International Journal of Computational Intelligence Systems, Vol. 8, No. 3 (2015) 490-501

Neural Incremental Attribute Learning in Groups

Fangzhou Liu
[1]*Department of Computer Science, University of Liverpool*
Liverpool, L69 3BX, UK
[2]*Department of Computer Science & Software Engineering, Xi'an Jiaotong-Liverpool University*
Suzhou, 214125, China
E-mail: fangzhou.liu@liverpool.ac.uk
www.liv.ac.uk

Ting Wang*
[3]*State Key Laboratory of Intelligent Technology and Systems,*
Tsinghua National Laboratory for Information Science and Technology,
Department of Computer Science and Technology, Tsinghua University
Beijing, 100084, China
[4]*Research Center of Web Information and Social Management,*
Wuxi Research Institute of Applied Technologies, Tsinghua University
Wuxi, 214072, China
E-mail: tingwang@tsinghua.edu.cn
www.tsinghua.edu.cn

Sheng-Uei Guan
Department of Computer Science & Software Engineering, Xi'an Jiaotong-Liverpool University
Suzhou, 214125, China
E-mail: steven.guan@xjtlu.edu.cn
www.xjtlu.edu.cn

Ka Lok Man
Department of Computer Science & Software Engineering, Xi'an Jiaotong-Liverpool University
Suzhou, 214125, China
E-mail: ka.man@xjtlu.edu.cn
www.xjtlu.edu.cn

Received 16 December 2013

Accepted 22 January 2015

Abstract

Incremental Attribute Learning (IAL) is a feasible approach for solving high-dimensional pattern recognition problems. It gradually trains features one by one. Previous research indicated that supervised machine learning with input attribute ordering can improve classification results. Moreover, input space partitioning can also effectively reduce the interference among features. This study proposed IAL based on Grouped Feature Ordering, which fused feature partitioning with feature ordering. The experimental results show that this approach is not only applicable for pattern classification improvement, but also efficient to reduce interference among features.

Keywords: Incremental Attribute Learning; Feature Ordering; Feature Grouping; Neural Networks; Pattern Classification; Feature Discrimination Ability.

* Corresponding author.

1. Introduction

High-dimensional problems are typically cursed with dimensional disasters in problem solving. To solve these problems, some dimensional reduction strategies like feature selection and feature extraction have been presented[1, 2]. However, these methods are invalid when the problem has a large number of features and all the features are significant. Thus feature reduction is not the ultimate solution to high dimensional problems.

A useful strategy for solving high-dimensional problems is "divide-and-conquer", where a complex problem is firstly separated into some smaller modules by features. These modules will be integrated after they have been tackled independently. A representative of such methods is Incremental Attribute Learning (IAL), which incrementally trains pattern features in one or more size. It has been shown as an applicable approach for solving machine learning problems in regression and classification using Genetic Algorithm (GA) [3, 4], Neural Network (NN) [5, 6], Support Vector Machine (SVM) [7], Particle Swarm Optimization (PSO) [8], Decision Tree [9], and so on. These previous studies also showed that IAL can exhibit better performance than conventional methods which often train all pattern features in one batch. IAL can outperform conventional methods, because IAL can reduce the interference brought by different input features[10]. If features are trained together by conventional methods in one batch, the interference between each other cannot be erased.

There are two different ways to reduce interference in IAL. One is Feature Ordering[6, 11-13], and the other is Feature Partition[10, 14, 15]. In previous studies, these two methods have been successively and independently verified as useful IAL preprocessing methods for final result improvement. However, what kind of influence will be brought by the integration of feature ordering and partition is still unknown. In previous research, few studies have been implemented to employ Feature Ordering and Feature Partition together in IAL. Thus in this study, it is very important to investigate whether they are applicable to improve the final classification performance.

In this paper, a new feature preprocessing method for IAL called as Grouped Feature Ordering is presented based on Accumulative Discriminability (AD), a metric for feature's discriminative ability calculation. This approach combines feature ordering and partition. Based on the grouped features partitioned and sorted by this approach, all the data will be trained by IAL algorithms. As a neural network algorithm of IAL, incremental neural network training with an increasing input dimension (ITID) [6] is employed to test the applicability and accuracy of this new approach. Literature review and background knowledge of IAL and its preprocessing will be introduced in Section 2. Section 3 will introduce Grouped Feature Ordering including its working model. Benchmarks with datasets from UCI will be tested out in Section 4 followed by some experimental result analysis, and conclusions will be drawn in the last section.

2. Literature Review and Previous Work

2.1. *IAL*

A number of previous studies have shown that IAL often exhibits better performance than other conventional machine learning techniques that train data in one batch. For example, based on datasets from University of California at Irvine (UCI) Machine Learning Repository, Guan et al. employed IAL to solve several classification and regression problems by NN [5, 6, 10-13, 16-19], PSO [8] and GA [3, 4]. Almost all of their results using IAL were better than those derived from traditional methods. For instance, based on Incremental Learning in terms of Input Attributes (ILIA) [5] and ITID[11], two IAL algorithms, final classification errors obtained by incremental neural networks for input feature learning of Diabetes, Thyroid and Glass datasets reduced by 8.2%, 14.6% and 12.6%, respectively [6, 11]. Furthermore, based on OIGA, the testing error rates derived by incremental genetic algorithms of Yeast, Glass and Wine declined by 25.9%, 19.4% and 10.8% [3], respectively, in classification. Further, Ang et al. proposed interference-less networks in his paper. He divided features into several groups without interference in the same group. Such an approach led to more acceptable results from the experiments [10].

Moreover, other researchers also confirmed that IAL is applicable to improve final pattern recognition results. For example, Chao et al. used a decision tree to implement IAL, and presented Intelligent, Incremental and Interactive Learning (i^+Learning) and i^+Learning regarding attributes (i^+LRA) in their paper [9]. These algorithms were employed to run in 16 different datasets supplied by UCI. The results indicated that the algorithms based on IAL performed better than ITI in 14 of the 16 datasets. Furthermore, Agrawal and Bala presented an incremental Bayesian classification approach for multivariate normal distribution data. In their experiments, features are imported one by one into Bayesian classifier. Their experimental results also

demonstrates that feature-based incremental Bayesian classifier is computationally efficient over batch Bayesian classifier in terms of time, although both of the results derived by these two methods are equivalent [20]. In addition, successful research on incremental SVM extended IAL to a wider application field[7]. All of these previous IAL studies showed that IAL can indeed improve the performance of pattern recognition. These studies denoted that different feature orderings can produce different pattern recognition results. Feature Ordering is a unique preprocessing step of IAL.

2.2. *Neural IAL*

In previous studies, ITID plays an important role in the training of neural IAL. It is applicable for both classification and regression. When ITID is employed, it divides the whole input space into several sub dimensions, each of which corresponds to an input feature. Instead of learning input features altogether as an input vector in a training instance, ITID learns input features one after another through their corresponding sub-networks, and the structure of NN gradually grows with an increasing input dimension based on ILIA [5]. During training, information obtained by a new sub-network is merged together with the information obtained by the old network. Such architecture is based on ILIA1. After training, if the outputs of NN are collapsed with an additional network sitting on the top where links to the collapsed output units and all the input units are built to collect more information from the inputs, this results in ILIA2 as shown in Fig. 1. Finally, a pruning technique is adopted to find out the appropriate network architecture. With less internal interference among input features, ITID achieves higher generalization accuracy than conventional methods [6].

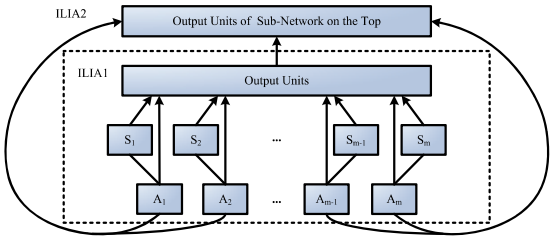

Fig. 1. The network structure of ITID

2.3. *IAL Feature Ordering*

Due to the fact that IAL incrementally imports features into systems, it is necessary to know which features should be introduced earlier. Thus feature ordering should be implemented as an independent preprocess apart from conventional preprocessing tasks like feature selection and feature extraction[21]. Feature Ordering aims to sort features based on feature's discriminative ability, so that when IAL is employed, features can be imported according to the ordering. Previous studies discovered that feature ordering relies on feature's discriminative ability. There are two approaches of feature ordering: contribution-based feature ranking[6, 11, 12] and metric-based feature ranking[21-26]. Both approaches can exhibit better performance than the conventional batch-training approach.

More specially, contribution-based methods focus on estimating each feature's individual contribution, where discriminative ability of each feature is calculated by some predictive algorithms like NN [6, 11, 12] and GA [4]. However, such an approach is time-consuming. In another hand, metric-based methods predict each feature's discriminative ability by some measurement such as mRMR [25] and correlations [22]. Comparing with contribution-based methods, metric-based methods are more efficient.

In previous research, several metrics for feature ordering have been discovered, such as mRMR [25, 27], Entropy [23], Single Discriminability (SD) [21], Evolving Linear Discriminant [24] and Fisher Score [26]. Furthermore, based on the accumulative feature discrimination ability, the Evolving Linear Discriminant exhibits the most stable performance.

2.4. *Maximum Mean Discriminative Criterion*

Maximum Mean Discriminative Criterion (MMDC)[24] is very useful to judge whether an IAL feature ordering is optimum or not. According to this criterion, an optimum feature ordering should have the greatest feature discrimination ability all the time when features are successively imported into IAL training system. Namely, when the feature space is growing along with the import of new features, pattern distribution in this growing feature space should always have the greatest discrimination ability. For this purpose, AD was developed based on some single feature discrimination ability metrics, like mRMR [25, 27], Entropy [23], and SD [21]. These single feature discrimination ability metrics aim to measure a single feature's discrimination ability, for example, to find a hyperplane in a one-dimensional feature space for classification. However, AD is quite

different. It aims to measure the discrimination ability of a multi-dimensional space consisted by more than one features. Namely, AD aims to find a hyperplane in a multi-dimensional feature space for classification. Specially, in one-dimensional feature space, the value of AD equals to that of SD for the same feature. Obviously, if there are *n* features existing in the classification problem, they will bring *n* different AD values along with the increase of feature importing.

Assuming that the discrimination ability of a feature or a feature space is a predictive symbol for final classification rates, namely, greater discrimination ability refers to lower classification error rates, it is necessary to ensure that pattern datasets should always have the greatest discrimination ability in every feature importing step. Thus according to MMDC, the feature with the largest AD should be selected as the first feature. Then the second feature will be selected. In this selection, it is necessary to make sure that the new feature selected from those remaining features except the first feature have the largest AD in the feature space consisted by itself and the first feature. Further, in the next steps, it is also necessary to guarantee that the newly imported feature can make the growing feature space have the largest AD with all previously imported features. In this way, feature ordering with the largest AD all the time can guarantee that different classes can be separated in the easiest way.

Therefore, with the aim for optimum classification results, each intermediate step will produce an optimal feature with the greatest discrimination ability for each round of feature importing. Obviously, after all features are imported, the resulting feature ordering will have the largest sum or mean of accumulative feature discrimination ability calculated in each step of the process. Here, AD and MMDC are employed to obtain the optimum feature ordering can be given with maximum discrimination ability mean by

$$\max \frac{1}{d}\sum_{d=1}^{d} AD(\mathbf{f}_{1:d}), (1 \le d \le m) \tag{1}$$

where $\mathbf{f}_{1:d}$ is the feature subset of $\{f_1, f_2, \dots, f_m\}$ during the feature importing process. The mean with a greater value indicates that the corresponding feature ordering has greater discrimination ability than the others. Hence, MMDC is able to select the optimum feature ordering.

In Eq.(1), AD is the ratio in *d*-feature space between the multi-dimensional standard deviation of all class centers and the sum of all multi-dimensional standard deviations of all patterns in each class.

If $\{f_1, f_2, \dots, f_m\}$ is the pool of input features, $\mathbf{f} = \{f_{k,d}\}_{k=1,d=1}^{k=r,d=m} \in R_{feature}^{r\times m}$, when the $d^{th}(1 \le d \le m)$ feature is imported, AD is

$$AD(f_1, f_2, \dots, f_d) = \frac{std\left[\left(\tilde{\boldsymbol{\mu}}_j\right)_{j=1}^{j=n}\right]}{\sum_{j=1}^{j=n} std\left[(f_i)_{i=1}^{i=d}\right]_j}, \quad (1 \le d \le m) \tag{2}$$

where $\tilde{\boldsymbol{\mu}}_\mathbf{j}$ is the centroid of vector $(f_1, f_2, \dots, f_d)$ with patterns belonging to *j*.

Therefore, the results of Eq.(2) are calculated on the run when new features are gradually imported into training. To obtain better classification results, it is necessary to ensure the result of Eq.(2) is the maximum in every step of feature importing. Here, *std* denotes the standard deviation in multi-dimensional space, which is derived by the standard deviation and Euclidean norm.

Let **x** be the vector for standard deviation calculation, the standard deviation of **x** is

$$std(\mathbf{x}) = \sqrt{\frac{\sum_{k=1}^{k=r}(x_k - \mu)^2}{r-1}} \tag{3}$$

where the vector $\mathrm{x} = \{x_k\}_{k=1}^{k=r}$, x_k is the value of k^{th} pattern, and *r* is the total number of patterns. Obviously, in Eq.(3), the component $(x_k - \mu)$ is a distance between k^{th} pattern and its mean. Thus, let *dist* replace this part, then Eq.(4) can be re-written as:

$$std(\mathbf{x}) = \sqrt{\frac{\sum_{k=1}^{k=r} dist_{x_k,\mu}{}^2}{r-1}} \tag{4}$$

where $dist_{x_k,\mu}$ denotes the distance of k^{th} pattern in **x** and its mean μ. If $\|D\|$ is the Euclidean norm of *d*-dimensional feature space, Eq.(4) can be given in a high-dimensional style by:

$$std(\mathbf{x}) = \sqrt{\frac{\sum_{k=1}^{k=r}\left\|D_{x_k,\tilde{\boldsymbol{\mu}}}\right\|^2}{r-1}} \tag{5}$$

where $\tilde{\boldsymbol{\mu}}$ is the centroid of **x**, and

$$\left\|D_{x_k,\tilde{\boldsymbol{\mu}}}\right\| = \sqrt{\sum_{i=1}^{d}\left(x_{k,i} - \mu_i\right)^2}. \tag{6}$$

Here *d* is the total number of features imported so far. Therefore, to calculate the standard deviation of *r* patterns in two dimensions, Eq.(5) can be written as

$$std(\mathbf{x}) = \sqrt{\frac{\sum_{k=1}^{k=r}\left[\left(x_{k,1} - \mu_1\right)^2 + \left(x_{k,2} - \mu_2\right)^2\right]}{r-1}}, \tag{7}$$

and for a tri-dimensional space, the equation is

$$std(\mathbf{x}) = \sqrt{\frac{\sum_{k=1}^{k=r}\left[\left(x_{k,1}-\mu_1\right)^2+\left(x_{k,2}-\mu_2\right)^2+\left(x_{k,3}-\mu_3\right)^2\right]}{r-1}}, \quad (8)$$

Accordingly, multi-dimensional standard deviation used in Eq.(2) of r patterns in an m-dimensional space is

$$std(\mathbf{x}) = \sqrt{\frac{\sum_{k=1}^{k=r}\sum_{i=1}^{i=m}\left(x_{k,i}-\mu_i\right)^2}{r-1}}, \quad (9)$$

$$\mathbf{x} = \left\{x_{k,d}\right\}_{k=1,d=1}^{k=r,d=m} \in \mathrm{R}_{feature}^{r\times m}$$

3. Grouped Feature Ordering

In previous batch training studies, interference among features is regarded as one of the main reasons of high error rates. It is very difficult to reduce error rates in pattern recognition. Ang et al. employed input space partitioning to get rid of the interference among input features [10, 14], which showed that assembling features with less interference in one group can decrease pattern recognition error rates. Similarly, IAL gradually and individually trains features one after another according to the feature ordering, which also can be treated as a feature separating process in training. Thus both feature ordering and feature grouping are able to reduce interference and produce lower error rates. Hence if feature ordering and feature grouping are employed together, it is likely to produce better results than each individual approach.

In this research, feature grouping is carried out together with feature ordering, in order to simplify the calculation. In previous research, feature grouping depends on calculating all feature's single and pairwise contribution, which is too complex and time-consuming to handle for large-scale problems. In the integrated process of feature grouping and ordering, all feature's contributions are not detected in pairs. Based on the obtained ordering, only features in the neighboring place will be considered whether it is necessary to be put in one group and be trained by batch in this group. It is manifest that such a process is much more efficient. The steps of Grouped Feature Ordering process is show as follows with Figure 2.

Step 1: Feature Ordering based on training dataset is calculated according to MMDC and AD.

Step 2: Training dataset is employed, and features are introduced into the predictive systems one by one. Training error rates of each importing step are obtained.

Step 3: Training Repetition. If the error rate derived by the later step is ***equal*** to or ***greater*** than that in the previous step, then the later one should be grouped with the previous features, and they will be trained again in one batch for the prediction; otherwise, the later one should be solely imported and trained again.

Step 4: Validation and Testing.

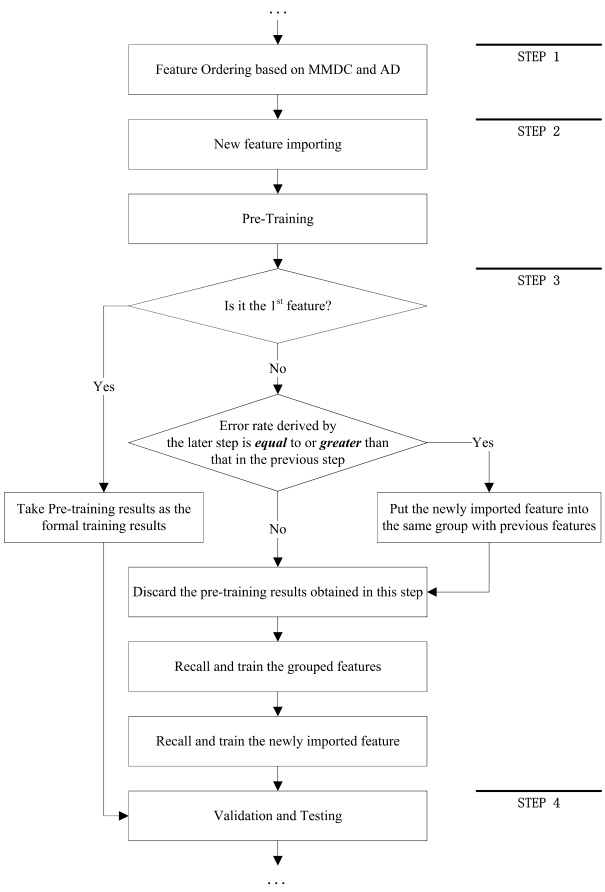

Fig.2. Whole Process of Grouped Feature Ordering

4. Benchmarks

The proposed IAL method with Grouped Feature Ordering using ITID was tested on six benchmarks from UCI machine learning datasets. They are Diabetes, Glass, Thyroid, Ionosphere, Musk1, and Semeion. All these six datasets are classification problems. In these experiments, all the patterns were randomly divided into three groups: training set (50%), validation set (25%) and testing set (25%). Especially, the training data were firstly used to rank feature ordering and sort with groups based on AD in the first place as a preprocessing task. After that, ILIA2 was employed for classification according to this feature ordering in the following steps.

Furthermore, to evaluate the performance, results are compared with those derived by AD feature ordering, contribution-based ordering, original orderings, and

conventional batch training. Different from the first three approaches which are based on IAL, the last approach employed neural networks and trained all the features in one batch. More specifically, the first approach employed the feature ordering method based on AD to sort features according to their discrimination ability in descending order, and then it was trained by ITID. The second approach also trained by ITID after it sorted all features by their single classification contributions which were derived by each single training and testing before the formal training. The third approach did not sort features, it trains all features directly based on IAL, while the last approach was not an IAL approach, which was a traditional neural network training approach.

4.1. *Diabetes*

Figure 3 shows error rates derived in each step of training for Diabetes. In the testing, if the error rate derived by the later step is equal to or greater than that in the previous step, then the later imported feature should be grouped with the previous features, and they will be trained again in one batch for the prediction; otherwise, the later imported feature should be solely introduced into the predictive system and trained again.

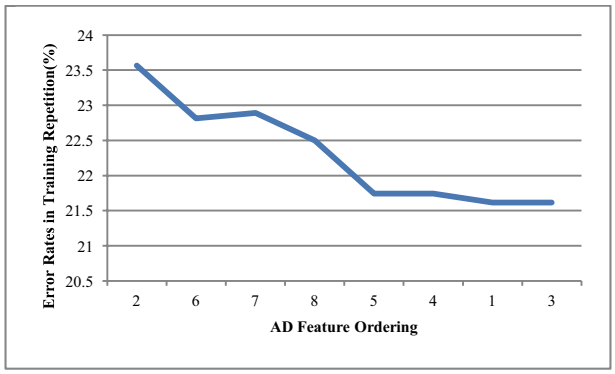

Fig.3. Error Rates in Training Repetition of Diabetes

Table 1 presents the classification error rates of different experiments using Diabetes dataset and the improvement percentages versus the conventional batch-training method. It is manifest that the approach using grouped feature ordering derived by AD achieved the best result with the lowest classification error as well as the IAL approach with the contribution-based feature ordering. Other approaches are not as good as these two approaches, and the conventional batch-training approach exhibited the worst performance.

4.2. *Glass*

In the benchmark of Glass, Figure 4 shows error rates derived in each step of training of Glass according to AD feature ordering. Features which got higher error rates than its previous features were trained together with the previous one in one group. The grouped feature ordering and its classification results are shown in Table 2. According to this table, the approach with grouped feature ordering based on AD and the approach using AD feature ordering without groups obtained the same lowest error rates (29.24530%). Classification results of other approaches are much worse than those derived from the AD-based approaches.

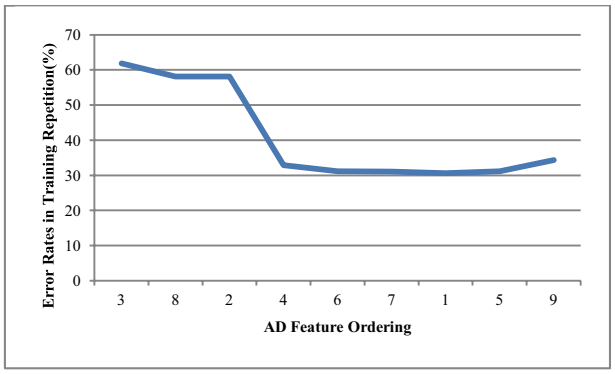

Fig.4. Error Rates in Training Repetition of Glass

4.3. *Thyroid*

The performance of Thyroid in training repetition is shown in Figure 5, where four feature groups were obtained at last. They were trained with other single features according to AD feature ordering. The results of classification are shown in Table 3. Two AD-based feature ordering approaches outperformed others with the lowest error rate of 1.21667%.

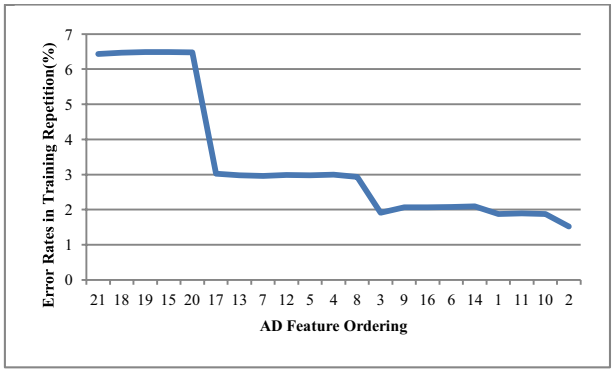

Fig.5. Error Rates in Training Repetition of Thyroid

4.4. *Ionosphere*

Figure 6 and Table 4 present the process and results of Ionosphere. In these experiments, compared with the conventional batch-training approach, the approach with feature partitions based on AD feature ordering got the greatest improvement, where the error rate is 4.88636%.

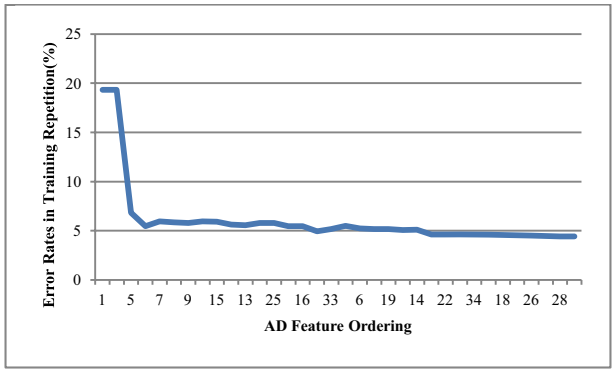

Fig.6. Error Rates in Training Repetition of Ionosphere

4.5. *Musk1*

Musk 1 is the first Musk dataset in UCI benchmarks. The error rates derived in training repetition are shown in Figure 7. Table 5 illustrates the final classification results derived by different approaches, where the proposed feature grouped AD feature ordering approach outperformed other approaches, and obtained the lowest classification error rate in 22.2689%.

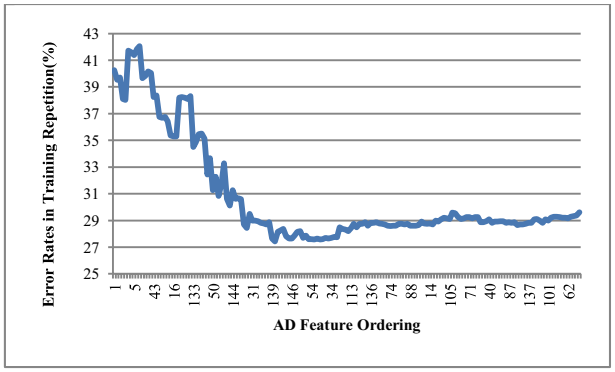

Fig.7. Error Rates in Training Repetition of Musk1

Table 1. Results of Diabetes

	Approach	Grouped Ordering	Classification Error (%)	Improvement (%)
1	Grouped Ordering based on AD	2-(6-7)-8-(5-4)-(1-3)	22.03125	7.94
2	AD	2-6-7-8-5-4-1-3	22.60416	5.55
3	Contribution-based	2-8-1-5-7-4-3-6	22.03125	7.94
4	Original Ordering	1-2-3-4-5-6-7-8	23.80209	0.54
5	Conventional Method	No Feature Ordering	23.93229	0.00

Table 2. Results of Glass

	Approach	Grouped Ordering	Classification Error (%)	Improvement (%)
1	Grouped Ordering based on AD	3-(8-2)-4-6-7-(1-5-9)	29.24530	29.06
2	AD	3-8-2-4-6-7-1-5-9	29.24530	29.06
3	Contribution-based	4-2-8-3-6-9-1-7-5	33.11322	19.68
4	Original Ordering	1-2-3-4-5-6-7-8-9	36.03775	12.59
5	Conventional Method	No Feature Ordering	41.22641	0.00

Table 3. Results of Thyroid

	Approach	Grouped Ordering	Classification Error (%)	Improvement (%)
1	Grouped Ordering based on AD	(21-18-19-15-20)-17-13-(7-12-5-4)-8-(3-9-16-6-14)-(1-11-10)-2	1.21667	34.72
2	AD	21-18-19-15-20-17-13-7-12-5-4-8-3-9-16-6-14-1-11-10-2	1.21667	34.72
3	Contribution-based	17-21-19-18-1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-20	1.72222	7.60
4	Original Ordering	1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21	1.59167	14.60
5	Conventional Method	No Feature Ordering	1.86389	0.00

Table 4. Results of Ionosphere

	Approach	Grouped Ordering	Classification Error (%)	Improvement (%)
1	Grouped Ordering based on AD	(1-2)-5-(3-7-23-9-29-15-31-13-21-25-11-16)-(8-33-4-6-17-19-27-14)-(10-22-12-34-24)-(18-32)-(26-20)-(28-30)	4.88636	46.25
2	AD	1-2-5-3-7-23-9-29-15-31-13-21-25-11-16-8-33-4-6-17-19-27-14-10-22-12-34-24-18-32-26-20-28-30	5.73864	36.88
3	Contribution-based	5-3-33-14-7-6-1-28-27-22-24-32-4-15-20-8-9-31-18-30-26-16-34-25-10-29-12-17-21-11-2-19-23-13	5.28409	41.87
4	Original Ordering	1-2-...-34	5.34091	41.25
5	Conventional Method	No Feature Ordering	9.09091	0.00

Table 5. Results of Musk1

	Approach	Grouped Ordering	Classification Error (%)	Improvement (%)
1	Grouped Ordering based on AD	1-(66-165)-116-(129-164-76-5-147-94-37-63-132-67-43-145)-140-(13-126)-97-141-16-(56-22-95-124-83-82)-(133-157-26-51-36)-(127-86)-(50-52)-(48-108-162)-10-(21-144-18-7-166)-53-(134-9-31-163-102-107-92-143-49)-139-(25-118-81-28-20-24-146-98-77-73-122-119-114-54-158-112-135-55-131-160-34-72-47-12-154-19-121-113-100-117-60-99-68-57-136-46-33-42-23-29-59-74-89-151-70-79-110-8-88-106-45-148-142-111-75-14-3-38-96-109-78-39-105-17-11-30-115-128-149-71-15-155-6-85-103-35-40-153-58-41-123-65-138-87-44-152-61-80-156-104-137-150-84-27-4-91-130-101-161-120-93-2-69-125-62-159-32-90-64)	22.2689	7.67
2	AD	1-66-165-116-129-164-76-5-147-94-37-63-132-67-43-145-140-13-126-97-141-16-56-22-95-124-83-82-133-157-26-51-36-127-86-50-52-48-108-162-10-21-144-18-7-166-53-134-9-31-163-102-107-92-143-49-139-25-118-81-28-20-24-146-98-77-73-122-119-114-54-158-112-135-55-131-160-34-72-47-12-154-19-121-113-100-117-60-99-68-57-136-46-33-42-23-29-59-74-89-151-70-79-110-8-88-106-45-148-142-111-75-14-3-38-96-109-78-39-105-17-11-30-115-128-149-71-15-155-6-85-103-35-40-153-58-41-123-65-138-87-44-152-61-80-156-104-137-150-84-27-4-91-130-101-161-120-93-2-69-125-62-159-32-90-64	25.88236	-7.32
3	Contribution-based	116-140-141-10-94-83-21-50-129-112-43-13-23-51-133-60-12-127-79-49-19-114-46-113-134-74-81-139-73-66-20-44-98-11-57-1-18-48-121-25-119-54-144-135-117-103-17-39-115-85-143-100-106-24-137-56-8-7-69-53-158-111-153-128-2-3-4-29-32-34-59-61-62-64-65-70-71-72-75-76-87-90-92-93-99-102-123-124-126-130-132-136-138-148-149-152-155-156-157-159-161-162-163-89-91-84-30-41-96-154-55-16-27-40-37-82-142-166-101-95-88-14-63-150-110-15-35-36-165-125-68-122-52-58-38-47-33-120-105-151-118-22-131-80-28-31-107-104-6-77-9-42-108-26-45-146-160-78-86-109-67-164-5-147-145-97	24.70588	-2.44
4	Original Ordering	1-2-...-166	22.68907	5.92
5	Conventional Method	No Feature Ordering	24.11764	0.00

4.6. *Semeion*

In the experiment of Semeion, Figure 8 shows the error rates of each training step based on AD feature ordering. Similar to previous experiments, features were partitioned into several groups. Classification results presented in Table 6 interpret that the approach with grouped AD feature ordering exhibited the best performance, where the classification error rate is 12.5%. In addition, classification error rate derived by conventional batch-training approach is 13.32915%, which is the worst one.

Table 6. Results of Semeion

	Approach	Grouped Ordering	Classification Error(%)	Improvement (%)
1	Grouped Ordering based on AD	112-96-162-178-146-128-111-95-79-161-(145-177)-(130-256-80)-127-194-63-1-(82-129)-(98-66-113-163)-(9-47-114-193)-(64-8-81-179)-(93-97-65)-(144-10)-231-230-229-2-11-(77-195-62)-143-(3-232)-(17-147)-78-(83-7-228-50)-99-233-(255-92-210-4)-(105-67)-191-(76-48)-(234-51-109)-152-175-84-103-(108-46)-240-(192-94-159-91-174-18)-107-254-167-151-(136-104)-16-(6-12)-75-(5-135)-188-(150-207-246-183-168-166)-(106-61)-121-(68-102)-149-(182-100-189-245)-227-(119-120-153-208-164-90-211)-49-(247-115-184)-(59-31-101-165-209-110-131-235-89)-60-33-180-(137-190)-35-36-(187-241-32-45)-69-(37-52-74-253-134)-(158-181-122)-(148-13)-(169-160-19)-(238-124)-58-(118-199-176-85-34-226)-185-123-173-(53-15-248-125-237-154-196-212-224-236)-(22-204-186-239)-(23-170)-(138-198-206)-(172-126)-88-155-(171-30-142-21-244-222-157)-(205-14-20-249-225)-(54-55-70-223-56-141-73)-(38-139-140-203-57-242)-197-(200-156-71-86-252-41-116-87-44-24-221-243-117-133-220-40-251-202-42-72-250-43-213-201-39-25-132-216-217-218-215-219-214-26-27-29-28)	12.5	6.22
2	AD	112-96-162-178-146-128-111-95-79-161-145-177-130-256-80-127-194-63-1-82-129-98-66-113-163-9-47-114-193-64-8-81-179-93-97-65-144-10-231-230-229-2-11-77-195-62-143-3-232-17-147-78-83-7-228-50-99-233-255-92-210-4-105-67-191-76-48-234-51-109-152-175-84-103-108-46-240-192-94-159-91-174-18-107-254-167-151-136-104-16-6-12-75-5-135-188-150-207-246-183-168-166-106-61-121-68-102-149-182-100-189-245-227-119-120-153-208-164-90-211-49-247-115-184-59-31-101-165-209-110-131-235-89-60-33-180-137-190-35-36-187-241-32-45-69-37-52-74-253-134-158-181-122-148-13-169-160-19-238-124-58-118-199-176-85-34-226-185-123-173-53-15-248-125-237-154-196-212-224-236-22-204-186-239-23-170-138-198-206-172-126-88-155-171-30-142-21-244-222-157-205-14-20-249-225-54-55-70-223-56-141-73-38-139-140-203-57-242-197-200-156-71-86-252-41-116-87-44-24-221-243-117-133-220-40-251-202-42-72-250-43-213-201-39-25-132-216-217-218-215-219-214-26-27-29-28	12.83922	3.68
3	Contribution-based	162-178-146-145-95-96-194-8-79-112-7-111-129-230-161-143-191-9-113-127-114-80-82-229-231-130-6-63-179-109-207-98-177-193-128-22-37-52-67-106-110-121-157-10-36-78-108-122-228-195-47-211-4-97-107-236-104-173-221-222-103-163-5-159-136-94-119-137-152-192-206-144-210-105-235-232-83-62-92-141-153-180-247-237-171-93-189-12-246-11-14-46-115-118-151-154-156-172-238-248-15-77-61-16-102-13-66-138-158-212-50-175-135-205-123-167-164-234-20-35-89-120-134-176-223-227-3-147-181-174-31-45-68-142-188-131-48-21-169-170-203-51-29-30-34-38-165-208-168-60-23-245-124-53-44-88-140-233-239-204-2-64-125-196-139-81-24-99-100-116-133-150-160-187-74-226-65-55-85-59-148-126-166-87-183-69-249-91-190-209-90-76-220-70-185-26-40-84-186-218-219-182-75-244-19-184-58-73-117-201-217-1-202-155-54-43-49-225-243-240-253-18-25-32-42-86-199-39-198-213-17-71-132-56-216-250-215-254-33-214-41-57-101-27-200-224-197-251-149-252-242-28-241-255-72-256	12.95226	2.83
4	Original Ordering	1-2-...-256	13.00251	2.45
5	Conventional Method	No Feature Ordering	13.32915	0.00

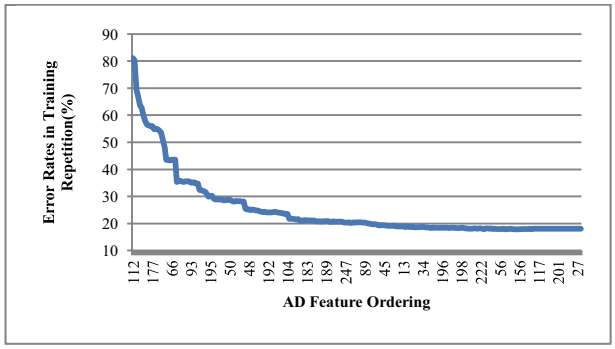

Fig.8. Error Rates in Training Repetition of Semeion

4.7. *Statistical Significance Testing*

In this part, the Wilcoxon signed-rank test is adopted to assess whether the method of grouped ordering based on AD is better than the other four methods over the tested six datasets. The Wilcoxon signed-rand test is a non-parametric pairwise comparison test with the assumption that the distribution of the difference scores is symmetric about the median.[28] The lower-sided Wilcoxon test is performed with the null hypothesis that the median of the difference scores of the compared two variables is equal to zero, while the alternative hypothesis that the median is less than zero, where the

difference score is between the method of grouped ordering based on AD and the other tested method. The Wilcoxon T statistic test is used to check whether the null hypothesis can be refuted for the desired significance level. The threshold significance level is 0.05. If the P-value of the Wilcoxon T statistic is smaller than 0.05, then the null hypothesis is rejected. The results of the Wilcoxon signed-rank test are given in Table 7, where the bold figures show the results which are not statistically significant at 5% confidence level. The statistically insignificant results are caused by two reasons. One reason is due to the same mean error rate result achieved by the compared two approaches. The other reason can be explained by the negative effect by small sample size problem, as the sample size in each approach is 20. Due to the small sample, the small difference in the different scores in mean classification error is difficult to detect. Besides, a little change in the sample may have a big influence on the symmetric of the median, which then leads to a statistically insignificant result. The limitation of small sample size problem could not be eliminated in this paper as it follows the sample size applied in the previous study of the compared approaches.

Hence, it can be summarized that, the Wilcoxon test results are consistent with the mean error rate results within 95% confidence interval, with all the statistically insignificant results having been considered. Since the statistical significance testing results are supported by the metrics of mean error rate, and from the mean error rate result, the method of grouped ordering based on AD has the lowest error rate among all the datasets, it shows that the method of grouped ordering based on AD outperformed the other methods with 95% level of confidence.

4.8. *Analysis*

According to Tables 1-7 and statistical testing, it is manifest that the proposed approach with grouped feature ordering based on AD obtained the lowest classification error rates and the most effective improvements comparing with those derived by some other approaches. Such a phenomenon indicates that grouping features can reduce the interference among input features, which is an important reason why grouped feature ordering can exhibit such a good performance. More specifically, Grouped Feature Ordering for IAL based on AD is applicable to reduce the classification error rates by 7.94%, 29.06%, 34.72%, 46.25%, 7.67%, and 6.22% in Diabetes, Glass, Thyroid, Ionosphere, Musk1, and Semeion, respectively. However, non-grouping approaches based on AD and Contribution-based Feature Ordering cannot always improve classification results. For example, results derived by these two methods in Musk1 performed worse than that obtained by conventional batch training approach. Nonetheless, the improvement brought by the approach with original feature ordering is minor. Take Diabetes as an example, it only improve 0.54% in final classification error totally. Therefore, the proposed Grouped Feature Ordering approach is not only feasible for pattern classification improvement, but also stable in the performance. Such an approach also indicates that feature grouping with ordering is an efficient way to reduce interference among features, which is beneficial to enhance the accuracy of pattern classification.

Table 7. Results of Wilcoxon Signed Ranks Test

	Compared pairwise Approaches	Diabetes	Glass	Thyroid	Ionosphere	Musk1	Semeion
1	Grouped Ordering based on AD - AD	0.018	**0.458**	**0.4715**	0.005	0.01	**0.099**
2	Grouped Ordering based on AD - Contribution-based Method	**0.421**	0.002	0	**0.102**	0.002	**0.071**
3	Grouped Ordering based on AD - Original Ordering -	0	0	0	**0.116**	**0.452**	**0.057**
4	Grouped Ordering based on AD - Conventional Method	0	0	0	0	**0.092**	0.029

5. Conclusions

IAL is a novel machine learning approach which gradually trains input attributes one by one. Previous research showed a number of unique preprocessing approaches such as feature ordering and feature partition. These approaches have been employed in IAL to enhance the accuracy of final classification results. However, current approaches are too complex to be used in large-scale classification problems. In this study, Grouped Feature Ordering for IAL, which fused feature partition with feature ordering, is presented. With

Grouped Feature Ordering, it is unnecessary to detect whether a feature is redundant. Unlike previous work, features are not arranged in different groups by comparing features' contribution in pairs. Grouped Feature Ordering provides a simple and effective way to obtain acceptable classification results. Only neighboring features in descending order can be arranged in one group after the Feature Ordering is calculated. Experimental results showed that Grouped Feature Ordering approach can not only improve the classification performance, but also reduce interference among input features, which is beneficial for accuracy enhancement in pattern classification.

Acknowledgements

This research is supported by National Natural Science Foundation of China under Grant 61070085 and Jiangsu Provincial Science and Technology under Grant No.BK20131182 .

Appendix A. Definitions and Stopping Criteria [5]

The error measure *E* used in the ILIA algorithms is *the squared error percentage*[29], derived from the normalization of the mean squared error to reduce the dependency on the number of coefficients in the problem representation and on the range of output values used:

$$E = 100 \cdot \frac{o_{\max} - o_{\min}}{K \cdot P} \sum_{p=1}^{P} \sum_{k=1}^{K} (o_{pk} - t_{pk})^2$$

where $o_{\max}$ and $o_{\min}$ are the maximum and minimum values of output coefficients in the problem representation.

$E_{tr}(t)$ is the average error per pattern of the network over the training set, measured after epoch *t*. The value $E_{va}(t)$ is the corresponding error on the validation set after epoch *t* and is used by the stopping criterion. $E_{te}(t)$ is the corresponding error on the test set; it is not known to the training algorithm but characterizes the quality of the network resulting from training.

The value $E_{opt}(t)$ is defined to be the lowest validation set error obtained in epochs up to epoch *t*:

$$E_{opt}(t) = \min_{t' \le t} E_{va}(t')$$

The *generalization loss* [28] at epoch *t* is defined as the relative increase of the validation error over the minimum so far (in percentage):

$$GL(t) = 100 \cdot (\frac{E_{va}(t)}{E_{opt}(t)} - 1)$$

A high generalization loss is one candidate reason to stop training because it directly indicates overfitting.

To formalize the notion of training progress, a *training strip of length k* [29] is defined to be a sequence of k epochs numbered $n+1$ ... $n+k$ where n is divisible by k. The training *progress* measured after a training strip is:

$$P_k(t) = 1000 \cdot (\frac{\sum_{t' \in t-k+1 \ldots t} E_{tr}(t')}{k \cdot \min_{t' \in t-k+1 \ldots t} E_{tr}(t')} - 1)$$

It is used to measure how much larger the average training error is than the minimum training error during the training strip.

During the process of growing and training sub-networks, heuristic overall stopping criteria are adopted as the following: $E_{opt} < E_{th}$ **OR** *(Reduction of training set error due to the last new hidden unit is less than 0.01%* **AND** *Validation set error increased due to the last new hidden unit)*. The first part ($E_{opt} < E_{th}$) means that the optimal validation set error is below the threshold and the result has been acceptable. The other part means the last insertion of a hidden unit resulted in hardly any progress. The criteria for adding a new hidden unit are as follows: *At least 25 epochs reached for the current network* **AND** *(Generalization loss* $GL(t) > 5$ ***OR*** *Training progress* $P_k(t) < 0.1$*)*. The first part means that the current network should be trained for at least a certain number of epochs before a new hidden unit is installed because the error curves will be turbulent in the beginning. The second part means that the current network has been overfit or training has little progress.

In addition, in order to minimize the cost function, the RPROP algorithm [30] is adopted. The parameters are set as: $\eta^+ = 1.2$, $\eta^- = 0.5$, $\Delta_0 = 0.1$, $\Delta_{\max} = 50$, $\Delta_{\min} = 1.0e-6$, with in ITID weights from –0.25 ... 0.25 randomly.

References

1. H. Liu, E. R. Dougherty, J. G. Dy, K. Torkkola, E. Tuv, H. Peng, C. Ding, F. Long, M. Berens, L. Parsons, Z. Zhao, L. Yu, and G. Forman, 'Evolving Feature Selection', *Intelligent Systems, IEEE,* 20 (2005), 64-76.
2. S.M. Weiss, and N. Indurkhya, *Predictive Data Mining: A Practical Guide*Morgan Kaufmann, 1998).
3. S. U. Guan, and F. M. Zhu, 'An Incremental Approach to Genetic-Algorithms-Based Classification', *IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics,* 35 (2005), 227-39.

4. F. M. Zhu, and S. U. Guan, 'Ordered Incremental Training with Genetic Algorithms', *International Journal of Intelligent Systems,* 19 (2004), 1239-56.
5. S. U. Guan, and S.C. Li, 'Incremental Learning with Respect to New Incoming Input Attributes', *Neural Processing Letters,* 14 (2001), 241-60.
6. S. U. Guan and J. Liu, 'Incremental Neural Network Training with an Increasing Input Dimension', *Journal of Intelligent Systems,* 13 (2004), 45-69.
7. X. Liu, G. Zhang, Y. Zhan, and E. Zhu, 'An Incremental Feature Learning Algorithm Based on Least Square Support Vector Machine', in *2nd International Frontiers in Algorithmics Workshop, FAW 2008, June 19, 2008 - June 21, 2008* (Changsha, China: Springer Verlag, 2008), pp. 330-38.
8. W. Bai, S. Cheng, E. M. Tadjouddine, and S. U. Guan, 'Incremental Attribute Based Particle Swarm Optimization', in *2012 8th International Conference on Natural Computation, ICNC 2012, May 29, 2012 - May 31, 2012* (Chongqing, China: IEEE Computer Society, 2012), pp. 669-74.
9. S. Chao, and F. Wong, 'An Incremental Decision Tree Learning Methodology Regarding Attributes in Medical Data Mining', in *2009 International Conference on Machine Learning and Cybernetics, July 12, 2009 - July 15, 2009* (Baoding, China: IEEE Computer Society, 2009), pp. 1694-99.
10. J. H. Ang, S. U. Guan, K. C. Tan, and A. Al Mamun, 'Interference-Less Neural Network Training', *Neurocomputing,* 71 (2008), 3509-24.
11. S. U. Guan, and J. Liu, 'Incremental Ordered Neural Network Training', *Journal of Intelligent Systems,* 12 (2002), 137-72.
12. S. U. Guan, and J. Liu, 'Feature Selection for Modular Networks Based on Incremental Training', *Journal of Intelligent Systems,* 14 (2005), 353-83.
13. S. U. Guan, J. Liu, and Y. Qi, 'An incremental approach to contribution-based feature selection', *Journal of Intelligent Systems*, 13(2004), 15-44.
14. S. U. Guan, and J. H. Ang, 'Incremental Training Based on Input Space Partitioning and Ordered Attribute Presentation with Backward Elimination', *Journal of Intelligent Systems,* 14 (2005), 321-51.
15. S. Guo, S. U. Guan, W. Li, et al, 'Input Space Partitioning for Neural Network Learning', *International Journal of Applied Evolutionary Computation (IJAEC)*, 4(2013), 56-66.
16. S. U. Guan, S.C. Li, 'Parallel growing and training of neural networks using output parallelism', *IEEE Trans Neural Netw*, 13(2002), 542-550.
17. S. U. Guan, P. Li, 'Incremental learning in terms of output attributes', *Journal of Intelligent Systems*, 13(2002), 95-122.
18. S. U. Guan, C. Bao, T. Neo, 'Reduced pattern training based on task decomposition using pattern distributor', *IEEE Trans Neural Netw*, 18(2007), 1738-1749.
19. C. Bao, T. Neo, S. U. Guan, 'Reduced pattern training in pattern distributor networks', J *Res Pract Inf Technol*, 39(2007), 273-286.
20. R.K. Agrawal, R. Bala ,'Incremental Bayesian classification for multivariate normal distribution data', *Pattern Recognition Letters*, 29(2008):1873-1876.
21. T. Wang, S. U. Guan, and F. Liu, 'Feature Discriminability for Pattern Classification Based on Neural Incremental Attribute Learning', in *Foundations of Intelligent Systems: Proceedings of the Sixth International Conference on Intelligent Systems and Knowledge Engineering, Shanghai, China, Dec 2011 (ISKE2011)* (Tiergartenstrasse 17, Heidelberg, D-69121, Germany: Springer Verlag, 2011), pp. 275-80.
22. T. Wang, S. U. Guan, and F. Liu, 'Correlation-Based Feature Ordering for Classification Based on Neural Incremental Attribute Learning', *International Journal of Machine Learning and Computing,* 2 (2012), 807-11.
23. T. Wang, S. U. Guan, and F. Liu, 'Entropic Feature Discrimination Ability for Pattern Classification Based on Neural Ial', in *9th International Symposium on Neural Networks, ISNN 2012, July 11, 2012 - July 14, 2012* (Shenyang, China: Springer Verlag, 2012), pp. 30-37.
24. T. Wang, S. U. Guan, T. O. Ting, K. L. Man, and F. Liu, 'Evolving Linear Discriminant in a Continuously Growing Dimensional Space for Incremental Attribute Learning', in *9th IFIP International Conference on Network and Parallel Computing, NPC 2012, September 6, 2012 - September 8, 2012* (Gwangju, Korea, Republic of: Springer Verlag, 2012), pp. 482-91.
25. T. Wang, and Y. Q. Wang, 'Pattern Classification with Ordered Features Using Mrmr and Neural Networks', in *2010 International Conference on Information, Networking and Automation, ICINA 2010, October 17, 2010 - October 19, 2010* (Kunming, China: IEEE Computer Society, 2010), pp. V2128-V31.
26. T. Wang, and S. U. Guan, 'Feature Ordering for Neural Incremental Attribute Learning Based on Fisher's Linear Discriminant', in *Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2013 5th International Conference on*, 2013), pp. 507-10.
27. H. Peng, F. Long, and 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), 1226-38.
28. N. Japkowicz,and M. Shah, Evaluating Learning Algorithms: A Classification Perspective, *Cambridge University Press,* (2011).
29. L. Prechelt, 'PROBEN1: A set of neural network benchmark problems and benchmarking rules', *Technical Report 21/94*, Department of Informatics,University of Karlsruhe, Germany,(1994).
30. M. Riedmiller, and H. Braun, 'A direct adaptive method for faster backpropagation learning: the RPROP algorithm', in *Proceedings of the IEEE International Conference on Neural Networks*, (1993), 586–591.