Web Intelligence and Agent Systems: An international journal 4 (2006) 431–441
IOS Press

A study on rough set-aided feature selection for automatic web-page classification

Toshiko Wakaki[a,*], Hiroyuki Itakura[a], Masaki Tamura[b], Hiroshi Motoda[c] and Takashi Washio[c]
[a]*Shibaura Institute of Technology, 307 Fukasaku, Minuma-ku, Saitama-City 337-8570, Japan*
[b]*Japan Advanced Institute of Science and Technology, 1-1, Asahidai, Tatsunokuchi, Ishikawa 923-1292, Japan*
[c]*The Institute of Scientific and Industrial Reasearch, Osaka University, Mihogaoka, Ibaraki, Osaka 567-0047, Japan*

Abstract. Due to the recent explosive increase of Web-pages on World Wide Web, it is now urgently required for portal sites like Yahoo! service having directory-style search engines to classify Web-pages into many categories automatically. This paper investigates how rough set theory can help select relevant features for Web-page classification. Our experimental results show that the combination of the rough set-aided feature selection method and the Support Vector Machine with the linear kernel is quite useful in practice to classify Web-pages into multiple categories because not only our experiments give acceptable accuracy but also the high dimensionality reduction is achieved without the need to search for a threshold for feature selection.

Keywords: Rough set, feature selection, Web-page classification, Support Vector Machines, C4.5, dimensionality reduction

1. Introduction

Web-pages on the World Wide Web are now explosively increasing, and the portal site services including the search engine function on the World Wide Web become even more important accordingly. Especially, at portal sites such as Yahoo! service, Web-pages should be classified hierarchically into many categories since they have directory-style search engines. At present, however, the task of classifying tremendous amount of Web-pages into many categories relies on time-consuming and expensive man power. Therefore, the automatic Web-page classification is urgently required for such portal sites to reduce costs and man power.

To meet such requirement, recently Tsukada et al. [12] proposed a method for automatic Web-page classification by using machine learning methods. In their approach, Web pages are downloaded from 5 domains of top-categories on Yahoo Japan!, and then frequent itemsets are generated as attributes by using co-occurrence analysis based on basket analysis. Next, using Web pages whose categories are known, a decision tree is learned in the framework of the generated attributes based on the decision tree learning technique C4.5 [9]. It has been reported that their method achieves acceptable accuracy for the classification of Web-pages. However, in applying their method [12], the threshold called minimum support should be given in advance to generate frequent itemsets as attributes. Thus the optimal threshold whose frequent itemsets as attributes gives the highest performance for Web-page classification should be found in an ad hoc way with varying their threshold in experiments since it cannot be known in advance. For the practical purpose of Web-page classification, a set of relevant features giving acceptable performance is required to be found without searching for an appropriate threshold.

This paper investigates how rough set theory [8], which needs no threshold, can help select features relevant for Web-page classification. Our experimental results show that the *rough-set aided feature selection method* [3,6,10] in conjunction with *Support Vector Machines* [4] with the linear kernel is useful in prac-

*Corresponding author. Tel.: +81 48 687 5158; Fax: +81 48 687 5198; E-mail: twakaki@sic.shibaura-it.ac.jp.

tice for classifying Web-pages into categories because it achieves quite acceptable accuracy without the need for an appropriate threshold for feature selection. Precisely speaking, a set of relevant features is selected in our experiments based on the rough set-aided feature selection method (the RSDR method, for short) using almost the same Yahoo data used in the experiments of Tsukada et al.. The result shows that features (attributes) are reduced to 3% of the original attributes without depending on any threshold for feature selection, and the performance for the RSDR method and the trained SVM classifier with the linear kernel is comparable with or better than the best classification performance obtained in their experiments. Furthermore, we also give the comparison with the additional experiments using the TF-IDF weighting method and the SVM-based feature selection method (the SVM-FS method) [2,11] as alternative feature selectors and C4.5 [9] as an alternative classifier. Especially, our experimental results for the latter, i.e. the SVM-FS method are new ones, which are not given in our previous paper [13]. We obtained an interesting experimental result that the SVM-FS method can also reduce features as greatly as the RSDR method without deteriorating the classification performance although an optimal (or minimal) set of features should be found empirically.

The paper is organized as follows. Section 2 introduces feature selection methods studied in this paper, namely the rough set-aided feature selection method, the TF-IDF weighting method, SVM-based feature selection method and the method generating frequent itemsets. Section 3 gives an overview of two kinds of classifiers: C4.5 and SVMs. Section 4 presents the experimental results. Section 5 addresses related work and gives discussions. Section 6 ends the paper with concluding remarks.

2. Features selection methods

In a classification problem, the number of features can be quite large, many of which can be irrelevant or redundant. A relevant feature is defined in [5] as one removal of which deteriorates the performance or accuracy of the classifier, and an irrelevant or redundant feature is one which is not relevant. These irrelevant features could deteriorate the performance of a classifier that uses all features since irrelevant information is included inside the totality of the features. Thus the motivation of a feature selector is (i) *simplifying* the classifier by the selected features; (ii) *improving or not significantly reducing* the accuracy of the classifier; and (iii) *reducing* the dimensionality of the data so that a classifier can handle a large volume of data.

So far, many approaches as feature selectors have been proposed. Some of them depend on a threshold for feature selection and others do not. In this paper, we comparatively study three methods that need a respective threshold and one that does not need any threshold. The former is *The Term Frequency-Inverse Document Frequency* weighting method (TF-IDF method), *the SVM-based feature selection method* (SVM-FS method) [2] and the method of generating *frequent item sets* as attributes proposed by Tsukada et al. [12], and the latter is the *rough set-aided dimensionality reduction* (RSDR method) [3,6,10] as a feature selector.

2.1. Rough set-aided dimensionality reduction

An overview of rough set-aided dimensionality reduction [3,6,10] is given as follows. Suppose that a dataset is viewed as a decision table T where attributes are columns and objects are rows. Let U denote the set of all objects in the dataset and A the set of all attributes such that $a : U \to V_a$ for every $a \in A$ where V_a is the value set for attribute a. In a decision system, A is decomposed into the set C of conditional attributes and the set D of decision attributes which are mutually exclusive and $C \cup D = A$. For any $P \subseteq A$, there is an equivalence relation $I(P)$ as follows:

$$I(P) = \{(x, y) \in U^2 \mid \forall a \in P\ a(x) = a(y)\}.$$

If $(x, y) \in I(P)$, then x and y are indiscernible by attributes from P. The equivalence classes of the *P-indiscernibility equivalence relation* $I(P)$ are denoted $[x]_P$. Given an equivalence relation $I(P)$ for $P \subseteq C$, the lower approximation $\underline{P}X$ is defined for any $X \subseteq U$ as follows:

$$\underline{P}X = \{x \in U \mid [x]_P \subseteq X\}.$$

The *C-positive region* of D is defined as the following set of all objects from the universe U which can be classified with certainty into equivalence classes in U/D using the knowledge in attributes C:

$$\mathrm{POS}_C(D) = \bigcup_{X \in U/D} \underline{C}X.$$

where $U/D = \{\ [x]_D \mid x \in U\}$.

An attribute $a \in C$ is *dispensable* in a decision table T if $\mathrm{POS}_{(C-\{a\})}(D) = \mathrm{POS}_C(D)$; otherwise at-

$QuickReduct(C, D, R)$

Input: the set C of all conditional attributes
the set D of decision attributes.
output: the reduct R of C ($R \subseteq C$)

1. $R \leftarrow \phi$
2. **do**
3. $T \leftarrow R$
4. $\forall x \in (C - R)$
5. **if** $\gamma_{R \cup \{x\}}(D) > \gamma_T(D)$
6. $T \leftarrow R \cup \{x\}$
7. $R \leftarrow T$
8. **until** $\gamma_R(D) = \gamma_C(D)$
9. **return** R

Fig. 1. The QUICKREDUCT Algorithm.

tribute a is *indispensable* in T. A set $R \subseteq C$ of attributes is called a *reduct* of C if it preserves the condition: $\mathrm{POS}_R(D) = \mathrm{POS}_C(D)$. Especially a set $R \subseteq C$ of attributes is called a *minimal reduct* of C if it is minimal among all reducts with respect to $\subseteq$. With regard to computational complexity and memory requirements, the calculation of all reducts is an NP-hard task [10]. To solve this problem, we use QUICKREDUCT algorithm [3,6] shown in Fig. 1 for feature selection of Web-page classification. The algorithm uses the *degree of dependency* $\gamma_P(D)$ as a criterion for the attribute selection as well as a stop condition as follows:[1]

$$\gamma_P(D) = \frac{\parallel \mathrm{POS}_P(D) \parallel}{\parallel U \parallel},$$

This algorithm does not always generate a *minimal* reduct since $\gamma_P(D)$ is not a perfect heuristic. It does result in only one close-to-minimal reduct, though it is useful in greatly reducing dataset dimensionality. The average complexity of QUICKREDUCT algorithm was experimentally determined to be approximately $O(n)$ for a dimensionality of n though the worst-case runtime complexity is $O(n!)$.

2.2. TF-IDF weighting method

TF-IDF weighting is based on the heuristics that (i) the more times that a word appears in a document, the more relevant that a word is to the content of that document, and (ii) the more documents a word occurs in, the less relevant that a word is to the content of documents. We use the following function called *TF-IDF weighting* to compute the weights taking into account the above heuristics:

$$tfidf(t,d) = \frac{tf(t,d)}{\Sigma_{s \in d} tf(s,d)} \left(\log \frac{N}{df(t)} + 1 \right),$$

where t, d and N denote a term, a document and the total number of documents respectively, $tf(t,d)$ is the frequency of the term t in the document d and $df(t)$ is the number of documents in which the term t appears.

TF-IDF weighting method is a feature selector which selects a term t as a relevant feature if $tfidf(t,d)$ for some document d is greater than a given *threshold*. According to the vector space model, each coordinate axis is defined as the one mapped from the corresponding selected term (i.e. feature). Therefore, in feature space, a document d is represented by a vector whose each value of the coordinate axis corresponding to a selected feature t gives rise to a nonzero value (1 in our case) if $tfidf(t,d)$ is greater than the *threshold*; otherwise a zero.

2.3. SVM-based feature selection method

Brank et al. [2] proposed a feature selection method based on linear Support Vector Machines. In their approach, first, the linear SVM is trained on a subset of training data to compute a vector of weights $\mathbf{w} = (w_1, \ldots, w_n)$ which is normal to the hyperplane separating the positive from the negative examples. Second, considering that features with small values of $|w_j|$ do not have a large influence on the predictions of the classifier based on $\mathbf{w}$, the dimensionality of feature space is reduced in a way that, if the absolute value $|w_j|$ of a feature j is small enough, the corresponding jth feature is deleted. As a result, a set $\mathcal{A}$ of selected features is obtained as the one which contains all the features undeleted in this way

In general, a SVM is a classifier for binary classification whereas Web-pages have multiple categories. So, in our research, we slightly extended the SVM-based feature selection method (SVM-FS method, for short) proposed by Brank et al. in order to apply it to Web-page classification as follows.

Let $Page_i^c$ be a set of terms extracted from the ith page for the category c. Then a set of all terms contained in $\cup Page_i^c$ is regarded as an unreduced feature set. When its cardinality is n, $Page_i^c$ is represented as a vector having the class label c in a n-dimensional (unreduced) feature space.

In the following, our SVM-FS method is shown where $Page_i^+$ (or $Page_i^-$) for any $Page_i^{c'}$ denotes a set of items contained in $Page_i^{c'}$, but it has a class label $+$ (or $-$) instead of c'.

[1] For any set A, $\parallel A \parallel$ denotes the cardinality of A.

1. Given $\cup Page_i^{c'}$, a binary class dataset DS_c for each class c is constructed as follows:
   For any $Page_i^{c'}$, if its class label c' is equivalent to c, $Page_i^+ \in DS_c$; otherwise $Page_i^- \in DS_c$.
2. For each class c, a set $\mathcal{A}_c$ of selected features are computed by applying Brank et al.'s approach to the dataset DS_c.
3. A set $\mathcal{A}$ of selected features is obtained as $\cup_c \mathcal{A}_c$.

2.4. Frequent itemsets as attributes

In the Tsukada et al.'s approach [12], *frequent itemsets* are generated as attributes to design tabular data from Web-pages based on *basket analysis* which is well-known in the field of data mining. Basket analysis targets a set of transactions consisting of a set of items (i.e. terms) and derives itemsets having *support* greater than a user-specified *threshold*. The support of an itemset I means how frequently I appears, and it is defined as the ratio of the number of transactions including the itemset to the total number of transactions. Itemsets having support greater than the threshold *minimum support* are called *frequent itemsets*. Then frequent itemsets extracted from Web-pages are defined as the attributes of a decision table that reflect the features of Web-pages for each class label.

After attributes are generated in this way, their decision table, i.e., a matrix T is defined as follows. Let c be the label of some class, T_c be the decision table for class c, *Page*$_i^c$ be the ith page for the class c which is a set of items, (i.e., nouns) and *Attribute*$_j$ be the jth frequent itemsets *Itemset*$_j^c$ generated for the class c. Then $T_c[i, j]$ is 1 if *Itemset*$_j^c \subset$ *Page*$_i^c$; otherwise it is zero. T is constructed by integrating T_c for every class label c.

3. Classifiers

To assess the effectiveness of the feature selection methods described above, we use two kinds of classifiers. One is C4.5 [9] and the other is Support Vector Machines (SVMs, for short) [4].

3.1. A classifier based on decision tree learning

Given a decision table (i.e., a set of training examples), the decision tree learning algorithm creates a tree data structure that can be used to classify new instances whose class labels are unknown. Given the training examples, it begins with the root of the tree to evaluate each attribute using a statistical test to determine how well it alone classifies the training examples. As a result, the best attribute is selected and used as the test at the root node of the tree. A descendant of the root node is then created for each possible value of this attribute, and the training examples are sorted into the appropriate descendant node by descending the branch corresponding to the example's value for this attribute. If (almost) all the training examples associated with the descendant node belong to the same class, it makes the descendant a leaf node having the class label of its examples instead of test. Otherwise, the entire process is then repeated using the training examples associated with each descendant node to select the best attribute to test at the point in the tree. This forms a greedy search in which the algorithm never backtracks to reconsider earlier choices.

C4.5 [9] uses a statistical criterion called *gain ratio* to evaluate the goodness of the attribute. When a new instance reaches a leaf node of the decision tree, its class is determined using the label stored there.

3.2. Support vector machines

Firstly, we introduce SVMs with the linear kernel. Let $\mathbf{x}_1, \ldots, \mathbf{x}_m$ be training vectors representing training examples in a d-dimensional feature space, and $y_1, \ldots, y_m \in \{-1, 1\}$ be a class variable denoting that $\mathbf{x}_i$ $(1 \leqslant i \leqslant m)$ is a positive example if y_i is 1 and is a negative example if y_i is -1. Then the problem is to classify a vector of an unknown class by using a decision boundary learned from these training vectors. For linear SVMs, the decision boundary is a hyperplane whose functional form can be written as

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b,$$

where $\mathbf{w}$ is a d-dimensional weight vector normal to this hyperplane, and b is a bias term.

In general, the class predictor trained by a SVM has the form *prediction*$(\mathbf{x})$ = *sign*$(f(\mathbf{x}))$ where *sign*(z) is defined as 1 if $z \geqslant 0$; otherwise is -1, and $f(\mathbf{x})$ is derived using a set SVs of *support vectors* $\mathbf{s}_i$ as follows:

$$f(\mathbf{x}) = \sum_{s_i \in SVs} \alpha_i y_i \mathbf{K}(\mathbf{s}_i, \mathbf{x}) + b.$$

Here $\mathbf{K}(\mathbf{z}, \mathbf{x})$ is a *kernel* function. In case of a linear kernel $\mathbf{K}(\mathbf{z}, \mathbf{x}) = \mathbf{z}^T\mathbf{x}$, the class predictor can be rewritten as $sign(\mathbf{w}^T\mathbf{x} + b)$ where the weight vector is given by

$$\mathbf{w} = \sum_{s_i \in SVs} \alpha_i y_i \mathbf{s}_i.$$

In our experiments, we used SVMs with the linear kernel as well as SVMs with the polynomial kernels ($p = 2, 3$) defined as follows:

$$\textbf{Polynomial Kernel} \qquad \mathbf{K}(\mathbf{z}, \mathbf{x}) = (\mathbf{z}^T \mathbf{x} + 1)^p.$$

The SVM tool we used was TinySVM [15] developed at Nara Institute of Science and Technology.

4. Experimental evaluation

4.1. Datasets used

Tsukada et al. [12] performed the experiments for the classification of Web-pages on 5 domains of 14 top-categories in Yahoo! Japan: "*Arts & Humanities*", "*Business & Economy*", "*Education*", "*Government*" and "*Health*" (respectively abbreviated here as *Ah, Be, Ed, Go, He*). They randomly downloaded about 250 Web-pages per category, for a total of 1270 Web-pages (see Fig. 2).

According to their method, downloaded pages are first subjected to a pre-processing procedure including removal of the HTML tag, morphological analysis, and so on. The morphological analysis is done by using the system "*chasen*" [14], and the set $Page_i^c$ $(1 \leqslant i \leqslant n_c)$ of noun keywords is derived for each category c as denoted below:

$$\textit{class}_c \Leftrightarrow \{\textit{Page}_1^c, \ldots, \textit{Page}_i^c, \ldots, \textit{Page}_{n_c}^c\},$$

$$\textit{Page}_i^c = \{\textit{word}_{i,1}^c, \ldots, \textit{word}_{i,j}^c, \ldots\},$$

where $\textit{Page}_i^c$ indicates the i-th Web-page labeled $\textit{class}_c$, and $\textit{word}_{i,j}^c$ indicates the j-th item extracted from $\textit{Page}_i^c$.

In our experiments, we used 1270 Web-pages (i.e. $\sum_c n_c = 1270$) collected by Tsukada et al. [12] where pages belonging to multiple categories were included, whereas in the experiments of Tsukada et al., they used the dataset which was a subset of ours and comprised of only 1000 Web-pages.

Consequently in our experiments, we used the 11385 items (i.e., nouns) in total extracted from the 1270 Web-pages. Such a pre-processed dataset $\bigcup \textit{Page}_i^c$ can be used as a set of training and testing examples for Web-page classification, to which various feature selection methods addressed in Section 2 become applicable.

4.2. Performance measures

The following four quantities are used in several measures to evaluate the performance of a classifier.

- TP (True Positive): the number of documents *correctly classified* to that class.
- TN (True Negative): the number of documents *correctly rejected* from that class.
- FP (False Positive): the number of documents *incorrectly classified* from that class.
- FN (False Negative): the number of documents *incorrectly rejected* to that class.

Using these quantities, the performance of the classification is evaluated in terms of *Accuracy*, *Error rate*, *Precision*, *Recall*, and F_1 *measure* defined as follows [11]:

$$\textit{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN},$$

$$\textit{Precision} = \frac{TP}{TP + FP},$$

$$\textit{Recall} = \frac{TP}{TP + FN},$$

$$F_1\ \textit{measure} = \frac{2}{\dfrac{1}{\textit{Precision}} + \dfrac{1}{\textit{Recall}}}.$$

When distributions are highly skewed, as those in text categorization often are, using *Accuracy* or *Error rate* $= 1 -$ Accuracy may be inappropriate. Instead, *Recall*, *Precision*, and F_1 *measure* are commonly used as classification performance measures. There is a trade off relationship between *Precision* and *Recall* So, the F_1 *measure* which is the harmonic mean of *Precision* and *Recall* is mainly used in this study since it takes into account effects of both quantities.

4.3. Experimental settings

Using Yahoo data over 5 categories addressed in Section 4.1, we performed experiments in order to evaluate the following items:

(i) How effective for Web page classification is the rough set-aided feature selection method (the RSDR method) compared with SVM-based feature selection method (SVM-FS method), TF-IDF weighting method, and the method of generating frequent itemsets as attributes when these methods are used with classifiers such as C4.5, linear SVMs and polynomial SVMs?

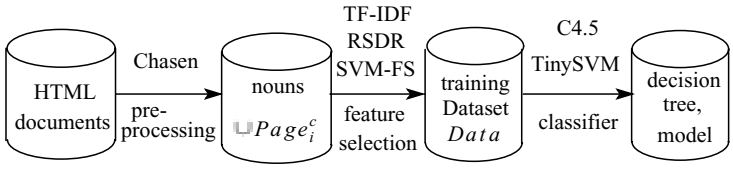

Fig. 2. The flow of data for Web page classification.

(ii) With what kernel do SVMs perform best for Web-page classification?
(iii) What is the performance of QUICKREDUCT algorithm? Is the QUICKREDUCT algorithm fast enough for practical use of the Web-page classification?

Applying the feature selection methods described in Section 2 to the pre-processed dataset $\bigcup Page_i^c$, the reduced dataset *Data* was generated as the attribute-reduced decision table (see Fig. 2). In our experiments, the TF-IDF weighting method, the RSDR method and the SVM-FS method as feature selectors were evaluated.

The classification performance is evaluated by using the reduced dataset *Data* and the classifier C4.5 or TinySVM as follows.

First, for each category c, the binary class dataset $Data_c$ is constructed from *Data*, each having examples of binary classes: positive and negative examples of $class_c$. Such $Data_c$ is used to evaluate the *binary class classification* in our experiments as was done in the experiments of Tsukada et al. The main reason is that a supervised learning methods like TinySVM can not classify a set of data into multiple classes at once though some documents may belong to multiple categories.

Next, the classification performance of the classifier C4.5 or TinySVM using the dataset $Data_c$ is evaluated based on *n-fold cross-validation*. This method divides all examples in $Data_c$ into n subsets of approximately equal size. Each time one of the n subsets is used as a set of testing examples and the other $n-1$ subsets are put together to form a set of training examples. The same trial is repeated n times, and the averaged performance in the n repetitions is the result used to asses our approach. Tsukada et al. applied 4-*fold cross-validation* in their experiments, so did we in our experiments.

The averaged F_1 values in 4 repetitions for each of 5 categories are shown for the classifier C4.5 in Fig. 3 and for TinySVM in Fig. 4, where four lines in each figure correspond to one case selected no features and three cases applied three kinds of feature selection methods, that is, the TF-IDF method whose threshold is 0.4, the RSDR method and the SVM-FS method whose total number of selected features is 903.

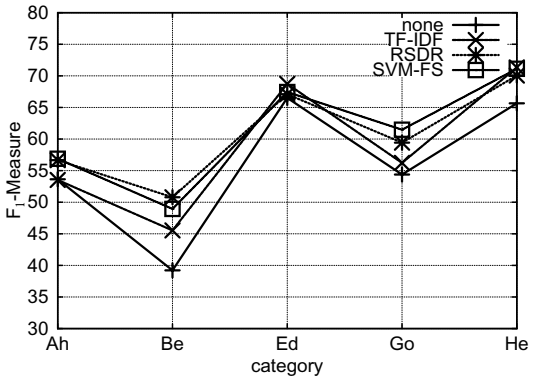

Fig. 3. F_1 value for each of 5 categories by C4.5.

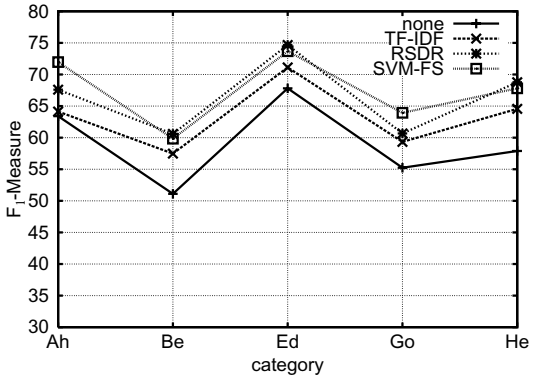

Fig. 4. F_1 value for each of 5 categories by SVM.

4.4. Experimental results

With respect to the TF-IDF and the RSDR methods, the averaged classification performance over 5 categories is shown in Table 1 for the classifiers, C4.5, linear SVMs and 2nd order polynomial SVMs respectively. Results of four different feature selection schemes are shown for each classifier: 1) no feature selection, 2) TF-IDF, 3) RSDR, 4) RSDF after TF-IDF. In this table, R means the RSDR method and T means the TF-IDF method, and whether they were used or not, is denoted

Table 1
Performance for RSDR and TF-IDF

classifier	feature selection			performance (%)			
	T	R	Num.	Accu	Prec	Recall	F_1
C4.5	×	×	11385	86.36	77.85	43.91	55.88
	0.4	×	1113	87.01	76.81	49.57	60.11
	×	◯	336	87.23	78.16	49.94	60.80
	0.25	◯	371	87.18	78.81	48.80	60.14
SVM (linear)	×	×	11385	85.26	67.17	52.91	59.09
	0.4	×	1113	87.10	72.63	58.47	64.76
	×	◯	336	88.19	77.62	58.16	66.46
	0.25	◯	371	88.65	70.92	59.05	67.69
SVM (2nd polynomial)	×	×	11385	81.78	55.62	48.04	51.08
	0.6	×	638	84.74	64.94	53.03	58.14
	×	◯	336	85.10	58.52	56.50	60.41

by the symbols ◯ or ×. Especially for any case the TF-IDF was used, the optimal threshold value of the TF-IDF offering the highest F_1 value is shown instead of ◯ along with the corresponding performance. Num., Accu., Prec., Recall and F_1 denote *Number* of selected features, *Accuracy*, *Recall*, *Precision* and F_1 *measure* respectively.

With respect to the SVM-FS method, the averaged classification performance over 5 categories is shown in Table 2 for the classifiers, C4.5 and linear SVMs where each case was evaluated by specifying the number of features from 301 to 1204. Especially, in order to compare the SVM-FS method with the RSDR method, SVM-FS was evaluated for the case whose number of features was specified as 336, i.e. the number of features selected by the RSDR method as is shown in Table 2.

On the other hand, the classification performance obtained by Tsukada et al.'s approach is shown in Table 3 in order to compare our results with theirs. In Table 3, Num. denotes *Number* of frequent itemsets (attributes) generated by minimum support level "Minsup", and F_1[2] denotes the corresponding averaged F_1 value over 5 categories.

Figure 5 shows the classification performances for the linear SVMs versus selected features with respect to RSDR, SVM-FS and TF-IDF method where F_1 values in Tables 1 and 2 are used. Especially, the performance of RSDR is depicted by the symbol $+$.

These performances indicates the following results:

1. *Effectiveness of feature selection*. Applying any feature selection method such as RSDR, TF-IDF (or the both) and SVM-FS can improve the classification performance because C4.5 and SVMs could have higher F_1 values when one of these feature selection methods was applied than they did when no feature selection method was used.
2. *Feature selection methods*. Comparing the RSDR method with the TF-IDF method for each classifier, the RSDR method is more effective than the TF-IDF method because the F_1 value obtained when the RSDR method was applied is higher than the highest F_1 value obtained by the TF-IDF method with varying threshold values.
   Comparing the RSDR method with the SVM-FS method for the linear SVM classifier, both achieves acceptable performances with keeping high dimensional reduction because the F_1 value 67.00% obtained by the SVM-FS method is a little better than or comparable with F_1 value 66.46% of the RSDR method for selected features, 336 (3% of the unreduced features). However, such an optimal (or minimum) set of features without deteriorating the classification performance for the SVM-FS method is not known in advance and should be found experimentally.

Table 2
Performance for SVM-FS

classifier	Num.	performance (%)			
		Accu	Prec	Recall	F_1
C4.5	301	86.82	77.37	48.53	59.52
	336	86.66	76.27	48.57	59.22
	397	86.65	76.58	48.01	58.92
	602	87.51	78.26	51.94	62.34
	903	87.18	77.20	50.80	61.17
	1204	86.85	77.12	48.11	59.12
SVM(linear)	301	87.84	77.21	56.12	64.97
	322	88.25	77.21	59.02	66.88
	336	88.28	77.15	59.28	67.00
	397	88.35	76.66	60.72	67.70
	602	87.70	74.30	59.87	66.20
	903	88.26	76.59	60.38	67.44
	1204	87.67	75.01	58.70	65.78

[2] F_1 values in Table 3 are calculated based on the performance for 5 categories shown in [12, p. 310].

Table 3
Performance for frequent itemsets

classifier	feature selection		performance (%)			
	Minsup	Num.	Accu	Prec	Recall	F_1
C4.5	10%	823	88.4	79.4	57.3	66.3
	20%	78	86.1	74.6	48.0	57.8

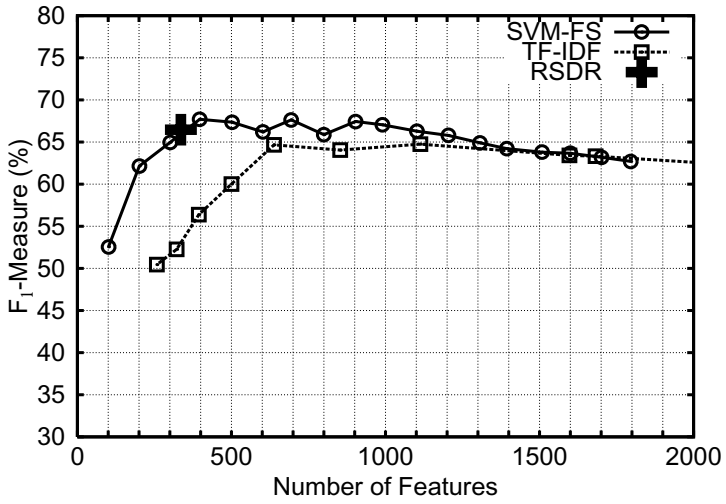

Fig. 5. Performances for 3 feature selection methods.

Comparing the SVM-FS method with the TF-IDF method for each classifier, the best performance of the SVM-FS method is always better than the highest performance of the TF-IDF method.
With respect to Tsukada et al.'s feature selection method, the RSDR method as well as SVM-FS method combined with the linear SVM classifier may be more effective than their method of generating frequent itemsets because it can obtain the better or comparable F_1 value by using the far less number (i.e., 336) of features than theirs (i.e. 823) for the case having the best F_1 value in their experiments (see Table 3).

3. *Number of selected features*. The RSDR method can achieve great *dimensionality reduction*, reducing the number of features or attributes to 336 (i.e., only 3% of the 11385 unreduced attributes) without the need to use any threshold as well as without deteriorating the classification performance using the linear SVM classifier.
The SVM-FS method can also reduce feature space as greatly as (or a little more than) that of RSDR without deteriorating the classification performance using the linear SVM classifier though such an optimal set of features should be searched experimentally.
4. *Classification performance*. In our experiments, using the linear SVM classifier, we obtained F_1 values 66.46% for RSDR, 67.69% for RSDR combined with TF-IDF and 67.44 % for SVM-FS in conjunction with linear SVMs whose values are better than or comparable with the best F_1 value 66.3% obtained for frequent itemsets of minimum support level 10% in the experiments of Tsukada et al. These are acceptable performances for Web-page classification.
5. *Classifiers and SVM kernels*. SVMs with the linear kernel achieve the best performance for Web-page classification among C4.5, the linear SVMs and SVMs with 2nd-order polynomial kernels since with respect to any feature selection method, its classification performance (F_1 value) is always the best among them.
6. *Computational complexity*. Figure 6 shows the runtime of our QUICKREDUCT program to compute the reduct for the number n of attributes. Given 11385 attributes, it takes 428 sec runtime to compute the reduct under Linux operating system on a 2.4 GHz Pentium IV computer, which is the acceptable performance since it is necessary to compute the reduct only one time for training SVMs or C4.5 classifiers.

Table 4
Number of Features for RSDR combined with TF-IDF

feature selection		TF-IDF Threshold				
R	T	0	0.25	0.40	0.50	0.60
×	◯	11385	1682	1113	835	638
◯	◯	336	371	388	470	308

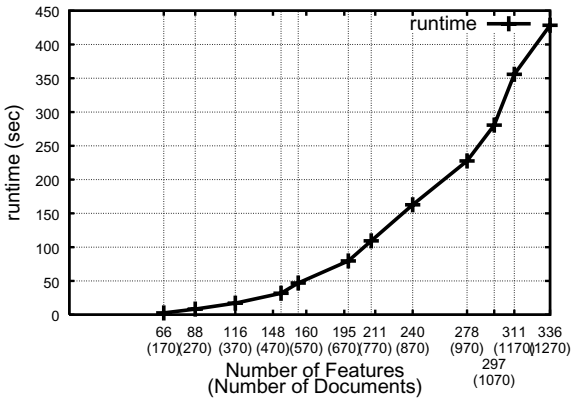

Fig. 6. RSDR runtime wrt dimensionality.

On the other hand, though it takes only few seconds for the SVM-FS method to find a set of the selected features for the given threshold, i.e. the number n of its cardinality, the task to search for the optimal number of features experimentally needs the human power such as preparing data sets used in experiments and is very time-consuming.

Figures 7, 8, and Table 4 are additional results of our experiments as follows.

Figure 7 shows how much the classification performance (i.e. F_1 value) depends on the number of selected features (or the TF-IDF threshold value) when decreasing it from 11385 to 258 (or increasing the threshold from 0.0 to 1.0) for the respective classifiers, C4.5, SVMs with the linear kernel and SVMs with 2nd and 3rd order polynomial kernels. We can see again that, for Web page classification, SVM with the linear kernel is the best of all, C4.5 is the 2nd best, SVM with the 2nd polynomial kernel is the 3rd best, and SVM with the 3rd polynomial kernel is the worst for the TF-IDF threshold values between 0.0 and 0.8. The optimal TF-IDF thresholds offering the highest F_1 values for the respective classifiers given in this figure are described in Table 1.

We also evaluated the effectiveness of the feature selection method for RSDR combined with TF-IDF method, which is shown in Fig. 8 and Table 4. Figure 8 shows how much the classification performance (i.e.

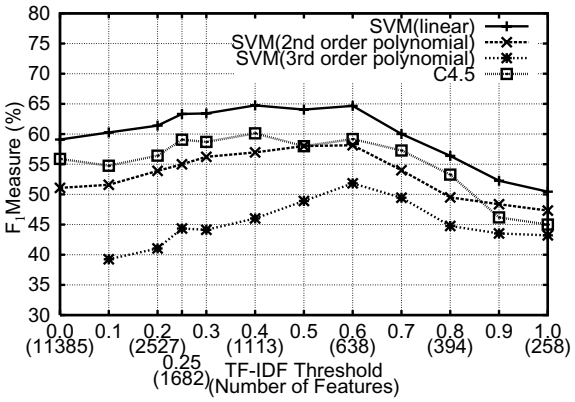

Fig. 7. Performance of classifiers and SVM kernels.

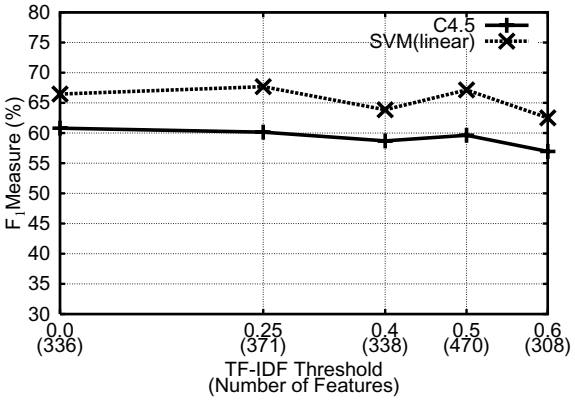

Fig. 8. Performance for RSDR combined with TF-IDF.

F_1 value) depends on the TF-IDF threshold for two classifiers, C4.5 and SVM with the linear kernel, where the threshold is increased from 0.0 to 0.6 as shown in Fig. 8 and Table 4. In each classifier, the RSDR method was used after the TF-IDF method was used. These results again confirm that the RSDR method can select the relevant features immediately without depending on features selected by the TF-IDF method with the varying TF-IDF threshold values.

5. Related work and discussions

In our experiments used Yahoo data, we found that the RSDR method combined with the linear SVM classifier can achieve the high dimensionality reduction, i.e. 336 selected features (3% of the original ones) with keeping the acceptable classification performance (F_1 value: 66.46%) without any need of threshold for feature selection. We also found that the SVM-

FS method combined with the linear SVM classifier can achieve dimensionality reduction as greatly as (or slightly greater than) the RSDR method without deteriorating classification performance. To our surprise, using Yahoo data, the number of features selected by the RSDR method, which was uniquely determined by QUICKREDUCT algorithm, nearly coincides with the optimal (minimum) number of features selected by the SVM-FS method though such an optimal set of features which does not deteriorate classification performance had to be searched by the SVM-FS method experimentally (see Fig. 5). Whether such coincidence of the number of the optimal feature set always holds between the RSDR method and the SVM-FS method should be examined and verified by conducting many more experiments using various kinds of datasets. However, we conjecture that such coincidence may occur because the RSDR method tries to find automatically the quasi minimal reduct R of attributes (i.e. features) which can classify all objects in a database into the sets for categories using the information of R-positive region of the decision attribute based on the rough set theory.

With respect to the SVM-FS method proposed by Brank et al., it is unclear to what extent the number of features can be reduced without deteriorating classification performance. For the practical purpose, however, the SVM-FS method may be more robust than RSDR method because it can handle a large volume of dataset [2].

Chouchoulas and Shen [3] proposed the text categorization system where features are selected by applying two processes, firstly keyword acquisition and secondly rough set-aided dimensionality reduction. In their system, keywords are acquired based on the weighting methods such as the term frequency-inverse document frequency (TF-IDF) metric, the fuzzy relevance metric (FRM) and so on. The rough set theory is not used for keyword acquisition, but is applied only for reducing feature space dimensionality in their approach. On the other hand, we explored the possibility that the rough set theory contributes to keyword acquisition as well as the dimensionality reduction. Our experiments corresponding to Chouchoulas and Shen's approach are shown in Fig. 8 and Table 4 where RSDR method is applied after applying the TF-IDF method for the feature selection. These results show that, performance of the combination of the TF-IDF method and the RSDR method does not always give the better classification performance than that of the case applied the RSDR method only (without the TF-IDF method). It is unclear for their method that how many keywords should be acquired before applying the rough set-aided dimensionality reduction in order not to deteriorate the classification performance.

Lingras and Butz [7] showed how the classification obtained from a support vector machine can be represented using rough sets. Though their formulation is very interesting, no experimental results and applications are given. Their future work is to verify how their approach is especially useful and effective for softmargin classifiers by performing experimental evaluation.

Recently, An et al. [1] also conducted similar experiments to ours using Yahoo Data to assess the effectiveness of the rough set feature selection method on Web-page classification. Using 7615 pages downloaded from 13 Yahoo categories as the training data, they first chose n frequent terms ($n = 20, 30, 40, 50, 60$ in their experiments) occurring Web-pages from each category, and then in the small n-dimensional vector space, rough set-aided feature selection and a binary classifier learning were done for each category using the average 590 pages from each category. Classification of a given new page was determined by combining the voting results obtained from 13 classifiers. Their experiments showed the effectiveness of the rough set-aided feature selection for a larger n (i.e. $n = 50, 60$). However, although an accurate comparison is difficult because of the differences in experimental setting, their classification performances (i.e. the F_1 values) obtained on the basis of their method are not so as good as ours. They did not study the optimal number n of the original features giving the best classification performance where n is regarded as a kind of threshold. The number n of the original features is so small in their experiments that the number of the reduced attributes is also very small (e.g. average 2.46 for 60 original features). Besides, their experiments do not show whether rough set-aided feature selection is more effective than the other feature selection methods for Web-page classification.

Tsukada et al. proposed the method to generate frequent itemsets as attributes (i.e. features) based on basket analysis, taking into account co-occurrence of words in Web-pages. Though the performance for minimum support level 10% in Table 3 is comparable with that of the linear SVM with the RSDR or SVM-FS method in Table 1, its number 823 of generated features (i.e. frequent itemsets) is two and half times of the number of the features selected by the RSDR method as well as SVM-FS method. Since almost all the frequent itemsets have only one element (i.e. noun) except about 10% of them having two elements, it seems

that co-occurrence of words does not strongly affect the classification of Web-pages but there may exist simple dependence between words and categories because not only both the RSDR method and the SVM-FS method do not take into account the co-occurrence of words but also the linear SVM classifier achieves the better performance than SVMs with polynomial kernels for classifying Web-pages.

6. Conclusion

To meet the requirements of the automatic Web-page classification, we evaluated the effectiveness of the rough set-aided feature selection method. We found that the combination of the RSDR method and the linear SVM classifier as well as the combination of the SVM-FS method and the linear SVM classifier significantly improves the classification performance and results in acceptable accuracy by using a reduced dataset comprising only 3% of the features in the unreduced data. It is often necessary for many feature selection and feature generation methods such as the SVM-FS method, TF-IDF method and the method of generating frequent itemsets to search the respective optimal threshold values offering the highest classification performance experimentally. Instead, the RSDR technique can immediately obtain a set of relevant features without depending on any ad hoc threshold. This is quite useful and desirable in practical applications like Web-page classification.

The worst-case runtime complexity O(n!) of QUICKREDUCT may not matter to the RSDR since the RSDR-based training process is not normally invoked so often. In our experiments, it took 482 sec runtime to compute the reduct.

At present, the size of the dataset is not crucial for the SVM-FS method because the SVM-FS method is robust with respect to increase of the dataset as shown in Brank et al.'s research [2] and Sima's research [11]. For the RSDR method, more work is needed to efficiently compute the reduct. One approach would be to use an adequate subset of the whole dataset for the respective class. It is our future plan to perform such experiments.

With respect to SVMs, we have obtained the result that SVM with the linear kernel has the better performance than those of SVMs with higher order polynomial kernels. This means that using higher-order polynomial kernel for SVMs does not improve classification performance to classify Web-pages.

Although in this study, to classify Web pages over Yahoo top-level categories is focused on, our future work is to investigate an approach for hierarchical classifications of Web pages along the hierarchy of the directory-style search engines.

Acknowledgments

We would like to thank Professor Ning Zhong (Maebashi Institute of Technology, Japan) for his helpful comments.

References

[1] A. An, Y. Huang, X. Huang and N. Cercone, Feature Selection with Rough Sets for Web Page Classification, *Trans Rough Sets* (2004), 1–13.

[2] J. Brank, M. Grobelnik, N. Milic-Frayling and D. Mladenic, *Feature selection using support vector machines*, Proceedings of the Third International Conference on Data Minig Methods and Databases for Engineering, Finance, and Other Fields, 2002, 25–27.

[3] A. Chouchoulas and Q. Shen, Rough set-aided keyword reduction for text categorization, *Applied Artificial Intelligence* **15** (2001), 843–873.

[4] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, Cambridge University Press, 2000.

[5] M. Dash and H. Liu, Consistency-based search in feature selection, *Artificial Intelligence* **151** (2003), 155–176.

[6] J. Katzberg and W. Ziarko, Variable precision extension of rough sets, W. Ziarko, ed., *Fundamenta Informaticae, Special Issue on Rough Sets* **27**(2–3) (1996), 155–168.

[7] P. Lingras and C. Butz, *Interval Set Classifiers using Support Vector Machines*, Proceedings of the 2004 Conference of the North American Fuzzy Information Processing Society, Vol. 2, 2004, 707–710.

[8] Z. Pawlak, *Rough sets: Theoretical Aspect of Reasoning About Data*, Kluwer Academic Publishers, 1991.

[9] J.R. Quinlan, *C4.5: Programs for Machine Learning*, San Mateo, CA: Morgan Kaufmann, 1991.

[10] Q. Shen and A. Chouchoulas, Rough set-based dimensionality reduction for supervised and unsupervised learning, *International Journal of Applied Mathematics and Computer Science* **11**(3) (2001), 583–601.

[11] K. Shima, *Identifying Discriminative Features form High-Dimensional Data using Support Vector Machines*, Ph. D. Thesis, University of Tokyo, Tokyo, Japan, 2003.

[12] T. Tsukada, M. Washio and H. Matoda, *Automatic web-page classification by using machine learning methods*, Proceedings of the First Asia-Pacific Conference on Web Intelligence (WI 2001), LNAI 2198:303–313, 2001.

[13] T. Wakaki, H. Itakura and M. tamura, *Rough Set-Aided Feature Selection for Automatic Web-Page Classification*, Proceedings of 2004 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2004), IEEE Computer Society, 2004, 70–76.

[14] chasen. http://chasen.aist-nara.ac.jp/.

[15] TinySVM. http://cl.aist-nara.ac.jp/~taku-ku/software/tinysvm/.