社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2003－ＩＣＳ－133 (17)
2003／9／15

Feature subset selection using restriction kernels

Ken SADOHARA†

† National Institute of Advanced Industrial Science and Technology (AIST)
AIST Tsukuba Central 2, 1–1–1 Umezono, Tsukuba-shi, Ibaraki, Japan
E-mail: †ken.sadohara@aist.go.jp

Abstract This paper presents a new feature subset selection algorithm than can take into account higher order correlation between variables. The algorithm is a kind of wrapper methods using Support Vector Machines (SVMs) for learning classifiers represented as hyperplanes spanned by combinations of variables. It is known that kernel functions enable efficient learning of the high dimensional hyperplanes, while this paper considers another use of kernel functions for analyzing the learned classifiers to determine irrelevant variables. In the analysis, the algorithm computes the restriction of a classifier obtained by removing the components containing a variable, and the variable is identified as irrelevant if the restriction discriminates data as well as the classifier. Although there exist numerous components to be removed, it is shown that the restriction can be computed efficiently by using restriction kernels. It is also shown that the presented algorithm outperforms existing algorithms in empirical studies on the synthetic data sets. Furthermore, the algorithm is applied to text categorization tasks and an encouraging result is obtained.
Key words feature selection, support vector machine, kernel methods, text categorization

制限カーネルを用いた特徴選択

佐土原 健†

† 産業技術総合研究所 〒 305–8568 つくば市梅園 1–1–1 つくば中央第 2
E-mail: †ken.sadohara@aist.go.jp

あらまし 本論文は, 変数の高次相関を考慮に入れた, 新しい特徴選択アルゴリズムを提案する. このアルゴリズムは, 学習エンジンとしてサポートベクトルマシンを用いるラッパー法の一種であり, カーネル関数を用いて, 変数の組み合わせが張る高次特徴空間上で効率の良い学習を行った後, 学習された分類器を分析し分類に寄与しない変数を同定する. この分析は, 特徴空間上の超平面として記述される分類器から, ある変数を含む全て成分を取り除いて得られる分類器の制限を計算し, 制限された分類器と元の分類器の分類性能を比較することで行われる. ある特定の変数が含まれる変数の組み合わせの数は, 一般に非常に大きいが, 制限カーネルと呼ばれるカーネル関数を用いることで, 分類器の制限を効率良く計算できることが示される. さらに, 人工的なデータを用いた実験と, ニュース記事分類タスクの実データを用いた実験により, 提案する特徴選択アルゴリズムが, 既存のアルゴリズムよりも優れていることが示される.
キーワード 特徴選択, サポートベクトルマシン, カーネル法, テキスト分類

1. Introduction

For data mining, which deals with overwhelming quantity of data, the problem of focusing on the most relevant information is quite important. As a specific task of the problem, feature subset selection has been received significant attention [6], [7]. In terms of supervised inductive learning, it is the problem of selecting a small subset of variables that ideally is necessary and sufficient to predict desired output. Selecting an appropriate subset of variables is important because not merely it conserves computational resources, but also it significantly improves prediction accuracy or affords better understanding of the data.

Among a number of feature subset selection algorithms [3]～[5] presented so far, perhaps the simplest approach is to evaluate each variable individually based on its correlation with the target concept and then to select k variables with highest value. As the evaluation metric, mutual

information between the target concept and each variable is typically used. However this approach easily mistakes relevant variables for irrelevant ones because it does not take into account higher order correlations. For example, let us consider Boolean variables X_1, X_2, X_3, X_4, and a target Boolean concept $Y = X_1\overline{X_2} \vee X_2X_3$. We see that the mutual information between Y and X_2 is zero though X_2 is indispensable for Y. The relevance of X_2 to Y is correctly estimated by taking into account X_1 or X_3 at the same time.

On the other hand, wrapper methods, which use induction algorithms as a subroutine to evaluate optimality of a given subset of variables, can take into account higher order correlations to some extent with the help of the induction algorithms. For example, a wrapper method described in [3] uses a decision tree learner as the induction algorithm. Since the induction algorithm constructs decision trees by identifying sets of variables correlated with the target concept, the wrapper method can utilize information about the correlation. However, as illustrated in the literature [8], there exist the cases that the decision tree learner fail to capture the correlation because of its univariate node-splits strategy: they split nodes by a single variable most relevant to the target concept during construction of a decision tree.

To capture the higher order correlations in a more direct way, this paper considers a new feature subset selection algorithm employing kernel methods. The algorithm is a kind of wrapper methods using Support Vector Machines (SVMs) [1]. To capture higher order correlations, it uses a SVM for training classifiers represented as hyperplanes in feature spaces[†1] spanned by combinations of variables. It is known that learning of the classifiers in the high dimensional spaces can be done efficiently by using kernel functions, this paper considers another use of kernel functions for efficient analysis of learned classifiers. In the analysis, the presented algorithm computes the restriction of a classifier obtained by removing the components containing a variable, and the variable is identified as irrelevant if the restriction discriminates data as well as the original classifier. Although there exist numerous components containing to be removed, it is shown that a class of kernel functions called *restriction kernels* enables efficient computation of the restriction.

It is also shown that the presented algorithm outperforms existing algorithms in empirical studies on the synthetic data sets of randomly generated Boolean concepts. Furthermore, the algorithm is applied to a real-world data set of text categorization tasks and an encouraging result is obtained.

†1 In the literatures of SVMs, the space where SVMs train classifiers is referred to as the feature space and coordinates spanning the space is referred to as features. To avoid confusion, variables selected by feature selection are not referred to as features.

2. Support Vector Machines

Although, SVMs produce non-linear discriminant functions in a data space that discriminate positive data from negative ones, the functions are not obtained directly in the data space. Instead, SVMs learn a linear discriminant function in a feature space spanned by derived features considered as effective in the discrimination. That is, by transforming data into a space where the classification task becomes easy, SVMs simply learn hyperplanes separating the data.

Among the hyperplanes separating the data, SVMs find *maximal margin hyperplanes* $f(\boldsymbol{z}) = \langle \boldsymbol{w} \cdot \boldsymbol{z} \rangle + b = 0$ that maximize Euclidean distance from the closest datum $\boldsymbol{z}^*$ in the feature space. Since the Euclidean distance of $\boldsymbol{z}^*$ equals to $\frac{1}{\|\boldsymbol{w}\|^2}$ under the normalization such that $|f(\boldsymbol{z}^*)| = 1$, SVMs tries to minimize $\|\boldsymbol{w}\|^2$ as shown in the following specification of quadratic programming problem:

$$\begin{aligned} \text{minimize} \quad & \|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\xi_i, \\ \text{subject to} \quad & y_i f(\boldsymbol{z}_i) \geqq 1 - \xi_i \quad 1 \leqq i \leqq n \\ & \xi_i \geqq 0, \quad 1 \leqq i \leqq n, \end{aligned}$$

where each datum $\boldsymbol{x}_i$ with a class label $y_i \in \{+1, -1\}$ is mapped to $\boldsymbol{z}_i$ in a feature space. Notice that the constraint $y_i f(\boldsymbol{z}_i) \geqq 1 \quad (1 \leqq i \leqq n)$ requires linear separability of data and variables ξ_i are introduced to relax the constraint to cope with non-separable cases, which are often in practical applications. It is known that minimizing the above objective function amounts to approximately minimizing a bound of generalization error for a suitable positive constant C.

According to the optimization theory, the above problem is transformed into the following dual problem:

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \langle \boldsymbol{z}_i \cdot \boldsymbol{z}_j \rangle, \\ \text{subject to} \quad & 0 \leqq \alpha_i \leqq C \quad (1 \leqq i \leqq n), \ \sum_{i=1}^{n}\alpha_i y_i = 0. \end{aligned}$$

It is known that the above convex quadratic programming can be solved efficiently. For a solution $\alpha_1^*, \ldots, \alpha_n^*$, the maximal margin hyperplane $f^*(\boldsymbol{z}) = 0$ can be expressed in the dual representation in terms of these parameters:

$$\begin{aligned} f^*(\boldsymbol{z}) &= \sum_{i=1}^{n}\alpha_i^* y_i \langle \boldsymbol{z}_i \cdot \boldsymbol{z} \rangle + b^* \\ b^* &= y_s - \sum_{i=1}^{n}\alpha_i^* y_i \langle \boldsymbol{z}_i \cdot \boldsymbol{z}_s \rangle \text{ for some } \alpha_s^* \neq 0 \end{aligned}$$

An advantage of using the dual representation is that we can side-step a difficulty in the transformation of data from the data space into the feature space. Since the feature space

tends to have numerous features considered as potentially effective, the dimension of the feature space tends to be very high and thus it is often infeasible to explicitly map the data into the feature space. Notice that, in the dual representation, the mapping ϕ of data appears only in the form of inner products $\langle \boldsymbol{z}_i \cdot \boldsymbol{z}_j \rangle = \langle \phi(\boldsymbol{x}_i) \cdot \phi(\boldsymbol{x}_j) \rangle$. Therefore, if we have a way of computing the inner product directly as a function of the input points, i.e. $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \phi(\boldsymbol{x}_i) \cdot \phi(\boldsymbol{x}_j) \rangle$, then we can side-step the computational problem inherent in evaluating the mapping. The use of such functions K called *kernel functions* makes it possible to map the data implicitly into a high dimensional feature space and to efficiently find the optimal hyperplane in the feature space.

In addition to the use of kernel functions for the learning of classifiers with derived features, the next section considers another use of them for analysis of learned classifiers.

3. Feature selection kernels

This section considers a use of kernel functions for selecting irrelevant variables with taking into account higher order correlations. In order to capture the higher order correlations, a classifier is first trained in the feature space spanned by combinations of variables. Then by removing the combinations containing a variable X, the restriction of the classifier is obtained, and its discriminative power is compared with the classifier to determine whether X is relevant or not. The restrictions of classifiers are obtained by computing the restrictions of discriminant functions defined as follows. For a linear discriminant function $f(z_1, \ldots, z_\ell) = \sum_{i=1}^{\ell} w_i z_i + b$ on a feature space Z spanned by features $V = \{z_1, \ldots, z_\ell\}$, and for a subspace Z' spanned by features $V' \subset V$, f' is the *restriction* of f onto Z' if $f'(z_1, \ldots, z_\ell) = \sum_{z_i \in V'} w_i z_i + b$. For our examples, let us consider the feature space Z spanned by all conjunctions of X_1, X_2, X_3 and X_4. Then, it is shown that any 4-variable Boolean function can be represented as a hyperplane with zero threshold i.e. $b = 0$, in this feature space, and the maximal margin hyperplane in this space is efficiently learned by using the DNF kernel [13]. For a Boolean concept $Y = X_1\overline{X_2} \vee X_2X_3$, we see that a hyperplane

$$f(X_1, X_2, X_3, X_4) =$$
$$X_1(1 - X_2) + X_2X_3 - (1 - X_1)(1 - X_2) - X_2(1 - X_3) = 0$$

discriminates all data of Y. From this feature space Z, by removing 52 conjunctions containing X_4, e.g. $X_4, \overline{X_4}, X_4X_1, X_4\overline{X_1}, X_4X_1X_2, \ldots$, we obtain the subspace Z_{-X_4} of Z, and by using a kernel function described later, we can compute the restriction f' of f onto Z_{-X_4} efficiently. In this case, because f does not contain any term containing X_4, f' is equivalent to f and has same discriminative power as f has. Therefore, we can conclude X_4 is irrelevant to Y. On the other hand, for the subspace Z_{-X_2} obtained by removing conjunctions containing X_2, the restriction f'' of f onto Z_{-X_2} is identically zero and has no discriminative ability since all terms in f'' contain X_2. Therefore, we can conclude that X_2 is relevant to Y and cannot be removed.

Now, we consider how to compute the restrictions. The following theorem establishes the principle of the computation.

[Theorem 1] Let K be a kernel function in a feature space Z, and K' be a kernel function in a subspace Z' of Z. Then, for a linear discriminant function in Z

$$f(\boldsymbol{x}) = b + \sum_{j=1}^{n} y_j \alpha_j K(\boldsymbol{x}_j, \boldsymbol{x}),$$

its restriction f' onto Z' can be computed as

$$f'(\boldsymbol{x}) = b + \sum_{j=1}^{n} y_j \alpha_j K'(\boldsymbol{x}_j, \boldsymbol{x}).$$

The kernel K' above is referred to as a *restriction kernel*. The literature [12] uses restriction kernels in the learning of Boolean functions for reducing the length of conjunctions of learned Boolean functions. This paper considers another use of restriction kernels for feature subset selection.

Let us consider again the feature space spanned by all possible conjunctions consisting of Boolean variables. In the feature space, it is shown that the inner-product can be computed with the DNF kernel [10], [13]: $K(\boldsymbol{u}, \boldsymbol{v}) \stackrel{\text{def}}{=} -1 + 2^{\mathrm{S}(\boldsymbol{u}, \boldsymbol{v})}$, where $\mathrm{s}(\boldsymbol{u}, \boldsymbol{v})$ denotes the number of bits that have the same value in $\boldsymbol{u}$ and $\boldsymbol{v}$. This is because non-zero-valued conjunctions must agree with both $\boldsymbol{u}$ and $\boldsymbol{v}$, and the number of such conjunctions except the empty conjunction is exactly $-1 + 2^{\mathrm{S}(\boldsymbol{u}, \boldsymbol{v})}$. In a similar argument, we see that for the subspace obtained by removing conjunctions containing variables $X_1, \ldots, X_m$, the inner-product in the subspace can be computed by using the following kernel function:

[Definition 1] (feature selection kernels for the DNF kernel)
$K^M(\boldsymbol{u}, \boldsymbol{v}) \stackrel{\text{def}}{=} -1 + 2^{\mathrm{S}(\boldsymbol{u}^{-M}, \boldsymbol{v}^{-M})}$,
where $\boldsymbol{u}^{-M}$ denotes the vector obtained by removing components corresponding to $X_1, \ldots, X_m$ from $\boldsymbol{u}$.

These arguments above are also valid for other Boolean kernels such as k-DNF kernel [10], [12] or the monotone k-DNF kernel [10], [12]. Furthermore, whereas the Boolean kernels are applicable only for nominal variables, we can define restriction kernels for the polynomial kernels that can be applicable for continuous variables.

Using the feature selection kernels, a new feature selection algorithm FERK is devised and shown in Table 1.

The distance d between f and f' captures the difference on their discriminative ability, and is defined as follows. For a discriminant function $f(\boldsymbol{x}) = b + \sum_j^n y_j \alpha_j K(\boldsymbol{x}_j, \boldsymbol{x})$, the distance d between f and its restriction $f'(\boldsymbol{x}) = b +$

0. Let V be a set of variables. Let K^M be a feature selection kernel insensitive to variables in M.
1. $M \stackrel{\text{def}}{=} \emptyset$, $R \stackrel{\text{def}}{=} V$.
2. Train a SVM using K^M and obtain a discriminant function f.
3. Analyze the learned discriminant function f.

For each $v \in R$, compute the restriction f' of f using $K^{M \cup \{v\}}$, and measure the distance d between f and f'. Let v be the variable that yields the minimum distance.

4. If the stopping condition holds then terminates and output R.
5. $M \stackrel{\text{def}}{=} M \cup \{v\}$, $R \stackrel{\text{def}}{=} R \setminus \{v\}$. Go to 2.

Table 1 A feature subset selection algorithm FERK

$\sum_j^n y_j \alpha_j K'(\boldsymbol{x}_j, \boldsymbol{x})$ is computed as follows:

$$
\begin{aligned}
d(f, f') &\stackrel{\text{def}}{=} \sum_i^n y_i \left(f(\boldsymbol{x}_i) - f'(\boldsymbol{x}_i) \right) \\
&= \sum_i^n y_i \left(\sum_j^n y_j \alpha_j K(\boldsymbol{x}_j, \boldsymbol{x}_i) - \sum_j^n y_j \alpha_j K'(\boldsymbol{x}_j, \boldsymbol{x}_i) \right) \\
&= \sum_i^n \sum_j^n y_i y_j \alpha_j \left(K(\boldsymbol{x}_j, \boldsymbol{x}_i) - K'(\boldsymbol{x}_j, \boldsymbol{x}_i) \right).
\end{aligned}
$$

Concerning the stopping condition, there may exist various candidates depending on the purpose of feature selection. In the empirical study described in the section 5.1, FERK stops when the number of the remaining variables reaches a given threshold. On the other hand, in the empirical study described in the section 5.2, FERK stops when the accuracy of f' on the training data becomes lower than that of f.

Finally, we should notice that FERK is extendable to remove more than one variable at each iteration in order to reduce computational cost. In fact, an extended FERK is applied to a text categorization task with thousands of variables in the section 5.2, and it is shown that removing multiple variables at each iteration saves computational costs.

4. Related works

Among a wealth of feature subset selection algorithms, perhaps the simplest algorithm is to evaluate each variable individually based on its correlation with the class variable and then to select k variables with highest value. As the evaluation metric, mutual information between the class and each variable is typically used. Since this method, denoted by MINFO, is computationally efficient, it is used for tasks with high dimensional data e.g. text categorization.

RELIEF [5] is another method that evaluates each variable individually, but its evaluation is done in a different way. RELIEF samples data randomly from the training data and updates relevance weights of each variable based on the difference between the selected data and the two nearest data of the same and opposite class.

Compared with these methods, recent works (e.g. [6]) show that wrapper methods perform better. In wrapper methods, a feature subset selection algorithm exists as a wrapper around an induction algorithm and uses it as a subroutine rather than as a post processor.

For example, in the literature [3], a wrapper method using decision tree learners e.g. C4.5 [9], is presented. It use the decision tree learners to estimate the accuracy of the learned classifier using only a given subset of variables. Based on the estimate for each subset of variables, it conducts greedy search for an optimal subset of variables.

In contrast to those methods that evaluate the relevance of each variable individually, the wrapper method can take into account higher order correlations to some extent with the help of decision tree learners. However, as illustrated in the literature [8], the decision tree learners have a risk of failing to capture the relevance of variables in the case that they are not relevant by themselves but they are relevant when other variables' values are known. This is because of their univariate node-splits strategy: they split nodes by a single variable most relevant to class membership during construction of a decision tree.

On the other hand, FERK takes into account higher order correlations by using hyperplanes spanned by combinations of variables. While this could be an advantage of FERK, the wrapper method also have an advantage. The method evaluates each subset of variables in a more elaborate way: it re-learns for every subset of variables and estimates accuracy of the learned classifier using the n-fold cross validation. The evaluation scheme is expected to make a more precise evaluation although it requires high computational cost and limits its application to data sets with many variables.

The use of kernel methods for feature subset selection also appears in [4]. The algorithm describe in the paper, denoted by SVM RFE, is similar to FERK. The difference is the way of evaluation of each subset of variables. FERK evaluates the discriminative ability of the restriction for a subset of variables, while SVM RFE evaluates the magnitude of coefficients of features containing variables in the subset.

5. Empirical studies

In this section, FERK is compared with the feature subset selection algorithms described in the previous section. To compare them in various aspects, experiments on synthetic data of randomly generated Boolean concepts are performed. Moreover, to show the applicability of FERK to real world data, experiments on classification of news articles are done.

5.1 Learning of Boolean concepts

FERK is compared with MINFO, RELIEF, SVM RFE and a wrapper method using C4.5 denoted by WCBE. WCBE uses the backward elimination of variables and performs 10-

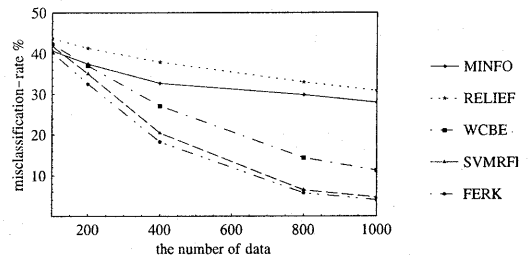

Figure 1 The error rates vs. the number n of data.

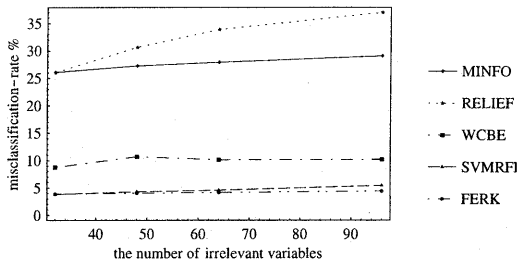

Figure 2 The error rates vs. the number r of irrelevant variables.

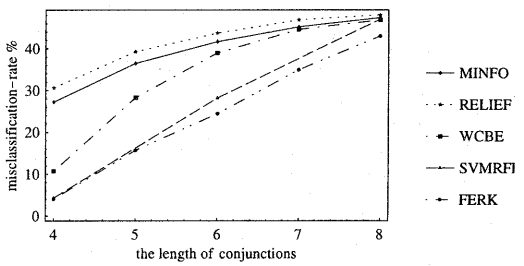

Figure 3 The error rates vs. the length ℓ of conjunctions.

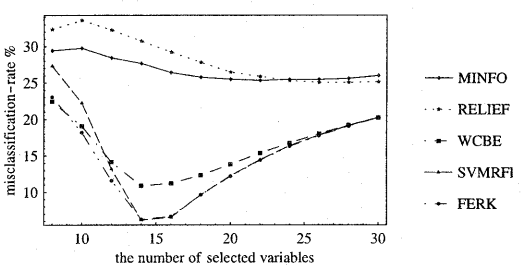

Figure 4 The error rate vs. the number k of selected variables.

fold cross-validation to estimate the accuracy of each classifier. Each of SVM RFE and FERK uses a Boolean Kernel Classifier (BKC) [12] as an inductive learning engine, where the BKC uses the DNF kernel with its capacity control capability. It should also be mentioned that we use a deterministic version of RELIEF that uses all instances, all near-hits and all near-misses of each instance.

Data sets used in the following experiments are synthetic ones obtained from randomly generated Boolean concepts. In the generation of the Boolean concepts, the following parameters are varied: the number r of irrelevant Boolean variables, the number n of training data, the complexity of the target Boolean concepts defined by the length ℓ of conjunctions. In a certain parameter setting, 160 different Boolean concepts generated as follows. First, a Boolean concept is generated in DNF using a fixed 16 variables. The DNF formula consists of conjunctions of randomly generated ℓ variables negated with probability $\frac{1}{2}$. The number of the conjunctions is set to $2^{\ell-1}$ so as to produce approximately equal numbers of positive and negative data. Then, for each Boolean concept, n training data and 2000 test data with dimension $16 + r$ are independently drawn from the uniform distribution. The training data are then fed to the feature subset selection algorithms and k variables are selected, where k is determined by the generated DNF formula in some experiments, or is explicitly controlled in the other experiments. From the selected variables and the training data, BKC with ℓ-DNF kernel learns a classifier and its misclassification rate is measured on the test data. Finally, The rate is averaged across 160 different Boolean concepts.

Figure 1 describes the result of the experiment varying n when $\ell = 4$, $r = 48$ and k is set to the number of variables appearing in each Boolean concept. Figure 2 describes the result of the experiment varying r when $\ell = 4$, $n = 1000$ and k is set to the number of variables appearing in each Boolean concept. Figure 3 describes the result of the experiment varying ℓ when $r = 48$, $n = 1000$ and k is set to the number of variables appearing in each Boolean concept.

In all of the experiments described above, the number k of variables selected by each feature selection algorithm is set to the correct value, i.e. the number of variables appearing in each DNF formula. Although the setting is for the purpose to clearly see the influence of other parameters, it is impractical to use the size of the correct feature subset. Therefore another experiment is performed to see the influence of the size of feature subsets. Figure 4 describes the result of the experiment varying the number k of variables selected by each feature subset selection algorithm when $r = 48$, $n = 1000$ and $\ell = 4$. The average number of variables appearing in each DNF formula is 14.5. We see that SVM RFE and FERK have sharp downward peaks around this value.

5.2 Text categorization

In order to investigate the applicability of FERK to real world data sets, experiments on the Reuter-21578 collection are performed. It is a collection of newswire stories classified under categories related to economics, and is widely used in text categorization [14]. The following experiments

category	method	dimension	accuracy(%)	BEP(%)	k
money		1,000	85.1	81.0	2
money	FERK	**999**	**85.5**	**81.5**	2
trade		1,000	92.8	82.0	3
trade	FERK	**310**	**93.2**	**84.6**	3
trade	MINFO	310	91.8	78.0	2
interest		1,000	90.0	68.4	8
interest	FERK	1,000	90.0	68.4	8
gnp		1,000	96.6	61.5	2
gnp	FERK	**298**	**97.2**	**61.5**	2
gnp	MINFO	298	95.6	53.8	1
cpi		1,000	98.6	88.2	1
cpi	FERK	1,000	98.6	88.2	1

Table 2 The result of experiments on Reuter-21578.

uses a date set named "re0" from pre-processed data sets of Reuter-21578 provide by G.Forman [2]. The data set contains 1504 stories, and each story is represented as a 2886 dimensional binary vector associated with a category. From the data set, 150 vectors are set aside for validation, and the remaining 1354 vectors are used for training. For computational reason, these vectors are further pre-processed by MINFO and converted into 1000 dimensional binary vectors. The preprocessed training data set is then fed into FERK that uses feature selection kernels for the monotone k-DNF kernel. Since the dimension of the data is still high, FERK is modified so as to remove more than one variable according to the dimension at each iteration. For the dimension d, the modified FERK removes $10^{\lfloor \log_{10} d-1 \rfloor}$ variables. During the iterations, if the restriction is less accurate than the learned classifier, FERK stops and outputs the remaining variables. From the variables and the training data, BKC [12] using the monotone k-DNF kernel learns a classifier. Then, its accuracy and its Precision/Recall Break Even Point (BEP) [14] are measured. In this way, experiments are done for the five most frequent categories. Table 2 describes the result of the experiments, where the rows with the empty method represent experiments without feature selection. We see that FERK does not remove any variable for two categories interest and cpi. But notice that according to the conservative stopping condition, FERK does not degrade accuracy and BEP. On the other hand, for categories money, trade and gnp, FERK reduces the dimension of each data set, and yields higher accuracy and higher BEP compared with experiments without feature selection. Especially, for trade and gnp, FERK removes a considerable number of variables. To see the appropriateness of the feature set selected by FERK, additional experiments using MINFO are performed for trade and gnp. In these experiments, by using MINFO, the dimension of each data set is reduced to the same dimension as FERK. As the result, MINFO performs worse than FERK, and therefore it seems that FERK selects a more appropriate feature subsets than MINFO does.

6. Conclusion

This paper presented a new feature subset selection algorithm that can take into account higher order correlations between variables. To identify irrelevant variables, the algorithm analyzes learned classifiers represented as hyperplanes spanned by combinations of the variables. In the analysis, it computes the restriction of a classifier obtained by removing components containing a variable, and the variable is identified as irrelevant if the restriction discriminates data as well as the classifier. Although there exist numerous components to be removed, it was shown that feature selection kernels enable efficient computation of the restriction. Furthermore, empirical studies using synthetic data sets showed that the presented algorithm outperforms several existing algorithms. Finally the algorithm was also applied to text categorization task and an encouraging result is obtained.

Acknowledgment

We would like to thank George Forman for the prepared datasets Reuters-21578. We also extend our thanks to the WEKA project for their open-source machine learning software. This work is partially supported by Grant-in-Aid for Young Scientists (B) (No.14780315) from the Ministry of Education, Culture, Sports, Science and Technology (MEXT).

References

[1] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines.* Cambridge Press, 2000.

[2] G. Forman. An extensive empirical study of feature selection metrics for text classification. *JMLR*, 1289–1305, 2003.

[3] G.H.John et al. Irrelevant features and the subset selection problem. *Proc. of ICML*, 121–129, 1994.

[4] I. Guyon et al. Gene selection for cancer classification using support vector machines.*Machine Learning*,389–422,2002.

[5] K. Kira and L.A. Rendell. A practical approach to feature selection. *Proc. of ICML*, 249–256, 1992.

[6] L.C.Molina et al. Feature selection algorithm: a survey and experimental evaluation. *Proc. of ICDM*, 306–313, 2002.

[7] H. Liu and H. Motoda. *Feature extraction construction and selection.* Kluwer Academic Publishers, 1998.

[8] J.R. Quinlan. An empirical comparison of genetic and decision-tree classifiers. *Proc. of ICML*, 135–141, 1988.

[9] J.R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, 1993.

[10] R.Khardon, D.Roth, and R.Servedio. Efficiency versus convergence of Boolean kernels for on-line learning algorithms. *NIPS*, volume 14, 423–430, 2002.

[11] K. Sadohara. Feature subset selection using restriction kernels. Technical Report AIST02-J00030-3, 2003.

[12] K. Sadohara. On a capacity control using Boolean kernels for the learning of Boolean functions. *Proc. of ICDM*, 410–417, 2002.

[13] Ken Sadohara. Learning of Boolean functions using support vector machines. *Proc. of ALT*, pages 106–118, 2001.

[14] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.