On Approaches to Discretization of Datasets Used for Evaluation of Decision Systems

Grzegorz Baron and Katarzyna Haręźlak

Abstract The paper describes research on ways of datasets discretization, when test datasets are used for evaluation of a classifier. Three different approaches of processing for training and test datasets are presented: "independent"—where discretization is performed separately for both sets assuming that the same algorithm parameters are used; "glued"—where both sets are concatenated, discretized, and resulting set is separated to obtain training and test sets, and finally "test on learn"—where test dataset is discretized using ranges obtained from learning data. All methods have been investigated and tested in authorship attribution domain using Naive Bayes classifier.

Keywords Discretization · Decision system · Classification · Naive Bayes classifier · Authorship attribution

1 Introduction

In the area of text analysis and processing very often research focuses on input data preparation methods to improve classification results. A scoring function can be used to evaluate quality of features affecting classification performance [10], a feature scaling method using Naive Bayes classifier can be applied [13], or a feature weighting method and text normalization can be attempted [5]. The paper addresses the issue of the influence of discretization methods applied to datasets used in evaluation of decision systems.

Considering the nature of numerical data, theoretically it can be infinitely dense. In many cases reduction of data density is beneficial or even necessary, and it can be obtained by discretization. Mainly it allows to convert continuous form of data into

G. Baron · K. Haręźlak (✉)
Silesian University of Technology, Akademicka 16, 44-100 Gliwice, Poland
e-mail: katarzyna.harezlak@polsl.pl

G. Baron
e-mail: grzegorz.baron@polsl.pl

I. Czarnowski et al. (eds.), *Intelligent Decision Technologies 2016*,
Smart Innovation, Systems and Technologies 57,
DOI 10.1007/978-3-319-39627-9_14

discrete domain, but it can also change the volume of data, delivering the smaller number of continuous values. The former approach is employed when a classifier chosen for further analysis cannot operate on continuous numbers. In other cases such data preprocessing is facultative and can be analyzed in respect to possible benefits, for example improvement of classification performance [1], or more succinct way of expressing knowledge learned from input data.

When evaluation of a classifier performance is executed by using test datasets, the question arises how these sets should be discretized in relation to learning datasets. Three approaches can be employed. The first one relies on independent processing of learning and test datasets. In the second approach test data values are assigned to the bins based on the bins boundaries calculated during the discretization of learning datasets. And thirdly, data from training and test sets can be concatenated together, discretization process performed for such set, and then the resulting dataset splitted back to obtain learning and test sets.

The paper focuses on an analysis of different discretization methods in conjunction with the way of input sets discretization, taking into consideration some most popular discretization algorithms. To determine the influence of data discretization approach on classification quality, the Naive Bayes classifier has been chosen. It is a simple but very useful tool used in various domains, including text analysis. The presented experiments were conducted in an attempt to answer if it is possible to formulate any rules supporting the process of choosing the most suitable discretization method for a specific task.

The task considered as the application domain for described algorithms and procedures is authorship attribution from stylometric analysis of text. It deals with recognition of authorship based on style, in order to determine an author of some anonymous or disputed text, detect plagiarism etc. Statistics or machine learning techniques are mainly used for performing such tasks [7, 11].

The paper is organized as follows. Section 2 presents the theoretical background and methods employed in the research. Section 3 introduces the experimental setup, datasets used and techniques employed. The test results and their discussion are given in Sect. 4, whereas Sect. 5 contains conclusions.

2 Theoretical Background

The background of the presented research includes discretization algorithms, approaches to discretization of test datasets, and Naive Bayes classifiers.

2.1 Discretization

Many machine learning applications operate only on discrete data, whereas the nature of information in real life is often continuous. On the other hand, a number of methods work well with continuous features but perform better in discrete domain. Discretization converts wide spectrum of continuous values into datasets of discrete attributes, constituted by finite sets of intervals. It can be considered as a data reduction method which simplifies information as well as reduces possible information noise. But it is important to notice that in the discretized data always some loss of information occurs, therefore the process must be applied with caution.

Discretization algorithms can be fundamentally divided into two categories: supervised which utilize class information, and unsupervised which omit such information during discretization process. Generally discretization can be considered as four-step process: sorting all values, determining cut-points for splitting (or intervals for merging), performing splitting or merging according to an algorithm criterion, and evaluating the stopping condition of the process. Attribute values from the input set are assigned to one of the evaluated intervals.

Discretization Algorithms. The two most popular unsupervised discretization methods are so-called equal width and equal frequency binning. The former method seeks the minimum and maximum values of an attribute and then divides the whole range into the desired number of discrete intervals of equal width. There is a modification of the algorithm that relies on leave-one-out estimation of entropy [4]. The resulting number of bins is optimized and depends on the nature of input data. The equal frequency algorithm sorts all attribute values in ascending order, evaluates the minimum and maximum values for the discretized attribute, and then divides the range into some required number of intervals so that each part contains the same number of discrete values [8].

For the purpose of the presented research two supervised discretization methods were selected. Both of them utilize the Minimum Description Length principle (MDL). The first one is based on research of Fayyad and Irani [3], whereas the second one uses Kononenko's MDL criterion [6]. Supervised methods are considered as more efficient and delivering better results [2, 8].

Test Datasets Discretization. Application of test datasets is one of the ways of evaluating classifiers. The aim of such approach is to use for that purpose data which was not utilized during the training stage of a decision system building process. In cases of discretized data, it is obvious that learning datasets are discretized applying some parameters like a type of algorithm, number of bins, width of bin, frequency of instances in the bin, class, etc. Similar parameters should be applied for test datasets, but results would be different depending on the type of algorithm, other required parameters, and relationship between discretization processes of training and test datasets, whether they are dependent on each other or not.

2.2 *Bayes Classifiers*

Bayes classifiers are relatively simple but powerful, often used as a reference model for other classification research. The basic Naive Bayes for authorship attribution can be utilized in two versions, depending on the nature of input data. If the features set consists of binary variables that inform only if a word (from the previously selected list) exists in the analyzed text, then the multivariate Naive Bayes classifier can be used. If the information about word occurrences is extracted, the multinomial Naive Bayes algorithm is suitable for problem solving. For big sizes of the vocabulary the second approach is considered better [9].

Bayes classifier is based on Bayes' rule of conditional probability:

$$p(c_j \mid d) = \frac{p(d \mid c_j)p(c_j)}{p(d)} \quad , \tag{1}$$

where: $p(c_j \mid d)$—a'posteriori probability of instance d being in class c_j, $p(d \mid c_j)$—probability of generating instance d given class c_j, $p(c_j)$—a'priori probability of occurrence of class c_j, $p(d)$—probability of instance d occurring, and

$$p(d \mid c_j) = p(d_1 \mid c_j)p(d_2 \mid c_j)\dots p(d_m \mid c_j) \quad . \tag{2}$$

The MAP (maximum a'posteriori) decision rule is applied to get the result of classification process $NBC(d_1, \dots, d_n)$:

$$NBC(d_1, \dots, d_n) = \underset{c}{\text{argmax}} p(C = c) \prod_{i=1}^{n} p(D_i = d_i \mid C = c) \quad . \tag{3}$$

It is commonly assumed that values of numeric attributes are normally distributed, so the probability density function for Gaussian (normal) distribution is utilized. For specific purposes other distributions could be more suitable.

3 Experimental Setup

Processing of datasets during experiments required execution of steps, as follows:

1. preparation of input data,
2. discretization of input data (unsupervised and supervised) using various approaches to learning and test instances,
3. classification using Naive Bayes classifier,
4. classifier evaluation during the test stage.

The following subsections present the main conditions of the performed experiments, the characterization of input datasets, and descriptions of discretization and classification techniques employed.

3.1 Input Datasets

The main condition while creating sets of characteristic features for authorship attribution is that they should uniquely describe all texts of a given author, and at the same time they need to enable distinction from other authors. In the research linguistic descriptors from lexical and syntactic groups were chosen, reflecting frequencies of usage for selected function words and punctuation marks [12]. It is assumed that lexical elements characterize literary style of authors, whereas the style of sentences building is described by syntactic features.

As the base for all experiments texts of two pairs of authors were chosen, male and female [12]. For each author several works were studied. To obtain input data source texts were splitted into blocks of comparable size, and frequencies for selected descriptors were calculated. Each dataset consisted of attributes belonging to one of two classes, corresponding to two recognized authors, and performed classification was binary.

The validation of classification results was performed using test sets. It was important to prepare training and test datasets basing on the disjunctive works of writers. Such approach allows to get objective results. As the result separate training and test datasets were obtained, with balanced classes in each set.

3.2 Approaches to Discretization of Test Datasets

When a quality of a decision system is evaluated by using test sets, and input data needs to be discretized, the relation between discretization procedures for learning and test sets can be considered in three ways:

- “independent” (*Id*)—training and test datasets are discretized separately,
- “glued” (*Gd*)—training and test datasets are concatenated, the resulting set is discretized applying required parameters, and finally data is divided back into learning and test set,
- “test on learn” (*TLd*)—firstly training dataset is discretized using chosen parameters, and then test set is processed using bin’s range values calculated for training data.

“Independent” way is the easiest to apply, but intuitively it can be considered as not good, because the way how test set is discretized can be very different from results obtained for training dataset. Since training and test sets are analyzed separately, it is very likely that the bin ranges in both sets are different, and the numbers of bins in both sets may vary. This can possibly lead to the situation, where the same attribute value is assigned to different bins in training and test sets. That seems to be a problem which can degrade the system performance.

“Glued” approach allows to discretize all data in more consistent way. However, test data should be totally independent from training and vice versa, and this assumption is not entirely true because of common processing of both sets. For example minimum and maximum values in test set can be significantly different than in training set. Resulting discretized learning set will be definitely different when compared to the one obtained for data without test set appended.

Discretizing in the “test on learn” manner seems to be more natural and potentially better than previous two, as the ranges of bins found for training dataset are applied for test data. But in this case the possible influence of training information onto the test data exists, which violates the assumption about independence of information used for evaluation of the decision system.

As can be seen all three approaches have possible advantages and disadvantages and deeper experimental investigation is necessary to assess considered solutions. For all three ways of processing several discretization algorithms were employed: unsupervised—equal width, optimized equal width, and equal frequency, and supervised—Fayyad & Irani MDL, and Kononenko MDL.

Some of the properties and relationships described above are illustrated by examples shown in Table 1. For presentation purposes only few instances of bigger datasets being processed are presented. The equal width algorithm was used with number of bins parameter set to 3. Notation used for describing bins reflects the

Table 1 Exemplary results of discretization of input datasets applying equal width algorithm using: “independent” (*Id*), “glued” (*Gd*), and “test on learn” (*TLd*) approach

(a) Bin ranges calculated for training data			
Training bin ranges (*Id/TLd*):	(-inf–0.008105], (0.008105–0.009427], (0.009427-inf)		
Training bin ranges (*Gd*):	(-inf–0.007057], (0.007057–0.008903], (0.008903-inf)		
(b) Input and discretized training data			
Training data	Discrete *Id*	Discrete *TLd*	Discrete *Gd*
0.006783	(-inf–0.008105]	(-inf–0.008105]	(-inf–0.007057]
0.006915	(-inf–0.008105]	(-inf–0.008105]	(-inf–0.007057]
0.010151	(0.009427-inf)	(0.009427-inf)	(0.008903-inf)
0.009330	(0.008105–0.009427]	(0.008105–0.009427]	(0.008903-inf)
(c) Bin ranges calculated for test data			
Test bin ranges (*Id*):	(-inf–0.006909], (0.006909–0.008608], (0.008608-inf)		
Test bin ranges (*TLd*):	the same as training bin ranges (*Id/TLd*)		
Test bin ranges (*Gd*):	the same as training bin ranges (*Gd*)		
(d) Input and discretized test data			
Test data	Discrete *Id*	Discrete *TLd*	Discrete *Gd*
0.009475	(0.008608-inf)	(0.009427-inf)	(0.008903-inf)
0.010135	(0.008608-inf)	(0.009427-inf)	(0.008903-inf)
0.007278	(0.006909–0.008608]	(-inf–0.008105]	(0.007057–0.008903]
0.007493	(0.006909–0.008608]	(-inf–0.008105]	(0.007057–0.008903]

lower and upper boundaries of respective bin (*inf*—infinity used for formal description of first and last intervals). Subtables (a) and (c) show bin ranges calculated for training and test data respectively. Subtable (b) contains input data and outcomes obtained for training data whereas part (d) presents results of test sets discretization.

For unsupervised equal width and equal frequency discretizations the only parameter required was the number of bins. For optimized equal width algorithm the obtained numbers of bins were lower or equal to the parameter value. For both equal width versions the number of bin parameter ranged from 2 to 10 with step 1, and from 10 to 1000 with step 10. For equal frequency the maximum value of this parameter is equal to the number of instances in a discretized dataset. Because of discretizing training and test sets together somehow the maximum value of parameter had to be fitted to lower cardinality of processed datasets. The supervised discretization was applied without any parameters and resulting number of bins depended on nature of data.

It is important to point out that for "independent" discretization of test dataset the resulting number of bins in training and test sets could be different. Such effect caused problems during the classifier evaluation stage, where the numbers of bins in both sets were expected to be equal. To overcome this problem the names of ranges (obtained as strings) were converted to their ordinal numbers, and numerical data type for each attribute was declared. So from a classifier perspective it operated on numerical data, but transformed during discretization. The same conversion was applied to dataset discretized using other methods to unify the experiments.

Naive Bayes classifier can deal with different types of attributes, in particular numeric and nominal ones. During the experiments it operated on numerical data thanks to conversion mentioned above. The normal distribution was used for numeric attributes.

Discretization and classification were performed separately for data based on male and female texts, both groups of results were averaged, and as such were the subject of further analysis. There were also experiments performed for datasets without discretization, to obtain some reference values for comparison.

4 Results and Discussion

To obtain a reference point for discussion, classification for datasets without discretization was performed. For the Naive Bayes classifier the predictive accuracy was 86.94 % (calculated as mean of results obtained for male and female authors separately). This value is indicated in all figures presenting experimental results. Figure 1 gives results for unsupervised methods.

The range of parameters variation for equal width and equal frequency algorithms was initially very wide. Experiments showed that for all ways of discretization the most promising classification results were obtained for relatively small values of a given parameter, typically below 10. For higher values performance was decreasing rapidly. Therefore diagrams presented in Fig. 1 were prepared for number of bins up to 10.

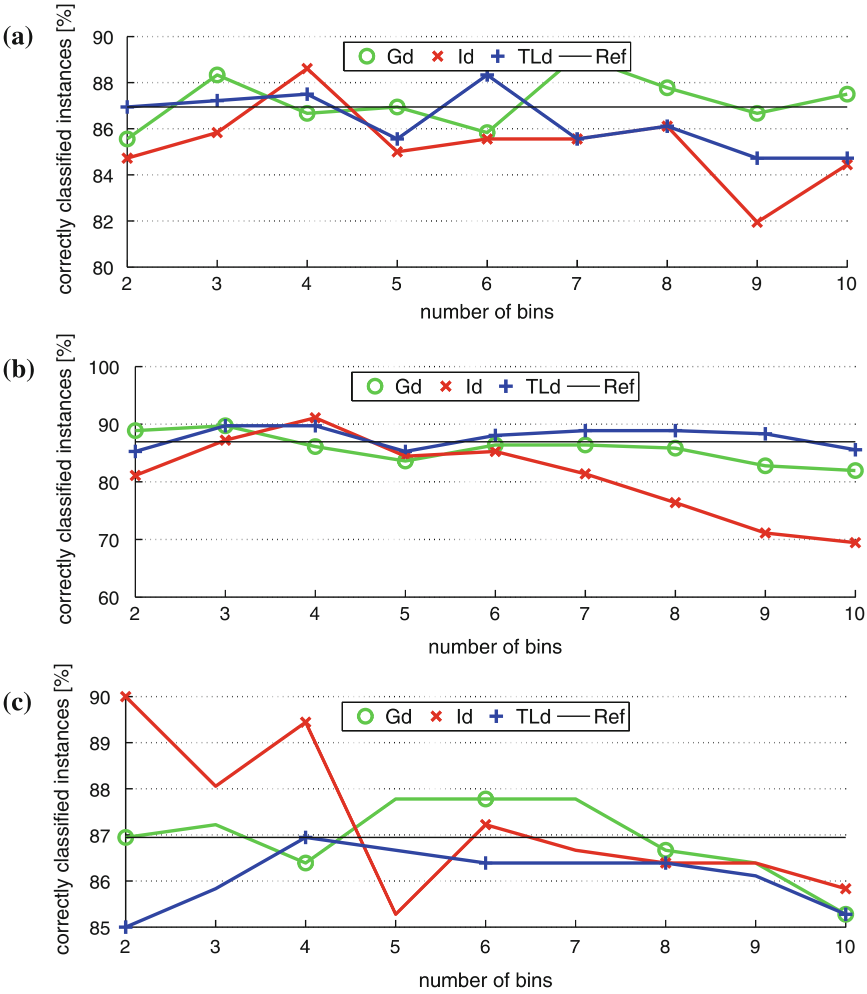

Fig. 1 Classifier performance for: **a** equal width, **b** optimized equal width, **c** equal frequency discretization, with training and test sets discretized using methods: "independent" (*Id*), "glued" (*Gd*), "test on learn" (*TLd*). *Ref* represents reference value obtained for non-discretized data

For equal width discretization (Fig. 1a) all three approaches to test set discretization delivered results better than reference for some values of a number of bins. But only "glued" approach performed well or almost well in the entire analyzed range. The similar observations could be made for optimized equal width discretization algorithm (Fig. 1b), except for the fact that all three discretization procedures delivered more stable results, when compared to that obtained for simple equal width algorithm. The best overall result (considering unsupervised methods) of correctly classified instances was obtained exactly for this algorithm using "independent"

Table 2 Results of Naive Bayes classification for experiments performed using three test datasets discretization approaches for supervised algorithms

Discretization algorithm	Test dataset discretization approach		
	"independent" (%)	"glued" (%)	"test on learn" (%)
Fayyad & Irani	95.56	93.33	91.11
Kononenko	83.33	94.44	91.11

discretization. Also equal frequency binning (Fig. 1c) gave better results for small parameter values. Especially "independent" discretization delivered good results, comparable to the best.

Table 2 presents results obtained for supervised discretization algorithms. Almost all algorithms applied for different approaches perform very well, exceeding the reference level. Only Kononenko MDL in combination with "independent" test datasets discretization delivered worse results.

As aforementioned, the most interesting range of discretization parameters lies below 10. Results changed there dynamically, therefore it was interesting to investigate that area more deeply. The idea was to observe only the range where classifier performance seems to be better than reference. Therefore average value of classifier efficiency for three analyzed ways of discretization in respect to values of algorithm parameters was calculated. Two of three algorithms performed better than reference for parameter equal or lower than 6. A set of boxplot diagrams presenting classification results is presented in Fig. 2, for discretization parameters limited up to 6 (value represents required number of bins).

The main aim of performed research was to find relations between classification accuracy, assessed during the classifier evaluation process, and a method of test datasets discretization. Intuitive analysis could lead to a conclusion that discretization of test sets performed in some reference to training dataset should deliver better

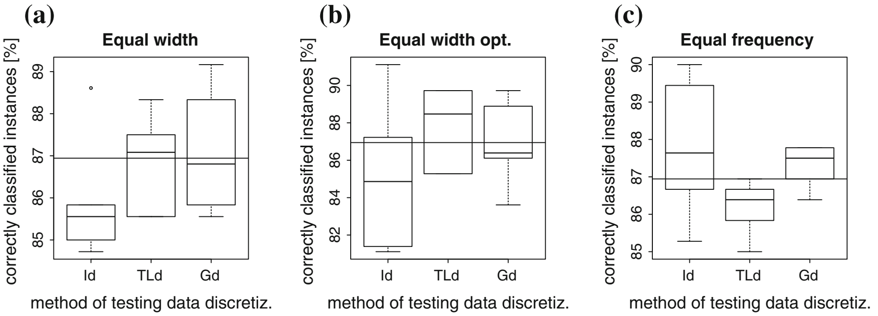

Fig. 2 Selected results of classifier evaluation for: **a** equal width, **b** optimized equal width, **c** equal frequency algorithms, with training set discretized using methods: "independent" (*Id*), "glued" (*Gd*), "test on learn" (*TLd*). Diagrams are based on results obtained for bin number ranged from 2 to 6. Reference level of 86.94 % is indicated

results. One of the reasons was that “independent” discretization outcomes for specific algorithms could have different number of bins in training and test datasets. Yet results presented in Fig. 2 show that a firm hierarchy between tested approaches does not exist. For both equal width algorithms “test on learn” method gives results better than others, whereas it is the worst approach for equal frequency binning. On the other hand, “independent” method, considered as a poor one, performs surprisingly well with this algorithm. It is important to point out that the best results during experiments were obtained for “independent” method for unsupervised as well as supervised algorithms.

Presented research results allow to state that there is no unequivocal rule allowing to select the best approach to test sets discretization. Depending on nature of data and chosen discretization algorithm all analyzed ways can be taken into consideration.

5 Conclusions

The paper presents research on the influence of way of test datasets discretization on results of classifier evaluation. For the executed tests the Naive Bayes was selected and all outcomes were analyzed in comparison with the reference value obtained for non-discretized datasets, using the same classifier. The experiments were binary classification tasks performed in authorship attribution domain.

Study results showed that good quality of decision system was obtained for relatively small number of bins in discretized data. But facts which must be taken into consideration to keep this conclusion valid are as follows: nature of analyzed data—stylometric datasets prepared as aforedescribed; system performing binary classification. Observation of discretized outcomes of supervised algorithms supports prove of such conclusion. Number of bins delivered by these methods, which analyze entropy of data along with its class attribution, were also small what means that such conversion of data (given relatively small number of bins) did not cause significant loss of information. Furthermore, discretization can have positive influence on efficiency of data exploration.

The research delivered results which allow to state that it is not possible to formulate one universal rule supporting process of selecting training and test sets discretization method. Depending on used discretization algorithm different approaches can be taken into consideration. Especially the “independent” approach, where training and test data are discretized separately delivered the best overall results. Therefore such way of discretization can be suggested as entry, preliminary approach in many applications.

Acknowledgments The research described was performed at the Silesian University of Technology, Gliwice, Poland, in the framework of the project BK/RAu2/2016. All experiments were performed using WEKA workbench [4].

References

1. Baron, G.: Influence of data discretization on efficiency of Bayesian Classifier for authorship attribution. Procedia Comput. Sci. **35**, 1112–1121 (2014)
2. Dougherty, J., Kohavi, R., Sahami, M.: Supervised and unsupervised discretization of continuous features. In: Machine Learning: Proceedings of the 12th International Conference, pp. 194–202. Morgan Kaufmann (1995)
3. Fayyad, U.M., Irani, K.B.: Multi-interval discretization of continuous-valued attributes for classification learning. In: Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI), pp. 1022–1029 (1993)
4. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. SIGKDD Explor. **11**(1), 10–18 (2009)
5. Kim, S.B., Han, K.S., Rim, H.C., Myaeng, S.H.: Some effective techniques for Naive Bayes text classification. IEEE Trans. Knowl. Data Eng. **18**(11), 1457–1466 (2006)
6. Kononenko, I.: On biases in estimating multi-valued attributes. In: 14th International Joint Conference on Articial Intelligence, pp. 1034–1040 (1995)
7. Kotsiantis, S.B.: Supervised machine learning: a review of classification techniques. In: Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth. HCI, Information Retrieval and Pervasive Technologies, pp. 3–24. IOS Press, Amsterdam, The Netherlands (2007)
8. Kotsiantis, S., Kanellopoulos, D.: Discretization techniques: a recent survey. Int. Trans. Comput. Sci. Eng. **1**(32), 47–58 (2006)
9. McCallum, A., Nigam, K.: A comparison of event models for Naive Bayes text classification. In: AAAI-98 Workshop On Learning For Text Categorization, pp. 41–48. AAAI Press (1998)
10. Schneider, K.M.: Techniques for improving the performance of Naive Bayes for text classification. In: Proceedings of 6th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing), pp. 682–693 (2005)
11. Stańczyk, U.: Rule-based approach to computational stylistics. In: Bouvry, P., Kłopotek, M., Marciniak, M., Mykowiecka, A., Rybiński, H. (eds.) Security and Intelligent Information Systems, LNCS (LNAI), vol. 7053, pp. 168–179. Springer, Berlin (2012)
12. Stańczyk, U.: Ranking of characteristic features in combined wrapper approaches to selection. Neural Comput. Appl. **26**(2), 329–344 (2015)
13. Youn, E., Jeong, M.K.: Class dependent feature scaling method using Naive Bayes classifier for text datamining. Pattern Recognit. Lett. **30**(5), 477–485 (2009)