were concentrated toward the minimum values and unbalanced. Therefore, it was necessary to normalize the distributions of the Gini values.

Those unbalanced Gini values were caused by the biased Gini-Index. The three specific reasons for the feature selection bias of Gini-Index are as follows.

1) For low-frequency features, the Gini values are always low (on purity measure) overall irrespective of the distribution of features among classes. Even though all the members of a low-frequency feature are distributed to the same class, it cannot always have a high Gini value, because the probability of that feature in the dataset is very low. The low-frequency features always have low Gini values. Thus, it is required to amend the Gini-Index expressions for the specific low-frequency features.
2) For high-frequency features, the Gini values are relatively high irrespective of the distribution of features

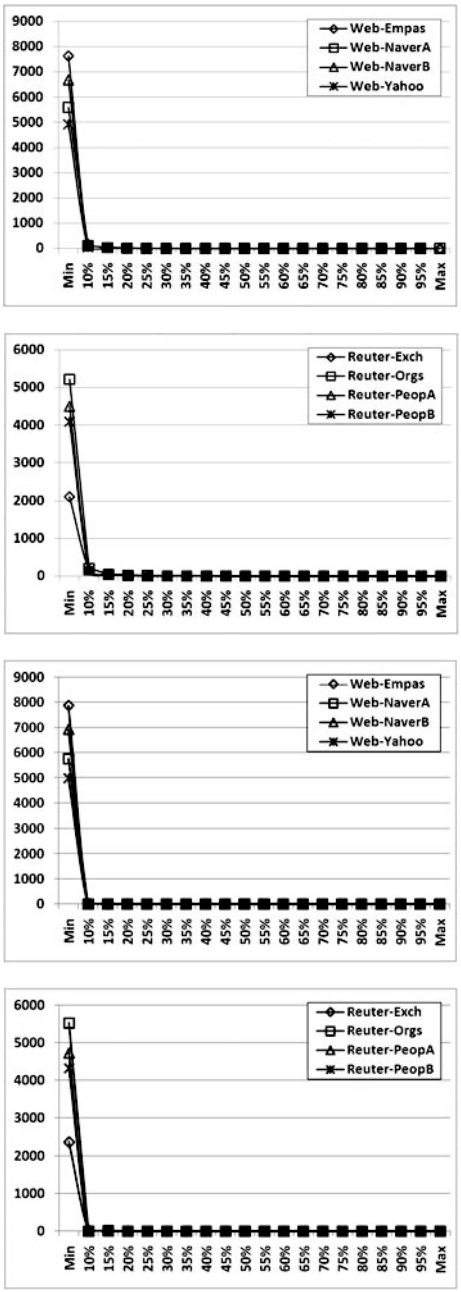

Fig. 1 Distribution of number of features for which Gini values are between minimum and maximum values, divided into 20 intervals, using *Gini-A* and *Gini-B*, respectively, for Reuters-21578 and Web datasets.

Table 1 Number of features in training datasets, numbers of features for which Gini values are below 1 % between minimum and maximum values, and their ratios to total number of training features, using *Gini-A* and *Gini-B*, for Reuters-21578 and Web datasets.

	Gini-A			Gini-B		
	Training Features	Features below 1%		Training Features	Features below 1%	
Empas	7,899	6,198	78.5%	7,899	7,817	99.0%
NaverA	5,774	4,475	77.5%	5,774	5,723	99.1%
NaverB	6,929	5,327	76.9%	6,929	6,890	99.4%
Yahoo	4,999	4,500	90.0%	4,999	4,979	99.6%
Exch	2,427	1,403	57.8%	2,427	2,221	91.5%
Orgs	5,547	4,089	73.7%	5,547	5,451	98.3%
PeopA	4,768	3,496	73.3%	4,768	4,651	97.5%
PeopB	5,547	4,089	73.7%	4,347	4,248	97.7%

among classes. If the high-frequency features are distributed among several classes, the Gini values can be higher than in the case of low-frequency features. The higher frequency features have higher Gini values than those of low frequency features overall. Thus, it is required to decrease the Gini-Index values for irrelevant general high-frequency features.

3) For the specific features belonging to large classes, the Gini values are relatively lower than those belonging to the small classes. Even though all the members of that feature are distributed to the same class, they are always influenced by the size of class. The Gini-Index expression presented by W. Shang et al. (2007) was considered for unbalanced class distribution and amended to expression (4). However, in feature selection for text classification, it has a bias, because if specific features are distributed to the same class, they must have the similar value irrespective of the size of classes to which features belong.

The problems of Gini-Index expressions *Gini-A* and *Gini-B* with regard to the bias of the Gini-Index are as follows.

- In expression (3) for *Gini-A*, the specific features are influenced by the feature probability $P(W)$. If the feature frequency in a class is high relative to other classes, $P(C_i|W)$ will be relatively high. However, because the total frequency of training dataset is high, most Gini values are influenced by the probability $P(W)$ irrespective of $P(C_i|W)^2$. Therefore, low-frequency features have lower Gini values than those of high-frequency features. On the other hand, high-frequency features have relatively high Gini values.
- In expression (4) for *Gini-B*, the expression was reformulated to consider the unbalanced class distribution by adopting $\sum P(W|C_i)^2$ in place of $P(W)$ [13]. In this formula, the Gini values are also influenced by $\sum P(W|C_i)^2$ and high-frequency features have higher Gini values than those of low-frequency features, irrespective of $P(C_i|W)^2$. Thus, high-frequency features have higher Gini values than those of low-frequency features and low-frequency features always have relatively low Gini values.

Table 2 Gini values for each feature and their frequencies for 'Exch' of Reuters-21578 dataset using *Gini-B*.

Features	Gini Value	Total Freq.	Frequency for each class						
			1	**2**	**3**	**4**	**5**	**6**	**7**
NAS*	**0.43448**	**40**	**0**	**0**	**0**	**0**	**0**	**40**	**0**
CBT*	**0.28877**	**55**	**0**	**0**	**50**	**5**	**0**	**0**	**0**
Trading	0.14855	297	28	7	52	83	40	22	65
Stock	0.14678	147	33	5	3	7	37	7	55
Contract	0.12680	107	5	3	41	50	2	0	6
AMEX*	**0.09099**	**11**	**10**	**0**	**0**	**0**	**0**	**0**	**1**
Petition*	**0.04913**	**19**	**0**	**0**	**0**	**18**	**0**	**0**	**1**
Technology	0.02236	8	6	0	0	0	0	2	0
Campaign*	**0.01761**	**4**	**4**	**0**	**0**	**0**	**0**	**0**	**0**
Hog*	**0.00422**	**5**	**0**	**0**	**0**	**5**	**0**	**0**	**0**
Statement	0.00219	19	3	0	1	0	2	6	7
Optimism	0.00110	1	1	0	0	0	0	0	0

- In expression (4), the $P(W|C_i)$ is presented as $P(W, C_i)/P(C_i)$ and $P(C_i)$ of large classes being larger than that of small classes. Thus, for specific features belonging to the large classes, the Gini values are relatively lower than those of the small classes.

Table 2 shows the Gini values of the twelve features (NAS, CBT, etc.) and their frequencies for 'Exch' dataset of Reuters-21578 using *Gini-B* expression (4), Shang's Gini-Index (2007). For each feature, it shows the total frequencies, the frequencies and Gini values for each class, ordered by the Gini values. In this Table, we can easily assess manually the goodness of features using term frequencies for each class. 'NAS', 'CBT', 'AMEX', 'Petition', 'Campaign' and 'Hog' are good features. On the other hand, 'Trading', 'Stock' and 'Statement' look to be irrelevant features. However, we can see that the Gini values of 'Stock' and 'Contract' (high frequency and distributed over several classes) are higher than those of 'AMEX' and 'Petition' (low frequency and distributed to one or two classes). Because the former features have a high total frequency, their Gini values are higher than those of the latter. Therefore, a reformulation of the Gini-Index expression was necessary to eliminate those irrelevant general terms and retain the specific terms.

3. Improved Gini-Index Algorithm for Feature Selection

As mentioned above, the Gini-Index for feature selection remains the bias with respect to unbalanced datasets. We have discussed the reasons and problems for the feature selection bias of the Gini-Index in text classification. Because most Gini values are more affected by $P(W)$ or $P(W|C_i)^2$ than by the distribution of feature frequencies among classes $P(C_i|W)$ in feature selection. Therefore, it is required to adjust the Gini-Index algorithm so that it can avoid the bias of Gini-Index and solve those problems. We introduce the new, Improved Gini-Index (I-GI) algorithm and three new, reformulated Gini-Index expressions to solve both the bias and the problems.

3.1 New Reformulated Gini-Index Expressions

We reformulated Shang's Gini-Index expression (2007) as three new Gini-Index expressions in order to solve the bias of Gini-Index. First, we amended expression (3) to expression (5), namely *IGini-A*:

$$IGini_A(W) = \sum_{i=1}^{m} P(C_i|W)^2 \tag{5}$$

In this expression, we eliminated the $P(W)$ from expression (3). The specific reasons for reformulating the *IGini-A* expression are as follows.

- Because most features in the datasets have low frequencies, the $P(W)$ are very small and the Gini values are more influenced by $P(W)$ than $P(C_i|W)^2$. For high-frequency features, the $P(W)$ is relatively much high, and thus the Gini values are influenced only by $P(W)$.
- Therefore, we eliminate the $P(W)$ from expression (3), and we calculate the Gini values using only $P(C_i|W)^2$. It is more efficient to estimate the representative features than to use expression (3), and this can solve the first two factors incurring the bias of the Gini-Index.

Second, we can normalize the Gini values by square of root, logarithm or other methods for the probability $P(W)$ or the posterior probability $P(W|C_i)$. We applied the logarithm base 2 of the probability $P(W)$ and its absolute value to reduce the range of the $P(W)$ and to keep it positive. We therefore reformulated expression (6), namely *IGini-B*:

$$IGini_B(W) = \left|\frac{1}{\log_2 P(W)}\right| \sum_{i=1}^{m} P(C_i|W)^2 \tag{6}$$

The specific reasons for reformulating this expression are as follows.

- We normalized the Gini values instead of eliminating $P(W)$ from expression (3), as in expression (6). We used $|1/\log_2 P(W)|$ instead of $P(W)$ from expression (3) to raise the $P(W)$ and the Gini value for low-frequency features. The Gini values are calculated using $P(C_i|W)^2$ and $|1/\log_2 P(W)|$. This lifts the Gini values for low-frequency features and enables an unbiased range of Gini values.
- Therefore, the new expression (6) can be more efficient at estimating specific features as well as general features, thereby solving the first two reasons for the biased Gini-Index.

Third, expression (4) is reformulated to yield expression (7), by normalizing the probability $P(W|C_i)$ with the logarithm base 2, which reduces the range of $P(W|C_i)$ and produces unbiased Gini values. This reformulation is expression (7), namely *IGini-C*:

$$IGini_C(W) = \left|\frac{1}{\log_2 P(W|C_i)^2}\right| \sum_{i=1}^{m} P(C_i|W)^2 \tag{7}$$

The specific reasons for reformulating the *IGini-C* expression are as follows.

- We normalized $P(W|C_i)^2$ by the logarithm base 2 from expression (4), calculating the Gini values by using $|1/\log_2 P(W|C_i)^2|$ and $P(C_i|W)^2$. This increases the variations of Gini values by $|1/\log_2 P(W|C_i)^2|$. The Gini values of low-frequency features are relatively more increased and those of high-frequency features.
- Additionally, for the specific features belonging to the same class, high Gini values can be obtained by using $|1/\log_2 P(W|C_i)^2|$ instead of $P(W|C_i)^2$, and are less influenced by the class sizes. Because the $P(W|C_i)$ is presented as $P(W, C_i)/P(C_i)$ and the Gini values are influenced by $P(C_i)$ for each size of class. Therefore, the variations of $P(W|C_i)$ among classes can be reduced.

We next calculated the Gini values using expressions (5), (6) and (7), estimated the representative feature subsets and applied the features to all of the datasets for classification.

3.2 Improved Gini-Index Algorithm for Feature Selection

In text classification, to improve performance, it is necessary to reduce a high dimensionality of the feature space using feature selection methods. In dimensionality reduction (DR), there are two distinct ways of viewing DR, depending on whether the task is performed locally (i.e., for each individual category) or globally. Local DR is that chooses feature sets of terms for each category for classification under a category. This means that different subsets of document sets are used when working with the different categories. Global DR is that chooses feature subsets for classification under all categories [4].

All functions of feature selection methods are specified "locally" to a specific category c_i ; in order to assess the value of a term t_k in a "global," category independent sense, either the sum $f_{sum}(t_k) = \sum_{i=1}^{|C|} f(t_k, c_i)$ or the maximum $f_{\max}(t_k) = \max_{i=1}^{|C|} f(t_k, c_i)$ of their category-specific values $f(t_k, c_i)$ usually are computed. According to feature selection methods, it was adopted the better of the two having the best performance generally [4]. Yang and Pedersen (1997) had shown that, with various classifiers and various initial corpora, sophisticated techniques, such as $IG_{sum}(t, c_i)$, $\chi^2_{\max}(t, c_i)$, OR_{sum}, $MI_{\max}$ and others can reduce the dimensionality of the term space. Collectively, the experiments reported that $\{OR_{sum}, NGL_{sum}, GSS_{\max}\} > \{\chi^2_{\max}, IG_{sum}\} > \{\chi^2_{wavg}\} > \{MI_{\max}, MIw_{sum}\}$, where ">" means "performs better than" for Odds Ratio (OR), Ng-Goh-Low coefficient (NGL), Galavotti-Sebastiani-Simi coefficient (GSS), χ^2, Information Gain (IG), and Mutual Information (MI) [4], [16].

Commonly used global goodness estimators are the maximum and average (or sum) functions. A well discriminated feature will have skewed distribution across the classes. However, these two functions do not capture how a feature is distributed over different classes [11].

In this study, we used two policies for global and lo-

cal DRs in feature selection. First, we adopted the sum of feature values for all classes in feature selection for global DR using Gini-Index expression (3)~(7), namely, $Gini_{sum}$: $Gini_{sum}(W) = \sum Gini(W, C_i)$. The process of feature selection using $Gini_{sum}$ for global DR is as follows.

- First, we calculate the Gini values using a Gini-Index expression among expressions (3)~(7) for all features. All of the features are ranked according to their Gini values.
- $F_{sum}(S_n)$ is the ordered features set and n is the expression number (3)~(7). We select nine representative feature subsets $F_{sum}(S_{nj})$ from $F_{sum}(S_n)$, j is the feature subset reduced by 10 %$*j$, for 10 %, 20 % and so on up to 90 % (the dimensionality of feature spaces was reduced by 10 %$*j$ for each subset from $F_{sum}(S_n)$).
- The feature subsets for all Gini-Index expressions (3)~(7) are selected recursively.

Second, we adopted the higher feature values among the classes instead of their maximum for local DR. Because if the maximum function is used for local DR, we can select only one representative class for each feature. Thus, the upper functions can obtain feature subsets for multiple classes and improve feature selection in text classification. We amended the following Gini Index expressions (8)~(12) for each class from expressions (3)~(7), namely, $Gini_{high}$ for local DR: $Gini_{high}(W) = Upper\{Gini(W, C_i)\}$.

$$Gini_A(W, C_i) = P(W)P(C_i|W)^2 \tag{8}$$

$$Gini_B(W, C_i) = P(W|C_i)^2P(C_i|W)^2 \tag{9}$$

$$IGini_A(W, C_i) = P(C_i|W)^2 \tag{10}$$

$$IGini_B(W, C_i) = \left|\frac{1}{\log_2 P(W)}\right| P(C_i|W)^2 \tag{11}$$

$$IGini_C(W, C_i) = \left|\frac{1}{\log_2 P(W|C_i)^2}\right| P(C_i|W)^2 \tag{12}$$

The process of feature selection using $Gini_{high}$ for local DR is as follows.

- First, we calculate the Gini values for each class for all features, using a Gini-Index expression among expressions (8)~(12). All features with their Gini values for each class, irrespective of classes, are ranked.
- Nine representative feature subsets $F_{high}(S_{mj}, C_i)$ of class C_i are selected from the ordered features $F_{high}(S_m)$, j is the feature subset reduced by 10 %$*j$, for 10 %, 20 % and so on up to 90 %, $F_{high}(S_m)$ is the ordered features set, m are the expression numbers (8)~(12) and C_i is a class i. The feature subsets for all of the Gini-Index expression (8)~(12) are selected recursively.
- We can obtain the feature subsets for each class by $Gini_{high}$ locally. For each class, independent feature subsets can be obtained from the ordered features, according or the Gini values. Thus, all features can belong to multi-classes.

We used both policies for feature selection: $Gini_{sum}$ for global DR using expressions (3)~(7), and $Gini_{high}$ for local DR using expressions (8)~(12). We obtained ninety feature subsets (9*10, including the original feature subset) by reducing the dimensionality of feature spaces by 10 % from the original feature subset, for all Gini-Index expressions (3)~(7) of $Gini_{sum}$ and (8)~(12) of $Gini_{high}$. Then, we applied all of those feature subsets to all of the datasets. Thus, the I-GI algorithm is as follows:

Input: *vector spaces of training datasets with term frequency and class labels*
Output: *vector spaces of all datasets, Gini values and feature subsets for each Gini expression*
for *all Gini expressions* (3)~(12)
  for *each feature W* do begin
    Calculate *Gini*(W) *using expression* (3)~(7) for $Gini_{sum}$ *DR*
    for i=1 to k do
      Calculate *Gini*(W, C_i) *using expression* (8)~(12) for $Gini_{high}$ *DR*
  end
  Obtain *ordered feature sets* $F_{sum}(S_n)$ *by Gini*(W)
  Obtain *ordered feature sets* $F_{high}(S_m)$ *by Gini*(W, C_i)
      $F_{sum}(S_n)$, $F_{high}(S_m) \in F(S)$
  for j=1 to 9 do begin
    for all features
     Select *feature subsets* $F_{sum}(S_{n.j})$ *upper* j*10 % *from* $F_{sum}(S_n)$
     Select *feature subsets* $F_{high}(S_{m.j}, C_i)$ *upper* j*10 % *from* $F_{high}(S_m)$, $F_{sum}(S_{n.j}) \in F_{sum}(S_n)$, $F_{high}(S_{m.j}) \in F_{high}(S_m)$
    end
  end
end
for *all feature subsets using Gini expressions* (3)~(12)
  for j=1 to 9 do begin
    Apply *features of* $F_{sum}(S_{nj})$ and $F_{high}(S_{mj}, C_i)$ *to vector spaces of all of datasets*
end

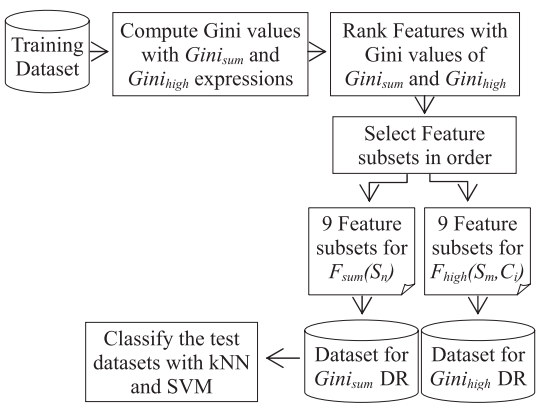

Fig. 2 Process of I-GI algorithm for feature selection using $Gini_{\text{sum}}$ for global DR and $Gini_{\text{high}}$ for local DR with new Gini-Index expression.

The process of the improved Gini-Index (I-GI) algorithm for feature selection using Ginisum for global DR and Ginihigh for local DR with the new Gini-Index expression is as shown in Fig. 2.

4. Experiments and Evaluations

4.1 Experimental Document Sets

We used two types of datasets, namely Reuters-21578 and Web document sets. Reuters-21578 contains Exchanges, Orgs, and People categories, with each category having its specified subcategories. We used the Exchanges, Orgs, and People categories for the experiments. We selected seven subcategories from the Exchanges category, each having more than ten documents (*Exch*). From the Orgs category, we selected eight subcategories, each having more than twenty documents (*Orgs*). From the People category, we selected sixteen (and ten) subcategories, each having more than ten (and twenty) documents (*PeopA* and *PeopB*, respectively), as shown in Table 3.

Web document sets were extracted at the *'Natural Science'* directory from *http://www.empas.com*, *http://www.yahoo.co.kr*, and *http://www.naver.com*, three well-known Korean portal sites, and well classified manually by an indexer. In this way we could easily evaluate the performance by comparing the pre-allocated directory with the results. We selected nine subdirectories from among the directory services from those sites, which were *Empas*, *Yahoo*, *NaverA*, and *NaverB*, and we extracted two types of document set at *http://www.naver.com*. The numbers of documents, the numbers of features and the numbers of classes for each dataset were as listed in Table 3.

4.2 Experiments and Evaluations

First, we selected the feature subsets using the new I-GI algorithm according to the global DR and local DR policies. In addition, to compare the performances of new I-GI algorithm with the existing feature selection methods, we tested χ^2, IG, OR and W. Shang's Gini Index (2007) by global and local DR. The χ^2, IG, OR expressions are as follows [16], [17].

Table 3 Numbers of documents, numbers of features, and numbers of classes for each document in Reuters-21578 and Web document sets.

Document Sets		Number of Documents	Number of Features	Number of Classes
Reuters-21578 Datasets	Exch	284	3,131	7
	Orgs	775	6,644	8
	PeopA	704	6,062	16
	PeopB	569	5,602	10
Web Datasets	Empas	1,036	7,969	9
	Yahoo	964	5,063	9
	NaverA	667	5,846	9
	NaverB	1,069	7,004	9

$$\chi^2(t, c_i) = \frac{N[P(t, c_i)P(\bar{t}, \bar{c}_i) - P(t, \bar{c}_i)P(\bar{t}, c_i)]^2}{P(t)P(\bar{t})P(c_i)P(\bar{c}_i)} \quad (13)$$

$$\begin{aligned} IG(t, c_i) = & -P(c_i)\log P(c_i) \\ & +P(t)P(c_i|t)\log P(c_i|t) \\ & +P(\bar{t})P(c_i|\bar{t})\log \Pr(c_i|\bar{t}) \end{aligned} \quad (14)$$

$$OR(t, c_i) = \log \frac{P(t|c_i)(1 - P(t|\bar{c}_i))}{(1 - P(t|c_i))P(t|\bar{c}_i)} \quad (15)$$

The $P(t, c_i)$ is the probability of feature t in a class c_i, the probability of features except t in the same class $P(\bar{t}, c_i)$, the probability of feature t in different classes $P(t, \bar{c}_i)$, the probability of features except t in different classes $P(\bar{t}, \bar{c}_i)$ and N is the total number of features.

Second, we applied the feature subsets to all of the datasets for each feature selection method by $Gini_{sum}$ for global DR and by $Gini_{high}$ for local DR policies. Third, we classified documents with the classification algorithms, kNN and SVM. The kNN classifier has been widely used and offers good performance in various data classification areas. In the classification performance evaluations, we employed the F1 measure using Recall and Precision, where F1=(2*Recall*Precision)/(Recall+Precision). We compared the micro-F1 and macro-F1 classification performance for each feature selection and classifier according to the datasets. For the purposes of the experiments, we developed the kNN classification tool. For the SVM classifications, we used the Multi-Class Support Vector Machine of SVMlight from Cornell University and the University of Dortmund [2], [9]. We tested the classifications using the 'linear' kernel function for SVMlight, k being the number of classes for each dataset for kNN. We used 10-fold cross-validation for all of the classifications.

4.3 Experimental Results

First, we compared the distributions of the number of features for which the Gini values are between the minimum and maximum values, using three new expressions *IGini-A*, *IGini-B* and *IGini-C*, respectively, for Reuters-21578 and Web datasets, as shown in Fig. 3. The X-axis marks the intervals of the percentages of the Gini values between the minimum and the maximum values divided into 20, and the Y-axis shows the feature counts belonging to each interval. The first two figures show the distributions of the Gini values using *IGini-A*, and they are concentrated toward the maximum value. The next four figures show the distributions of the Gini values using *IGini-B* and *IGini-C*, respectively, and those distributions are unbiased and not concentrated: the values were distributed and concentrated toward the median value overall. Therefore, when we normalized the Gini values using *IGini-B* and *IGini-C*, the Gini values were balanced, compared with the results for *Gini-B* in Fig. 1.

Table 4 lists the Gini values and their ranks ordered using *Gini-A*, *Gini-B*, *IGini-A*, *IGini-B* and *I-Gini-C* with $Gini_{sum}$ for global DR, for the 'Exch' training dataset of Reuters-21578. We assessed that 'NAS', 'CBT', 'AMEX'

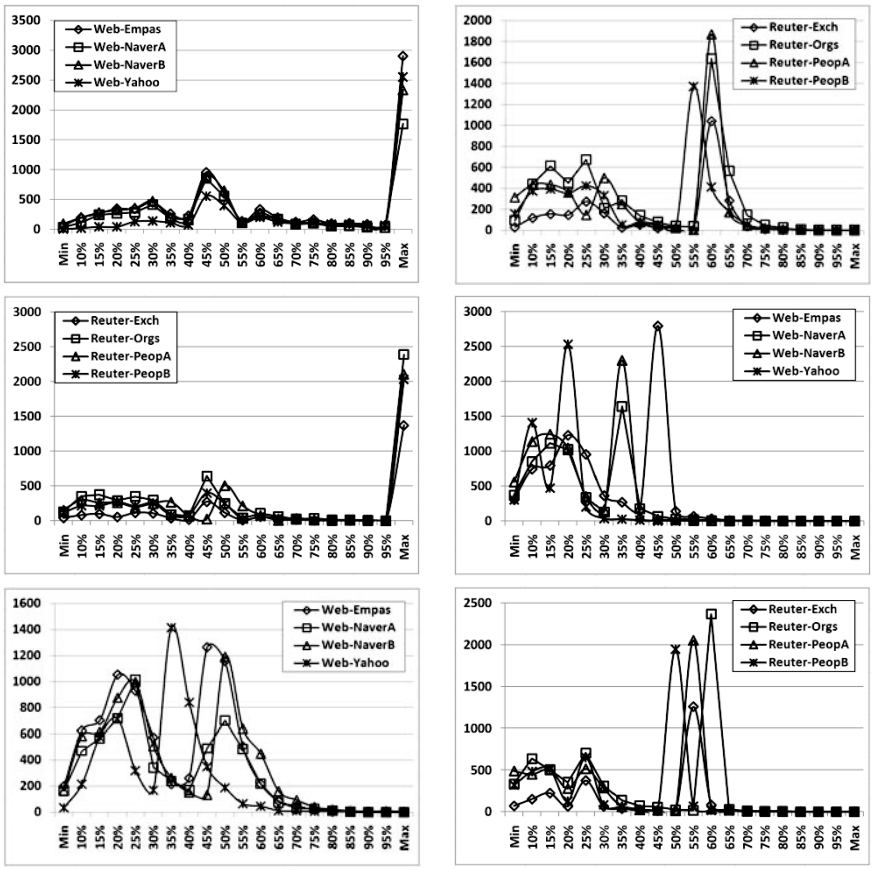

Fig. 3 Distribution of number of features for which Gini values are between minimum and maximum values, using three new *IGini-A*, *IGinit-B* and *IGini-C*, respectively, for Reuters-21578 and Web datasets.

Table 4 Gini values and their ranks (1∼12) of features using *Gini-A*, *Gini-B*, *IGini-A*, *IGini-B* and *I-Gini-C*, respectively for 'Exch' training dataset of Reuters-21578.

Features	Values and Ranks for each Gini-Index									
	Gini-A		Gini-B		IGini-A		IGini-B		IGini-C	
NAS	**0.00305**	**3**	**0.4345**	**1**	**1**	**1**	**0.00305**	**5**	**0.0895**	**1**
CBT	**0.00349**	**2**	**0.2888**	**2**	0.8347	6	**0.00419**	**4**	**0.0724**	**2**
Trading	**0.00429**	**1**	**0.1486**	**3**	0.1897	12	**0.02263**	**1**	0.0182	11
Stock	**0.0029**	**5**	**0.1468**	**4**	0.2598	11	**0.01120**	**2**	0.0237	10
Contract	**0.0030**	**4**	**0.1268**	**5**	0.3717	9	**0.00815**	**3**	0.0321	9
AMEX	0.0007	7	0.0910	6	0.8347	6	0.00084	8	**0.0633**	**5**
Petition	0.0013	6	0.0005	12	**0.90037**	**5**	0.00145	6	**0.0635**	**3**
Technology	0.00038	9	0.0224	7	0.6250	8	0.00061	9	0.0416	8
Campaign	0.00031	11	0.0176	8	**1**	**1**	0.00031	11	**0.0633**	**4**
Hog	0.00038	9	0.0042	9	**1**	**1**	0.00038	10	0.0560	**6**
Statement	0.00039	8	0.0022	10	0.2742	10	0.00145	6	0.0161	12
Optimism	0.000076	12	0.0011	11	**1**	**1**	0.000076	12	0.0505	7

'Campaign', 'Hog' and 'Petition' are good representative features and that 'Trading', 'Stock' and 'Statement' are irrelevant features in Sect. 2. In the cases of *Gini-A*, *Gini-B* and *I-Gini-B*, the high-frequency features have high Gini values, *IGini-A* showing that the features distributed to the same class have the highest Gini values. In the case of *IGini-C*, there are high Gini values for the features distributed to the one or two classes irrespective of feature frequency, which results are similar to those of the manual assessment in Sect. 2. Good representative features, in both general and specific cases, could be selected using *IGini-C*.

Second, we compared the classification performances of Micro-F1 and Macro-F1 using the kNN and SVM classifiers according to *tf*idf*, *Gini-A*, *Gini-B* and the new *I-GI* algorithm with *IGini-A*, *IGini-B* and *IGini-C* expressions. *tf*idf* indicates the performance using *tf*idf*, *Gini-A* and *Gini-B* are the results for the existing Gini-Index, and *IGini-A*, *IGini-B* and *IGini-C* are the results for the new I-GI algorithm with three new expressions. All of the results are presented the best performances for each feature selection algorithm.

Table 5 shows the classification performances of Micro-F1 and Macro-F1 with the kNN classifiers using the I-GI algorithm, the existing Gini-Index and *tf*idf*, by $Gini_{sum}$ for global DR. The bolded results highlight the best performances among the methods. However, there are no differences among the results overall, and the results for *tf*idf* are better than those for the others excepting some specific cases. There are no the notable methods using $Gini_{sum}$ for global DR.

Table 6 shows the classification performances of Micro-F1 and Macro-F1 with SVM classifiers according to each method by $Gini_{sum}$ for global DR, and the bolded results, again, highlight the best performances among the

Table 5 Micro-F1 and Macro-F1 with kNN classifiers, using *tf*idf*, *Gini-A*, *Gini-B*, *IGini-A*, *IGini-B* and *IGini-C* by $Gini_{sum}$ for global DR.

Micro-F1	tf*idf	Gini-A	Gini-B	**IGini-A**	**IGini-B**	**IGini-C**
Empas	**0.892**	0.877	0.877	0.877	0.875	0.873
NaverA	0.795	0.793	0.787	**0.805**	0.789	0.793
NaverB	0.803	0.791	0.791	**0.816**	0.814	0.813
Yahoo	**0.906**	0.895	0.894	0.893	0.891	0.895
Exch	0.690	0.724	0.736	0.793	0.793	**0.828**
Orgs	0.814	0.814	**0.819**	0.814	0.779	0.779
PeopA	0.621	0.605	0.617	0.710	0.718	**0.796**
PeopB	0.723	0.717	0.733	0.801	**0.806**	0.796
Macro-F1	tf*idf	Gini-A	Gini-B	**IGini-A**	**IGini-B**	**IGini-C**
Empas	**0.866**	0.847	0.848	0.844	0.846	0.846
NaverA	0.769	0.769	0.763	**0.782**	**0.782**	0.780
NaverB	0.750	0.726	0.724	0.771	**0.774**	0.770
Yahoo	**0.867**	0.850	0.854	0.850	0.850	0.770
Exch	0.610	0.579	0.584	**0.685**	0.675	0.675
Orgs	0.796	0.793	0.797	0.725	**0.803**	0.760
PeopA	0.674	0.640	0.642	0.770	0.770	**0.814**
PeopB	0.779	0.766	0.773	0.828	**0.828**	0.814

Table 6 Micro-F1 and Macro-F1 with SVM classifiers, using *tf*idf*, *Gini-A*, *Gini-B*, *IGini-A*, *IGini-B* and *IGini-C* by $Gini_{sum}$ for global DR.

Micro-F1	tf*idf	Gini-A	Gini-B	**IGini-A**	**IGini-B**	**IGini-C**
Empas	**0.871**	0.830	0.830	0.829	0.827	0.831
NaverA	**0.805**	0.794	0.790	0.797	0.800	0.797
NaverB	**0.810**	0.779	0.782	0.799	0.792	0.794
Yahoo	**0.910**	0.864	0.870	0.878	0.870	0.878
Exch	0.752	0.749	0.756	0.791	**0.813**	0.809
Orgs	**0.822**	0.789	0.784	0.765	0.774	0.766
PeopA	0.757	0.762	0.764	0.780	**0.792**	0.789
PeopB	0.840	0.835	0.845	0.848	0.847	**0.854**
Macro-F1	tf*idf	Gini-A	Gini-B	**IGini-A**	**IGini-B**	**IGini-C**
Empas	**0.839**	0.796	0.796	0.791	0.796	0.794
NaverA	**0.785**	0.768	0.770	0.774	0.774	0.772
NaverB	**0.759**	0.706	0.708	0.749	0.726	0.721
Yahoo	**0.880**	0.801	0.806	0.822	0.809	0.823
Exch	0.680	0.634	0.646	0.646	0.721	**0.727**
Orgs	0.780	0.725	0.725	0.669	**0.728**	0.705
PeopA	0.724	0.717	0.716	0.789	0.735	**0.757**
PeopB	0.851	0.847	0.848	0.851	0.856	**0.866**

methods. The results for *tf*idf* are better than those for others overall excepting some specific cases. Neither the existing Gini-Index nor the new I-GI algorithm showed good performances compared with *tf*idf*. There are no notable methods using $Gini_{sum}$ for global DR.

Table 7 and 8 show the Micro-F1 and Macro-F1 classification performances with the kNN and SVM classifiers according to each method by $Gini_{high}$ for local DR. The bolded results are highlight the best performances among the methods. Using the $Gini_{high}$ for local DR in feature selection, the performances of the I-GI algorithm are better than those of the others overall. Notably improved performances were obtained when $Gini_{high}$ for the local DR of feature selection was applied using the I-GI algorithm. The performances of *Gini-A* and *Gini-B* also were good overall. In addition, performances of *Gini-A* and *Gini-B* also were good, in some cases. Therefore, when we applied the $Gini_{high}$ for local DR in feature selection, using the I-GI algorithm with the three new expressions *IGini-A*, *IGini-B* and *IGini-C*, the best performances were shown.

Table 7 Micro-F1 and Macro-F1 with kNN classifiers, using *tf*idf*, *Gini-A*, *Gini-B*, *IGini-A*, *IGini-B* and *IGini-C* by $Gini_{high}$ for local DR.

Micro-F1	tf*idf	Gini-A	Gini-B	**IGini-A**	**IGini-B**	**IGini-C**
Empas	0.892	0.938	0.930	**0.962**	**0.960**	**0.961**
NaverA	0.795	0.914	0.925	**0.965**	**0.967**	**0.967**
NaverB	0.803	0.865	0.859	**0.910**	**0.911**	**0.912**
Yahoo	0.906	0.937	0.930	**0.969**	**0.970**	**0.970**
Exch	0.690	**0.998**	**0.977**	**0.977**	**0.977**	**0.989**
Orgs	0.814	**0.947**	**0.956**	**0.996**	**0.996**	**0.996**
PeopA	0.621	**0.996**	**0.984**	**0.998**	**0.996**	**0.998**
PeopB	0.723	**0.995**	**0.998**	**0.998**	**0.998**	**0.998**
Macro-F1	tf*idf	Gini-A	Gini-B	**IGini-A**	**IGini-B**	**IGini-C**
Empas	0.866	0.929	0.924	**0.953**	**0.951**	**0.952**
NaverA	0.769	0.902	0.916	**0.958**	**0.961**	**0.960**
NaverB	0.750	0.854	0.832	**0.897**	**0.902**	**0.902**
Yahoo	0.867	0.911	0.902	**0.951**	**0.951**	**0.951**
Exch	0.610	**0.998**	**0.970**	0.929	0.929	**0.964**
Orgs	0.796	**0.957**	**0.949**	**0.997**	**0.997**	**0.988**
PeopA	0.674	**0.998**	**0.970**	**0.998**	**0.998**	**0.998**
PeopB	0.779	**0.990**	**0.998**	**0.998**	**0.998**	**0.998**

Table 8 Micro-F1 and Macro-F1 with SVM classifiers, using *tf*idf*, *Gini-A*, *Gini-B*, *IGini-A*, *IGini-B* and *IGini-C* by $Gini_{high}$ for local DR.

Micro-F1	tf*idf	Gini-A	Gini-B	**IGini-A**	**IGini-B**	**IGini-C**
Empas	0.871	0.918	0.905	**0.946**	**0.946**	**0.947**
NaverA	0.805	0.904	0.910	**0.955**	**0.956**	**0.956**
NaverB	0.810	0.851	0.843	**0.900**	**0.904**	**0.897**
Yahoo	0.910	0.933	0.922	**0.958**	**0.958**	**0.959**
Exch	0.752	**0.958**	**0.947**	**0.972**	**0.972**	**0.972**
Orgs	0.822	0.898	0.882	**0.990**	**0.986**	**0.990**
PeopA	0.757	**0.975**	**0.957**	**0.979**	**0.979**	**0.979**
PeopB	0.840	**0.981**	**0.975**	**0.984**	**0.984**	**0.984**
Macro-F1	tf*idf	Gini-A	Gini-B	**IGini-A**	**IGini-B**	**IGini-C**
Empas	0.839	0.903	0.892	**0.945**	**0.942**	**0.943**
NaverA	0.785	0.900	0.905	**0.955**	**0.957**	**0.958**
NaverB	0.759	0.788	0.780	**0.844**	**0.850**	**0.864**
Yahoo	0.880	0.913	0.884	**0.943**	**0.948**	**0.945**
Exch	0.680	0.930	0.903	**0.958**	**0.958**	**0.958**
Orgs	0.780	0.880	0.854	**0.985**	**0.973**	**0.985**
PeopA	0.724	**0.959**	0.910	**0.953**	**0.965**	**0.968**
PeopB	0.851	**0.974**	**0.969**	**0.978**	**0.978**	**0.978**

Third, we compared the classification performances using *tf*idf*, χ^2, IG, and OR according to the kNN and SVM classifiers. Table 9 shows the classification performances of Micro-F1 and Macro-F1 for *tf*idf*, χ^2, IG, and OR. The results are the best performances for each feature selection method. Note that the IG and OR algorithms offered better performances than *tf*idf* and χ^2 overall. In Tables 7, 8 and 9, the bold-face scores mean the notably improved performances compared with those of the *tf*idf* or the others for each dataset.

In addition, we compared the averages of the Micro-F1 and Macro-F1 classification performances with the kNN and SVM classifiers, using *tf*idf*, *Gini-A*, *Gini-B*, *IGini-A*, *IGini-B* and *IGini-C*, as shown in Table 10. When we used the I-GI algorithm, we improved the classification performance by 19.2~19.4 % (kNN) and 14 % (SVM) in Micro-F1, and 19.6~19.8 % (kNN) and 15.8~16.3 % (SVM) in Macro-F1, compared with those of *tf*idf*. And the performances were enhanced by 19.4 %, 15.9 %, 3.3 %, 2.8 %

Table 9 Micro-F1 and Macro-F1 with kNN and SVM classifiers, using *tf*idf*, χ^2, IG, and OR.

kNN	Micro-F1				Macro-F1			
	tf*idf	χ^2	IG	OR	tf*idf	χ^2	IG	OR
Empas	0.892	0.887	**0.942**	**0.926**	0.866	0.859	**0.931**	**0.903**
NaverA	0.795	0.843	**0.934**	**0.900**	0.769	0.822	**0.922**	0.875
NaverB	0.803	0.800	**0.886**	**0.877**	0.750	0.749	**0.877**	0.819
Yahoo	0.906	0.905	**0.950**	**0.939**	0.867	0.871	**0.932**	**0.905**
Exch	0.690	0.828	**0.908**	**0.989**	0.610	0.689	0.866	**0.994**
Orgs	0.814	0.850	**0.978**	**0.965**	0.796	0.846	**0.989**	**0.964**
PeopA	0.621	0.649	**0.980**	**0.984**	0.674	0.691	**0.993**	**0.970**
PeopB	0.723	0.759	**0.953**	**0.990**	0.779	0.829	**0.979**	**0.991**
Average	0.781	0.815	**0.941**	**0.946**	0.764	0.795	**0.936**	**0.928**
SVM	**Micro-F1**				**Macro-F1**			
	tf*idf	χ^2	IG	OR	tf*idf	χ^2	IG	OR
Empas	0.871	0.854	**0.923**	**0.922**	0.839	0.827	**0.915**	**0.909**
NaverA	0.805	0.842	**0.911**	**0.920**	0.785	0.826	**0.917**	**0.907**
NaverB	0.810	0.836	**0.847**	**0.877**	0.759	0.753	0.810	0.805
Yahoo	0.910	0.859	**0.944**	**0.924**	0.880	0.799	**0.930**	**0.880**
Exch	0.752	**0.877**	0.799	0.816	0.680	0.759	0.824	0.818
Orgs	0.822	0.863	**0.921**	**0.991**	0.780	0.815	**0.914**	**0.986**
PeopA	0.757	0.862	0.750	**0.977**	0.724	0.800	0.825	**0.967**
PeopB	0.840	**0.907**	0.854	**0.984**	0.851	0.902	0.893	**0.978**
Average	0.821	**0.863**	**0.869**	**0.926**	0.787	0.810	**0.879**	**0.906**

Table 10 Averages of Micro-F1 and Macro-F1 with kNN and SVM classifiers, using *tf*idf* and Gini-Index algorithm; *Gini-A*, *Gini-B*, *IGini-A*, *IGini-B* and *IGini-C*, with $Gini_{high}$ for local DR.

$Gini_{high}$	Micro-F1		Macro-F1	
	kNN	SVM	kNN	SVM
tf*idf	0.780	0.821	0.764	0.787
χ^2	0.815	0.863	0.795	0.810
IG	0.941	0.869	0.936	0.879
OR	0.946	0.926	0.928	0.906
Gini-A	0.949	0.927	0.943	0.906
Gini-B	0.945	0.918	0.933	0.887
IGini-A	**0.972**	**0.961**	**0.960**	**0.945**
IGini-B	**0.972**	**0.961**	**0.961**	**0.946**
IGini-C	**0.974**	**0.961**	**0.964**	**0.950**

and 2.9 % (kNN) in Micro-F1, 14 %, 9.8 %, 9.2 %, 3.5 % and 4.3 % (SVM) in Micro-F1, 20 %, 16.9 %, 2.8 %, 3.6 % and 3.1 % (kNN) in Macro-F1, 16.3 %, 14 %, 7.1 %, 4.4 %, 6.3 % (SVM) in Macro-F1, compared with *tf*idf*, χ^2, *IG*, *OR* and the existing Gini-Index, respectively, according to each classifier.

Therefore, we could see good performance improvements using the I-GI algorithm with the three new Gini-Index expressions. In addition, we could obtain balanced Gini values using the I-GI algorithm with the three new Gini-Index expressions and we could solve the bias of the Gini-Index for the specific low-frequency features, for the general irrelevant high frequency features, and for the features of large classes, obtaining, thereby, good relevant and representative features.

5. Conclusions

The Gini-Index algorithm for feature selection is still biased with unbalanced datasets in text classification using the existing Gini-Index algorithm. For specific low-frequency features and the irrelevant general high-frequency features, the Gini values are biased. In addition, specific features belonging to large classes, those Gini values are relatively lower than those belonging to small classes. Therefore, we here propose the I-GI algorithm with three new, reformulated Gini-Index expressions to remove the bias of the Gini-Index and solve those problems in feature selection. According to the experimental results, when we used the I-GI algorithm with *IGini-C*, we could obtain unbiased Gini values, retaining specific representative features and eliminating many irrelevant general features from the feature subset. In addition, when we adopted the local DR in feature selection and the I-GI algorithm with *IGini-A*, *IGini-B* and *IGini-C*, we could improve the overall classification performances. Moreover, the performances were better than those of the χ^2, IG, and OR methods. In future work, to effect further improvements in performances, we will find and resolve other bias of the Gini-Index for feature selection in text classification, and will compare the results with those of various other studies.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (2010-0028784).

References

[1] B. Chandra and P.P. Varghese, "Fuzzifying Gini-Index based decision trees," Expert Systems with Applications, vol.36, no.4, pp.8549–8559, 2009.
[2] T. Joachims, Multi-Class Support Vector Machine, Department of Computer Science, Cornell University, http://svmlight.joachims.org
[3] C. Strobl, A.-L. Boulesteix, and T. Augustin, "Unbiased split selection for classification trees based on the Gini-Index," Computational Statistics & Data Analysis, vol.52, pp.483–501, 2007.
[4] F. Sebastiani, "Machine learning in automated text categorization," ACM Computing Surveys, vol.34, no.1, pp.1–47, 2002.
[5] H. Ogura, H. Amano, and M. Kondo, "Feature selection with a measure of deviations from Poisson in text categorization," Expert Systems with Applications, vol.36, no.3, Part 2, pp.6826–6832, 2008.
[6] L.E. Raileanu and K. Stoffel, "Theoretical comparison between the Gini-Index and information gain criteria," Annals of Mathematics and Artificial Intelligence, vol.41, pp.77–93, 2004.
[7] M. Sandri and P. Zuccolotto, "Bias correction algorithm for the Gini variable importance measure in classification trees," J. Computational and Graphical Statistics, vol.17, no.3, pp.611–628, 2008.
[8] M.A. Muharram and G.D. Smith, "Evolutionary feature construction using information gain and Gini-Index," Lect. Notes Comput. Sci., vol.3003, pp.379–388, 2004.
[9] P. Soucy and G.W. Mineau, "A simple KNN algorithm for text categorization," Proc. 2001 IEEE International Conference on Data Mining, pp.647–648, 2001.
[10] P.-N. Tan, M. Steinbach, and V. Kumar, Introduction to Data Mining, Addison-Wesley, 2006.
[11] S.R. Singh, H.A. Murthy, and T.A. Gonsalves, "Feature selection for text classification based on Gini coefficient of inequality," Fourth Workshop on Feature Selection in Data Mining, pp.76–85, 2010.
[12] W. Shang, Y. Qu, and H. Zhu, "An adaptive fuzzy kNN text classifier based on Gini-Index weight," 11th IEEE Symposium on ISCC apos;06. Proceedings, vol.26-29, pp.448–453, 2006.

[13] W. Shang, H. Huang, and H. Zhu, "A Novel feature selection algorithm for text categorization," Expert System with Application, vol.33, pp.1–5, 2007.
[14] W. Shang, H. Dong, and H. Zhuo, "Intelligent decision-making system based on data mining," International Joint Conference on Computational Sciences and Optimization (CSO 2009), pp.360–364, 2009.
[15] X.-H. Zhang, J. Wu, T.-J. Lu, and Y. Jiang, "A discretization algorithm based on Gini criterion," Proc. Sixth International Conference on Machine Learning and Cybernetics, pp.2557–2561, 2007.
[16] Y. Yang and J.P. Pedersen, "A comparative study on feature selection in text categorization," Proc. International Conference on Machine Learning, pp.412–420, 1997.
[17] Z.-H. Deng, S.-W. Tang, D.-Q. Yang, M. Zhang, X.-B. Wu, and M. Yang, "Two odds-radio-based text classification algorithms," Proc. Web Information Systems Engineering(Workshops), pp.23–231, 2002.

Heum Park received the M.S. in Interdisciplinary Program in Cognitive Science and Ph.D. degrees in Information System Engineering from Pusan National University, Busan, Korea, in 1998 and 2008, respectively. He has been a technical adviser in the Ubitec, Inc, since 2006. He has been a researcher of Center for U-Port IT Research and Education at Pusan National University, Busan, Korea, since 2003. His research interests include information retrieval, machine learning, ontology, and its application. Dr. Park is a member of the ACM. Dr. Park is a researcher of Center for U-Port IT Research and Education in Pusan National University.

Hyuk-Chul Kwon received the M.S. and Ph.D. degrees in computer engineering from Seoul National University, Seoul, Korea, in 1984 and 1987, respectively. He has been a professor at Pusan National University, Busan, Korea, since 1988. He was a Visiting Professor and Researcher at CSLI, Stanford University, Stanford, CA, from 1992 to 1993. He served as a Consultant at the Xerox Palo Alto Research Center, Palo Alto, CA, in 1993. He is the Head of the School of Electrical and Computer Engineering and the Director of the Specialized Group of Industrial Automation, Information, and Communication, Pusan National University. His research interests include natural language processing, information retrieval, machine learning, ontology, and its application.