Text Classification Method using a Named Entity Extractor

Shigeaki SAKURAI and Yoshimi SAITO
Corporate Research & Development Center, Toshiba Corporation
1, Komukai Toshiba-cho, Saiwai-ku, Kawasaki 212-8582, Japan
email:shigeaki.sakurai@toshiba.co.jp, yoshimi.saito@toshiba.co.jp

***Abstract*— We have previously proposed a text classification method based on lexical analysis and a key concept dictionary. Because creating a key concept dictionary is a time-consuming task, it is difficult to apply the method to many target tasks. This paper proposes a new text classification method based on expressions with classes of named entities. Here, the classes of the named entities are kinds of classes that gather relevant expressions with the same feature. This paper applies the method to an e-mail classification task and a questionnaire analysis task, and shows the effect of the method.**

I. Introduction

The development of computers and networks has enabled the gathering of large amounts of textual data. However, this data is not always used effectively. This has led to the active study of text mining techniques. Text classification included in the study classifies textual data into user-defined classes. Many text classification methods have been proposed [4][6].

We have proposed a method based on lexical analysis and a key concept dictionary created by experts [7][8]. This method is able to classify textual data with a high precision ratio and to discover the relationships between attribute values and classes. This method requires the creation of a key concept dictionary that is dependent on the target task. Because creating a key concept dictionary is a time-consuming task, it is difficult to apply the method to many target tasks. Thus, we have proposed a new method based on key phrases [9]. A key phrase is extracted from textual data using key phrase extraction rules created by experts. The experts create the rules based on linguistic features. The rules do not depend on the target task and can be applied to many target tasks. This method is also designed to realize high precision and to discover the relationships between attribute values and classes. This method requires the acquisition of a readable classification model. It acquires a model with a fuzzy decision tree format using a fuzzy inductive learning algorithm. This model is easy for users to understand. However, the precision ratios given by this method are lower than those of the method based on a key concept dictionary.

Many papers [1][5] have reported that SVM (Support Vector Machine) gives high precision ratios for text classification. However, SVM is not able to acquire a readable classification model because SVM acquires a hyperplane which identifies classes of textual data. On the other hand, techniques that extract expressions with classes of named entities included in the textual data have been studied. Although the execution of classes of named entities uses rules given by experts, the rules do not depend on the target task. Expressions with classes of named entities may provide appropriate features for textual data.

Thus, we propose a new text classification method based on SVM and a named entity extractor. This method aims to acquire high precision ratios without using dictionaries that depend on target tasks. We apply this method to three e-mail classification tasks and to an analysis task: a product classification task, a contents classification task, an address classification task, and a questionnaire analysis task. We examine whether our named entity extractor can extract features efficiently in these experiments. The experimental results are compared with the results based on words.

In the remainder of the paper, the proposed method is explained in section II and the numerical experiments using e-mail data and questionnaire data are shown in section III. A summary and details of future work appear in section IV.

II. Text classification

A. Flow

Text classification methods must perform two processes. One is the feature extraction and the calculation of the values that characterize textual data. The other is the learning of the classification model and the acquisition of the relationships between the extracted attribute values and the classes assigned to textual data. If we acquire a classification model, it is possible to infer a class of a new textual data item by evaluating attribute values based on the classification model. This paper proposes a method that uses a named entity extractor for the feature extraction and uses SVM for the learning of the classification model. Figure I shows the flow of our method. In this figure, the black arrows show the data flow in the learning phase and the white arrows show the data flow in the evaluation phase. Also, the text class stores classes assigned to the training textual data.

B. Feature extraction

Our named entity extractor deals with more kinds of words than previous named entity extractors. The extractor extracts expressions of proper nouns, numerical expressions, and general expressions that have important meanings in textual data. Here, a person's name, a company name, and a place name are expressions of proper nouns. A date, a time, and a sum of money are numerical expressions. A material name, a game name, and a food name are general expressions. For example, if the sentence "He has seen Hayao Miyazaki, who

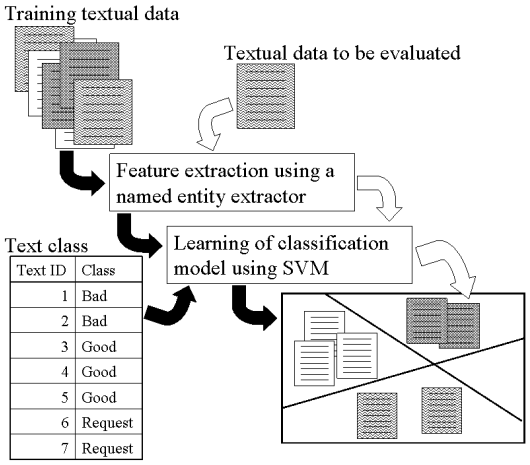


FIGURE I
FLOW OF TEXT CLASSIFICATION

is a famous movie director." is given to the named entity extractor, the extractor extracts "Hayao Miyazaki" and "movie director". The extractor also assigns the class of the named entity "person's name" to "Hayao Miyazaki" and assigns the class of the named entity "job name" to "movie director".

When textual data is input to the named entity extractor, the extractor outputs expressions with classes of named entities and their degrees of certainty. Here, each degree of certainty has an value from 1 to 100. The extractor is composed of three parts, a lexical analysis part, an extraction part, and an integration part. The extraction part is composed of three sub-parts: the extraction of basic expressions, the extraction of composite expressions, and the selection of extracted expressions. Figure II shows an outline of our named entity extractor.

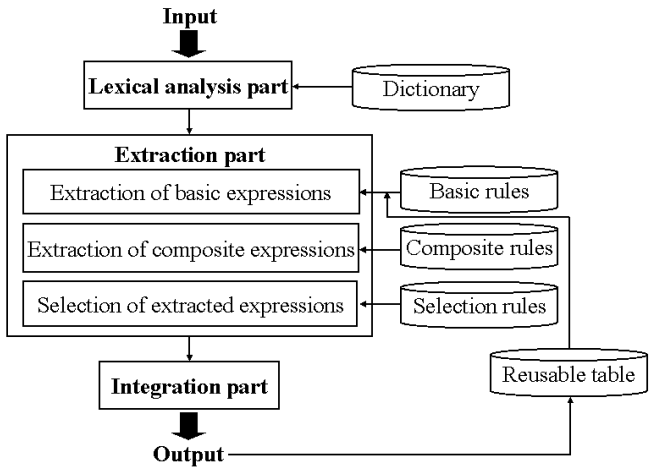


FIGURE II
FLOW OF NAMED ENTITY EXTRACTOR

Lexical analysis part:

Our named entity extractor deals with Japanese textual data. The data requires word segmentation. This part segments the data into words using lexical analysis [3] and uses a dictionary for the lexical analysis. The dictionary has 280 thousands words and each word has 346 attributes values, such as information regarding co-occurrence and named attributes of proper nouns.

Extraction part/extraction of basic expressions:

This sub-part extracts basic expressions by applying the results of lexical analysis to basic rules. Here, the basic rules have 537 rules and each rule describes the relationships that tie the patterns of articles and attributes to the classes of named entities.

Extraction part/extraction of composite parts:

This sub-part extracts composite expressions by applying a combination of basic expressions to composite rules. Here, the composite rules have 145 rules and each rule describes the relationships which tie the combination of basic expressions to the classes of the named entities.

Extraction part/selection of extracted expressions:

This sub-part evaluates the positions of extracted expressions, their degree of certainty, and their kinds of characters. This sub-part also applies the results of the evaluation to the selection rules and selects extracted expressions. The selection rules have 121 rules and each rule expresses the relationships which tie the positions, the degree of certainty, and the kinds of characters to the classes of named entities.

Integration part:

This part selects the most preferable expressions when there are some selected expressions in the same position. The selected expressions are also registered in the reusable table. This table is used to revise the extraction of basic expressions.

Here, the rules in these parts are created according to Japanese linguistic knowledge and are not dependent with specific tasks. Also, our named entity extractor was evaluated by using 250 articles from Japanese newspapers. The extractor could extract named entities and their classes with 75.0% recall ratio and 60.0% precision ratio.

The outputs of the named entity extractor are used in order to characterize each textual data item. The feature extraction process extracts expressions which have equal or higher degrees of certainty than the threshold from all textual data items. The process regards each extracted expression as an attribute. Also, the process regards whether the expression is included in the textual data item as an attribute value. That is, if the textual data item includes the expression, the attribute corresponding to the expression has 1 as an attribute value, otherwise the attribute has 0. For example, if three expressions "Hayao Miyazaki", "movie director", and "movie star" are given to characterize textual data and the former sentence is given, the feature extraction process sets attribute values as shown in Table I to the sentence.

TABLE I

EXAMPLE OF CHARACTERIZED TEXT

Hayao Miyazaki	movie director	movie star
1	1	0

C. SVM (Support Vector Machine)

SVM [10], proposed by Vapnik, is a method that generates a two pattern classifier. SVM maps training examples to a high dimensional space and determines a hyperplane that classifies a given training example in the high dimensional space. SVM uses the maximum margin principle to determine the hyperplane. Here, the margin is the minimum distance from the hyperplane to the training examples. The training examples that give the minimum distance are called support vectors. Figure III shows the relationship between the identification hyperplane, the margins, and the training examples. In this figure, circles indicate training examples with class 1, squares indicate training examples with class 2, and the sold line is the identification hyperplane. The examples on hyperplanes H1 and H2 indicate support vectors.

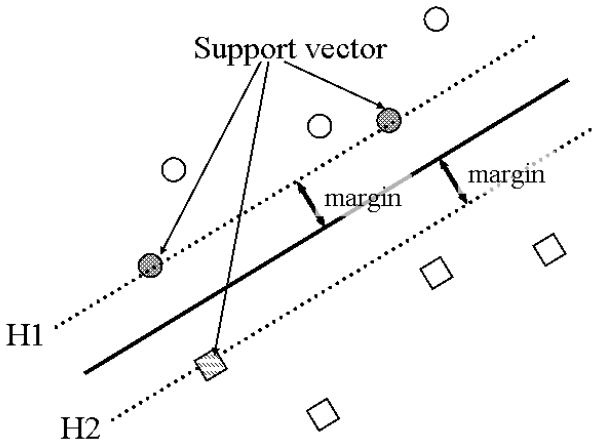


FIGURE III

CLASSIFICATION BASED ON SVM

It is possible for SVM to avoid calculating the positions of each training example in the high dimensional space by using a kernel function. Therefore, it is possible for SVM to determine the identification hyperplane with high speed.

On the other hand, SVM deals with training examples containing 2 classes. When SVM deals with training examples containing more than 2 classes, it is necessary to find ways of combining several SVMs. We use SVM, shown in [2], that is able to deal with training examples containing more than 2 classes.

III. EXPERIMENT

A. Experimental data

In our experiments, we used e-mail data collected by our customer center and questionnaire data collected by our advertising department. The e-mail data is composed of two kinds of data set. One data set has two kinds of classification criterion, product criterion and contents criterion. The product criterion analyzes the primary product in the e-mails and has 5 classes: washing machine, vacuum cleaner, refrigerator, microwave oven, and other home appliance. The contents criterion analyzes the type of contents described in the e-mails and has 5 classes: question, request, proposition, complaint, and other user communication. The other data set has an address criterion. The address criterion analyzes the most appropriate department for the customer center to send the e-mails and has 13 classes. Twelve of these classes correspond to a single department and the remaining class is for other departments that have few e-mails. On the other hand, the questionnaire data has an evaluation criterion. The evaluation criterion analyzes the evaluation that respondents give when they use a web site. The criterion has 5 classes: bad, complaint, good, request, and other evaluations. Table II, Table III, Table IV, and Table V show number of data points in each class. In this table, the last column shows the total number in each data set.

TABLE II

E-MAIL DATA: PRODUCT CRITERION

Class	Number
Washing machine	103
Vacuum cleaner	81
Refrigerator	84
Microwave oven	153
Other	45
Total	466

TABLE III

E-MAIL DATA: CONTENTS CRITERION

Class	Number
Question	266
Request	93
Proposition	10
Complaint	83
Other	14
Total	466

TABLE IV

E-MAIL DATA: ADDRESS CRITERION

Class	Number	Class	Number
A Dept.	43	H Dept.	112
B Dept.	11	I Dept.	12
C Dept.	39	J Dept.	25
D Dept.	30	K Dept.	20
E Dept.	94	L Dept.	42
F Dept.	26	Other	111
G Dept.	16	Total	581

B. Experimental method

In the experiments, we applied word-based features and expression-based features to textual data. The word-based features were generated using lexical analysis and tf-idf values defined by Formula (1).

$$\mathrm{tf}-\mathrm{idf}_i = \frac{1}{D}\cdot\log_2\left(\frac{D}{d_i}\right)\cdot\sum_j \frac{\log_2(t_{ij}+1)}{\log_2 w_j} \quad (1)$$

TABLE V
QUESTIONNAIRE DATA

Class	Number
Bad	1,168
Complaint	414
Good	1,132
Request	239
Other	490
Total	3,443

Here, D is the total amount of textual data, d_i is the amount of textual data that has the i-th word, w_j is the amount of words included in the j-th textual data, and t_{ij} is the amount of i-th words included in the j-th textual data. The formula revises word frequency by removing the effect of common words.

The word-based feature extraction method extracts words using lexical analysis. A tf-idf value was calculated for each word and if the tf-idf value was higher than the threshold, the word was selected as an attribute. In the experiments, the threshold was 0.005. The value was determined based on the results of preparatory experiments. Each textual data item was evaluated for each attribute to determine whether a corresponding word is included. If the textual data item included a corresponding word, the item was given an attribute value of 1. If the item did not include a corresponding word, the item was given an attribute value of 0.

The expression-based feature was applied to textual data as described in sub-section II-B. In the experiments, the threshold for the degree of certainty was 1. The value was determined based on the results of preparatory experiments.

Table VI shows the number of attributes for each criterion given by these feature extraction methods. In this table, Product, Contents, Address, and Evaluation indicate each criterion: product criterion, contents criterion, address criterion, and evaluation criterion. Word and NE indicate the feature extraction methods: the word-based method and the expression-based method. Here, the number corresponding to the product criterion is equal to the number corresponding to the contents criterion, because these criteria deal with the same data set.

TABLE VI
NUMBER OF ATTRIBUTES

	Word	NE
Product	2,099	1,057
Contents	2,099	1,057
Address	2,418	2,071
Evaluation	524	337

The experiments were performed using 10 cross validations. That is, each data set was decomposed into 10 data subsets. Classification models were acquired from 9 data subsets using SVM [2]. Here, SVM uses default parameters and a linear kernel option. Each textual data item included in the remaining data subset was evaluated and a class was inferred. If the class was equal to the original class that corresponded to the textual data item, we determined that the inference was correct. The experiment was repeated 10 times by changing the data subset to be evaluated. Lastly, we calculated precision ratios, which are the amount of textual data items with correctly inferred classes divided by the total amount of textual data items.

These experiments were performed for each criterion and each feature extraction method. We also performed experiments for a method that combines the word-based method with the expression-based method. In these experiments, each textual data item was characterized by two types of features. For example, the product criterion had 3,156 (=2,099+1,057) attributes.

C. Experimental results

Figure IV shows the precision ratios for each data set. In this figure, the x-axis gives the experimental number and the y-axis gives the precision ratios. Line graphs of Word, NE, and Word+NE indicate the experimental results corresponding to the feature methods: the word-based method, the expression-based method, and the method combining both words and expressions with classes of named entities.

Table VII shows the average precision ratios obtained by 10 experiments. In this table, Product, Contents, Address, and Evaluation indicate each criterion. Word, NE, and Word+NE indicate the feature extraction methods.

TABLE VII
AVERAGE PRECISION RATIO

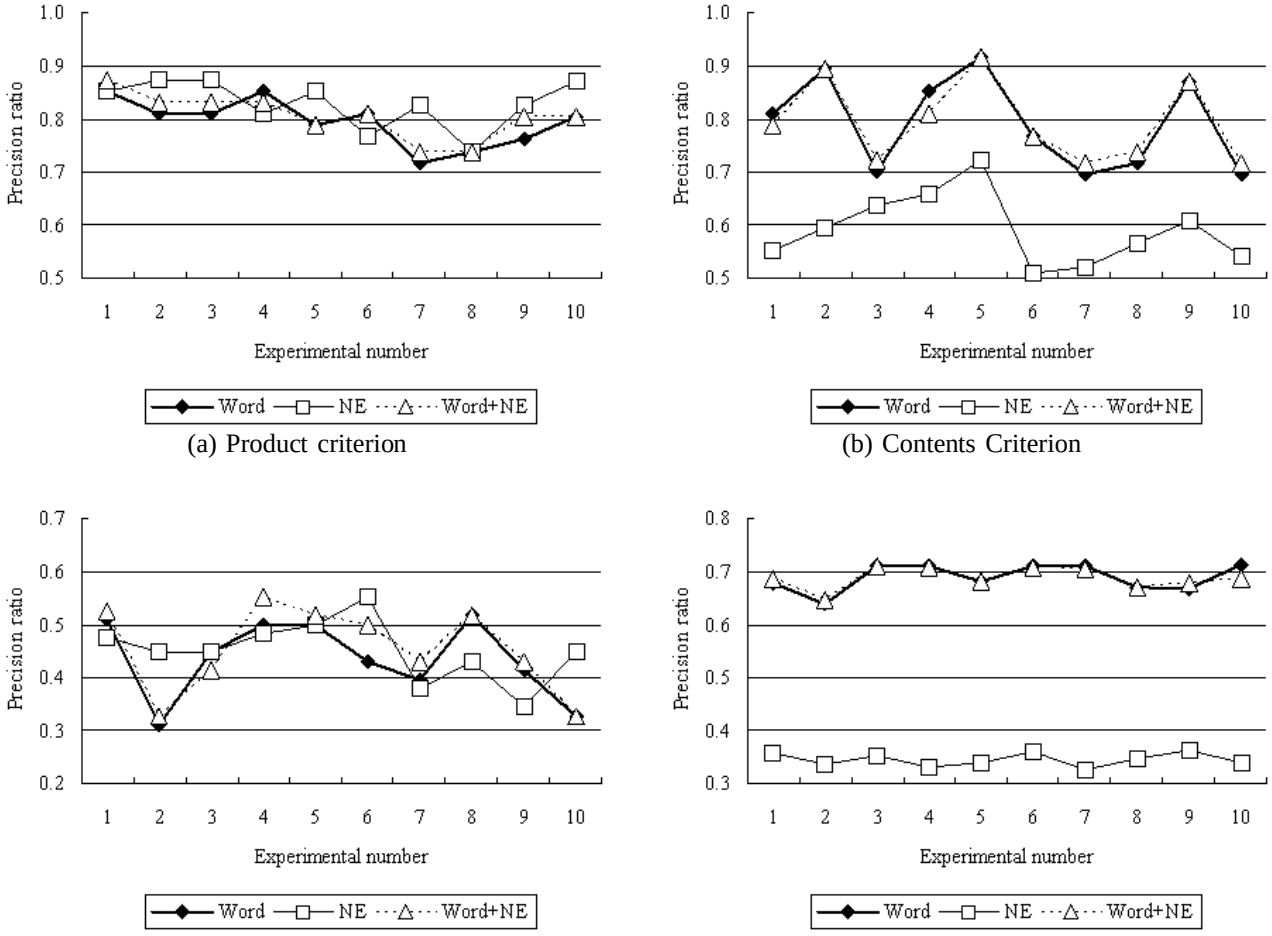
	Word	NE	Word+NE
Product	79.4%	82.8%	80.5%
Contents	79.2%	59.2%	79.4%
Address	43.5%	45.1%	45.4%
Evaluation	68.9%	34.6%	68.7%

D. Discussion

Precision ratio:

First, we considered the case where only expressions with classes of named entities were used. The experimental results show that the precision ratios are higher for the product criterion and the address criterion. This is why noun expressions are important in the classification of textual data and why it is possible for the named entity extractor to extract important noun expressions. That is, products such as washing machines and refrigerators are suitable for the product criterion and the types of products which the departments deal with are suitable for the address criterion. The named entity extractor expands the proper noun extraction technique and is suitable for extracting such expressions. However, it is important for the contents criterion and the evaluation criterion to extract adjectival expressions in the classification of textual data. It is difficult for the named entity extractor to extract adjectival expressions. Therefore, precision ratios are worse for these criteria.

Next, we considered the case where both words and expressions with classes of named entities were used. The experimental results show that the precision ratios are almost equal and sometimes higher than the method using only words in each data set. This is why it is possible to obtain adjectival expressions from the words and to obtain better noun expressions from the expressions with classes of named entities. However, the method based on both words and expressions



(a) Product criterion (b) Contents Criterion

(c) Address Criterion (d) Evaluation criterion

FIGURE IV
PRECISION RATIO

with classes of named entities gave worse precision ratios than the method using only expressions with classes of named entities in the product criterion. We think this is why noun expressions were also extracted from the words and worse expressions were mixed.

Number of attributes:

The number of attributes increases when using both words and expressions with classes of named entities. The combination method requires a large amount of time in order to acquire classification models however the time is much shorter than the sum of the time for lexical analysis and the time for named entity analysis. Therefore, the whole learning time increases only slightly, and it is not a concern that the learning time increases as the number of attributes increases.

Task-dependent dictionary:

We created key concept dictionaries for an e-mail data set [8]. The dictionaries contain important expressions and integrate similar expressions to classify textual data according to each criterion. The dictionaries were created by an expert, but each expression is simply described without using regular expressions. In the cases of the product criterion and address criterion, the dictionary gives a precision ratio of 86.7% and 61.1%, respectively. On the other hand, for the contents criterion, the proposed method gives a slightly higher precision ratio than the precision ratio based on the dictionary. However, the result for the contents criterion does not always give a sufficient precision ratio, because there is significant room for improvement in the dictionary, i.e., the dictionary is not able to extract interrogative expressions or negative expressions. These expressions are important for the classification of e-mails based on the contents criterion. If we use regular expressions, we can extract these expressions. We may be able to get a higher precision ratio. Therefore, in order to aim for a higher precision ratio, it is necessary to revise the text classification method without using a task-dependent dictionary.

IV. SUMMARY AND FUTURE WORK

This paper has proposed a new text classification method using SVM and a named entity extractor and applied this

method to e-mail data and questionnaire data. The experimental results show that features based on the named entity extractor are efficient for data sets in which noun expressions are important features for the classification of textual data. Also, the experimental results show that features combining words and expressions with the classes of the named entities provide reliable high precision ratios. The named entity extractor is important for performing text classification without using a task-dependent dictionary.

However, the precision ratios are not always sufficiently high. The feature extraction method and the acquisition method of the classification model need to be revised. We will attempt to use the classes of the named entities corresponding to expressions extracted by the named entity extractor and will also attempt to deal with more words including adjectival expressions by our named entity extractor. Moreover, we will attempt to consider a learning method based on the combining of multiple classification models.

REFERENCES

[1] A. Cardoso-Cachopo and A. L. Oliveira. An Empirical Comparison of Text Categorization Methods, Proc. of the 10th International Symposium on String Processing and Information Retrieval, pp. 183-196, 2003.

[2] C. -W. Hsu, C. -C. Chang, and C. -J. Lin. A Practical Guide to Support Vector Classification, http://www.csie.ntu.edu.tw/˜ cjlin/libsvm/

[3] Y. Ichimura, Y. Nakayama, M. Miyoshi, T. Akahane, T. Sekiguchi, and Y. Fujiwara. Text Mining System for Analysis of a Salesperson's Daily Reports, Proc. of Pacific Association for Computational Linguistics 2001, pp. 127-135, 2001.

[4] J. Joachims. Text Categorization with Support Vector Machines: Learning with Many Relevant Features, Technical Report LS-8 Report 23, Computer Science Department, University of Dortmund, Dortmund, Germany, 1997.

[5] L. M. Manevitz and M. Yousef. One-Class SVMs for Document Classification, Journal of Machine Learning Research, vol. 2, pp. 139-154, 2001.

[6] B. Raskutti, H. Ferrá, and A. Kowalczyk. Combining Clustering and Co-training to Enhance Text Classification using Unlabelled Data, Proc. of 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 620-625, 2002.

[7] S. Sakurai, Y. Ichimura, A. Suyama, and R. Orihara. Inductive Learning of a Knowledge Dictionary for a Text Mining System, Proc. of 14th International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems, pp. 247-252, 2001.

[8] S. Sakurai, A. Suyama, and K. Fume. Acquisition of a Concepts Relation Dictionary for Classifying E-mails, Proc. of the IASTED International Conference on Artificial Intelligence and Applications, AIA2003, pp. 13-19, 2003.

[9] S. Sakurai and A. Suyama. Rule Discovery from Textual Data based on Key Phrase Patterns, Proc. of the ACM Symposium on Applied Computing SAC 2004, pp. 606-612, 2004.

[10] V. N. Vapnik. The Nature of Statistical Learning Theory, Springer, 1995.