International Journal of Innovative
Computing, Information and Control
Volume **8**, Number **3(B)**, March **2012**

 ISSN 1349-4198
pp. **2089–2100**

MODIFIED SEQUENTIAL FLOATING SEARCH ALGORITHM WITH A NOVEL RANKING METHOD

Chien-Hsing Chou[1,*], Yi-Zeng Hsieh[2] and Chi-Yi Tsai[1]

[1]Department of Electrical Engineering
Tamkang University
No. 151, Yingjhuan Rd., Danshuei, Taipei 25137, Taiwan
*Corresponding author: chchou@mail.tku.edu.tw

[2]Department of Computer Science and Information Engineering
National Central University
No. 300, Jhongda Rd., Jhongli City, Taoyuan 32001, Taiwan

Received December 2010; revised April 2011

Abstract. *Feature selection plays a critical role in pattern classification. Of the various feature selection methods, the sequential floating search (SFS) method is perhaps the most well-known and widely adopted. This paper proposes a feature selection method combining feature ranking and SFS. The proposed feature ranking approach adopts the new idea of false features to rank features based on their importance, and then applies SFS to features that are less important or of lower rank. This approach overcomes issues with the original SFS and extracts more critical features. In addition, most feature selection methods do not consider the problem of multi-class classification. As a result, these methods have difficulty achieving good performance when dealing with a greater variety of classes. Therefore, this study adopts a one-against-all strategy to address this issue. The proposed approach divides multi-class classification into several binary classifications and adopts feature selection to derive individual feature subsets. This strategy achieves satisfactory performance in experimental simulations.*

Keywords: Feature selection, Sequential floating search, False feature, One-against-all, Pattern classification

1. **Introduction.** The choice of a suitable classification algorithm and feature selection is often the key to success in pattern classification. There are currently a variety of pattern classification methods, including nearest neighbors (NN) [1,2], k-nearest neighbors (KNN) [3,4], condensed nearest neighbor (CNN) [5], multilayer perception (MLP) [6] and support vector machines (SVM) [7,8]. The process of selecting an appropriate classification algorithm should consider not only the accuracy of the classifier, but also the equally important considerations of the time required for training and testing. For more discussions on classification algorithms, refer to [2]. This study focuses on the process of feature selection. A successful feature selection improves classification accuracy and extracts the critical features that users are concerned with. For instance, in the analysis research of DNA sequence, feature selection makes it possible to locate the segments on the sequence or the types of amino acids that may lead to certain diseases [9,10], or select the genes that may lead to certain diseases from the data of microarray [11]. Another example is text categorization, in which feature selection makes it possible to extract the keywords contributing to text classification [12-15]. In addition, feature selection not only selects the features that users are most interested in, but also saves time in training and testing the classifier, and reduces the memory space required for data storage.

Researchers have proposed many feature selection methods in recent years [9-24] while other have tested and compared these methods [12-20]. A feature selection algorithm involves the following four important steps (Figure 1) [16]. First, create a feature subset generated from an original feature set, then evaluate the subset to see if it matches the stopping criterion. If it does, proceed to result validation; otherwise, regenerate a new subset for evaluation and re-estimate it until the stopping criterion is met.

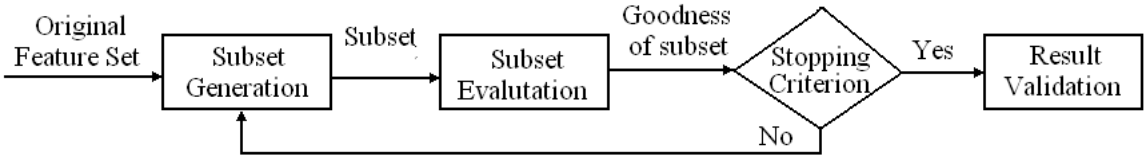


FIGURE 1. Four important steps in a feature selection algorithm

Guyon and Elisseeff [17] categorized feature selection algorithms as wrappers, filters and hybrid algorithms. The following discussion provides a brief introduction to these feature selection algorithms.

(1) Wrappers: The wrappers method extracts the best feature subset by adopting a specific searching strategy and performing constant evaluation [25-28]. These strategies include sequential floating search [25], adaptive floating search [26], branch and bound [27] and genetic algorithm [28], etc.
(2) Filters: The filters method ranks the importance of the features according to their statistical criteria or information-theoretic criteria [29,30]. This method usually uses information gain or X^2 statistic to extract the features in text categorization [12-14].
(3) Hybrid: The hybrid method extracts several feature subsets by combining both filters and wrappers through an independent feature evaluation method. The hybrid method extracts the best feature subset by processing the classification algorithm. This strategy is performed repeatedly until it is unable to obtain any better feature subsets [31,32].

The sequential forward floating search (SFFS) and sequential backward floating search (SBFS) algorithms proposed by Pudil et al. are the most widely adopted of these three methods [25]. In these two methods, users first set up a criterion function (e.g., the classification accuracy) and search for the best feature subset by constantly adding or removing a certain feature. When SFFS (or SBFS) evaluates the criterion function after adding (or removing) a feature from the feature set, the same performance of criterion function may be derived by different features. However, users do not have the necessary information to decide which feature to add (or remove) first. For example, to classify the Chinese characters "犬" and "大" in Figure 2, this study adopts SBFS and classification accuracy as the criteria function for SBFS. The accuracy rate can be increased by removing a single feature in the character areas of a, b or c. However, no additional information was provided to the users to distinguish the feature in which character areas is to be first removed. Therefore, in the end, the users must decide which character areas to be removed first. This issue often arises in classification problems, such as character recognition.

This study proposes a hybrid feature selection algorithm to overcome the problems listed above. Feature ranking is the first stage of this algorithm, and feature selection is the second. This study proposes a new feature ranking algorithm based on the concept of false features. This new feature ranking algorithm ranks each feature according to its importance, giving users the necessary information to decide which feature in the character areas should be removed first when they encounter the problems listed in Figure 2. In

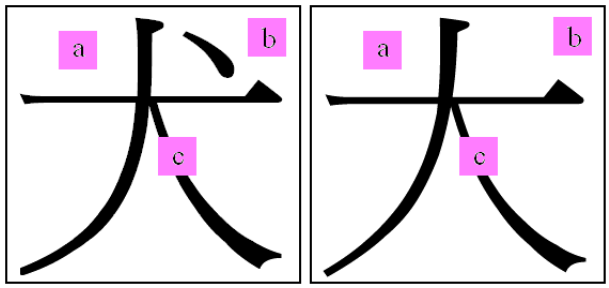
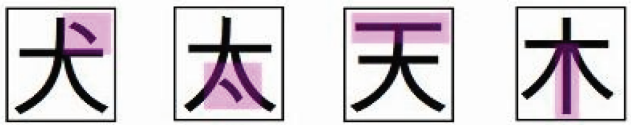


FIGURE 2. Removing either one of the features in the character area of a, b or c obtains the same accuracy rate

addition, the feature selection stage chooses those less important features to process the SBFS algorithm. As a result, this method can accurately extract critical features and increase the accuracy rate at the same time.

This study attempts to solve another problem occurring in feature selection algorithms. For most feature selection algorithms, the classification problem, which is either binary or multi-class classification, is not considered prior. When the classes of classification increase, it is difficult to achieve satisfactory performance for many feature selection methods. Consider the following example to illustrate this issue. Figure 3 shows four similar classes of Chinese characters to be classified. The critical area of character in Figure 3(a) on the right upper part (the block in pink) is for identifying the character "犬". For character "太", (Figure 3(b)) the critical area is the lower middle part. However, if we want to deal with these four characters at the same time, most feature selection algorithms cannot classify them by selecting just a few critical areas. Our simulations indicate that when there are a greater variety of character classes, most feature selection algorithms select more features as the feature subset for classification. Therefore, the meaning of feature selection is lost and the expected results cannot be achieved.



(a) (b) (c) (d)

FIGURE 3. Four similar Chinese characters. The pink block represents the critical area for classifying the corresponding character.

This study proposes a one-against-all strategy for feature selection to overcome this problem. The key point of this strategy is to partition the multi-class classification into several binary sub-classifications. Assuming that we have N classes, it is possible to generate N binary sub-classifications from the training dataset. An individual feature subset can be extracted for each binary sub-classification. During the result validation stage, a more appropriate feature subset is used for the corresponding sub-classification to improve the testing performance.

2. **The Proposed Hybrid Sequential Backward Floating Search.** This section presents a hybrid sequential backward floating search (HSBFS) algorithm. This algorithm includes two stages: feature ranking and feature selection. HSBFS gradually extracts a critical feature subset through the iterative process of feature ranking and feature selection. This study uses the nearest neighbor (NN) method as the classification method,

and uses the accuracy rate as the criterion function for evaluation. The following section introduces detailed steps of the HSBFS.

2.1. **Feature ranking method with false feature.** This study proposes a novel feature ranking method based on false features. The purpose of this approach is to rank the importance of all the features and select those of lesser importance to further execute the SBFS algorithm. The steps are listed as below:

Step 1: Assume that the original feature set contains D features. Then put m false features artificially into the D original features as the new full feature set. "False feature" means the feature is set to 0 in this case. The full feature set includes $D + m$ features, and the value of m can be customized by the users. It was set as $m = 5$ in this paper.

Step 2: Randomly generate K feature subsets from the full feature set. Denote each feature subset as S, which includes n features to be selected randomly from the full set (a feature can be reselected from the full set). The values K and n can be customized by users, and were set at $K = 640$ and $n = 0.25D$ in this paper.

Step 3: For a feature subset S, the classifier is constructed by nearest neighbor method and obtains a corresponding accuracy rate $\mathrm{Acc}(S)$.

Step 4: After obtaining accuracy rates $\mathrm{Acc}(S)$ for all feature subsets, evaluate the importance of each feature f (includes m false features) using the following formula:

$$importance(f) = \frac{\sum_{S \in \Lambda(f)} \mathrm{Acc}(S)}{|\Lambda(f)|} \tag{1}$$

where $\Lambda(f)$ is the collection of all generated subsets that incorporate f, and $|\Lambda(f)|$ is the number of these subsets. The larger the $importance(f)$ is, the more important the feature f is.

Step 5: Calculate the average importance of false features, T, from m false features.

Step 6: In D original features, generate the subset U by selecting the features with the importance less than T, and then process the feature selection with SBFS.

2.2. **Processing feature selection with SBFS.** Although the process of the feature selection stage is generally the same as the original SBFS, there are still some small details that are different. We list all these differences below.

(1) When executing SBFS, the removable feature is only for the features in the subset U.

(2) In the stage of searching for the removable features, removing a feature f can increase the accuracy rate (criterion function) or let it remain the same. We marked feature f as the removable candidate.

(3) For these features marked as removable candidates, the feature is selected to remove if it achieves the highest accuracy rate after the removal. If some features reach the same accuracy rate after the removal, remove the lowest importance feature first.

(4) In searching for an addable feature, the requirement is that the accuracy rate must be improved after adding any feature f.

(5) After executing SBFS, terminate the HSBFS algorithm if no more features need to be removed or added. Otherwise, re-execute the feature ranking stage and feature selection stage.

A key to the success of this algorithm is whether the feature ranking is effective. Our previous research on attractive faces classification [33] (in Appendix I) used similar computing concepts to Equation (1) and successfully ranked the influential features in attractive faces classification. The current study further combines a new concept of false features. If any particular feature is less important than the average of the false features,

we can reasonably assume that information for that feature is lacking. Without examining all the features, only a subset consisting of less important features is given to execute SBFS.

In addition, because the removable features are already extracted from the less important ones in feature selection stage, we only add a feature for sure when we know that the accuracy rate will increase. This kind of requirement is much stricter than that for removing features. Compared with the original SBFS, HSBFS achieves better accuracy rates and selects more critical features in our simulations.

3. **The Feature Selection Strategy to Address the Multi-Class Classification.** Most feature selection methods do not distinguish whether issues are caused by binary classification or multi-class classification. Experimental simulations indicate that the greater the number of classes for multi-class classification, the greater the difficulty in feature selection. Therefore, this study adopts a "one-against-all" strategy to overcome this issue. This strategy has been successfully adopted for pattern classification techniques such as SVM [7,8]. Therefore, we used the same concept to design a feature selection strategy to address the multi-class classification problem.

Assuming that there are N classes in the dataset, generate N binary sub-classifications, where each binary sub-classification (class i for instance) involves the dataset of class i and class *non-i*. Class *non-i* represents all other data patterns not belonging to class i. Next, use HSBFS and NN to extract individual feature subset to each sub-classification. Because the selected feature subsets of each sub-classification are different from the feature amounts, this study proposes a process to classify data pattern in the testing stage (Figure 4). We sent the data pattern $\underline{x}$ to each independent sub-classifier to calculate the corresponding $membership(i)$. The formula is listed below:

$$membership(i) = \frac{d_{non\text{-}i}}{d_i + d_{non\text{-}i}} \tag{2}$$

$$\text{where } d_i = \min_{\underline{x}_j \in Class\ i} ||\underline{x} - \underline{x}_j||, \tag{3}$$

$$d_{non\text{-}i} = \min_{\underline{x}_k \in Class\ non\text{-}i} ||\underline{x} - \underline{x}_k|| \tag{4}$$

where d_i is the shortest distance between $\underline{x}$ and the data pattern belongs to class i in the ith sub-classifier, and $d_{non\text{-}i}$ is the shortest distance to class *non-i*. The bigger the membership, the higher the possibility that $\underline{x}$ belongs to class i. Select the class with the largest membership as the classified class to data pattern $\underline{x}$. The formula is shown as below:

$$i^* = Arg \max_{i=1,\ldots,N} membership(i) \tag{5}$$

where i^* is the classified class to data pattern x.

4. **Experimental Results.** The experimental simulations in this study compared the SBFS and HSBFS algorithms. The parameters of the HSBFS algorithm were $M = 5$, $n = 64$ and $K = 640$. The datasets for the experiment consisted of combinations of ten similar Chinese characters selected from the ETL9b [34] handwritten Chinese characters database, where each class includes 200 handwritten Chinese characters. Figure 5 shows some examples of these characters, while Table 1 shows the four different datasets generated for simulation. The experimental simulations in this study partitioned each dataset into a training dataset and testing dataset. To extract feature subsets using feature selection methods, we used half of the data patterns in the training dataset and the nearest neighbor (NN) method [1,2] as the classification method to build a classifier. The

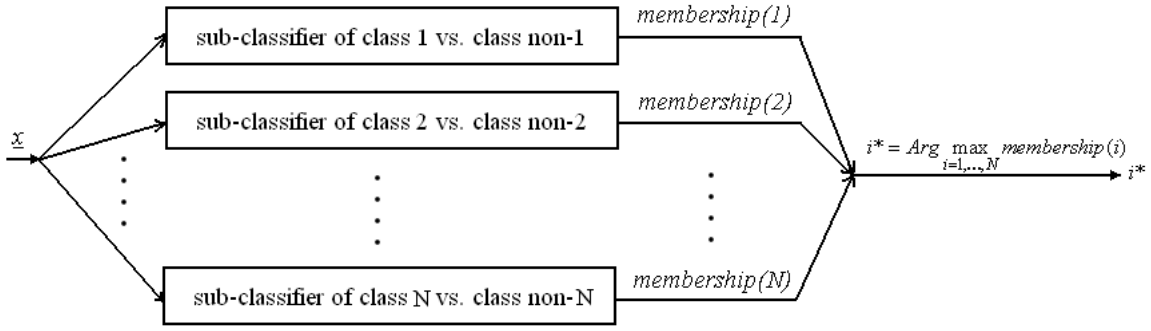


FIGURE 4. The process of classifying data pattern $\underline{x}$ in the testing stage

remaining data patterns in the training dataset were used to estimate the accuracy rate as a criterion function of the feature selection method. After extracting the feature subset using the specific feature selection method, the training dataset and the extracted feature subset were integrated to build a classifier for testing and obtain validation results from the testing dataset.

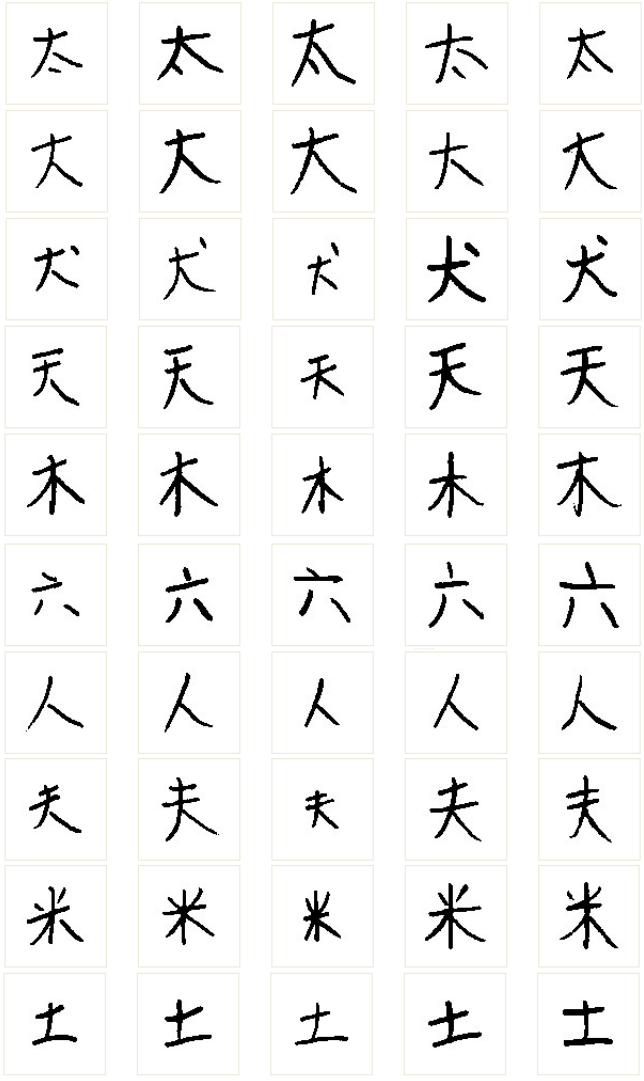


FIGURE 5. Some examples of ten classes of Chinese characters from ETL9b database

TABLE 1. Four datasets generated for experimental simulations

	Number of Class	Character Label	Number of Training Data	Number of Testing Data
Dataset 1	2	太, 大	267	133
Dataset 2	3	太, 大, 犬	400	200
Dataset 3	5	太, 大, 犬, 天, 木	667	333
Dataset 4	10	太, 大, 犬, 天, 木, 六, 人, 夫, 米,土	1,333	667

For each character image, we adopted the non-linear normalization technique [35] to normalize it to a size of 64×64. Each character image was then divided into 16×16 blocks (Figure 6). Each block consisted of 4×4 pixels, and compiled the numbers of the black pixels in each blocks to be the features [36]. If a block contains more black pixels, then the corresponding feature value is large. For examples, Figure 6 shows that the feature value of block 'a' is 10 because more than half of pixels are black pixels within block 'a'. And the feature value of block 'b' is 0, because there are no black pixels within block 'b.' A character image finally consists of 256 features (i.e., 16×16 blocks), whose values range from 0 to 16; each feature represents the information of a certain area (block) in a character image. Therefore, if a certain block is the critical area for classifying (e.g., block 'a'), the feature selection method should extract the corresponding feature. On other hand, the corresponding feature would not be extracted if the block is not the critical area for classifying (e.g., block 'b'). The following discussion presents the results of the experimental simulation.

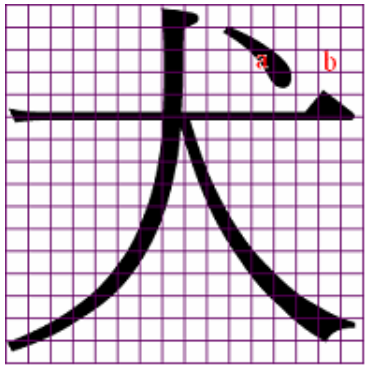


FIGURE 6. Character image divided into 16×16 blocks

Dataset 1: Dataset 1 included the Chinese characters '太' and '大'. Figure 5 shows some examples of two Chinese characters from the ETL9b database. For further analysis, the bottom of the central area of character image is the critical area to classify the characters '太' and '大' (Figure 7). If a feature selection method extracted more critical features from this area, the recognition result should be more accurate. Table 2 shows the simulation results of Dataset 1. Each of the two feature selection algorithms improved the accuracy rate after selecting its feature subset. SBFS extracted fewer features, but HSBFS obtained a higher accuracy rate than SBFS. Besides, HSBFS also spent less time than SBFS in seeking the corresponding feature subset.

Figure 8 displays the corresponding feature subsets selected by the two algorithms to verify the simulation results. If an algorithm selects a feature in the feature subset, the corresponding character area to this feature is labeled in pink. Examining the pink areas in the character image reveals which areas (i.e., features) are selected by the feature selection

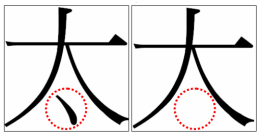
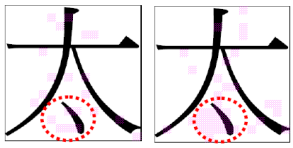


FIGURE 7. The critical area to classify the characters '太' and '大'

TABLE 2. Simulation results from Dataset 1

Dataset 1	Without Feature Selection	SBFS	HSBFS
Number of Selected Features	256	21	35
Accuracy of Test Data (%)	68.66	82.09	**91.04**
Computational Time for Feature Selection		0.8 hour	**0.4 hour**



(a) (b)

FIGURE 8. Feature subset selected by (a) SBFS; (b) HSBFS

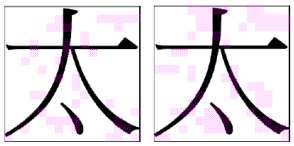
algorithms, and whether the algorithms really select the critical features for classification. A comparison of HSFBF and SBFS clearly shows that most of the features extracted by HSBFS are located in the bottom of the central area (Figure 8(b)). Obviously, this is the critical area for classifying the characters '太' and '大'. This is a reasonable explanation why HSBFS obtained a higher accuracy rate than SBFS.

Dataset 2: Dataset 2 includes the character '犬' in addition to the characters '太' and '大'. Table 3 shows the simulation results from Dataset 2, while Figure 9 shows the two feature subsets chosen by SBFS and HSBFS, respectively. These two feature selection methods increased the accuracy rate, but the effectiveness of SBFS was more limited. SBFS also extracted fewer features, while HSBFS obtained a higher accuracy rate and required less time for computation.

TABLE 3. Simulation results from Dataset 2

Dataset 2	Without Feature Selection	SBFS	HSBFS
Number of Selected Features	256	44	67
Accuracy of Test Data (%)	63.5	69.5	**79.5**
Computational Time for Feature Selection		1.8 hours	**0.8 hour**

Because the classes have been increased to three groups in this problem, both methods extracted more features. Therefore, we adopted HSBFS and the one-against-all strategy. Table 4 shows the result after applying HSBFS in combination with the one-against-all strategy. The one-against-all strategy individually extracted 34, 59 and 39 features for the '太', '大' and '犬' sub-classifications, respectively. Figure 10 shows the selected feature subsets of these three classes. The feature subset selected to discriminate class

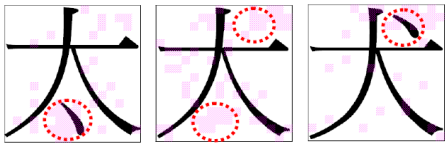


(a) (b)

FIGURE 9. Feature subset selected by (a) SBFS; (b) HSBFS

TABLE 4. Simulation result of Dataset 2 by applying one-against-all strategy

Dataset 2	HSBFS with one-against-all strategy		
Number of Selected Features	Class 太	Class 大	Class 犬
	34	59	39
Accuracy of Test Data (%)	**88**		
Computational Time for Feature Selection	1.4 hours		



(a) (b) (c)

FIGURE 10. Feature subset selected for (a) Class '太'; (b) Class '大'; (c) Class '犬'

'太' from another two classes '大' and '犬' is mainly located in the lower central area (see Figure 10(a)); the features mainly located in the upper right part are selected for the feature subset to discriminate class '犬' from '太' and '大' (see Figure 10(c)). The feature subset selected to discriminate class '大' from the other two classes covers the two above-mentioned areas (see Figure 10(b)). These results perfectly matched our intuition and expectations. In addition, the accuracy rate increased from 79.5% to 88%.

Dataset 3: Dataset 3 includes the characters '太', '大', '犬', '天' and '木'. Table 5 shows the simulation results of Dataset 3. Figure 11 shows the selected feature subset from SBFS and HSBFS, respectively. The number of features increased, while the disparity between HSBFS and SBFS in terms of the accuracy rate decreased. Table 6 lists the results of implementing the one-against-all strategy with HSBFS. There are 62, 67, 61, 16 and 18 features to be extracted from the five sub-classifications, respectively. Figure 12 shows the selected feature subsets of the five classes. When using the one-against-all strategy, the accuracy rate increased from 81.74% to 89.22%. At the same time, comparing the results of Figures 11(b) and 12(a)-12(e) shows that adopting the one-against-all strategy can really help users identify the critical features they are interested in. This method is also perfectly suited to overcoming the multi-class classification problem.

Dataset 4: Dataset 4 includes 10 similar characters. Table 7 shows the simulation results from Dataset 4. There was a greater number of selected features and not such a significant increase in the accuracy rate of SBFS or HSBFS. After adopting the one-against-all strategy (see Table 8), the accuracy rate increased from 80.51% to 88.61% and there were less selected features. For classes '人' and '土', only 15 and 9 features were selected to deal with the corresponding sub-classifications, respectively.

TABLE 5. Simulation results from Dataset 3

Dataset 3	Without Feature Selection	SBFS	HSBFS
Number of Selected Features	256	85	106
Accuracy of Test Data (%)	73.35	79.04	**81.74**
Computational Time for Feature Selection		7 hours	**2.1 hours**

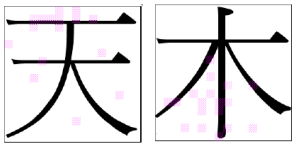
TABLE 6. Simulation result of Dataset 3 by applying one-against-all strategy

Dataset 3	HSBFS with one-against-all strategy				
Number of Selected Features	Class 太	Class 大	Class 犬	Class 天	Class 木
	62	67	61	16	18
Accuracy of Test Data (%)	**89.22**				
Computational Time for Feature Selection	11.6 hours				

(a) (b)

FIGURE 11. Feature subset selected by (a) SBFS; (b) HSBFS

(a) (b) (c)



(d) (e)

FIGURE 12. Feature subset selected for (a) Class '太'; (b) Class '大'; (c) Class '犬'; (d) Class '天'; (e) Class '木'

TABLE 7. Simulation results from Dataset 4

Dataset 4	Without Feature Selection	SBFS	HSBFS
Number of Selected Features	256	138	185
Accuracy of Test Data (%)	77.06	77.51	**80.51**
Computational Time for Feature Selection		68 hours	**12 hours**

TABLE 8. Simulation result of Dataset 4 by applying one-against-all strategy

Dataset 4	HSBFS with one-against-all strategy									
Number of Selected Features	太	大	犬	天	木	六	人	夫	米	土
	80	115	62	29	69	32	15	67	57	9
Accuracy of Test Data (%)	88.61									
Computational Time for Feature Selection	80 hours									

5. **Conclusion.** This study is the first to introduce the HSBFS algorithm. This algorithm is able to overcome the problems involved in sequential floating search and can extract the critical feature subset more accurately and effectively. This study also adopts a one-against-all strategy to improve the effectiveness of feature selection methods and address the multi-class classification problem. The accuracy rates in three experimental simulations increased by at least 8%. Although these simulations show that the proposed one-against-all strategy provides satisfactory performance, the computational cost is expensive if the class number is large. To overcome this limitation, the one-against-all strategy makes it possible to accomplish training on decomposed classes in parallel processing. Another way to decrease computational cost is to reduce the amount of data in the training data set. Therefore, for training a binary sub-classification (Class i for instance), it is possible to select part of relative classes as Class *non-i*, instead of using all other data patterns that do not belong to Class i.

Acknowledgement. This work was supported by the National Science Council, Taiwan, under the Grant NSC 99-2221-E-238-017, NSC 99-2622-E-032-004-CC3 and NSC 100-2221-E-032-069.

REFERENCES

[1] T. Cover and P. E. Hart, The nearest neighbor rule for small samples drawn from uniform distributions, *IEEE Trans. Information Theory*, vol.13, no.1, pp.21-27, 1967.
[2] R. O. Duda, P. E. Hart and D. G. Stork, *Pattern Classification*, Wiley, New York, 2001.
[3] A. Levine, L. Lustick and B. Saltzberg, The nearest neighbor rule for small samples drawn from uniform distributions, *IEEE Trans. Information Theory*, vol.19, no.5, pp.697-699, 1973.
[4] J. F. O'Callaghan, An alternative definition for "neighborhood of a point", *IEEE Trans. Computers*, vol.24, no.11, pp.1121-1125, 1975.
[5] P. Hart, The condensed nearest neighbor rule, *IEEE Trans. Information Theory*, vol.14, pp.515-516, 1968.
[6] S. Haykin, *Neural Network*, 2nd Edition, Pretince Hall, 1999.
[7] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer Verlag, New York, 1995.
[8] C. Cortes and V. Vapnik, Support vector machines, *Machine Learning*, vol.20, pp.1-25, 1995.
[9] H. Liu, J. Li and L. Wong, A comparative study on feature selection and classification methods using gene expression profiles and proteomic patterns, *Genome Informatics*, vol.13, pp.51-60, 2002.
[10] T. Li, C. Zhang and M. Ogihara, A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression, *Bioinformatics*, vol.20, no.15, pp.2429-2437, 2004.
[11] E. P. Xing, M. I. Jordan and R. M. Karp, Feature selection for high-dimensional genomic microarry data, *International Conference on Machine Learning*, pp.601-608, 2001.
[12] Y. Yang and J. O. Pedersen, A comparative study on feature selection in text categorization, *International Conference on Machine Learning*, pp.412-420, 1997.
[13] T. Liu, S. Liu, Z. Chen and W. Y. Ma, An evaluation of feature selection for text categorization, *International Conference on Machine Learning*, 2003.
[14] G. Forman, An extensive empirical study of feature selection metrics for text classification, *Journal of Machine Learning Research*, vol.3, pp.1289-1305, 2003.

[15] T. Joachims, Text categorization with support vector machines: Learning with many relevant features, *Proc. of the European Conference on Machine Learning*, 1998.
[16] H. Liu and L. Yu, Toward integrating feature selection algorithms for classification and clustering, *IEEE Trans. on Knowledge and Data Engineering*, vol.17, no.4, pp.491-502, 2005.
[17] N. Guyon and A. Elisseeff, An introduction to variable and feature selection, *Journal of Machine Learning Research*, vol.3, pp.1157-1182, 2003.
[18] R. Kohavi and G. John, Wrappers for feature selection, *Artificial Intelligence*, vol.97, no.1-2, pp.273-324, 1997.
[19] A. Blum and P. Langley, Selection of relevant features and examples in machine learning, *Artificial Intelligence*, vol.97, no.1-2, pp.245-271, 1997.
[20] A. Jain and D. Zongker, Feature selection: Evaluation, application, and small sample performance, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.19, no.2, pp.153-158, 1997.
[21] G.-J. Mun, B.-N. Noh and Y.-M. Kim, Enhance stochastic learning for feature selection in intrusion classification, *International Journal of Innovative Computing, Information and Control*, vol.5, no.11(A), pp.3625-3635, 2009.
[22] C.-C. Lai, C.-H. Wu and M.-C. Tsai, Feature selection using particle swarm optimization with application in spam filtering, *International Journal of Innovative Computing, Information and Control*, vol.5, no.2, pp.423-432, 2009.
[23] Y. Chen, J. Chang and C. Cheng, Forecasting IPO returns using feature selection and entropy-based rough sets, *International Journal of Innovative Computing, Information and Control*, vol.4, no.8, pp.1861-1875, 2008.
[24] H. Benítez-Pérez and A. Benítez-Pérez, The use of armax strategy and self organizing maps for feature extraction and classification for fault diagnosis, *International Journal of Innovative Computing, Information and Control*, vol.5, no.12(B), pp.4787-4796, 2009.
[25] P. Pudil, J. Novovičová and J. Kittler, Floating search methods in feature selection, *Pattern Recognition Letters*, vol.15, no.11, pp.1119-1125, 1994.
[26] P. Somol, P. Pudil, J. Novovičvá and P. Paclík, Adaptive floating search methods in feature selection, *Pattern Recognition Letters*, vol.20, no.11-13, pp.1157-1163, 1999.
[27] P. Somol, P. Pudil and J. Kittler, Fast branch & bound algorithms for optimal feature selection, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.26, no.7, pp.900-912, 2004.
[28] W. Siedlecki and J. Sklansky, A note on genetic algorithms for large-scale feature selection, *Pattern Recognition Letters*, vol.10, no.5, pp.335-347, 1989.
[29] I. Dhillon, S. Mallela and R. Kumar, A divisive information-theoretic feature clustering algorithm for text classification, *Journal of Machine Learning Research*, vol.3, pp.1265-1287, 2003.
[30] K. Torkkola, Feature extraction by non-parametric mutual information maximization, *Journal of Machine Learning Research*, vol.3, pp.1415-1438, 2003.
[31] S. Das, Filters, wrappers and a boosting-based hybrid for feature selection, *Proc. of the 18th Int. Conf. Machine Learning*, pp.74-81, 2001.
[32] E. Xing, M. Jordan and R. Karp, Feature selection for high-dimensional genomic microarray data, *Proc. of the 15th Int. Conf. Machine Learning*, pp.601-608, 2001.
[33] F. Chang and C. H. Chou, Bi-prototype theory for facial attractiveness, *Neural Computation*, vol.21, no.3, pp.890-910, 2009.
[34] R. Collobert, S. Bengio and J. Mariéthoz, Torch: A modular machine learning software library, *Technical Report IDIAP-RR 02-46*, 2002.
[35] H. Yamada, K. Yamamoto and T. Saito, A nonlinear normalization method for handprinted Kanji character recognition – Line density equalization, *Pattern Recognition*, vol.23, no.9, pp.1023-1029, 1990.
[36] C. H. Chou, C. Y. Kuo and F. Chang, Recognition of fragmented characters using multiple feature-subset classifiers, *The 9th International Conference on Document Analysis and Recognition*, vol.1, pp.198-202, 2007.