Information Sciences 223 (2013) 102–118

Contents lists available at SciVerse ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Feature subset selection using separability index matrix

Jeong-Su Han [a], Sang Wan Lee [b,*], Zeungnam Bien [c]

[a] Samsung Electronics, Suwon, Republic of Korea
[b] Computation & Neural Systems, Behavioral & Social Neuroscience, California Institute of Technology, Pasadena, CA, USA
[c] School of Electrical and Computer Engineering, Ulsan National Institute of Science and Technology (UNIST), Ulsan, Republic of Korea

ARTICLE INFO

Article history:
Received 1 July 2009
Received in revised form 23 November 2011
Accepted 26 September 2012
Available online 10 October 2012

Keywords:
Feature subset selection
Filter method
Separability index matrix
EMG signal
Gait phase recognition

ABSTRACT

Effective Feature Subset Selection (FSS) is an important step when designing engineering systems that classify complex data in real time. The electromyographic (EMG) signal-based walking assistance system is a typical system that requires an efficient computational architecture for classification. The performance of such a system depends largely on a criterion function that assesses the quality of selected feature subsets. However, many well-known conventional criterion functions use less relevant features for classification or they have a high computational cost. Here, we propose a new criterion function that provides more effective *FSS*. The proposed criterion function, known as a *separability index matrix* (*SIM*), provides features pertinent to the classification task and a very low computational cost. This new function produces to a simple feature selection algorithm when combined with the forward search paradigm. We performed extensive experimental comparisons in terms of classification accuracy and computational costs to confirm that the proposed algorithm outperformed other filter-type feature selection methods that are based on various distance measures, including inter–intra, Euclidean, Mahalanobis, and Bhattacharyya distances. We then applied the proposed method to a gait phase recognition problem in our EMG signal-based walking assistance system. We demonstrated that the proposed method performed competitively when compared with other wrapper-type feature selection methods in terms of class-separability and recognition rate.

© 2012 Elsevier Inc. All rights reserved.

1. Introduction

Feature subset selection (FSS) is a process for identifying n most informative features $V = \{\upsilon_1, \ldots, \upsilon_n\}$ from N known features $S = \{x_1, x_2, \ldots, x_N\}$ ($n < N$), assuming that the C target classes $W = \{w_1, w_2, \ldots, w_C\}$ are given and an observed dataset described by M samples (instances) is available. The selection of the most informative feature set leads to an improvement in classification accuracy, faster and more cost-effective classification performance, and a better understanding of the underlying process of the observed dataset [11,19].

It would not be an overstatement to claim that FSS is the most important step when designing an engineering system because to its necessity for online classification of complex data. The electromyographic (EMG) signal-based walking assistance system is a typical system that requires an efficient computational architecture with effective *FSS*. Previous reports of EMG signal applications to lower body movements [4,8,10,16,17], suggest that the limited computational capacity of hardware and the high computational costs of the recognition task are crucial determinants of success when developing a walking assistance system [9]. Many factors affect the computational costs (e.g., input dimensionality, and the complexity of a

* Corresponding author.
E-mail address: swlee@caltech.edu (S.W. Lee).

0020-0255/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ins.2012.09.042

feature extraction and classification) but the size of a selected feature subset is of great importance mainly because the feature subset size dictates the memory allocation, which is the main bottleneck in mobile computing systems.

The division of effective FSS into a *filter* method and a *wrapper* method [19] is a key paradigm that provides a functional difference when evaluating the quality of a selected feature subset. In the filter method, the criterion function utilizes quantitative information such as the interclass distance of selected features [11]. However, the criterion function used by the wrapper method relies on performance metrics for the classifier such as accuracy, specificity, and precision.

The wrapper method often performs better at classification compared with the filter method, but it requires significantly higher computational costs because the fitness evaluation of a subset requires cross-validation or a bootstrapping procedure during the error estimation for each subset [19,23]. Furthermore, the choice of the classifier inevitably biases the characteristics of the selected feature subset, which often leads to the loss of any generalization capability [15].

Various criterion functions have been used for the filter method and they can be categorized into two groups, i.e., *distance-based* measures and *relation-based* measures. The Fisher ratio [3,26], Mahalanobis distance [5], and Bhattacharyya distance [5] are typical examples of distance-based measures that use a distance metric to measure class separability. These measures assign an average value of separability to classes for each feature. Thus, they may select features that are highly correlated if the selected features have a large average separability value. Relation-based measures are represented by correlation, mutual information [1,23], or fuzzy dependency based on a penalized Euclidean distance [7]. They extract the relation between features and classes. Using these approaches, a good feature is highly correlated within classes, but uncorrelated with other features. However, there are high computational costs when acquiring good features using these measures, especially when mutual information is involved, although some heuristics can reduce this cost [1,13,14,27].

We propose a computationally efficient criterion function based on a novel concept of a "separability index matrix (SIM)". The proposed method effectively distinguishes relevant features from irrelevant and/or redundant features.

This paper is organized as follows: In Section 2, we review various well-known criterion functions that are typically used by the filter method. In Section 3, we explain the concept of the separability index matrix (SIM) and we propose a new criterion function for effective FSS. Section 4 details the SIM-based feature selection method (SIMF) and its properties. Section 5 provides extensive experimental results, which are compared with benchmark datasets. We demonstrated the validity of the proposed method in a realistic situation and we present its application to an EMG signal-based gait phase recognition problem in Section 6. Concluding remarks are provided in Section 7.

2. Criterion functions of a filter method

Choosing a good feature subset is an important step in pattern classification tasks. A specific criterion function is required to evaluate the quality of a selected feature subset. In particular, it is always possible to find an optimal feature subset using the Branch-and-Bound technique if a monotonic criterion function is provided [5,25]. In general, the overall performance of FSS depends largely on the selected criterion function.

Various criterion functions have been reported, such as interclass distance [11], statistical dependence [19], and information-theoretic measures [1,23]. These measures can be categorized into two groups depending on the method they use to evaluate the quality of feature subsets. We refer to these groups as "*distance-based* measures" and "*relation-based* measures".

Distance-based measures characterize the quality of a feature based on its ability to discriminate instances of a class from instances of other classes. Instances from different classes (between-class) should have feature values that are more distinctive than values from the same class (within-class). Measures such as the Fisher ratio, Euclidean distance, Mahalanobis distance, and Bhattacharyya distance are representative examples in this group.

For example, it is well-known that the Fisher ratio is defined as the ratio of the between-class difference to the within-class spread [3] as follows:

$$\lambda_{i,j,l} = \frac{(m_{i,l} - m_{j,l})^2}{(\sigma_{i,l}^2 + \sigma_{j,l}^2)} \tag{1}$$

where $m_{i,l}$, $m_{j,l}$, $\sigma_{i,l}^2$, and $\sigma_{j,l}^2$ are the means and variances of instances of the ith class and jth class in the direction of the lth feature, while $\lambda_{i,j,l}$ indicates the class separation degree between the ith class and jth class in the direction of the lth feature. The Fisher ratio provides a good measure of class separability because it increases as the between-class difference increases and as the within-class spread decreases. The following generalized Fisher ratio [12] can be used in the case of a multi-class problem:

$$\lambda_l = \frac{1}{C(C-1)} \frac{\sum_{i=1}^{C}\sum_{j=1}^{C} \phi_i \phi_j \lambda_{i,j,l}}{\sum_{i=1}^{C}\sum_{j=1}^{C} \phi_i \phi_j}, \quad i \neq j. \tag{2}$$

Here, λ_l is the average class separability measure in the direction of the lth feature and C is the total number of classes, while i and j indicate the class indices with $1 \leqslant i, j \leqslant C$. Also, φ_i and φ_j are the mixing weights for the ith and jth class, respectively, and $\lambda_{i,j,l}$ is the Fisher ratio defined in (1).

The rationale for devising relation-based measures is that a good feature subset contains features that are highly correlated with classes, but uncorrelated with other features. The relevance of this approach is usually characterized in terms of

correlation or mutual information [1,23]. Mutual information is more widely used to define the dependency of variables than the correlation because the output of the correlation measure is limited to linear dependency.

If we know two variables for the kth feature x_k and ith class w_i, their mutual information [3] is defined in terms of their probabilistic density functions $p(x_k)$, $p(w_i)$, and $p(x_k, w_i)$ as follows:

$$I(x_k; w_i) = I(w_i; x_k) = \sum_{w_i} \int p(x_k, w_i) \log \frac{p(x_k, w_i)}{p(w_k)p(w_i)} dx_k \tag{3}$$

If one of the variables x_k and w_i is continuous, it is difficult to compute their mutual information because calculation of the integral is often a complex task when only a limited number of instances is available. At least one of the variables is typically continuous in many realistic classification problems. Furthermore, as the number of the selected features increases, the task of obtaining mutual information often entails significantly high computational costs due to the requirement of estimating multivariate densities $p(x_1, \ldots, x_n)$ and $p(x_1, \ldots, x_n, w_i)$, which may become an ill-posed problem with large n. As in [14], one can adopt a density estimation method as a preprocessing step to efficiently compute mutual information such as the 'Parzen windows' technique. Relation-based measures generally have the major drawback of high computational costs, so they can be inefficient when handling a large number of features, although relation-based measures may provide more relevant features and remove redundant features more effectively when compared with distance-based measures.

However, distance-based measures can be utilized with relatively low computational costs when compared with relation-based measures. It is difficult to remove redundant features and there is a tendency to select less relevant features because these distance-based measures typically utilize the average (or sum) of the separability for classes. For example, the selected feature shown in Fig. 1 assumes a large distance-based measure if there is a relatively high separability between certain pairs of classes. Most separability measures used in this group find that $\{x_1, x_2\}$ in case (a) is a better feature combination than $\{x_1, x_3\}$ in case (b), because of their high separability values. However, feature set $\{x_1, x_3\}$ in case (b) is a more relevant subset because it is associated with a lower classification error than case (a) as shown in Table 1. Classification errors usually occur in pairs of classes that have low separability.

The total classification error is the sum of classification errors for every distinct pair of classes. Each error found in every distinct pair of classes is a function of the separability between associated classes. This implies that each classification error can be reduced only by increasing the separability of the corresponding classes. For example, the highly separable state between class 1 and class 2 of feature x_1 in Fig. 1 does not help to reduce the classification error between classes 2 and 3. Therefore, the quality of a feature must be evaluated based on its ability to classify every distinct pair of classes. We can determine

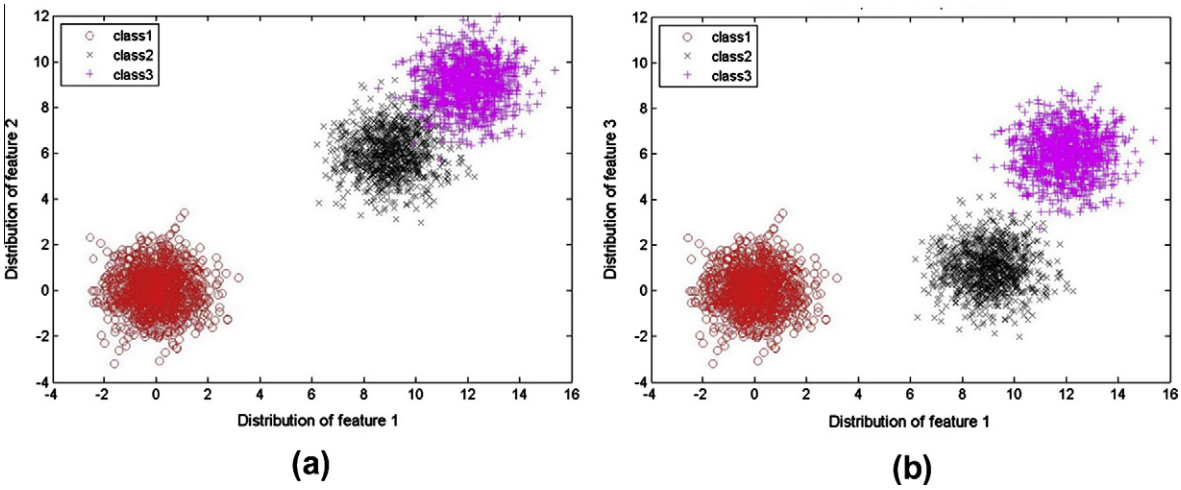

Fig. 1. (a) Scatter plot of features x_1 and x_2; (b) scatter plot of features x_1 and x_3. Each dataset was generated by sampling the Gaussian distribution: $N([000], 1^2)$ for class 1, $N([960.5], 1^2)$ for class 2, and $N([1296], 1^2)$ for class 3.

Table 1
Classification percentage errors obtained using various classifiers. Note that the average classification error (%) and standard deviation were averaged over 30 trials using a random sub-sampling method.

	Feature subset $\{x_1, x_2\}$	Feature subset $\{x_1, x_3\}$
k-Nearest neighbor classifier	1.1844 ± 0.2553	0.1333 ± 0.0678
Parzen classifier	1.2111 ± 0.2290	0.1000 ± 0.1222
Naive Bayes classifier	1.6533 ± 0.2781	0.3644 ± 0.2149

the classification capability of a certain feature by comparing the separability values of every pair of classes for a certain feature with the separability values of the corresponding pairs of classes for all available features. As explained in the next section, our proposed notion of a "separability index matrix" will be used to provide a classification method for a certain feature for every distinct pair of classes.

3. New criterion function based on a separability index matrix (SIM)

In this section, we define the notion of the "classifiability of a feature" as a new criterion function based on the concept of a separability index matrix (SIM). To achieve this purpose, we introduce new notations and operations for matrix calculation.

Definition 1. *Element-wise operations of matrices* Let C be a given integer, which denotes the total number of classes. For $C \times C$ matrices $A = [a_{i,j}]_{C\times C}$, $B = [b_{i,j}]_{C\times C}$, $P = [p_{i,j}]_{C\times C}$, and $Q = [q_{i,j}]_{C\times C}$ where $a_{i,j}$, $b_{i,j} \in R$ (the set of real numbers) and $p_{i,j}$, $q_{i,j} \in \{0,1\}$, $1 \leqslant i, j \leqslant C$, we define various element-wise operations of matrices denoted by $\otimes$, $\oslash$, $\oslash\!\!\vee$, $\oslash\!\!\wedge$, $\oslash\!\!\sim$, and $\oslash\!\!\geqslant$, as follows:

$$A \otimes B = [r_{i,j}]_{C\times C} \text{ where } r_{i,j} = a_{i,j} \times b_{i,j} \quad \text{(multiplication)}$$
$$A \oslash B = [r_{i,j}]_{C\times C} \text{ where } r_{i,j} = a_{i,j} \div b_{i,j} (b_{i,j} \neq 0) \quad \text{(division)}$$
$$P \oslash\!\!\vee Q = [r_{i,j}]_{C\times C} \text{ where } r_{i,j} = p_{i,j} \vee q_{i,j} \quad \text{(logical OR operation)}$$
$$P \oslash\!\!\wedge Q = [r_{i,j}]_{C\times C} \text{ where } r_{i,j} = p_{i,j} \wedge q_{i,j} \quad \text{(logical AND operation)}$$
$$\oslash\!\!\sim P = [r_{i,j}]_{C\times C} \text{ where } r_{i,j} = \sim p_{i,j} \quad \text{(logical NOT operation)}$$

$$A \oslash\!\!\geqslant B = [r_{i,j}]_{C\times C} \text{ where } r_{i,j} = \begin{cases} 1, & a_{i,j} \geqslant b_{i,j} \\ 0, & a_{i,j} < b_{i,j} \end{cases} \quad \text{(COMPARE operation)}$$

Definition 2. Separability Degree Matrix (SDM) Let C be the total number of classes. For $1 \leqslant i, j \leqslant C$ let w_i and w_j be the ith and jth class, respectively. Let there be given a criterion function $J(\bullet)$ of distance-based measure type, such as the Bhattacharyya distance [5]. For a feature x_k, we denote $J(w_i, w_j; \{x_k\})$ as the separability (or, distance) value between class w_i and class w_j when the specific criterion function $J(\bullet)$ is applied and the feature x_k is used. Then the separability degree matrix (SDM) is defined as follows:

$$SDM_k = [J(w_i, w_j; \{x_k\})]_{C\times C} \tag{4}$$

Each element of SDM_k (the separability degree matrix) represents a separability value of a pair of all distinct classes evaluated by using the feature x_k. As an example, we can use the generalized Fisher ratio in Eq. (2) to form the SDM as follows:

$$SDM_k = \begin{bmatrix} \lambda_{1,1,k} & \lambda_{1,2,k} & \cdots & \lambda_{1,C,k} \\ \lambda_{2,1,k} & \lambda_{2,2,k} & \cdots & \lambda_{2,C,k} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{C,1,k} & \lambda_{C,2,k} & \cdots & \lambda_{C,C,k} \end{bmatrix} \tag{5}$$

The definition of the SDM also can be applied to a feature subset. Let $V = \{x_1, \ldots, x_n\}$ be a selected feature subset in the feature selection process. We can obtain SDM_V as follows:

$$SDM_V \leftarrow [J(w_i, w_j; V)]_{C\times C} \tag{6}$$

In this case, each element of SDM_V represents a separability value of a pair of all distinct classes not in a single feature direction but in V feature space.

In Eq. (4), it is obvious that $J(w_i, w_j; \{x_k\}) = 0$, $J(w_i, w_j; \{x_k\}) = J(w_j, w_i; \{x_k\})$, and therefore $SDM_k = SDM_k^T$. Here, A^T means the transpose of matrix A. Thus, we shall omit the entries below the diagonal of the matrix SDM_k and show only the upper triangular elements of each matrix for notational simplicity.

Example 1. We calculate SDM (separability degree matrix) of the example dataset shown in Fig. 1 which has Gaussian distribution with the class mean $m_1 = [0 \quad 0 \quad 0]$, $m_2 = [9 \quad 6 \quad 0.5]$, $m_3 = [12 \quad 9 \quad 6]$, and unit variance. In further examples, we will use the Bhattacharyya distance [5] as a criterion function $J(\bullet)$. The Bhattacharyya distance is defined as follows:

$$BD = \frac{1}{8}(m_2 - m_1)^2 \left[\frac{\sigma_1^2 + \sigma_2^2}{2}\right]^{-1} + \frac{1}{2}\ln \frac{\left|\frac{\sigma_1^2+\sigma_2^2}{2}\right|}{\sqrt{|\sigma_1^2||\sigma_2^2|}}.$$

Let us calculate each entity of SDM_1.

$$SDM_1(1,2) = J(w_1,w_2;\{x_1\}) = \frac{1}{8}(0-9)^2\left[\frac{1^2+1^2}{2}\right]^{-1} + \frac{1}{2}\ln\frac{\left|\frac{1^2+1^2}{2}\right|}{\sqrt{|1^2||1^2|}} = \frac{1}{8}(0-9)^2 = 10.125,$$

$$SDM_1(1,3) = J(w_1,w_3;\{x_1\}) = \frac{1}{8}(0-12)^2 = 18,$$

$$SDM_1(2,3) = J(w_2,w_3;\{x_1\}) = \frac{1}{8}(9-12)^2 = 1.125.$$

By calculating other SDM_2 and SDM_3, we can obtain that

$$SDM_1 = \begin{bmatrix} 0 & 10.125 & 18 \\ & 0 & 1.125 \\ & & 0 \end{bmatrix}, \quad SDM_2 = \begin{bmatrix} 0 & 4.5 & 10.125 \\ & 0 & 1.125 \\ & & 0 \end{bmatrix}, \text{ and } SDM_3 = \begin{bmatrix} 0 & 0.03 & 4.5 \\ & 0 & 3.78 \\ & & 0 \end{bmatrix}$$

Note that we omit entries in the lower half of the *SDM*, which is symmetric.

After observing the *SDM* for each feature, we find that feature x_1 and x_2 have similar classification contents because they have big separability values in {class1,class2} and {class1,class3} but small values in {class2,class3}. Feature x_3 provides a relatively larger separability value in {class 2,class 3} than other features.

The degree of relevance of a feature (with regard to classifiability) can be expressed as a separability value of any criterion function. Without loss of generality, we use the term "*relevant feature*" to mean a feature with big separability values in comparison with other features. If a feature is a relevant one in certain pairs of classes, we expect that this feature gives fewer classification errors in those classes than other features. This type of relevant feature is *between-class separable*on the corresponding classes. The between-class separability of a certain feature can be either computed by comparing the classification accuracy of a specific classifier with that of other features or estimated by its separability value of related classes. Therefore the most relevant feature is a feature which has as many between-class separable cases as possible. By using the separability index matrix (SIM) defined below, we can extract such information systematically.

Definition 3. Separability Index Matrix (SIM) For a given SDM_k, $1 \leqslant k \leqslant N$, where N is the total number of features, let $SDM_{avg} = \frac{1}{N} \times (\sum_k SDM_k)$. For $\Delta > 0$, let $SDM_\Delta = \Delta \times [1_{i,j}]_{C\times C}$. Here Δ is a threshold value given by the designer. Then the separability index matrix (SIM) of feature x_k is defined as follows:

$$SIM_k^\Delta = [r_{i,j}]_{C\times C} = (SDM_k \,\overline{\gtrsim}\, SDM_{avg}) \,\overline{\vee}\, (SDM_k \,\overline{\gtrsim}\, SDM_\Delta) \tag{7}$$

From the properties of SDM_k, we note that $r_{i,i} = 0$, $r_{i,j} = r_{j,i}$, and $SIM_k = SIM_k^T$. Recalling the definition of operation $\overline{\gtrsim}$, we also note that each elements of SIM_k has a binary value {0,1}. Each element with '1' indicates that the corresponding classes are *between-class separable* and '0' means that they are *not between-class separable*. By virtue of the notion of the SIM, we can easily distinguish relevant features from irrelevant and/or redundant features with respect to previously selected features.

We suggest that the value of Δ is selected between the minimum of each SDM_k and the maximum of each SDM_k, which would render a between-class separable condition of SIM_k.

Example 2. *For each SDM_k $k = 1, 2$ and 3 in Example 1, the minimum value is 1.125, 1.125, 0.03, while the maximum value is 18, 10.125, and 4.5, respectively. We set $\Delta = 3$. Then,* the SIM of each feature in Example 1 is calculated as follows.

Since

$$SDM_{avg} = \begin{bmatrix} 0 & 4.9 & 11.1 \\ & 0 & 1.8 \\ & & 0 \end{bmatrix} \quad \text{and} \quad SDM_{\Delta=3} = \begin{bmatrix} 3 & 3 & 3 \\ & 3 & 3 \\ & & 3 \end{bmatrix},$$

$$SIM_1^{\Delta=3} = \begin{bmatrix} 0 & 1 & 1 \\ & 0 & 0 \\ & & 0 \end{bmatrix}, \quad SIM_2^{\Delta=3} = \begin{bmatrix} 0 & 1 & 1 \\ & 0 & 0 \\ & & 0 \end{bmatrix}, \quad \text{and} \quad SIM_1^{\Delta=3} = \begin{bmatrix} 0 & 1 & 1 \\ & 0 & 0 \\ & & 0 \end{bmatrix}$$

Feature x_1 can be used to classify {class1, class2} and {class1, class3} by scrutinizing each element of SIM_1^Δ. Feature x_2 presents the same classification knowledge of feature x_1; therefore one of them does not need to be selected for pattern classification. As shown in Fig. 1, feature x_3 shows complementary classification information in {class 2, class 3}, which is also revealed well in SIM_3^Δ.

Definition 4. Classifiability For the given $SIM_k^\Delta, 1 \leqslant k \leqslant N$ with N being the total number of features, let WM_k be a matrix given by $WM_k = SIM_k^\Delta \,\overline{\wedge}\, \left(\sum_{i=1}^N SIM_i^\Delta\right)$. Then the function defined as

$$G(x_k) = \sum_i \sum_j \left(SIM_k^{\Delta} \otimes WM_k \right) \tag{8}$$

is called the classifiability of a feature x_k and will serve as the new criterion function.

WM_k is called the weight matrix for k-th feature x_k. The weight matrix represents the relative importance of each feature with respect to all available features. Note that a conventional distance-based criterion function tells how separate the feature distribution is, whereas the new criterion function of classifiability reveals classification capability of a specific feature for classes.

Definition 5. Irrelevance, Relevance, and Redundancy of Features Let SIM_k^{Δ} and SIM_V^{Δ} be the SIM (Separability Index Matrix) of a feature x_k and the SIM of previously selected feature subset $V = \{x_1, \ldots, x_n\}$ in the feature selection process, respectively. Then:

(1) Feature x_k is said to be an irrelevant feature if $SIM_k^{\Delta} = 0$.
(2) Feature x_k is said to be a fully relevant feature if $SIM_k^{\Delta} \wedge SIM_V^{\Delta} = 0$ and $SIM_k^{\Delta} \neq 0$.
(3) Feature x_k is said to be a fully redundant feature if $SIM_k^{\Delta} \wedge SIM_V^{\Delta} = SIM_k^{\Delta}$ and $SIM_k^{\Delta} \neq 0$.
(4) Feature x_k is said to be a partially relevant (redundant) feature if $SIM_k^{\Delta} \wedge SIM_V^{\Delta} \neq SIM_k^{\Delta}$ and $SIM_k^{\Delta} \neq 0$.

A feature is considered as irrelevant if it has no between-class separable cases. A fully relevant feature has between-class separable cases which are not between-class separable by the previously selected feature subset V. We can discover such a feature from the remaining features by conducting simple Boolean AND operation between SIM_k^{Δ} and SIM_V^{Δ}. A redundant feature is a feature that provides between-class separable cases which are not needed in the previously selected feature subset V. Except for the above three cases, we regard them as partially relevant (redundant) features. Based on these concepts, we present in the next section a feature selection algorithm which is a forward search type.

4. Separability index matrix (SIM)-based feature selection algorithm

The proposed criterion function, i.e., classifiability based on the SIM, provides an efficient method for evaluating the quality of a feature. The proposed feature selection method is based on a forward search paradigm, where we add the feature to each search that is the most relevant of the remaining features. The complete algorithm is described below. There are two termination policies in the proposed selection algorithm. The algorithm terminates either (1) when the SIM of the selected feature subset satisfies a "*fully separable condition*" (refer to Definition 6) or (2) when the algorithm has selected the desired number of features.

Algorithm 1. Proposed feature selection algorithm

Let V be the selected feature subset, V_I an index set of the selected feature subset, F a given feature set, and F_I an index set of given feature set. We denote $V = \{\varphi\}, V_I = \{\varphi\}$, $F = \{x_1, x_2, \ldots, x_N\}$, and $F_I = \{1, 2, \ldots, N\}$.

1. Initialization
   $SDM_V \leftarrow [0_{i,j}]_{C \times C}$ where $1 \leqslant i,j \leqslant C$
   $SDM_k \leftarrow [J(w_i, w_j; \{x_k\})]_{C \times C}$ for all $k \in F_I$
   $SIM_k^{\Delta} \leftarrow (SDM_k \gtrdot SDM_{avg}) \vee (SDM_k \gtrdot SDM_{\Delta})$ for all $k \in F_I$
   $WM_k \leftarrow SIM_k^{\Delta} \oslash \left(\sum_{i=1}^{N} SIM_i^{\Delta} \right)$ for all $k \in F_I$
2. Search of the next best feature
   Find $\arg\max G(x_k)_{k \in S_I}$ where $G(x_k) = \sum_i \sum_j \left(SIM_k^{\Delta} \otimes WM_k \right)$
   $V_I \leftarrow V_I \cup \{k\}$
   If $(\|V_I\| \neq 1)$
      then $\arg\max S(x_k)_{k \in V_I}$ where $S(x_k) = \sum_i \sum_j (SDM_k \otimes WM_k)$
      and $V_I \leftarrow \{\varphi\}$
   3. Update of the feature subset with the next best feature
   $V \leftarrow V \cup \{x_k\}, F \leftarrow F - \{x_k\}, F_I \leftarrow F_I - \{k\}$
   $SDM_V \leftarrow [J(w_i, w_j; V)]_{C \times C}$
   $SIM_V^{\Delta} \leftarrow (SDM_V \gtrdot SDM_{avg}) \vee (SDM_V \gtrdot SDM_{\Delta})$
4. Update of the relevance of remaining features with respect to the selected feature subset
   $SIM_k^{\Delta} \leftarrow \left(\backsim SIM_V^{\Delta} \right) \wedge SIM_k^{\Delta}$ for all $k \in F_I$
   $WM_k \leftarrow SIM_k^{\Delta} \oslash \left(\sum_{i=1}^{N} SIM_i^{\Delta} \right)$ for all $k \in F_I$
   If "STOP" condition were satisfied, then go to 5; else go to 2.
5. Return V as the final subset

Our algorithm has three important components. First, the proposed algorithm considers the interactions of features during the evaluation step (Step 3), rather than the selection step (Step 2). During the selection step, the algorithm tries to find the most relevant feature with respect to the selected feature subset simply by evaluating its classifiability and the effect of the interaction is evaluated later. The updated SDM_V and SIM_V^A have interaction effects in Step 3. Furthermore, we simply modify this paradigm by checking whether the most recently selected feature interacts well with those that have already been selected. Secondly, our algorithm updates the relevance of the remaining features with respect to the selected feature subset using a simple Boolean operation in Step 4. After this update, we can easily find the next relevant features without intensively processing the remaining features in turn, as is found in other search methods. The third component relates to the computational time of our algorithm. The individual best search method, i.e., the fastest method, has a time complexity of $O(N)$. Here N is the total number of features. This method selects n individual best features from the complete feature sets after evaluating the goodness of N features independently. During the initial stage of the proposed algorithm, we have to evaluate the quality of N features like every other algorithm. After determining the quality of each feature, our algorithm repeats the selection of the best feature and a relevance update of the remaining features until the size of the selected feature subset reaches n. The computational time for the relevance update depends largely on the size of the target class, C, but the relevance update does not require much time because of the simple Boolean operation. Therefore the time complexity of the proposed algorithm is $O(C) \times O(N)$. It should be noted that $N > C$ is found in most classification problems.

Definition 6. Fully Separable Condition Let SIM_V^A be the separability index matrix of the selected feature subset V. Then, SIM_V^A is said to satisfy the *fully separable condition* if all elements of SIM_V^A except the diagonal terms is between-class separable in all distinct pairs of classes. This means that all elements of SIM_V^A are '1' except the diagonal terms being '0'.

Theorem 1. *Let SIM_V^A be the separability index matrix of the selected feature subset V. The selected feature subset V satisfies the "fully separable condition" if*

$$abs\left(\det\left(SIM_v^A\right)\right) = C - 1,$$

where abs(·) is an absolute value function, det (A) is a determinant of matrix A, and C is the total number of classes.

Proof. Let $P_C = [p_{ij}]_{C\times C}$ with

$$p_{ij} = \begin{cases} 1 & (i \neq j) \\ 0 & (i = j) \end{cases},$$

where $1 \leqslant i, j \leqslant C$. Note that the separability index matrix of the selected feature subset with the total number of classes C is P_C if the matrix satisfies a fully separable condition. Let 1_{mn} be the m-by-n matrix of all ones and let I_m be the m-by-m identity matrix. Then

$$\det(P_C) = \det(1_{CC} - I_C) = \det(1_{C1}1_{1C} - I_C) = (-1)^C \det(I_C - 1_{C1}1_{1C}) = (-1)^C \det(I_C + (-1_{C1})1_{1C}). \tag{9}$$

By the well-known Sylvester's determinant theorem [6], we have

$$\det(I_C + (-1_{C1})1_{1C}) = \det(I_1 + 1_{1C}(-1_{C1})). \tag{10}$$

By inserting Eq. (10) into (9), we obtain

$$\det(P_C) = (-1)^C \det(I_1 + 1_{1C}(-1_{C1})) = (-1)^C \det(1 - C) = (-1)^{C+1}(C - 1). \tag{11}$$

Thus, for $C \geqslant 1$,

$$abs(\det(P_C)) = C - 1. \quad \square \tag{12}$$

The Fully Separable Condition states a minimum property that the selected feature subset must possess. Any element with '0' except the diagonal terms in SIM shows that its corresponding classes are not separable by the selected feature subset. Therefore we must try to find a feature with *between-class separable* in those classes.

Example 3. For the SIM given in Example 2, we find a feature subset satisfying the *fully separable condition*.

From the results of Example 2, we know that $SIM_1^{A=3} = \begin{bmatrix} 0 & 1 & 1 \\ & 0 & 0 \\ & & 0 \end{bmatrix}$, $SIM_2^{A=3} = \begin{bmatrix} 0 & 1 & 1 \\ & 0 & 0 \\ & & 0 \end{bmatrix}$, and $SIM_1^{A=3} = \begin{bmatrix} 0 & 1 & 1 \\ & 0 & 0 \\ & & 0 \end{bmatrix}$. To find the best feature, we calculate classifiability $G(x_k)$ first. We obtain $G(x_1) = 0.83$, $G(x_2) = 0.83$, and $G(x_3) = 1.33$, for

$WM_1 = \begin{bmatrix} 0 & 0.5 & 0.33 \\ & 0 & 0 \\ & & 0 \end{bmatrix}$, $WM_2 = \begin{bmatrix} 0 & 0.5 & 0.33 \\ & 0 & 0 \\ & & 0 \end{bmatrix}$, and $WM_3 = \begin{bmatrix} 0 & 0 & 0.33 \\ & 0 & 1 \\ & & 0 \end{bmatrix}$.

Therefore we select x_3 first. Then, we need to update the relevance of the remaining features x_1 and x_2 with the guide of Step 4 of the algorithm. By applying a Boolean operation to the SIM of features x_1 and x_2, we obtain

$$SIM_1^{A=3} = \begin{bmatrix} 0 & 1 & 1 \\ & 0 & 0 \\ & & 0 \end{bmatrix} \wedge \left\{ \sim \begin{bmatrix} 0 & 0 & 1 \\ & 0 & 1 \\ & & 0 \end{bmatrix} \right\} = \begin{bmatrix} 0 & 1 & 0 \\ & 0 & 0 \\ & & 0 \end{bmatrix}$$

$$SIM_2^{A=3} = \begin{bmatrix} 0 & 1 & 1 \\ & 0 & 0 \\ & & 0 \end{bmatrix} \wedge \left\{ \sim \begin{bmatrix} 0 & 0 & 1 \\ & 0 & 1 \\ & & 0 \end{bmatrix} \right\} = \begin{bmatrix} 0 & 1 & 0 \\ & 0 & 0 \\ & & 0 \end{bmatrix}$$

The SIM of the remaining features are the same and it renders the same classifiability of the remaining features in this example, therefore we need $S(x_k)$ to choose the best one. We obtain $S(x_1) = 4.95$ and $S(x_2) = 2.35$ by calculating $S(x_k) = \sum_i \sum_j (SDM_k \otimes WM_k)$ with the SDM of each feature in Example 1 and updated weight matrices of each feature $WM_1 = WM_2 = \begin{bmatrix} 0 & 0.5 & 0 \\ & 0 & 0 \\ & & 0 \end{bmatrix}$. After adding feature x_1, the selected feature subset satisfies the *fully separable condition*, for $abs(\det\left(SIM_V^A\right)) = 2$.

Remark 1. In the feature selection problem with the two classes ($C = 2$), the proposed algorithm turns into *the individual best selection method*. The first best feature makes SIM_V^A satisfy the *fully separable condition* when WM_k. All remaining features become irrelevant features after the relevance update with the first best feature. The proposed algorithm finds the next best feature based not on SIM_k^A but on SDM_k which is the same criterion function of the individual best selection method.

Remark 2. If the desired number of features (N_d) given by the user is bigger than the number of selected features (N_f) which satisfy the *fully separable condition*, the proposed algorithm searches for additional features ($N_d - N_f$) based on a modified SIM_V^A, which is updated by $SIM_V^A \leftarrow (SDM_V \gtrsim SDM_{Vavg})$. Here SDM_{Vavg} is a mean matrix of SDM_{Vavg} given by $SDM_{Vavg} = m_{avg} \times [1_{i,j}]_{C\times C}$ where $m_{avg} = \frac{1}{2C}\sum_i \sum_j SDM_V(i,j)$. Because the SIM of the feature subset satisfying the *fully separable condition* has all '1's except the diagonal terms, all remaining features become irrelevant ones after the relevance update in the algorithm. In this situation, we can say that a relevant feature is a feature that contains a substantial discriminating force between classes which is less separable in the selected feature subset. To find relatively less separable information between classes, we use the modified SIM_V^A. The algorithm in this case runs in the same way except for this modification on SIM_V^A.

5. Experimental results

We tested our proposed feature selection algorithm using 10 continuous datasets, as shown in Table 2. We selected these datasets from the UCI repository [20] because they are well-known benchmark datasets with the following three conditions: (i) the number of features is $\geqslant 10$, (ii) there are no missing features, and (iii) all of the feature values are continuous.

We evaluated the performance of the proposed feature selection algorithm using the following two performance indicator values [2,22]: (1) classification accuracy and (2) computational costs.

Table 2
Datasets used for experiment.

Name of data set	Number of instances	Number of classes	Number of features	Feature size[b]
Glass	214	7	10	Small
Vowel	990	11	10	Small
Wine	178	3	13	Small
Letter	20,000	26	16	Small
Vehicle	846	4	18	Small
Segmentation	2310[a]	7	19	Small
Wdbc	569	2	30	Medium
Ionosphere	351	2	34	Medium
Satellite	6435[a]	6	36	Medium
Sonar	208	2	60	Large

[a] Training dataset and test dataset are merged into one dataset for experiment.
[b] Feature size is decided upon the results of Kudo's study [15].

Table 3
Criterion functions of feature selection algorithm used for experiment.

Alias	Criterion functions	Alias	Criterion functions
In–in	Inter–intra distance	eucl-s	Sum of squared Euclidean distance
Maha-s	Sum of Mahalanobis distance	eucl-m	Minimum of squared Euclidean distance
Maha-m	Minimum of Mahalanobis distance	bhatta	Bhattacharyya distance

5.1. Classification accuracy

We have tested the proposed feature selection method using various feature selection algorithms. A typical feature selection algorithm consists of a specific criterion function and a specific search method. There are many different types of criterion functions and search methods, with numerous combinations. To evaluate the classification accuracy, we considered the criterion functions shown in Table 3 when using a feature selection algorithm as well as the Branch-and-Bound search method. The latter is known to deliver an optimal solution for the given criterion function.

To determine the classification accuracy of the selected feature subset using various feature selection algorithms, we tested three different classifiers, i.e., the *k*-nearest neighbor classifier (*knnc*), Parzen classifier (*parzenc*), and a naive Bayes classifier (*naievebc*). The classification accuracy of the selected feature subset was affected by its size. In general, the classification accuracy improves as the number of features increases when the selected features are relevant. We set the maximum size of the selected feature subset as 9, because no further improvement of classification accuracy was required.

We split the original dataset into a training dataset and a test dataset using a random subsampling method. Several training datasets were generated by taking 20%, 30%, 50%, and 80% of the whole dataset. The classification accuracy of the selected feature subset was averaged over 30 independent trials. Thus, the total number of simulations for classification accuracy of all the dataset was 252,000, i.e., 10 (datasets) $\times$ 7 (feature selection algorithms, including the proposed one) $\times$ 3 (classifiers) $\times$ 10 (cases of feature size, including all feature cases) $\times$ 4 (ratio of training datasets) $\times$ 30 (iterations).

Table 4 shows the average classification error with the standard deviation for the *k*-nearest neighbor classifier of each dataset at a 50% training data ratio. We omit the classification results with the other two classifiers and with different ratios of training dataset, because results were similar to those shown in Table 4.

Figs. 2–4 shows representative experimental results when using the *k*-nearest neighbor classifier with the segmentation dataset, the Parzen classifier with the satellite dataset, and the naive Bayes classifier with the wine dataset, respectively. We compared the results using a paired *t*-test to test for statistical differences [24]. The significance level was set at $p < 0.05$, so the confidence interval was 0.95. Table 5 shows the results of the paired *t*-test when using the *k*-nearest neighbor classifier with the segmentation dataset. Here '1' means that the proposed algorithm produced a significantly better performance with fewer classification errors than other algorithms (i.e., $p < 0.05$). Where other algorithms performed better than the proposed algorithm, the comparison is indicated as '−1'. Note that '0' indicates no statistical difference between the proposed algorithm and other algorithms. We performed a *t*-test on the classification results when using the other classifiers and other datasets. Table 6 summarizes the *t*-test results with the various criterion functions for all datasets.

In terms of classification accuracy, the experimental results showed that the proposed feature selection algorithm performed better than the Branch-and-Bound search method with various criterion functions, except for the few cases shown in Table 6. In particular, the datasets letter, vehicle, segmentation, ionosphere, and satellite were improved by more than 50% in all cases. Note that the criterion functions 'in–in' and 'maha-s' always produced the same feature selection results in the experiments. This confirms that the proposed classifiability method consistently produced a better evaluation of the quality of a feature subset. The proposed feature selection algorithm was also the best performing individual selection algorithm with a two-class classification problem. Other feature selection algorithms may produce an optimal subset using the Branch-and-Bound method, but they produced lower classification accuracies than the proposed algorithm. Unlike the multi-class problem, the quality of a specific feature in a two-class problem increased with the separate feature distribution. Our experimental results confirmed that it was sufficient to individually check the separability of features and to select the *n* best features when $C = 2$. In multi-class problems, the notion of separability was not sufficient to determine the quality of a feature, as shown in Fig. 1. We need to consider the separability and the relation between feature distributions when evaluating the quality of a specific feature. Thus, the term "classifiability" is defined in this study based on a consideration of these issues.

After satisfying the fully separable condition, the proposed algorithm finds the next best feature based on the modified SIM_V^A. Table 7 shows the feature subsets from each dataset that satisfy the fully separable condition and their classification results. We concluded that the modified SIM_V^A performed this task adequately because classification errors decreased as more features were added. A practical recommendation of the number of selected feature subsets is $N_f + 2$, where N_f is the number of features that satisfy the fully separable condition. In this case, errors are found within 10% of the classification error of all features.

5.2. Computational costs

We measured the time taken by each feature selection algorithm to find a given number of features. We repeated this process over 30 trials for each algorithm and we used the average time for comparisons. In terms of computational costs,

Table 4
Averaged classification error (%) and standard deviation for 30 trials of knnc for each dataset.

Dataset	n	Proposed	In–in/maha-s	Maha-m	Eucl-s	Eucl-m	Bhatta
Glass ($N = 10$, $C = 7$)	1	59.9 ± 4.18	60.5 ± 4.76	58.8 ± 4.47	59.5 ± 4.51	59.0 ± 4.21	59.5 ± 4.74
	2	43.9 ± 3.80	39.4 ± 3.74	44.2 ± 4.39	50.7 ± 4.11	44.0 ± 3.49	45.8 ± 4.47
	4	35.6 ± 3.55	35.8 ± 4.13	37.5 ± 3.96	35.9 ± 4.02	38.3 ± 3.51	33.3 ± 4.66
	6	31.3 ± 3.79	40.5 ± 3.39	35.1 ± 3.88	31.6 ± 4.30	32.1 ± 4.15	35.4 ± 4.02
	8	31.5 ± 4.00	31.8 ± 4.53	31.3 ± 5.23	29.9 ± 4.78	31.3 ± 4.57	31.7 ± 3.84
	All	31.8 ± 4.39	31.9 ± 4.18	30.6 ± 3.85	32.1 ± 4.81	31.5 ± 4.38	32.0 ± 4.15
Vowel ($N = 10$, $C = 11$)	1	68.0 ± 1.29	66.3 ± 2.46	80.1 ± 2.96	66.3 ± 2.91	80.4 ± 2.95	69.3 ± 1.91
	2	40.6 ± 1.68	40.9 ± 2.12	40.8 ± 2.53	40.8 ± 1.44	41.3 ± 2.66	40.7 ± 1.84
	4	13.3 ± 1.28	14.3 ± 1.78	14.2 ± 1.63	13.3 ± 1.38	19.5 ± 1.73	14.1 ± 1.54
	6	7.45 ± 1.64	8.88 ± 1.76	8.59 ± 1.24	8.51 ± 1.31	7.91 ± 1.22	7.92 ± 1.39
	9	5.08 ± 1.50	5.25 ± 1.39	5.26 ± 1.35	5.26 ± 1.24	5.37 ± 1.74	4.24 ± 0.88
	All	4.98 ± 1.66	4.50 ± 1.39	4.55 ± 1.41	4.60 ± 1.23	4.42 ± 1.50	4.60 ± 1.35
Wine ($N = 13$, $C = 3$)	1	22.8 ± 4.09	22.1 ± 3.55	29.0 ± 4.55	33.0 ± 3.49	34.4 ± 3.24	23.6 ± 4.83
	2	8.90 ± 2.57	9.28 ± 2.43	30.5 ± 3.56	30.4 ± 3.94	31.7 ± 3.60	17.7 ± 3.44
	4	31.2 ± 3.27	31.7 ± 4.40	30.9 ± 3.66	30.9 ± 4.72	30.2 ± 3.05	29.3 ± 4.11
	6	32.0 ± 3.87	31.2 ± 4.20	30.2 ± 4.26	31.2 ± 3.68	30.5 ± 3.16	29.7 ± 4.55
	9	30.4 ± 4.40	30.1 ± 3.89	29.9 ± 4.66	31.2 ± 3.43	29.7 ± 3.48	30.8 ± 2.83
	All	30.8 ± 3.70	31.1 ± 3.81	30.1 ± 4.81	30.2 ± 3.04	29.6 ± 3.86	31.2 ± 3.54
Letter ($N = 16$, $C = 26$)	1	86.2 ± 1.00	86.3 ± 1.11	86.3 ± 0.88	86.2 ± 1.15	86.3 ± 0.81	86.0 ± 1.04
	2	74.1 ± 1.98	72.0 ± 1.73	72.3 ± 1.65	77.2 ± 1.37	76.0 ± 1.52	72.2 ± 1.64
	4	45.9 ± 1.66	45.5 ± 1.30	57.7 ± 1.62	48.6 ± 2.02	57.4 ± 1.60	45.5 ± 1.29
	6	28.3 ± 1.48	32.9 ± 1.23	41.1 ± 1.36	27.4 ± 1.24	40.0 ± 1.69	24.6 ± 1.16
	9	19.5 ± 1.30	17.7 ± 1.25	40.5 ± 1.50	17.6 ± 1.20	28.7 ± 1.43	17.3 ± 1.12
	All	22.3 ± 1.50	22.4 ± 1.47	22.4 ± 1.52	22.4 ± 1.74	22.2 ± 1.43	22.4 ± 1.68
Vehicle ($N = 18$, $C = 4$)	1	61.0 ± 2.61	63.3 ± 1.66	61.0 ± 2.52	54.8 ± 2.56	54.4 ± 2.33	53.7 ± 2.67
	2	53.5 ± 2.08	51.7 ± 2.11	55.0 ± 1.89	48.2 ± 1.76	44.5 ± 1.86	47.2 ± 2.11
	4	42.7 ± 1.51	44.1 ± 1.84	45.8 ± 1.69	45.6 ± 2.22	41.8 ± 1.65	42.6 ± 1.72
	6	40.1 ± 2.26	35.0 ± 1.24	45.0 ± 1.84	42.1 ± 1.79	42.4 ± 2.03	41.3 ± 2.12
	9	37.4 ± 1.88	33.4 ± 1.89	45.2 ± 1.52	41.1 ± 2.17	40.0 ± 1.68	39.4 ± 2.01
	All	37.4 ± 1.64	37.5 ± 2.02	37.1 ± 1.89	38.0 ± 2.14	37.8 ± 2.60	38.1 ± 2.24
Segmentation ($N = 19$, $C = 7$)	1	53.2 ± 2.34	85.7 ± 0.00	85.7 ± 0.00	55.0 ± 2.30	55.1 ± 2.68	48.1 ± 1.12
	2	19.7 ± 0.92	38.8 ± 1.32	44.6 ± 3.11	16.0 ± 0.74	46.8 ± 1.27	35.4 ± 0.97
	4	8.70 ± 0.94	19.2 ± 1.05	19.5 ± 1.40	12.5 ± 0.80	11.9 ± 0.86	20.6 ± 1.10
	6	5.99 ± 0.70	18.0 ± 0.94	8.30 ± 0.81	8.10 ± 0.65	9.88 ± 0.85	14.6 ± 0.90
	9	6.46 ± 0.79	6.69 ± 0.60	6.12 ± 0.73	7.43 ± 0.62	6.31 ± 0.76	6.55 ± 0.63
	All	5.96 ± 0.60	5.87 ± 0.75	5.93 ± 0.80	5.92 ± 0.73	5.94 ± 0.56	5.96 ± 0.82
Wdbc ($N = 30$, $C = 2$)	1	9.14 ± 1.46	9.54 ± 1.23	9.54 ± 1.23	8.78 ± 1.70	8.78 ± 1.70	8.91 ± 1.23
	2	8.83 ± 1.36	27.0 ± 2.15	27.0 ± 2.15	7.66 ± 1.19	7.66 ± 1.19	9.88 ± 1.47
	4	8.65 ± 1.69	9.58 ± 1.62	9.58 ± 1.62	7.70 ± 1.37	7.70 ± 1.37	9.12 ± 2.10
	6	8.67 ± 1.62	8.91 ± 1.34	8.91 ± 1.34	7.08 ± 1.21	7.08 ± 1.21	8.98 ± 1.33
	9	7.48 ± 1.34	8.83 ± 1.26	8.83 ± 1.26	7.37 ± 1.56	7.37 ± 1.56	7.14 ± 1.15
	All	7.28 ± 1.41	6.94 ± 1.03	6.94 ± 1.03	7.41 ± 1.21	7.41 ± 1.21	7.41 ± 1.39
Ionosphere ($N = 34$, $C = 2$)	1	24.9 ± 1.47	17.9 ± 1.67	36.0 ± 0.00	18.4 ± 3.02	18.4 ± 3.02	25.4 ± 1.61
	2	12.4 ± 1.62	12.6 ± 1.83	21.8 ± 7.16	13.8 ± 2.14	13.8 ± 2.14	12.3 ± 2.09
	4	11.2 ± 2.61	10.8 ± 2.06	18.1 ± 2.06	15.2 ± 2.40	15.2 ± 2.40	11.7 ± 2.09
	6	11.1 ± 1.75	9.92 ± 1.62	15.8 ± 1.64	14.4 ± 2.26	14.4 ± 2.26	11.4 ± 2.18
	9	13.2 ± 1.92	11.5 ± 1.74	14.4 ± 2.33	14.4 ± 2.69	14.4 ± 2.69	14.2 ± 2.44
	All	15.7 ± 2.30	15.2 ± 2.16	15.5 ± 1.86	15.3 ± 2.00	15.3 ± 2.00	15.5 ± 2.34
Satellite ($N = 36$, $C = 6$)	1	45.2 ± 1.60	42.8 ± 2.21	44.9 ± 1.40	42.6 ± 2.25	43.3 ± 2.24	45.2 ± 1.60
	2	23.1 ± 2.89	28.8 ± 4.14	25.2 ± 3.66	42.7 ± 4.46	20.9 ± 4.84	23.1 ± 2.89
	4	14.6 ± 1.04	16.1 ± 1.36	20.5 ± 1.19	39.9 ± 1.91	17.0 ± 1.47	14.6 ± 1.04
	6	14.0 ± 1.16	14.6 ± 1.40	14.8 ± 1.29	38.6 ± 1.94	14.6 ± 1.22	14.0 ± 1.16
	9	13.9 ± 1.68	14.8 ± 2.02	14.4 ± 1.35	37.8 ± 2.29	14.4 ± 1.20	13.9 ± 1.68
	All	13.5 ± 1.75	13.5 ± 1.67	13.3 ± 1.63	13.8 ± 1.70	13.3 ± 1.66	13.5 ± 1.75
Sonar ($N = 60$, $C = 2$)	1	27.0 ± 4.87	25.5 ± 4.21	25.5 ± 4.21	37.0 ± 3.12	37.0 ± 3.12	25.5 ± 4.21
	2	25.0 ± 3.09	26.1 ± 3.11	26.1 ± 3.11	30.1 ± 3.87	30.1 ± 3.87	26.1 ± 3.11
	4	24.5 ± 3.40	21.4 ± 3.86	21.4 ± 3.86	31.0 ± 3.94	31.0 ± 3.94	21.4 ± 3.86
	6	24.7 ± 3.19	20.3 ± 3.71	20.3 ± 3.71	26.0 ± 3.72	26.0 ± 3.72	20.3 ± 3.71
	9	25.9 ± 3.09	21.3 ± 4.32	21.3 ± 4.32	21.6 ± 3.22	21.6 ± 3.22	21.3 ± 4.32
	All	19.4 ± 3.30	20.7 ± 4.03	20.7 ± 4.03	20.7 ± 3.72	20.7 ± 3.72	20.7 ± 4.03

Table 8 shows that the feature selection algorithms using the 'eucl-s' criterion function required less time than the other criterion functions.

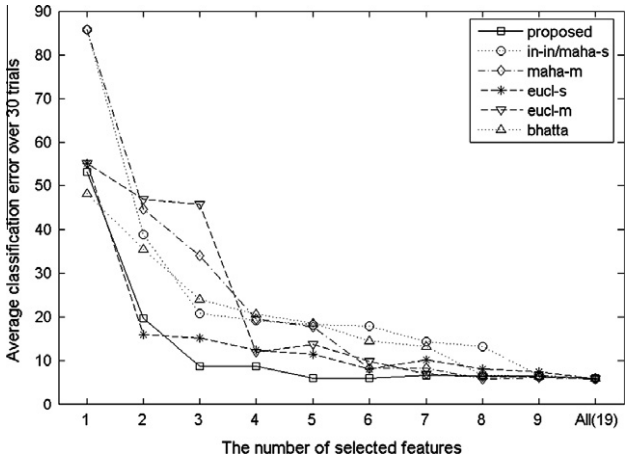

Fig. 2. Performance of *k*-nearest neighbor classifier for segmentation dataset.

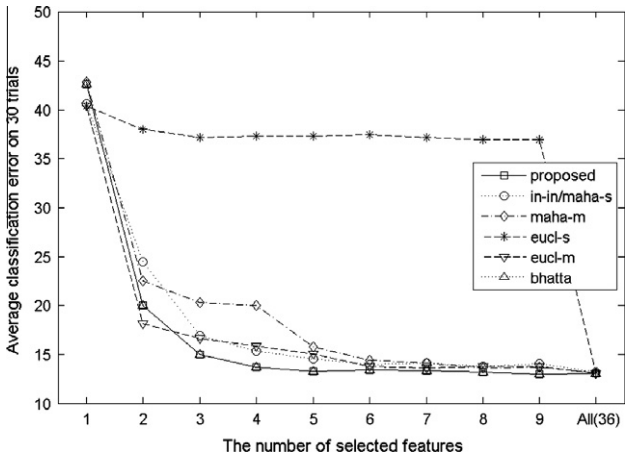

Fig. 3. Performance of Parzen classifier for satellite dataset.

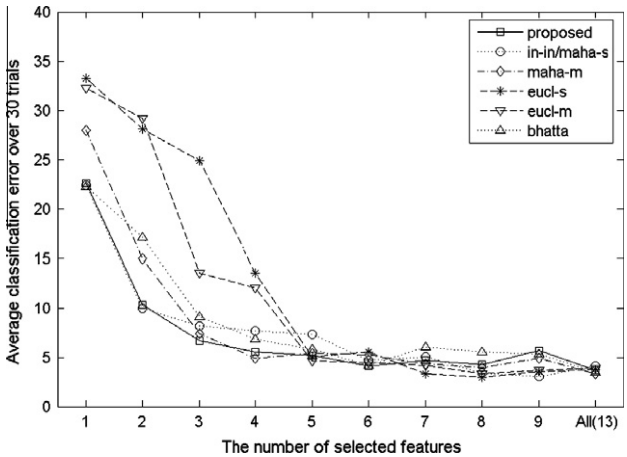

Fig. 4. Performance of naive Bayes classifier for wine dataset.

Table 5
Results of paired *t*-test on *k*-nearest neighbor classifier for segmentation dataset.

Feature size	1	2	3	4	5	6	7	8	9	All (19)	Sum
Proposed vs. in–in/maha-s	1	1	1	1	1	1	1	1	0	0	8
Proposed vs. maha-m	1	1	1	1	1	1	1	−1	0	0	6
Proposed vs. eucl-s	1	−1	1	1	1	1	1	1	1	0	7
Proposed vs. eucl-m	1	1	1	1	1	1	0	0	0	0	6
Proposed vs. bhatta	−1	1	1	1	1	1	1	0	0	0	5

Table 6
Results of paired *t*-test for all datasets across various criterion functions.

Dataset	Criterion function	Wins (*W*) (%)	Ties (%)	Loses (*L*) (%)	*W*/(*W* + *L*) (%)
Glass (*N* = 10, *C* = 7)	In–in/maha-s	19 (5/27)	74 (20/27)	7 (2/27)	71 (5/7)
	Maha-m	26 (7/27)	70 (19/27)	4 (1/27)	88 (7/8)
	Eucl-s	11 (3/27)	81 (22/27)	7 (2/27)	60 (3/5)
	Eucl-m	15 (4/27)	78 (21/27)	7 (2/27)	67 (4/6)
	Bhatta	4 (1/27)	89 (24/27)	7 (2/27)	33 (1/3)
Vowel (*N* = 10, *C* = 11)	In–in/maha-s	17 (5/30)	73 (22/30)	10 (3/30)	63 (5/8)
	Maha-m	47 (14/30)	53 (16/30)	0 (0/30)	100 (14/14)
	Eucl-s	10 (3/30)	77 (23/30)	13 (4/30)	43 (3/7)
	Eucl-m	43 (13/30)	57 (17/30)	0 (0/30)	100 (13/13)
	Bhatta	10 (3/30)	73 (22/30)	17 (5/30)	38 (3/8)
Wine (*N* = 13, *C* = 3)	In–in/maha-s	10 (3/30)	87 (26/30)	3 (1/30)	75 (3/4)
	Maha-m	23 (7/30)	73 (22/30)	3 (1/30)	88 (7/8)
	Eucl-s	30 (9/30)	57 (17/30)	13 (4/30)	69 (9/13)
	Eucl-m	27 (8/30)	67 (20/30)	7 (2/30)	80 (8/10)
	Bhatta	23 (7/30)	67 (20/30)	10 (3/30)	70 (7/10)
Letter (*N* = 16, *C* = 26)	In–in/maha-s	33 (10/30)	27 (8/30)	40 (12/30)	45 (10/22)
	Maha-m	67 (20/30)	17 (5/30)	17 (5/30)	80 (20/25)
	Eucl-s	40 (12/30)	33 (10/30)	27 (8/30)	60 (12/20)
	Eucl-m	80 (24/30)	17 (5/30)	3 (1/30)	96 (24/25)
	Bhatta	27 (8/30)	40 (12/30)	33 (10/30)	44 (8/18)
Vehicle (*N* = 18, *C* = 4)	In–in/maha-s	43 (13/30)	20 (6/30)	37 (11/30)	54 (13/24)
	Maha-m	80 (24/30)	13 (4/30)	7 (2/30)	92 (24/26)
	Eucl-s	63 (19/30)	23 (7/30)	13 (4/30)	83 (19/23)
	Eucl-m	53 (16/30)	27 (8/30)	20 (6/30)	73 (16/22)
	Bhatta	40 (12/30)	40 (12/30)	20 (6/30)	67 (12/18)
Segmentation (*N* = 19, *C* = 7)	In–in/maha-s	83 (25/30)	13 (4/30)	3 (1/30)	96 (25/26)
	Maha-m	70 (21/30)	13 (4/30)	17 (5/30)	81 (21/26)
	Eucl-s	63 (19/30)	17 (5/30)	20 (6/30)	76 (19/25)
	Eucl-m	53 (16/30)	30 (9/30)	17 (5/30)	76 (16/21)
	Bhatta	67 (20/30)	20 (6/30)	13 (4/30)	83 (20/24)
Wdbc (*N* = 30, *C* = 2)	In–in/maha-s/maha-m	30 (9/30)	40 (12/30)	30 (9/30)	50 (9/18)
	Eucl-s/eucl-m	27 (8/30)	50 (15/30)	23 (7/30)	53 (8/15)
	Bhatta	7 (2/30)	93 (28/30)	0 (0/30)	100 (2/2)
Ionosphere (*N* = 34, *C* = 2)	In–in/maha-s	7 (2/30)	60 (18/30)	33 (10/30)	17 (2/12)
	Maha-m	87 (26/30)	13 (4/30)	0 (0/30)	100 (26/26)
	Eucl-s/eucl-m	70 (21/30)	20 (6/30)	10 (10/30)	88 (21/24)
	Bhatta	0 (0/30)	97 (29/30)	3 (1/30)	0 (0/1)
Satellite (*N* = 36, *C* = 6)	In–in/maha-s	63 (21/30)	17 (5/30)	20 (6/30)	76 (19/25)
	Maha-m	63 (21/30)	30 (9/30)	7 (2/30)	90 (19/21)
	Eucl-s	80 (21/30)	10 (3/30)	10 (3/30)	89 (24/27)
	Eucl-m	47 (21/30)	20 (6/30)	33 (10/30)	58 (14/24)
	Bhatta	0 (0/30)	100 (30/30)	0 (0/30)	–
Sonar (*N* = 60, *C* = 2)	In–in/maha-s/				
	Maha-m/bhatta	3 (1/30)	47 (14/30)	50 (15/30)	6 (1/16)
	Eucl-s/eucl-m	47 (14/30)	37 (11/30)	17 (5/30)	74 (14/19)

Computation time largely depends on the complexity of the algorithm and the feature size, but it also depends on other factors such as hardware capacity. This means that the absolute running time is not a good measure, so we used the time ratio (*TR*) of the elapsed running time as follows:

$$TR = \sum_{o=1}^{30} t_o \Big/ \sum_{p=1}^{30} t_p \tag{13}$$

Table 7
Feature subsets that satisfy the fully separable condition and its classification error of knnc for each dataset.

	Feature subset satisfying fully separable cond. (set A)	C.E.* of first best feature	C.E. of set A	C.E. of set A + 2**	C.E. of all features
Glass	$\{x_3,x_6,x_7\}$	59.9 ± 4.18	36.8 ± 3.13	33.3 ± 4.17	31.8 ± 4.39
Vowel	$\{x_1,x_2,x_5\}$	68.0 ± 1.29	21.9 ± 1.60	10.6 ± 1.81	4.98 ± 1.66
Wine	$\{x_7\}$	22.8 ± 4.09	22.8 ± 4.09	30.7 ± 2.66	30.8 ± 3.70
Letter	$\{x_{11},x_{12},x_{13}\}$	86.2 ± 1.00	63.5 ± 1.29	32.5 ± 1.10	22.3 ± 1.50
Vehicle	$\{x_6,x_1\}$	61.0 ± 2.61	53.5 ± 2.08	42.7 ± 1.51	37.4 ± 1.64
Segmentation	$\{x_{16},x_{17},x_2\}$	53.2 ± 2.33	8.70 ± 0.65	6.04 ± 0.82	5.96 ± 0.60
Wdbc	$\{x_{28}\}$	9.14 ± 1.46	9.14 ± 1.46	8.56 ± 1.51	7.28 ± 1.41
Ionosphere	$\{x_1\}$	24.9 ± 1.47	24.9 ± 1.47	11.4 ± 2.30	15.7 ± 2.30
Satellite	$\{x_2,x_6,x_1\}$	45.2 ± 1.60	15.5 ± 1.36	14.0 ± 1.12	13.5 ± 1.75
Sonar	$\{x_{11}\}$	27.0 ± 4.87	27.0 ± 4.87	25.0 ± 3.61	19.4 ± 3.30

C.E.* means classification error, set A + *n* means that next best *n* features are added to set A.

Table 8
Search methods of feature selection algorithm used in experimental evaluation.

Alias	Search methods
Bwd	Backward sequential search method
Fwd	Forward sequential search method
l,r	Plus *l*–take away *r* method
Indi	Individual best search method
bb	Branch and Bound method
Float	Floating search method

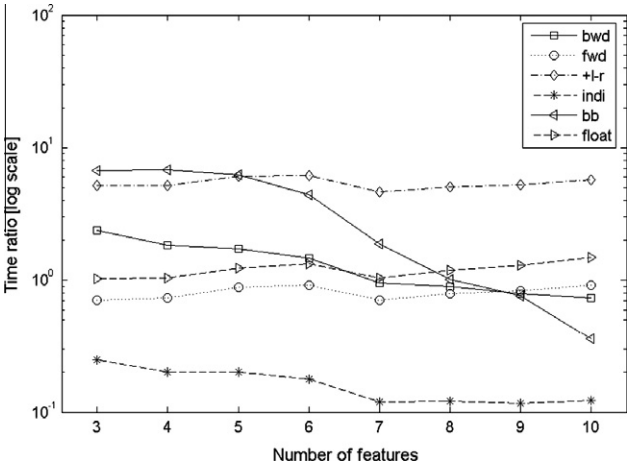

Fig. 5. Computational time ratio between conventional methods and proposed algorithm for letter dataset.

where t_p is the elapsed time for the proposed algorithm and t_o is elapsed time for other algorithms.

Figs. 5–8 show the computational time ratio for the letter, vehicle, ionosphere, and sonar datasets. These datasets were selected as representative datasets with a small feature size (letter, vehicle), medium feature size (ionosphere), and large feature size (sonar).

Other than the letter dataset, the proposed feature selection algorithm found a feature subset as quickly as the best individual search method. The two algorithms differed only in the step where the relevance update was applied to the remaining features. The relevance update performs a Boolean operation on the remaining features to identify the next best feature. The time cost of the relevance update process was sufficiently small that its effect could be ignored when the total number of classes was less than 10, regardless of the feature size. The time cost of the proposed feature selection algorithm depended mainly on the total number of features, rather than the total number of classes. The letter dataset had more target classes ($C = 26$) than the total number of features ($N = 16$), which was an unusual scenario in a classification problem.

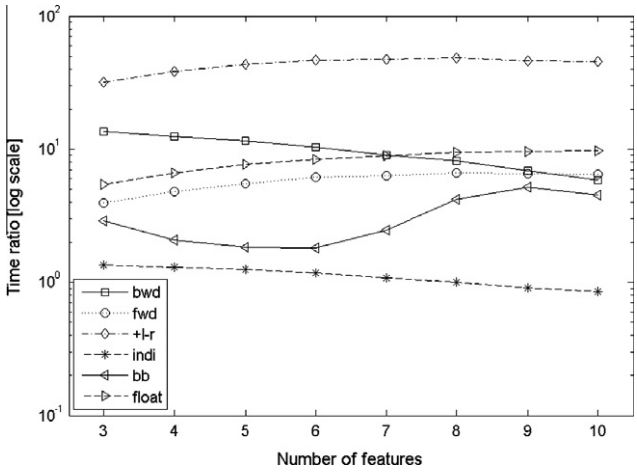

Fig. 6. Computational time ratio between conventional methods and proposed algorithm for vehicle dataset.

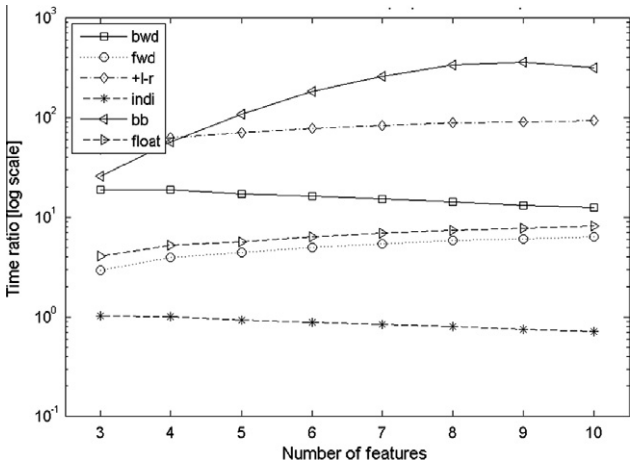

Fig. 7. Computational time ratio between conventional methods and proposed algorithm for ionosphere dataset.

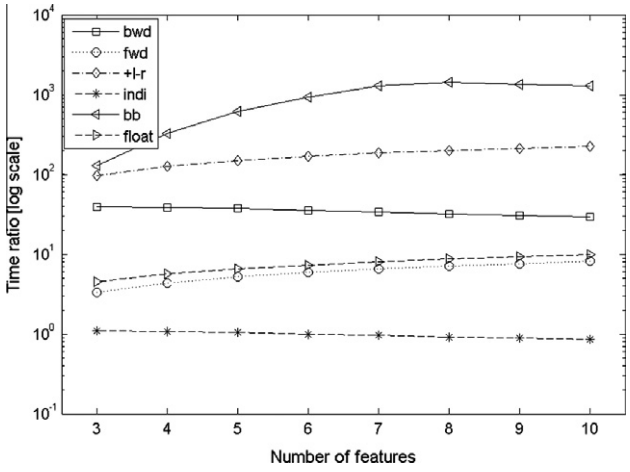

Fig. 8. Computational time ratio between conventional methods and proposed algorithm for sonar dataset.

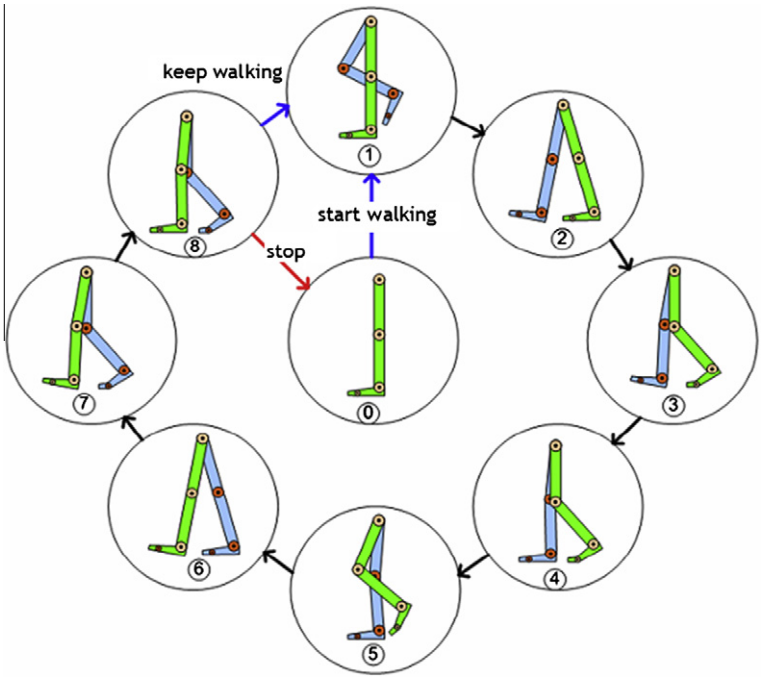

Fig. 9. State-transition diagram of walking phase.

6. A real-world application: EMG signal-based gait phase recognition

The proposed feature selection algorithm was applied to the real-world problem of electromyographic (EMG) signal-based gait phase recognition. We demonstrated that the proposed method (filter type) could be applied to an EMG signal-based walking assistance system in a simple and straightforward manner, while still delivering comparable performance to conventional wrapper-type methods that may incur higher computational costs. To ensure a regular and consistent dataset, we designed a gait-pattern recording system [18] with a sensing suit that tightly fixed electrodes at the sensing location while notch filter-based EMG amplifiers ensured a high signal-to-noise ratio. We then captured EMG signals from the 1-min gait pattern of three subjects (two males and one female). The measurements were conducted for the *semimembranosus* and *vastus medialis* muscles, where low-noise EMG signals can be obtained [10]. The walking speed was maintained at 1 km h^{-1} using a treadmill while the ground inclination was fixed at 0°. We defined the walking cycle as nine states depending on the on/off status of pressure sensors located on the toe and heel beneath both feet, [18] and the number of classes was also set as nine (see Fig. 9). Note that the states were typically identified in the walking speed of people with lower limb disabilities.

The data were then used to extract the following 13 types of feature sets: integral of absolute value (IAV), variance (VAR), Wilson amplitude (WAMP), zero crossing (ZC), number of turns (NT), mean amplitude (MA), mean frequency (MF), histogram (HIST), auto-regressive coefficient (AR), auto-regressive coefficient from third order cumulant (ARCU), energy of wavelet coefficient (EWT), energy of wavelet packet coefficient (EWP), zero crossing of wavelet coefficient (ZCWT). For more details, refer to [18].

A combination of all these 13 features constituted a bundle of feature sets with 416 dimensions. Three different feature selection algorithms were then applied: recursive feature elimination using a Support Vector Machine (SVM) [8], recursive feature elimination using the Maximum Margin Criterion (MMC) [21], and the proposed method. Note that SVM and MMC were wrapper-types, whereas the developed method was a filter-type. We considered 100- and 150-dimensional feature spaces for each feature selection method to test the effects of dimensionality. The selected feature sets were denoted as SVM 100/150, MMC 100/150, and Proposed 100/150.

Fig. 10 shows our qualitative comparison in terms of class-separability. A Fisher ratio [3] was used as a separability measure. Due to the great differences in dimensionality, the feature sets produced by some of the feature extraction techniques appeared to have better class-separability than feature sets produced by the our feature selection algorithms. Thus, the comparison was made either among the feature extraction groups (IAV, VAR, WAMP, ZC, NT, MA, MF, HIST, AR, ARCU, EWT, EWP, and ZCWT) or among the feature selection groups (SVM 100/150, MMC 100/150, and Proposed 100/150). When a comparison is made among the feature selection algorithms, the Fisher ratios of the feature sets produced using the proposed method were better than those produced using the SVM and MMC methods. Overall, the proposed method (filter-type) provided better class-separability than the SVM and MMC (wrapper-type) methods.

We performed a rigorously empirical evaluation of the EMG feature sets using four different types of representative classification techniques, i.e., Gaussian kernel-based Linear Discriminant Analysis (GK-LDA) [8], Polynomial kernel-based Linear

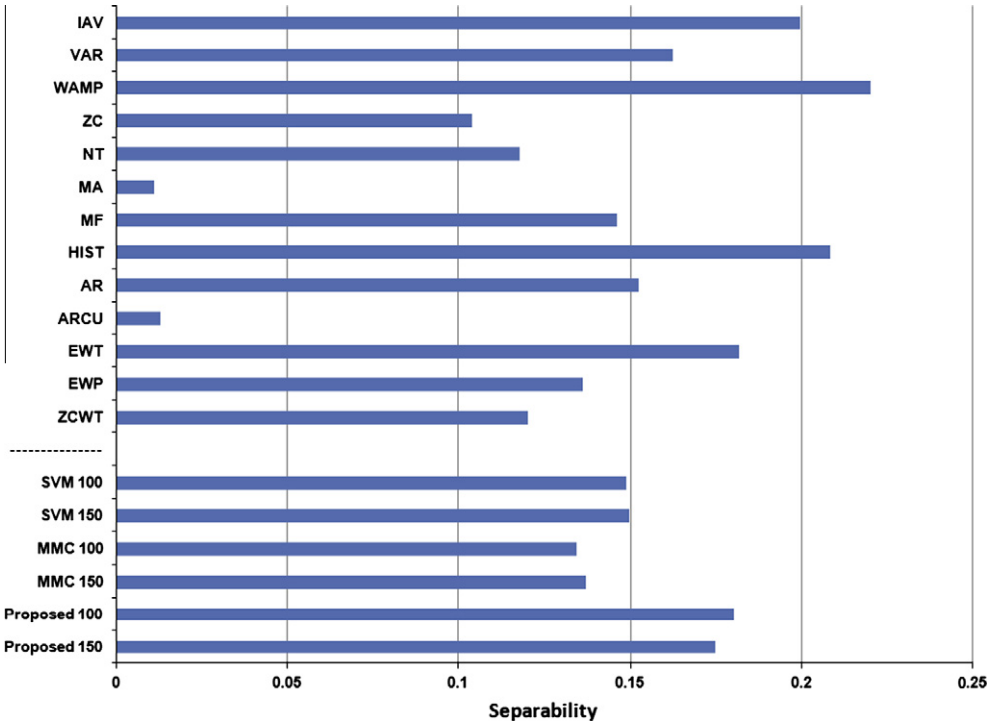

Fig. 10. Seperability value of various feature sets.

Table 9
Averaged classification error (%) and standard deviation of various classifiers for each dataset (note: all simulation underwent a 10-fold cross validation).

Classifier feature	GK-LDA	PK-LDA	GK-SVM	PK-SVM
IAV	22.62 ± 0.12	23.65 ± 0.12	18.85 ± 0.11	20.46 ± 0.12
VAR	27.10 ± 0.14	29.93 ± 0.08	22.43 ± 0.11	22.14 ± 0.10
WAMP	24.52 ± 0.08	26.91 ± 0.07	22.54 ± 0.07	25.06 ± 0.06
ZC	38.63 ± 0.15	39.69 ± 0.15	32.02 ± 0.13	37.00 ± 0.11
NT	42.17 ± 0.15	42.82 ± 0.14	37.76 ± 0.17	38.97 ± 0.16
MA	49.98 ± 0.08	52.67 ± 0.06	49.05 ± 0.07	49.36 ± 0.06
MF	34.69 ± 0.12	34.82 ± 0.12	28.57 ± 0.09	30.06 ± 0.08
HIST	21.22 ± 0.12	20.78 ± 0.13	19.89 ± 0.11	19.52 ± 0.12
AR	24.83 ± 0.10	28.50 ± 0.12	24.18 ± 0.12	27.89 ± 0.14
ARCU	58.55 ± 0.03	58.60 ± 0.04	55.46 ± 0.03	55.69 ± 0.01
EWT	24.58 ± 0.09	26.11 ± 0.06	21.26 ± 0.12	22.19 ± 0.11
EWP	27.86 ± 0.09	27.72 ± 0.08	22.57 ± 0.09	25.78 ± 0.10
ZCWT	30.44 ± 0.12	32.18 ± 0.13	29.75 ± 0.14	30.32 ± 0.14
SVM 100	15.75 ± 0.08	16.76 ± 0.09	16.45 ± 0.08	16.88 ± 0.08
SVM 150	15.17 ± 0.08	15.20 ± 0.09	14.24 ± 0.07	14.68 ± 0.07
MMC 100	14.30 ± 0.08	14.39 ± 0.07	14.30 ± 0.07	15.39 ± 0.08
MMC 150	15.21 ± 0.08	15.35 ± 0.07	14.72 ± 0.08	14.96 ± 0.08
Proposed 100	16.81 ± 0.08	16.89 ± 0.08	16.38 ± 0.07	17.77 ± 0.07
Proposed 150	15.73 ± 0.10	15.31 ± 0.09	14.95 ± 0.10	15.47 ± 0.09

Discriminant Analysis (PK-LDA) [8], Gaussian kernel-based Support Vector Machines (GK-SVM) [7], and Polynomial kernel-based Support Vector Machines (PK-SVM) [8]. To ensure a fair comparison, all the optimal parameters for the kernel functions, LDA, and SVM were chosen by an exhaustive search method, and a 10-fold cross validation was conducted. In terms of the recognition rate, Table 9 shows that the proposed method performed competitively compared with the SVM and MMC method.

7. Conclusion

In this study, we proposed a new criterion function for the "classifiability of a feature," which was based on a separability index matrix (*SIM*). The "classifiability" represents the classification capability of a feature set and we found that it was an

effective measure for evaluating the quality of a feature. We then proposed a new feature selection algorithm that performs relevance updates to identify the next best feature among the remaining features, which had lower computational costs than existing methods. We conducted extensive simulations to compare the performance of the proposed algorithm with other filter-type feature selection methods. The experimental results showed that the proposed algorithm outperformed other filter-type algorithms in terms of classification accuracy and computational costs. When applied we the developed method to an EMG signal-based gait phase recognition system, the proposed algorithm achieved better class-separability than several well-known wrapper-type feature selection methods and it also delivered competitive performance in terms of the recognition rate.

We envision that the proposed criterion function will be implemented with other types of search methods such as backward search or floating search. Our real-world example suggested that the proposed algorithm could be adopted for applications involving discrete datasets as well as for very large scale problems such as gene selection from microarray data.

Acknowledgement

This research was partly supported by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2010-C1090-1021-0010)

References

[1] R. Battiti, Using mutual information for selecting features in supervised neural net learning, IEEE Transactions on Neural Networks 5 (1994) 537–550.
[2] M. Dong, R. Kothari, Feature subset selection using a new definition of classifiability, Pattern Recognition Letters 24 (2003) 1215–1225.
[3] R.O. Duda, P.E. Hart, D.G. stork, Pattern Classification, second ed., John Wiley & sons, 2001.
[4] C. Fleischer, C. Reinicke, G. Hommel, Predicting the intended motion with EMG signals for exoskeleton orthosis controller, in: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems 2005, pp. 2029–2034.
[5] K. Fukunaga, Introduction to Statistical Pattern Recognition, second ed., Academic Press, 1990.
[6] D. Harville, Matrix Algebra From a Statistician's Perspective, Springer, 2008.
[7] Q. Hu, S. An, D. Yu, Soft fuzzy rough sets for robust feature evaluation and selection, Information Sciences 180 (2010) 4384–4400.
[8] T.-M. Huang, V. Kecman, I. Kopriva, Kernel Based Algorithms for Mining Huge Data Set: Supervised, Semi-supervised, and Unsupervised Learning, Springer, 2006.
[9] H. Jang, et al., Development of assistive walking device for disability, Final Report for YUDO Robotics. Inc., January 2007. (The copy of this reference was obtained with the permission of YUDO Robotics. Inc.)
[10] H. Kawamoto, S. Lee, Kanbe, Y. Sankai, Power assist method for HAL-3 using EMG based feedback controller, in: Proceedings of IEEE International Conference on Systems, Man and Cybernetics 2003, vol. 2, pp. 1648–1653.
[11] J. Kittler, Feature selection and extraction, in: Andrew Young (Ed.), Handbook of Pattern Recognition and Image Processing, Academic Press, San Diego, CA, 1986, pp. 60–83.
[12] S. Krishnan, K. Samudravijaya, P.V.S. Rao, Feature selection for pattern classification with Gaussian mixture models: a new objective criterion, Pattern Recognition Letters 17 (1996) 803–809.
[13] M. Kudo, J. Sklansky, Comparison of algorithms that select features for pattern classifiers, Pattern Recognition 33 (2000) 25–41.
[14] N. Kwak, C.H. Choi, Input feature selection by mutual information based on Parzen window, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (2002) 1667–1671.
[15] S. Maldonado, R. Weber, J. Basak, Simultaneous feature selection and classification using kernel-penalized support vector machines, Information Sciences 181 (2011) 115–128.
[16] S. Lee, Y. Sankai, Power assist control for walking aid with HAL-3 based on EMG and impedance adjustment around knee joint, in: Proceedings of IEEE International Conference on Intelligent Robots and Systems 2002, vol. 2, pp. 1499–1504.
[17] S. Lee, Y. Sankai, Power assist control for leg with HAL-3 based on virtual torque and impedance adjustment, in: Proceedings of IEEE International Conference on Systems, Man and Cybernetics 2002, vol. 4, p. 6.
[18] S. Lee, T. Yi, J.-S. Han, H. Jang, H.-H. Kim, J.-W. Jung, Z.Z. Bien, Walking phase recognition for people with lower limb disability, in: Proceedings of 10th IEEE International Conference on Rehabilitation Robotics, Noordwijk, The Netherlands, June 2007, pp. 60–67.
[19] H. Liu, H. Motoda, Feature Selection for Knowledge Discovery and Data Mining, Kluwer Academic Publishers, Boston, 1998.
[20] P. Murphy, D.W. Aha, UCI Repository of machine learning databases, University of California, Department of Information and Computer Science, Irvine, CA, 1994 <http://www.ics.uci.edu/~mlearn/MLRepository.html> (accessed 01.06.2007).
[21] S. Niijima, S. Kuhara, Recursive gene selection based on maximum margin criterion: a comparison with SVM-RFE, BMC Bioinformatics 7 (2006) 543.
[22] I.-S. Oh, J.-S. Lee, B.-R. Moon, Hybrid genetic algorithms for feature selection, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (2004) 1424–1437.
[23] H. Peng, F. Long, C. Ding, Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (2005) 1226–1238.
[24] B. Rosner, A generalization of the paired *t*-test, Journal of the Royal Statistical Society Series C, Applied Statistics 31 (1982) 9–13.
[25] P. Somol, P. Pudil, J. Kittler, Fast branch & bound algorithms for optimal feature selection, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (2004) 900–912.
[26] H. Tang, H. Maitre, No. Boujemaa, W. Jiang, On the relevance of linear discriminative features, Information Sciences 180 (2010) 3422–3433.
[27] R. Xia, C. Zong, S. Li, Ensemble of feature sets and classification algorithms for sentiment classification, Information Sciences 181 (2011) 1138–1152.