International Journal of Neural Systems, Vol. 24, No. 6 (2014) 1450021 (21 pages)

DOI: 10.1142/S012906571450021X

World Scientific
www.worldscientific.com

SENSOR (GROUP FEATURE) SELECTION WITH CONTROLLED REDUNDANCY IN A CONNECTIONIST FRAMEWORK

RUDRASIS CHAKRABORTY
Electronics and Communication Sciences Unit
Indian Statistical Institute, Calcutta, India
rudrasischa@gmail.com

CHIN-TENG LIN
Electrical and Control Engineering and Computer Science
National Chiao-Tung University, Taiwan
ctlin@mail.nctu.edu.tw

NIKHIL R. PAL*
Electronics and Communication Sciences Unit
Indian Statistical Institute, Calcutta, India
nikhil@isical.ac.in

Accepted 16 May 2014
Published Online 20 June 2014

For many applications, to reduce the processing time and the cost of decision making, we need to reduce the number of sensors, where each sensor produces a set of features. This sensor selection problem is a generalized feature selection problem. Here, we first present a sensor (group-feature) selection scheme based on Multi-Layered Perceptron Networks. This scheme sometimes selects redundant groups of features. So, we propose a selection scheme which can control the level of redundancy between the selected groups. The idea is general and can be used with any learning scheme. We have demonstrated the effectiveness of our scheme on several data sets. In this context, we define different measures of sensor dependency (dependency between groups of features). We have also presented an alternative learning scheme which is more effective than our old scheme. The proposed scheme is also adapted to radial basis function (RBS) network. The advantages of our scheme are threefold. It looks at all the groups together and hence can exploit nonlinear interaction between groups, if any. Our scheme can simultaneously select useful groups as well as learn the underlying system. The level of redundancy among groups can also be controlled.

Keywords: Sensor selection; feature selection; neural networks; redundancy control.

1. Introduction

Feature selection is a key step in designing pattern recognition and function approximation type systems.[1–16] The necessity of selecting a small number of "useful" features is fourfold. It reduces the cost of design and decision making, makes the learning task simpler and often improves the classification performance. Dimensionality reduction also enhances interpretability of the system. In particular, interpretability of decision trees and rule-based systems including fuzzy systems is enhanced significantly with dimensionality reduction. Not only selection of a small feature set is desired but also their relevancy/usefulness plays a vital role. The relevancy might be in terms of improving classification/prediction performance, if the target is to design a classifier/predictor or might be optimizing some criteria like extent of preservation of cluster structure. More features usually increase the degree of freedom of the system and hence the

*Corresponding author.

system gets better freedom to memorize the data. In many bioinformatics applications,[17–21] where the number of samples is very less compared to the number of genes, feature selection is important to overcome the "curse of the dimensionality" problem. Several researchers have explored in different directions for the feature selection problem[3–43] and not to mention that this search is still on.

Feature selection methods can be classified into Wrapper and Filter methods. While wrapper methods need a feedback from the target predictor/classifier, filter methods do not need a target to assess the utility of the features. Generally, wrapper methods perform better as the importance of a feature lies on the problem and also the tool that is used to solve the problem.[12] To select the optimal feature subset, one needs to go through all possible subsets of the original feature set which is computationally inefficient. There are broadly two possible directions to overcome this problem. One is "Forward Selection" method and the other is "Backward Elimination" method[2] but the interaction between features are not accounted for in these methods. There are only a handful of embedded methods, which simultaneously select useful features as well as learn the underlying systems.[4,12,24,48,50,51] Some researchers[47,49] also proposed rough set-based feature selection techniques.

Group sparsity-based feature selection has been addressed by few researchers.[44–46] Though these methods appear to be very close to our formulation, a careful analysis reveals that this framework fails to prioritize between nearly equal important groups, as they look at the redundancy within each group. These methods do not explicitly try to control redundancy using measures of dependency between groups.

Among the features present in the data set, besides useful features, there might be some bad/derogatory, indifferent and redundant/correlated features. Bad features are those whose removal from the original set might enhance the system performance. Indifferent features do not cause any harm other than increasing the cardinality of the feature set. A feature having the same value for all data points is an indifferent feature. There might be some features, which are useful but strongly dependent on each other (for example, linearly correlated features). Such features are redundant in the sense that all of them are not needed and selection of only a few of them is sufficient for the target application. A feature selection method should pick the useful features and discard all kind of "not-useful features" that we have mentioned. While most of the feature selection schemes do not focus on the redundant features, there are few methods[6,27,7,17,52,42,24] which remove the correlated/redundant features. We would like to mention that feature selection with controlled redundancy[7] is desirable as complete removal of redundancy would make the system vulnerable to measurement errors.

In this work, we focus on a different kind of feature selection. Here, we assume that features are partitioned into several groups. There are some real-life problems where data are collected from several sensors; for example, in case of an intelligent welding inspection system, the inputs come from different sensors such as radiograph, thermograph and eddy-current. And from each sensory input signal, some features are extracted. So, use of all these sensors would increase the cost of system design as well as the decision making time. In this kind of application, the designer always tries to reduce the required number of sensors. In a more general setting, we can view this problem as group feature selection problem where each group of features may correspond to a sensor or the group can be decided by the designer. We can think of the conventional feature selection problem as the group feature selection, with one feature in each group. In other words, group feature/sensor selection is a generalized feature selection problem. The sensor selection problem is introduced and solved by Chakraborty and Pal.[12] In this work, we extend this work in terms of controlling redundancy between selected groups.

2. MLP Network for Group-feature Selection with Controlled Redundancy (GFSMLP-CoR)

Chakraborty and Pal[12] developed a generalized network for group-feature selection. In this work, we move it one step further for controlling the redundancy among the groups of selected features. A sketchy description about the group feature selection network by Chakraborty and Pal[12] is given at first. Then we discuss our redundancy control scheme in detail. An MLP network for group feature selection is given in Fig. 1. Note that, the architecture in Fig. 1

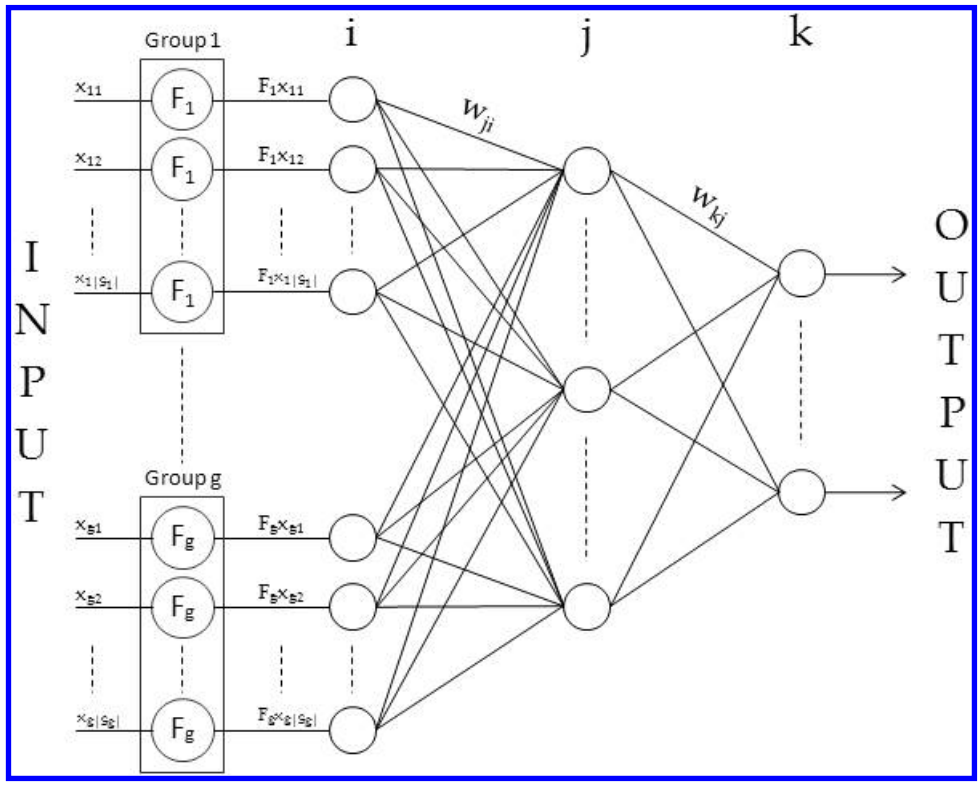

Fig. 1. GFS-MLP: A group-feature (sensor) selection MLP network.

is similar to that of an MLP except that each input is modified by the gate value of its associated group.

Let (X, Y) be the training data set with $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}^T$; $\mathbf{x}_j \in \mathbb{R}^p$ and $Y = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\}^T$; $\mathbf{y}_j \in \mathbb{R}^c$. N, p and c are the number of data points, number of features and number of classes, respectively. We denote a data point by $\mathbf{x}_i$ or $\mathbf{x}$ in the rest of paper, unless otherwise stated, the kth component of $\mathbf{x}$ is denoted by x_k. The network consists of p input nodes and c output nodes. Assume that this set of p features is partitioned into g groups where each group is denoted by S_i, $i \in \{1, 2, \ldots, g\}$ such that $\sum_{i=1}^{g} |S_i| = p$, $|S_i|$ is the cardinality of the set S_i (Note that $S_i \neq \Phi$ and $S_i \cap S_j = \Phi, i \neq j$). Each group S_i is associated with an attenuator function $F_i \in [0, 1]$. So any feature $l \in S_i$ is multiplied by the attenuator function F_i. Thus, for a data point $\mathbf{x}$ and a feature $l \in S_i$, the attenuated output would be

$$x'_l = x_l F_i. \tag{1}$$

When the attenuator function F_i attains the 0 value then $x'_l = 0, \forall l \in S_i$, on the other hand, with $F_i = 1$, all features belonging to the ith group enter the network unattenuated. Each F_i is a monotonic differentiable function of β_i (a tunable parameter), which is adjusted during the training. The parameter β is unrestricted and the range of F_i is $[0, 1]$. Two such attenuator functions are:

$$F_i = \frac{1}{1 + \exp^{-\beta_i}}, \tag{2}$$

$$F_i = \exp^{-\beta_i^2}. \tag{3}$$

We use the function F_i in (2), i.e. the sigmoidal function.

The symbols used in the following discussion are given below:

- x'_i: The attenuated value of the ith component, x_i, of an input vector $\boldsymbol{x} \in X$.
- o_i^k: The output of the ith node of the kth hidden layer.
- w_{ij}^k: The weight connecting the jth node of the kth layer to the ith node of the $k+1$th layer. ($k = 0$ refers to the input layer, n is the total number of hidden layers and $n + 1$th layer refers to the output layer).
- ϕ: The activation function.

ϕ'^k_i: The derivative of ϕ^k_i at the ith node of the kth hidden layer.

ϵ: The instantaneous error for the input vector $\mathbf{x}$.

Note that, in Fig. 1, the input x_{ij} denotes jth feature in the ith group, $i = 1, 2, \ldots, g$ and $j = 1, 2, \ldots, |S_i|$, where $|S_i|$ is the size of the ith group. Since each feature is generated by a particular sensor, each feature belongs to exactly one group. With these notations, the output from the ith node of the output layer is o^{n+1}_i and the desired/target output for the ith output node is y_i, when the input is $\mathbf{x}$.

Given an input vector $\mathbf{x}$, the output from the ith node of the first hidden layer is:

$$o^1_i = \phi\left(\sum_{j=1}^{p} x'_j w^0_{ij}\right). \tag{4}$$

This signal is then propagated through the network.

2.1. *Derivation of the learning rules*

Let μ be the learning rate for the attenuator parameter and η be the same for the weights.

The attenuated inputs computed in layer one are:

$$x'_i = F_l x_i; \quad i = 1, 2, \ldots, p. \tag{5}$$

Let there be n_1 nodes in the first hidden layer. The output of the ith node of the first hidden layer is

$$o^1_i = \phi\left(\sum_{j=1}^{p} x'_j w^0_{ij}\right); \quad i = 1, 2, \ldots, n_1. \tag{6}$$

The output of the ith node of the kth layer can be written as:

$$o^k_i = \phi\left(\sum_j o^{k-1}_j w^{k-1}_{ij}\right); \quad i = 1, 2, \ldots, n_k; \quad k = 2, 3, \ldots, n+1. \tag{7}$$

The final output from the ith output node of the network is o^{n+1}_i.

For an input vector $\mathbf{x}$, let the instantaneous error be E_x. The learning algorithm updates the weights and βs to minimize the system error E:

$$E = \sum_{\mathbf{x}\in X} E_x = \frac{1}{2}\sum_{\mathbf{x}\in X}\sum_{i=1}^{c} (o^{n+1}_i - y_i)^2. \tag{8}$$

Differentiating E w.r.t. w^k_{ij} we get the update rule for weights:

$$\Delta w^k_{ij} = -\eta \sum_{\mathbf{x}\in X} \frac{\partial E_x}{\partial w^k_{ij}}. \tag{9}$$

Using the chain rules, it is easy to derive the detailed update rules.

Similarly, the update rule for β's can be written as

$$\Delta \beta_l = -\mu \sum_{\mathbf{x}\in X} \frac{\partial E_x}{\partial \beta_l}. \tag{10}$$

Detailed update rules can be found in Ref. 12.

In order to learn the weights and β values, all the weight values are initialized with random values from $[-0.5, 0.5]$. Whereas, for the β values, we initialize them such that attenuator values are close to zero.[12] So at the beginning of the training, all groups are unimportant. As we choose the attenuator function to be sigmoidal function, the β values are randomly selected from $[-5.05, -4.95]$.

Here we do not penalize selection of redundant groups. So in a particular run two groups may be selected which are highly correlated (linear or nonlinear). In Sec. 2.2, we address this problem of redundancy control between feature-groups.

2.2. *Controlling redundancy*

Let us now clarify the notion of redundancy between two groups of features or between two sensors. Suppose feature F is a useful feature and feature F' is strongly related (dependent) to feature F. By "related/dependent" we mean that any one of two features would suffice. So clearly, the dependency between two features is symmetric. On the other hand, suppose there are two groups of features G and G', and for every feature in G, there is a strong dependent feature in G'. In addition, G' also has some extra features. In this case, G is highly correlated to G' whereas G' is less correlated to G. Then with respect to group G', G is redundant but the converse is not true. An extreme example could be when $G \subset G'$. In such a case when G' is selected, we do not need G, but when G is selected G' may also be needed. Thus, group dependency or sensor dependency is asymmetric. The dependency could be linear or nonlinear. Note that, redundancy between groups is relevant for our purpose only if they are

useful groups of features. A group is called useful if there exists at least one useful feature in the group. Let G and G' be two groups, G consists of only one feature which is a useful feature say F and G' consists of two derogatory features and F. Then group G is more useful than G' as the number of features is less in G than G' with the same utility.

We want to modify our group-feature selection scheme so that the redundancy can be controlled while selecting features. So, the learning must penalize selection of redundant groups. A natural way is to augment the system error E in (8) by a penalty term so that use of many redundant groups of features increases the system error as in (11):

$$\mathrm{TE} = E + \lambda P(X). \tag{11}$$

In (11), P defines the penalty for using redundant groups and λ is a regularizing constant that determines the severity of the penalty term. The term $P(X)$ should be defined in such a way that the redundancy between groups is captured.

Let there be g groups of features. One possible choice of $P(X)$ is

$$P(X) = \frac{1}{g(g-1)} \sum_{l=1}^{g} F(\beta_l) \times \sum_{m \neq l} F(\beta_m) \mathrm{dep}(S_l, S_m). \tag{12}$$

In (12), $\mathrm{dep}(S_l, S_m) \geq 0$ is a measure of dependency between the set S_l and S_m. As a simple measure of dependency, we have used $\mathrm{dep}(S_l, S_m) = \mathcal{G}_{1 i \in S_l}\{\mathcal{G}_{2 j \in S_m}\{\rho^2(x_i, x_j) \forall\, j \in S_m\} \forall\, i \in S_l\}$, where $\rho(x_i, x_j)$ is the Pearson's correlation coefficient between x_i and x_j. $\mathcal{G}_1$ and $\mathcal{G}_2$ are aggregation functions. Note that, in this work we have used $\mathcal{G}_1$ as the min and $\mathcal{G}_2$ to be the max functions. This dependency measure is asymmetric, i.e. $\mathrm{dep}(S_l, S_m) \neq \mathrm{dep}(S_m, S_l)$. Also, note that $\mathrm{dep}(S_l, S_l), \forall\, l$ is 1 which is the highest possible level of dependency.

As mentioned before, a gate is associated to every feature-group (sensor). So, if a group S_l is important, the associated gate would be open, i.e. F_l would be close to 1. Now if another group S_m is redundant with respect to group S_l, then we do not want the gate for S_m to be open. Thus, when F_l is close to 1, and $\mathrm{dep}(S_l, S_m)$ is high, then if F_m is also open, it will add a high penalty to the system error. Thus the training based on gradient descent technique, prevents opening of both gates F_l and F_m. Whereas, if $\mathrm{dep}(S_l, S_m)$ is high but the two groups are not useful, then the selection of them will not reduce the system error which in turn assures that these groups are not selected. *We start our training with all gates almost closed*. The factor $g(g-1)$ is used just to make the penalty term independent of the number of groups. This will make the choice of λ a bit easier. Here, if we set $\lambda = 0$, then (11) reduces to our original system and higher the value of λ higher is the effect of redundancy on the system error.

Consequently, this will change the learning rules. Since the penalty function does not involve any term containing weights, the learning rule for weights remains unchanged. While the learning rule for attenuator parameters changes slightly. It is given by,

$$\Delta\beta_l = -\mu \frac{\partial \mathrm{TE}}{\partial \beta_l} = -\mu \left(\frac{\partial E}{\partial \beta_l} + \lambda \frac{\partial P}{\partial \beta_l} \right),$$

where

$$\frac{\partial P}{\partial \beta_l} = \frac{1}{g(g-1)} F_l' \times \sum_{m \neq l} F_m [\mathrm{dep}(S_l, S_m) + \mathrm{dep}(S_m, S_l)].$$

We note here that our scheme is a generalized feature selection scheme and hence can be applied as a feature selection scheme treating every feature as a sensor or a group. The dependency measure, $\mathrm{dep}(S_l S_m)$ is a very general one and it does not have to be based on correlation. For example, it could be defined using a measure of mutual information. During learning, β's will change in such a manner that they will facilitate selection of those groups that help to solve the learning problem and at the same time control the use of redundant groups.

2.3. *Experimental results*

In this investigation, sigmoidal function is used as both attenuator and activation functions. The batch mode learning is used. We consider a group to be useful if its attenuation is below 90%; in other words, the gate is opened more than 10%. In this work, as the main concern is to control redundancy among groups, the threshold for the attenuation value is decided in an *ad hoc* manner. One can use the cross-validation mechanism for choosing an appropriate threshold value. But in practice, for a real

application, it will depend on the user and the problem. For example, if we want to design an intelligent weld inspection system, we can use sensors like X-ray images, thermal images, visual images, eddy current and acoustic emission. In this particular case, the cost of design, the decision making time as well as physical constraints (size of the equipment) will determine the desired set of sensors that the user would use.

We have used altogether 10 data sets as summarized in Table 1. This includes two variants of Iris data set denoted by Iris_1, Iris_2. In the Iris data set[53] (this data set is also used by Chakraborty and Pal[12]), the four features are sepal length (f_1), sepal width (f_2), petal length (f_3) and petal width (f_4). We group sepal length and width as the first group and the remaining two features as the second group. This is a natural grouping. This data set is denoted by Iris_1. The second variant of the Iris dataset, i.e. Iris_2 contains three groups. The first group contains two features f_1 and f_2, the second group consists of three features $f_1 \pm N(0, 0.05)$, f_3 and f_4. The last group has two features $f_3 \pm N(0, 0.05)$ and $f_4 \pm N(0, 0.05)$.

The electroencephalography (EEG) data are used in many applications.[55–63] The problem of selection of useful channels/independent components arises in most of the applications of EEG. To investigate the effect of dual task on EEG while driving, we collected EEG data by a virtual-reality (VR) based on highway driving environment. The VR driving environment has 3D surrounded scenes projected by seven projectors and a real car mounted on a platform with 6-degree-of-freedom to provide the kinesthetic stimuli to make the subject feel realistic driving conditions. The subjects are given stimulus which are similar to what they would face in a real driving scenario. During driving, all scenes move according to the displacement of the car and the subject's maneuvering of the wheels, which make the subject interact directly with the virtual environment. For this dual-task study, the drivers are asked to respond to two different kinds of events: unexpected car deviation and simple mathematical questions. Five cases (conditions) are used to investigate the interaction of these two tasks and their effect on the brain waves. The five cases are as follows.

- **Case-1** – A math question is asked at 400 ms before the occurrence of the car deviation (math-400 ms-deviation)
- **Case-2** – Two tasks (deviation and math question) appear simultaneously (deviation and math)
- **Case-3** – A math question is asked at 400 ms after the occurrence of the deviation (deviation-400 ms-math)
- **Case-4** – Only the math question is asked (single math)
- **Case-5** – The car is subjected to only sudden deviance (single deviation)

In the actual experiment, each subject took part in four sessions, each of 15 min duration. Here we can accommodate a total of about 100 trials in each condition. Subjects are given a break between every two sessions to avoid fatigue. The EEG data used in this paper are collected from 11 healthy subjects who are students of the National Chiao Tung University, Taiwan. The standard ethics protocol was followed and consent of each student was taken. The physiological data are collected by a 32-channel EEG module (Neuronscan, Inc.) arranged according to international 10-20 system. The reference channels used are A1 and A2. The EEG data are recorded with 16-bit quantization levels at 500 Hz sampling rate. The collected EEG data are first pre-processed to remove noise. To reduce the computational overhead, we lower the sampling rate of EEG data from 500 to 250 Hz. Since the five different cases appear in a randomly mixed order, we first separate the EEG signals related to different cases from the raw EEG

Table 1. Summary of all data sets used.

Data set	#Class	#Features	Data set size
Iris_1	3	4	150
Iris_2	3	7	150
Distraction	5	70	393
RS-Data_1	8	7	262,144
RS-Data_2	8	9	262,144
RS-Data_3	8	14	262,144
RS-Data_4	8	18	262,144
LandSat	6	44	6435
LRS	10	93	531
Gas sensor	6	128	13,790

data. An epoch is defined as the length of EEG signals corresponding to a particular case (trial), and it includes data from a baseline to the end of response by the subject. Here, each epoch is of 6s and thus consists of 1500 sample points. For this work, we took only 1000 samples by removing the first and last 250 sample points. After extracting all epochs, ICA is applied to separate independent brain sources. To reduce the effect of blinking of eyes, we have ignored the 2 channels which are near the eye position. The remaining 30 of the 32 channels are used for the IC analysis. From this 30 independent components, we have chosen seven components, namely, Frontal, Central, Parietal, Right Occipital, Left Occipital, Left Motor and Right Motor. We treat these seven components as seven sensors. Sensors are numbered 1 through 7 in the order in which they are listed. Features computed from the ith senor will be referred to as either ith senor data or ith group features. On each of the seven components, we have applied multiclass common spatial patterns (CSP) to extract 10 features. The two-class CSP technique is extended to multiclass paradigm by using joint approximate diagonalization (JAD) (see Ref. 54 and the references therein). Thus, from each of the seven components, we have extracted 10 features resulting a 7-sensor (groups of features) problem with each sensor (group) consisting 10 features. We denote this data set by *Distraction* having 393 points distributed in 5 classes.

The next four data sets are different variations of RS-Data.[65] This is also a true sensor selection problem. RS-Data set is generated from a 256 level satellite image of size 512×512 taken by seven sensors operating in different spectral bands. Thus, the data set consists of 262,144 data points. The gray values of a pixel from the seven images (channels) correspond to a 7–D feature vector. There are eight classes in this data set. Our first variant of the RS-Data, denoted by RS-Data_1, consists of seven groups (each group corresponds to one sensor) with one feature in each group. The second variant used is RS-Data_2, which has two more sensors over the first variant. These two sensors, namely, groups 8 and 9, consist of one feature each which are the noise added values of features 4 and 6, respectively. So, RS-Data_2 consists of nine groups with one feature per group.

The next variant is denoted by RS-Data_3. Here the number of groups is the same as that of RS-Data_1. In addition to the pixel(p) values (the first feature), another feature is introduced per group, which is the standard deviation of the pixel values over the 3×3 neighborhood centered at the pixel p under consideration. So RS-Data_3 consists of seven groups, each with two features per group. Note that the same RS-Data_1 and RS-Data_3 were used in Chakraborty and Pal,[12] there these data sets were named as RS-Data and RS14, respectively. The fourth variant, RS-Data_4 is an augmented version of RS-Data_3. As in case of RS-Data_2, here groups 8 and 9 are added, which are the noise added versions of group 4 and 6, respectively. Thus, there are two redundant sensors. For each of these variants we have used 1600 points for training taking 200 randomly selected points from each class (as several earlier studies used this protocol). The remaining points are used for testing classification accuracy.

The next data set is a variant of Statlog (Landsat Satellite) data set.[53] This data set consists of multi-spectral values of pixels in 3×3 neighborhoods in a satellite image. There are six classes and the class label corresponds to the center pixel. The data set contains images in four spectral bands, each band has nine features correspond to nine pixel values. We modified this data set by augmenting with two additional features, mean and standard deviation of the pixel values of the 3×3 neighborhood. Thus, the data set used here, named as *LandSat*, contains 4 groups each with 11 features. There are 6435 sample points distributed in 6 classes. We have used 4435 samples in the training set and 2000 samples to test the classification accuracy as suggested in Ref. 53.

The data set,[53] low resolution spectrometer (LRS) contains 531 high quality spectra derived from IRAS-LRS database. This data set contains features from two bands namely blue and red bands. These two bands consist of 44 and 49 flux measurements, respectively. Thus, LRS is a 93-dimensional data set having two groups/sensors.

The last data set is the Gas Sensor data, which consists of 13,910 measurements from 16 chemical sensors. From each gas sensor 8 features have been extracted forming a 128-dimensional data set generated by 16 sensors and divided into 6 classes. We

removed the missing values from this original data set which results in a data set consisting of 13,790 points.

For all our experiments with GFSMLP-RoC, we use just one hidden layer. To make our results more reliable, we use two-level cross-validation mechanism as explained next. In the outer level, first we randomly partition the data into 10 folds, each of equal size (to the extent possible), $X = X_1 \cup X_2 \cup \cdots \cup X_{10}, X_i \cap X_j = \Phi \forall\, i \neq j$. One of the folds, say X_j, is kept out for testing. While the remaining 9 folds data, $Y = \bigcup_{i,i\neq j} X_i$, are now used for selection of features as well as for designing a network to test the effectiveness of the selected features on the data left out in the outer loop, i.e. on X_j. This is repeated for all $j = 1, \ldots, 10$.

In the inner loop, we use only Y and perform two tasks. First, we again use the cross-validation mechanism on Y to find the desirable architecture for a conventional MLP. Let n_1 be the optimal number of hidden nodes found using Y by varying n_1 from 2–20. Now we run the feature selection MLP (with n_1 hidden neurons) on Y. After the groups of features are selected, we project Y on the selected feature space to obtain Y'.

In order to assess how good these selected features are, we train a conventional MLP using the selected groups of features, i.e. using the data set Y'. In the reduced feature space, we again use the 10-fold cross validation mechanism on Y' to find the most desirable architecture, n_2 for Y'. For this, we vary the number of hidden nodes from 2 to 15. Next, we train an MLP with n_2 hidden nodes using data set Y' and test it on X'_j, where X'_j is the projected version of X_j that was left out in the outer loop. This process is repeated for all $X_j; j = 1, \ldots, 10$ in the outer loop to get the misclassification rate using the selected features. Finally, the entire process is repeated 10 times, every time using a different random partition in the outer loop. We report the average of the error rates.

The training is terminated when either the misclassification error reduces to less than 10% or the number of iterations reaches 1000. We have followed this protocol for all data sets. Note that, such a uniform principle may not be best for all data sets. A group of features (sensor) is considered useful if the associated gate opens more than 10%. If for more than one choice of number of hidden nodes, the error (number of misclassifications) attains the minimum value, then the smallest number is selected as the number of hidden nodes.

Table 2. Selection of Groups(%) for Iris_1 data for different values of penalty.

Penalty	Groups 1	Groups 2	Misclassification error (%)	Average no. of groups
All groups	100.00	100.00	3.33	2.00
0	70.00	90.00	4.00	1.60
2	40.00	60.00	4.67	1.00
5	30.00	70.00	4.67	1.00
10	50.00	50.00	6.67	1.00

For Iris_1 data, we conducted our experiments with four different penalty levels: $\lambda = 0, 2, 5, 10$. We have mentioned earlier that for $\lambda = 0$, the proposed method reduces to the method by Chakraborty and Pal[12] and hence it compares our results with those of Chakraborty and Pal.[12] We note from Table 2 that for $\lambda = 0$, on an average 1.6 groups are selected resulting 96% classification accuracy, whereas, the classification accuracy taking all groups is 96.67%. This insignificant compromise in classification accuracy results from 20% reduction in group selection. In Table 2, we find that group 2 usually has a higher frequency than the first group, which is intuitive as group 2 comprises features 3 and 4, the best two features of the Iris data set. With increase in λ value, the two groups are selected mutually exclusively. For this data set, the correlation between Group 1 and Group 2 is 0.18 and that between Group 2 and Group 1 is 0.67. With further increase in λ value, i.e. with $\lambda = 10$, both groups 1 and 2 get selected 50% of time, which results in an increase in misclassification error rate.

If we use a very high value of λ, no groups may be selected because then the penalty term will dominate over the error term. Figure 2 depicts how the β values change with iterations. We find that with iteration, β value for group 1 becomes more negative, on the contrary, group 2's β value becomes more positive. The variation of total error (TE) with iterations is shown in Fig. 3. Comparing Figs. 2 and 3 we find that with iteration as group 2 is selected, i.e. as the attenuator value of group 2 becomes close to one, the error rate decreases rapidly.

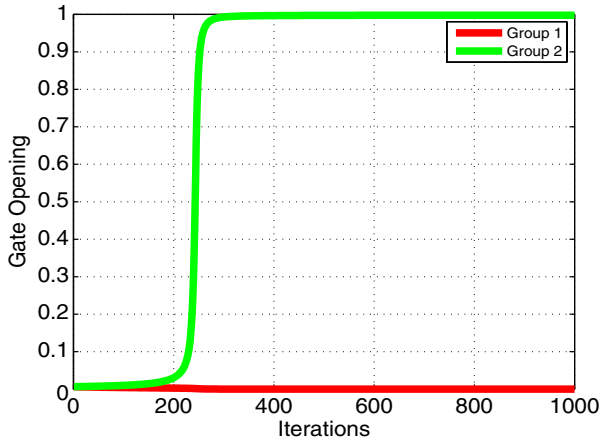

Fig. 2. Gate Opening with iteration for Iris_1 data set with penalty 2.

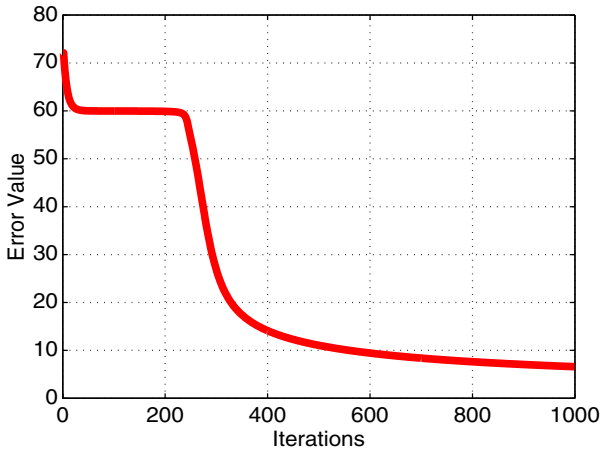

Fig. 3. Variation of Misclassification Error with iteration for Iris_1 data set with penalty = 2.

Table 3. Selection of Groups(%) for Iris_2 Data for different values of penalty.

Penalty	Groups 1	Groups 2	Groups 3	Misclassification error (%)	Average # groups
All	100.00	100.00	100.00	4.00	3.00
0	100.00	80.00	80.00	4.00	2.60
2	10.00	100.00	0.00	4.67	1.10
5	0.00	50.00	50.00	6.67	1.00
10	30.00	40.00	30.00	10.00	1.00

For Iris_2, the effect of λ values on redundancy control is shown in Table 3. When the λ value is zero, the first group is selected in all runs, while each of the second and third groups is selected 80% times. When $\lambda = 2$, the first and second groups got selected with frequency 10% and 100%, respectively whereas group 3 is not selected at all. The correlation in Table 4 depicts that groups 2 and 3 have significant correlations with all others. As a result on an average a single group is selected with higher λ value. Group 1 is completely rejected when λ value is 5 as this group is less discriminative than the other two.

Table 4. Group correlation (Eq. (13)) values and alternative correlation (Eq. (14)) values (shown in ()) for the Iris_2 Data.

Groups	1	2	3
1	1.00	0.18 (0.47)	0.17 (0.43)
2	0.67 (0.71)	1.00	0.49 (0.76)
3	0.53 (0.61)	0.80 (0.86)	1.00

The results for the four variants of RS data sets are given in Table 5. From this table, we can see that with a positive penalty value, the average number of groups selected decreases. By inspecting the correlation table of RSData_1 (Table 6), we find five pairs of significantly correlated groups, namely, {1, 2}, {1, 3}, {3, 7}, {2, 3} and {5, 7}. And by inspecting Fig. 4, we see that with no control on redundancy groups {1, 2}, {1, 3} are selected together for few runs. But with a positive λ value, inspection of detailed results (data not shown) reveals that those groups

Table 5. Selection of features for RS datasets using GFSMLP-RoC.

Dataset	Penalty (λ)	Average no. of groups	Misclassification error (%)
RSData_1	0	3.80	16.92
	2	2.80	20.71
	5	2.50	21.23
	10	1.90	23.03
RSData_2	0	5.50	17.01
	2	2.80	18.42
	5	2.40	22.94
	10	2.50	20.63
RSData_3	0	3.90	19.80
	2	2.90	20.67
	5	1.90	23.23
	10	1.60	23.66
RSData_4	0	5.40	19.10
	2	3.50	20.42
	5	3.10	23.23
	10	2.70	25.05

Table 6. Group correlation matrix for RS-Data_1.

Groups	1	2	3	4	5	6	7
1	1.00	**0.93**	**0.84**	0.07	0.27	0.21	0.55
2	**0.93**	1.00	**0.93**	0.13	0.38	0.26	0.66
3	**0.84**	**0.93**	1.00	0.19	0.51	0.32	**0.78**
4	0.07	0.13	0.19	1.00	0.67	0.17	0.40
5	0.27	0.38	0.51	0.67	1.00	0.37	**0.84**
6	0.21	0.26	0.32	0.17	0.37	1.00	0.45
7	0.55	0.66	**0.78**	0.40	**0.84**	0.45	1.00

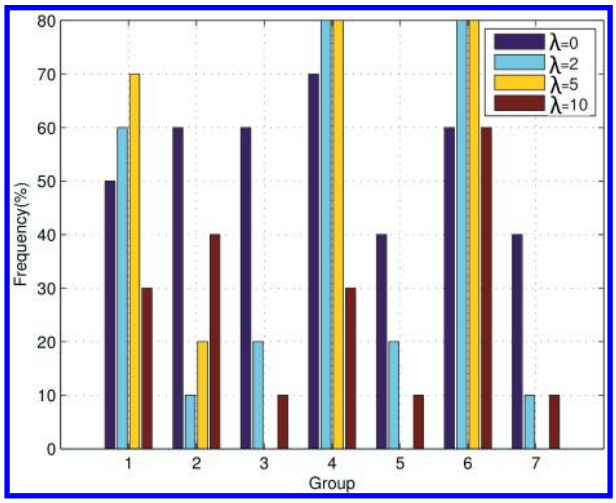

Fig. 4. Selection Frequency of different groups for the RSData_1 data set.

are selected in a mutually exclusive manner, which proves the effective control of redundancy. Similarly, from the information about simultaneous selection of different features (data not shown) for RSData_2, we find that group 9 and group 6 are selected together when there is no redundancy control but since they are highly correlated (group 9 is a noise-added version of group 6) they are selected mutually exclusively with positive penalty values. Similar results are observed for RSData_3 and RSData_4. In case of RSData_3, like RSData_1, group 1 and group 3 have a high correlation value. Analogously, groups 1 and 3 are selected in an alternate manner with increase in the penalty factor.

For the distraction Data set, the sensor/group selection result is given in Table 7. Consulting this table, we see that without any redundancy control, our scheme reduces the average number of groups (components) to more than 50% with a marginal sacrifice in the misclassification error rate. By inspection, we found that in case of penalty value 0, groups 1 (Frontal component) and 7 (Right Motor component) got selected together for a few runs. But they got selected mutually exclusively with increase in penalty value. Analyzing the pairwise group correlation values we find that those two groups are highly correlated (the correlation value is of 0.93). For a penalty factor of 5, the misclassification error value is improved with less number of groups on an average, which in turn suggests that there might be some derogatory group(s) present in the data set which are removed with higher penalty values.

The average number of groups selected for the LandSat data with different penalty level are given in Table 8. This table reveals that with increase in penalty, the average number of groups decreases with a slight increase in misclassification error. For instance, with penalty value 2, on an average 43% groups are selected with an increase of 7% misclassification error. For the Land Sat data, group 1 and group 2 have the highest correlation of 0.65. So, with an increase in the penalty value, the selection frequency of group 2 decreased drastically, as found in Fig. 5, which suggests the importance of group 1 over group 2 and also the effectiveness of control of redundancy by the proposed method.

Table 7. Selection of Groups(%) for distraction data for different values of penalty.

	Groups							Misclassification	Average no.
Penalty	1	2	3	4	5	6	7	error (%)	of groups
All groups	100.00	100.00	100.00	100.00	100.00	100.00	100.00	0.11	7.00
0	62.00	68.00	32.00	44.00	48.00	43.00	30.00	0.33	3.27
2	35.00	58.00	7.00	14.00	42.00	16.00	17.00	0.56	1.89
5	23.00	44.00	3.00	13.00	16.00	8.00	4.00	0.31	1.11
10	20.00	32.00	6.00	10.00	17.00	7.00	8.00	0.56	1.00

Table 8. Selection of groups for Land Sat data.

Penalty (λ)	Average no. of groups	Misclassification error (%)
All	4.00	12.10
0	2.90	15.05
2	1.70	19.00
5	1.30	22.00
10	1.20	24.98

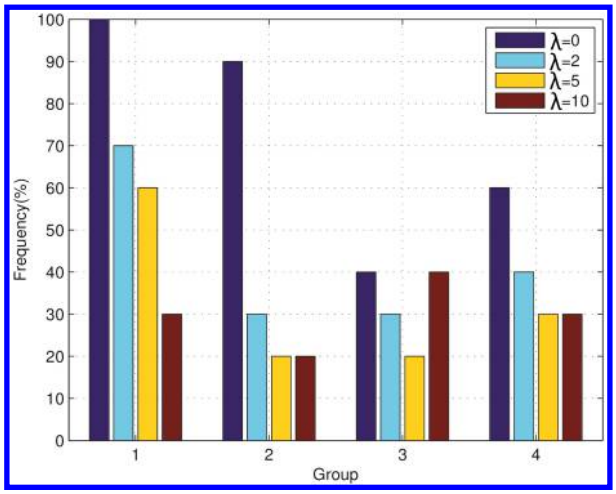

Fig. 5. Selection frequency of different groups of Land Sat dataset.

Table 9. Selection of groups for LRS Data.

Penalty (λ)	Average no. of groups	Misclassification error (%)
All	2.00	11.11
0	2.00	11.11
2	1.00	12.24
5	1.00	12.43

For the LRS data, the two groups are moderately correlated. So, with increase in the penalty value, one group is selected on an average, with almost 1% increase in the misclassification error (Table 9).

Table 10 depicts the significant correlation values for the gas sensor data. For this data set, with no redundancy control, our scheme selects almost 33% sensors with 1.3% increase in misclassification error; while with positive penalty values, the average number of groups decreases drastically with an increase in error value, as can be seen from Table 11.

Table 10. Significant Group correlation values (≥ 0.7) for gas sensor data.

F_1	F_2	ρ	F_1	F_2	ρ
12	3	0.77	7	8	0.92
11	3	0.79	8	7	0.92
12	15	0.83	9	10	0.93
12	16	0.83	10	9	0.93
15	12	0.83	11	12	0.96
16	12	0.83	12	11	0.96
11	16	0.86	13	14	0.97
16	11	0.86	14	13	0.97
11	15	0.87	15	16	0.97
15	11	0.87	16	15	0.97

Table 11. Selection of groups for gas sensor data.

Penalty (λ)	Average no. of groups	Misclassification error (%)
All	16.00	4.30
0	6.00	5.61
2	2.30	13.44
5	2.00	13.44
10	1.90	13.01

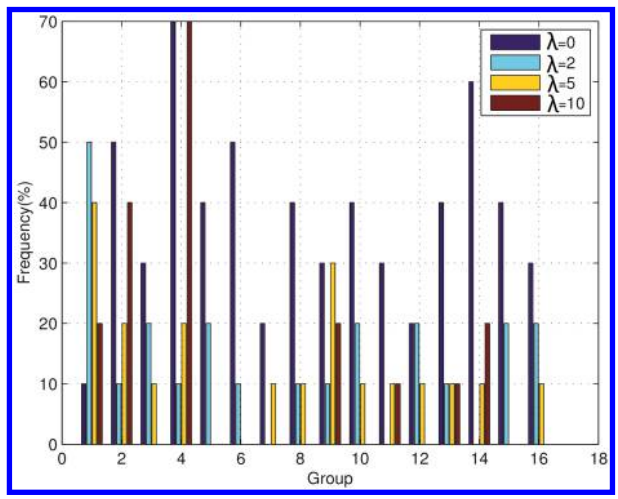

Fig. 6. Selection frequency of different groups for the gas sensor data set.

This suggests that there are useful dependent sensors. Consulting the Table 10, we do find that the pairs {15, 16}, {13, 14}, {11, 12}, {9, 10}, {8, 7} have high dependencies. So, with a positive penalty value, members from these group pairs are usually selected (Fig. 6) mutually exclusively (detailed data are not shown). With $\lambda = 0$, groups 13 and 14 are

selected together a few times (data not shown), but with a penalty factor of 2, group 13 has been selected with only 10% frequency whereas group 14 has not been selected at all.

3. An Alternative Definition of Group Dependency

In Eq. (12), we have defined $\text{dep}(S_l, S_m) = \mathcal{G}_{1 i \in S_l}\{\mathcal{G}_{2 j \in S_m}\{\rho^2(x_i, x_j) \forall\, j \in S_m\} \forall\, i \in S_l\}$, where $\mathcal{G}_1$ and $\mathcal{G}_2$ are the two aggregation functions. Previously, we have used $\mathcal{G}_1$ and $\mathcal{G}_2$ as min and max, respectively. Thus, the definition is as follows.

$$\text{dep}(S_l, S_m) = \min_{i \in S_l} \max_{j \in S_m} \rho^2(x_i, x_j). \qquad (13)$$

We have also demonstrated that this definition of group dependency works well. However, here we point out an example where these definition of aggregation functions behave differently than expected. Then, we propose an alternative definition of aggregation function $\mathcal{G}_1$ to address this issue.

Suppose there are two groups S_1 and S_2 consisting of four features each. Let us denote these features by $f_1, \ldots f_8$ where first four belong to group S_1 and the rest in S_2. Also assume that the features in the second group are the features in the first group with some added random noise from $\mathcal{N}(0, 0.5)$. Additionally, we have the information that features 1, 2 and 3 (and therefore feature 5, 6, 7) are very important features for the classification task. On the other hand, feature 4 (therefore feature 8) is completely random feature from $\mathcal{N}(0, 0.05)$. Thus, the feature pairs $\{f_1, f_5\}$, $\{f_2, f_6\}$, $\{f_3, f_7\}$ have high Pearson's correlation values whereas the feature pair $\{f_4, f_8\}$ has low correlation value. So, by going through the group penalty definition, Eq. (13), both $\text{dep}(S_1, S_2)$ and $\text{dep}(S_2, S_1)$ is very low due to $\{f_4, f_8\}$ correlation value. But, intuitively, the two groups are equally good and selection of only one is sufficient. As by definition (13), the dependency between them is low, it might result in selection of both. This is not very desirable and we do not want that. Note that, here the system will work, but the redundancy control will not be good. To resolve this problem, we have proposed an alternative definition of group correlation as follows.

$$\text{dep}(S_l, S_m) = \text{avg}_{i \in S_l} \max_{j \in S_m} \rho^2(x_i, x_j). \qquad (14)$$

Now, we demonstrate the utility of this definition on three variants of Iris data. The group dependency values for Iris_1 are displayed in Table 12; while the same for Iris_2 are included in Table 4. The average number of selected groups and misclassification error using different penalty levels are given in Table 13. The results using the old definition are given in {}. For these two data sets, the new definition works equally well as that of the old definition.

Table 12. Alternative Group dependency matrix for Iris_1 Data.

Groups	1	2
1	1.00	0.47
2	0.71	1.00

Table 13. Selection of features for datasets using the old and new dependency measure.

Dataset	Penalty (λ)	Average no. of groups (Grp. Freq.)	Misclassification error (%)
Iris_1	All	2.00	2.67
	0	1.80 $(80+100)$	3.33
		$\{1.60\ (70+90)\}$	4.00
	2	1.00 $(20+80)$	4.67
		$\{1.00\ (40+60)\}$	4.67
	5	1.00 $(20+80)$	5.33
		$\{1.00\ (30+70)\}$	4.67
	10	1.00 $(30+70)$	9.33
		$\{1.00\ (50+50)\}$	6.67
Iris_2	All	3.00	4.00
	0	2.20 $(100+70+50)$	4.00
		$\{2.60\ (100+80+80)\}$	4.00
	2	1.00 $(30+40+30)$	4.67
		$\{1.10\ (10+100+0)\}$	4.67
	5	1.00 $(30+60+10)$	6.67
		$\{1.00\ (0+50+50)\}$	6.67
	10	1.00 $(50+30+20)$	10.00
		$\{1.00\ (30+40+30)\}$	10.00

In order to further demonstrate the usefulness of our alternative definition of group dependency over the earlier one, we took a third variant, Iris_3 data consisting of three groups. The first group contains the first two features of Iris data. The second group consists of third and fourth features of Iris data with a third random feature following $N(0, 0.05)$ distribution. While, the last group is the noise added version of group 2 with three features. In this case, there are three groups in which the second and third groups are almost equally good groups. The two groups

Table 14. Alternative group dependency matrix for Iris_3 Data, the values in () represent the group correlation using Eq. (13).

Groups	1	2	3
1	1.00	0.47 (0.18)	0.47 (0.18)
2	0.48 (0.01)	1.00	0.67 (0.0)
3	0.48 (0.02)	0.68 (0.04)	1.00

should have a high dependency value as they consist of practically the same features. But our earlier definition fails to capture this situation as evident from the dependency shown in () Table 14, which also depicts the group dependency using the alternative definition. In Table 14, we see that groups 2 and 3 have a significant level of dependency as per the new definition. This demonstrates that this alternative definition of group dependency is quite effective. The results of group selection using both our old and modified definitions of group dependency are shown in Table 15. This result reveals that using our modified definition of dependency, with high λ value, the dependent groups are selected mutually exclusively. On the contrary, using the old definition, with $\lambda = 2$, groups 2 and 3 got selected together 10% of the time as the dependency between them is not significant. Moreover, with $\lambda = 2$, using the old definition the average number of groups selected is 1.3 which is reduced to 1.0 for the new definition. This suggests that our new dependency definition matches our intuition.

Table 15. Selection of features for Iris_3 using the two dependency measures.

Measure of dependency	Penalty (λ)	Average no. of groups (Group Freq.)	Misclassification error (%)
Old definition	All	3.00	4.00
	0	2.30 (70 + 80 + 80)	4.00
	2	1.30 (20 + 40 + 70)	4.67
	5	1.00 (10 + 50 + 40)	7.73
	10	1.00 (10 + 30 +60)	9.33
Alternative definition	All	3.00	4.00
	0	2.60 (100 + 70 + 90)	2.00
	2	1.00 (20 + 40 + 40)	4.67
	5	1.00 (10 + 20 + 70)	7.73
	10	1.00 (10 + 40 +50)	8.67

4. An Alternative to Pearson's Correlation

Here we describe an alternative to Pearson's correlation coefficient. Why an alternative to Pearson's correlation is needed? The Pearson's correlation measure between two random variables A and B is linear and the $\text{corr}(A, B) = 0$ does not mean that A and B are completely independent. In order to address this independence issue, Gebelein[66] had introduced a new dependence coefficient:

$$\rho_{\text{GM}}(A, B) = \sup_{fg} \rho(f(A), g(B)),$$

where $\rho_{\text{GM}}(A, B)$ is the Gebelein's Maximal Correlation (GMC)[66] and $\rho(\cdot, \cdot)$ is the Pearson's correlation. Here, the supremum is taken over all possible functions f, g with finite variance. Unlike Pearson's correlation, GMC has the independence property, i.e. A and B are independent, iff $\rho_{\text{GM}}(A, B) = 0$. GMC takes value between 0 and 1 with equality on either side and it is also symmetric. This is a very general definition of dependency.

Kursun and Favorov[67] have shown that using set of INteractive BAckpropagating Dendrities (SIN-BAD) strategy,[68,69] one can (approximately) measure the GM correlation between two random variables A and B. The detail of the SINBAD algorithm is given in Ref. 67. This algorithm requires Support Vector Machines, so we have used "SVM light" toolbox[70] for this purpose. Using the GM correlation, we repeat the group selection experiment on the two variants of Iris and report the results in Table 17. As an illustration, we have provided the group dependency using GMC definition for Iris_2 data in Table 16. As expected, the GMC between two groups is usually higher than the measure of dependency computed using Pearson's correlation and hence the impact of redundant groups is more severe when we use GMC. From these limited results, we see that our scheme works equally well to remove the level of redundancy among selected feature-groups.

Table 16. Group dependency Matrix using GMC for Iris_2 Data.

Groups	1	2	3
1	1.00	0.95	0.89
2	0.95	1.00	0.97
3	0.89	0.97	1.00

Table 17. Selection of features for datasets using the GM correlation in measure of dependency.

Dataset	Penalty (λ)	Average no. of groups (Group Freq.)	Misclassification error (%)
Iris_1	0	1.80 (90 + 90)	4.67
	2	1.00 (40 + 60)	8.00
	5	1.00 (40 + 60)	8.00
Iris_2	0	2.30 (80 + 80 + 70)	6.00
	2	1.00 (30 + 40 + 30)	9.33
	5	1.00 (20 + 50 + 30)	9.33

The group correlation definition using GMC, is given in Eq. (15).

$$\text{dep}(S_l, S_m) = \min_{i \in S_l} \max_{j \in S_m} \rho^2_{\text{GM}}(x_i, x_j). \qquad (15)$$

Since we could not find much advantage using the GMC and finding of GMC is computationally quite expensive, we do not experiment with other data sets using GMC.

5. An Alternative Learning Scheme

So far, we have learned the gate opening and network weights simultaneously by gradient descent technique. We began our training keeping all gates almost closed. In this section, we use an alternative and more effective learning scheme, named as mGFSMLP-CoR.

This learning scheme[7] comprises two stages. In the first stage, we learn only network weights keeping all gates completely opened. In the second stage, on the trained network we learn both network weights and gates simultaneously starting with all gates almost closed. This learning scheme bears some advantages over the earlier scheme. First, this learning scheme is less sensitive to initialization of weights. Second, if there are some bad or derogatory groups, those are easily removed in the second stage. Also, the selection of groups is consistent over different runs.

The result of mGFSMLP-CoR on several data sets are given in Table 18. Consulting the results in Table 18, we find that the results are very much comparable with that of the old learning scheme. Also, it is evident that the distinct number of groups is very close to the average number of groups selected. This suggests selection of almost a fixed set of groups over different runs. In other words, the new learning scheme reduces the impact of the initialization significantly. The misclassification error value is also very much comparable to the original method which suggests the effectiveness of mGFSMLP-CoR.

Next, we adapt our method to a radial basis function (RBF) network.

6. RBF NETWORK for Group Feature Selection with Controlled Redundancy (GFSRBF-CoR)

In this section, we propose a RBF network for group feature selection. Like MLP, RBF and other probabilistic neural networks are useful in various applications.[63,64,75–84] Given a training data set (X, Y), where $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\} \subset \mathbb{R}^p$, a typical RBF network computes the function $F^*(\mathbf{x}) = \sum_{i=1}^{N} w_i \Phi_i(\mathbf{x})$, where Φ_i is the ith basis function, $\Phi_i(\mathbf{x}) = \exp\{-\frac{\|\mathbf{x}-\boldsymbol{\mu}_i\|}{\sigma_i^2}\}$ in case of a Gaussian basis function. $\boldsymbol{\mu}_i$ and σ_i are the center and spread of the ith basis function, respectively. $\|\cdot\|$ is the Euclidean norm. Let, $\mathbf{x} = [x_1, x_2, \ldots, x_p]^T$ and $\boldsymbol{\mu}_i = [\mu_{i1}, \mu_{i2}, \ldots, \mu_{ip}]^T$. then,

$$\Phi_i(\mathbf{x}) = \prod_{j=1}^{p} \exp\left\{-\frac{(x_j - \mu_{ij})^2}{\sigma_i^2}\right\}. \qquad (16)$$

As discussed earlier, here also the data are collected from g sensors. Let, for a data point $\mathbf{x}$, the features from the jth sensor be denoted by $\mathbf{s}^j$. Hence $\mathbf{x} = [\mathbf{s}^1 \mathbf{s}^2 \cdots \mathbf{s}^g]^T$. Similarly, $\boldsymbol{\mu}_i$ can be written as $\boldsymbol{\mu}_i = [\mathbf{m}_i^1 \mathbf{m}_i^2 \cdots \mathbf{m}_i^g]$, where $\mathbf{s}^j$ and $\mathbf{m}_i^j$ have the same dimensionality, $\forall\, j \in \{1, \ldots, g\}$. Now, the jth component of the ith basis function, i.e. the contribution of jth group/sensor to the ith basis function, C_i^j can

Table 18. Selection of features for datasets using mGFSMLP-CoR.

Dataset	Penalty (λ)	Average # of features	Unique # of features	Misclassification error (%)
Iris_1	0	2.00	2	3.47
	2	1.00	1	3.93
	5	1.00	1	3.93
	10	1.00	1	3.93
Iris_2	0	2.00	2	3.47
	2	1.00	1	3.93
	5	1.00	1	3.93
	10	1.00	1	3.93
LRS	0	2.00	2	9.43
	2	1.00	1	11.32
	5	1.00	1	11.32
	10	1.00	1	11.32
LandSat	0	3.90	4	13.06
	2	2.30	3	13.53
	5	2.00	2	15.56
	10	1.90	2	18.35
Dist. Data	0	3.40	5	0.23
	2	2.00	2	0.31
	5	1.10	2	0.51
	10	1.00	2	0.74
Gas Sensor	0	12.60	15	4.87
	2	4.20	8	11.20
	5	3.20	8	12.49
	10	1.80	4	12.89
RSData_1	0	6.70	7	16.91
	2	4.00	4	20.67
	5	2.60	3	21.16
	10	1.60	3	22.66
RSData_2	0	8.90	9	16.81
	2	3.00	5	17.85
	5	2.50	4	20.26
	10	2.00	3	21.40
RSData_3	0	6.60	7	19.42
	2	3.00	3	20.55
	5	2.20	3	23.02
	10	2.00	3	23.47
RSData_4	0	8.70	9	18.81
	2	3.10	5	20.02
	5	2.60	3	23.24
	10	1.60	3	25.05

be written as

$$C_i^j = \exp\left\{-\frac{\|\mathbf{s}_j - \mathbf{m}_i^j\|^2}{\sigma_i^2}\right\}. \qquad (17)$$

Then, Eq. (16) can be written as $\Phi_i(\mathbf{x}) = \prod_{j=1}^{g} C_i^j$.

Now, our goal is to eliminate the bad feature groups or sensors, i.e. the effect of a bad group should not be propagated into the network. So, if the jth group is bad, C_i^j should have no effect on $\Phi_i(\mathbf{x})$, $\forall\, i$. In order to ensure that, we introduce attenuator/gate function associated with each feature group[12] like the GFSMLP-CoR network discussed earlier.

Let, F_j is the attenuator function with the jth sensor. As done in Ref. 12, here also the modulated

output from the sensory basis function is written as

$$C_i^j = \left[\exp\left\{-\frac{\|\mathbf{s}_j - \mathbf{m}_i^j\|^2}{\sigma_i^2}\right\}\right]^{F_j}. \quad (18)$$

For a bad feature group j, F_j should take the value 0, so that C_i^j is 1 and hence it would not have any effect on $\Phi_i(\mathbf{x})$. On the other hand, for a good feature j, F_j should take the value 1, so that Eq. (18) reduces to Eq. (17). As in case of GFSMLP-CoR, here also we have taken

$$F_j = \frac{1}{1+\exp(-\beta_j)},$$

where β_j is the tunable gate parameter.

This network is realized using four layers (Fig. 7) as in Ref. 12. Here we follow the same notation as in Ref. 12. We denote the output of the ith layer by $z^{(i)}$.

The first layer is the input layer that consists of p nodes. The second layer is the component function layer. If there are m basis functions, then this layer consists of $g \times m$ nodes. Thus, $z_{ij}^{(2)}$ denotes the output of the component function related to the jth group of the ith basis function. The third layer is the basis function layer where the number of nodes is the number of basis functions, m. The output of the ith node is $z_i^{(3)} = \prod_{j=1}^{g} z_{ij}^{(2)}$.

The last layer is the output layer consisting of c nodes where c is the number of classes in the data set. This layer is fully connected with the previous layer. Let w_{ij} be the weight associated with the link between the jth node of the third layer to the ith node of the output layer. Then the output of the ith node in the output layer is

$$z_i^{(4)} = f\left(-\sum_{j=1}^{m} w_{ij} z_j^{(3)}\right),$$

where f is the sigmoidal function in order to ensure the value of $f(\cdot)$ between 0 and 1. Thus,

$$z_i^{(4)} = \frac{1}{1+\exp(\sum_{j=1}^{m} w_{ij} z_j^{(3)})}.$$

6.1. *Learning rules*

Given a training data (X, Y), let for a data point $\mathbf{x}$ the output vector be $\mathbf{y} = [y_1, y_2, \ldots, y_c]^T$ and E_x be the instantaneous error for the data point $\mathbf{x}$. Thus, the total system error on the training data is

$$E = \sum_{\mathbf{x}\in X} E_x = \frac{1}{2}\sum_{\mathbf{x}\in X}\sum_{i=1}^{c}(z_i^{(4)} - y_i)^2.$$

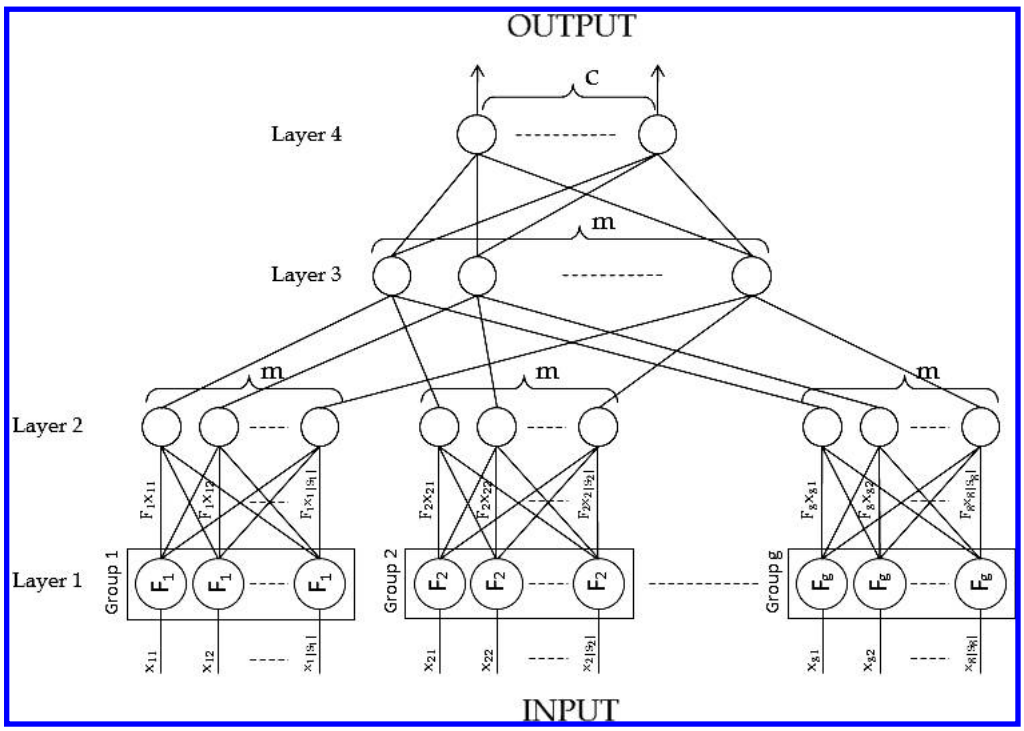

Fig. 7. GFS-RBF: Group feature selection RBF network.

In this work, we shall assume fixed center and spread for the basis functions. We use gradient-descent technique to learn group modulator β_j and w_{ij}.

Now, in order to control the redundancy among the selected groups, we add a penalty term as in Eq. (11). So, the total error, TE becomes

$$\text{TE} = E + \lambda P(X),$$

where

$$P(X) = \frac{1}{g(g-1)} \sum_{j=1}^{g} F_j \sum_{l \neq j} F_l \, \text{dep}(S_j, S_l).$$

The learning rules can be derived in a straightforward manner by differentiating TE with respect to w_{ij} and β_j. Like the GFSMLP-CoR, the initial βs are assigned random values between $[-5.05, -4.95]$.

To obtain the spread and center parameters we run the Fuzzy c-means (FCM)[71] clustering algorithm on X, with fuzzifier (m) value as 2, to get the cluster centers. Then we assign the cluster centers to the basis centers $\boldsymbol{\mu}_i$. The spread function, σ_i is calculated as follows: $\sigma_i = \min_{j \neq i} \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|$. Note that, there are other ways of initializing these parameters.

6.2. *Experimental results*

As an illustration, we use a few data sets to validate our GFSRBF-CoR scheme. The group selection on Iris_1 data is given in Table 19. From this table, we see that the second group is selected consistently for all penalty values, even when the penalty is 0! So, does it suggest that RBF network is more consistent than the MLP in sensor selection? We shall address this question after discussion of the results. In case of Iris_2 data, with no penalty value, the top two important groups (group 2 and group 3) are selected for all runs. But as they are highly dependent on each other, with a positive penalty value, only one of them should get selected. And as group 3 contains only features 3 and 4, group 3 is a better discriminator than group 2. So it should select group 3. By looking at the Table 20, we see that this is indeed the scenario, i.e. only group 3 is selected consistently over different runs. From Table 21 (along with inspection of detailed results — data not shown), we find that for LRS data with a positive penalty factor, one of the two groups is selected as they are moderately correlated. The result on LandSat data set in Table 22 reveals that as groups 1 and 2 are maximally correlated (the correlation between them is indeed the highest among all group pairs), they are selected

Table 19. Selection of groups(%) for Iris_1 Data using GFSRBF-CoR.

	Groups		Misclassification	Average no.
Penalty	1	2	error (%)	of groups
0	0.00	100.00	4.00	1.00
2	0.00	100.00	4.00	1.00
10	0.00	100.00	4.00	1.00

Table 20. Selection of Groups(%) for Iris_2 Data using GFSRBF-CoR.

	Groups			Misclassification	Average no.
Penalty	1	2	3	error (%)	of groups
0	0.00	100.00	100.00	6.00	2.00
2	0.00	0.00	100.00	6.67	1.00
10	0.00	0.00	100.00	6.67	1.00

Table 21. Selection of Groups(%) for LRS Data using GFSRBF-CoR.

	Groups		Misclassification	Average no.
Penalty	1	2	error (%)	of groups
0	100.00	100.00	14.88	2.00
2	60.00	40.00	14.50	1.00
10	40.00	60.00	18.83	1.00

Table 22. Selection of Groups(%) for LandSat Data using GFSRBF-CoR.

Penalty	Groups 1	2	3	4	Misclassification error (%)	Average no. of groups
0	100	100	60	80	16.16	3.40
2	10	60	30	40	23.43	1.40
10	40	0	50	10	28.52	1.00

disjointly with positive penalty values. Now, we are in a position to address the question we raised earlier: Is RBF network more consistent in selecting groups than the MLP network? Unlike an MLP, we do not initialize the RBF weight vectors randomly, rather they are initialized using centroids of clusters. So if the cluster centroids are not significantly affected by the initialization, the performance of RBF is also not going to be affected much. In this particular case, we use the FCM algorithm to find the cluster centroids and FCM centroids are less sensitive to initialization.

7. Conclusion

In this paper, we have proposed a group feature/sensor selection scheme which can control redundancy. Group selection approach is very important in many real-life application for reducing complexity and cost. As an example, suppose there are two sets (groups) of features, one from MRI scan and the other from X-ray. But, for the target application one is sufficient, so selection of the most important image modality between the two is important as it reduces the design cost and complexity of decision making for the target application. Chakraborty and Pal[12] proposed a scheme for group feature selection. Their scheme can select useful groups and remove bad/derogatory groups. In this paper, we have extended that work to control the redundancy between groups. We have proposed our scheme in two connectionist frameworks, MLP and RBF. We have also used an alternative learning scheme for MLP network which gives more consistent results than the first scheme. Two alternative definitions of dependency between groups have also been proposed. In place of Pearson's correlation, Gebelein's correlation has also been used to demonstrate the effectiveness of our scheme. This correlation measure is nonlinear in contrast to the linear Pearson's measure. And using this nonlinear measure, we have also shown the effectiveness of our proposed scheme. In this work, the penalty factors have been chosen in an *ad hoc* manner, but any systematic method like cross-validation scheme can also been used to select this parameter, if needed.

Acknowledgment

The authors gratefully acknowledge the cloud computing services provided by the Chunghwa Telecom Co. under the "Networked Communications Program".

References

1. K. Fukunaga, *Introduction to Statistical Pattern Recognition* (Academic, New York, 1990).
2. P. A. Devijver and J. Kittler, *Pattern Recognition, A Statistical Approach* (Prentice Hall International, Inc., London, 1982).
3. A. Jain and D. Zongker, Feature selection: Evaluation, application, and small sample performance, *IEEE Trans. Pattern Anal. Mach. Intell.* **19**(2) (1997) 153–158.
4. N. R. Pal and K. K. Chintalapudi, A connectionist system for feature selection, *Neural Parallel Sci. Comput.* **5**(3) (1997) 359–381.
5. Y. Aksu, D. J. Miller, G. Kesidis and Q. X. Yang, Margin-Maximizing feature elimination methods for linear and nonlinear Kernel-based discriminant functions, *IEEE Trans. Neural Netw.* **21**(5) (2010) 701–717.
6. L. Zhou, L. Wang and C. Shen, Feature selection with redundancy-constrained class separability, *IEEE Trans. Neural Netw.* **21**(5) (2010) 853–858.
7. R. Chakraborty and N. R. Pal, Feature selection using a neural framework with controlled redundancy, *IEEE Trans. Neural Netw.* (2014), doi 10.1109/TNNLS.2014.2308902.
8. C. Shen, H. Li and M. J. Brooks, Supervised dimensionality reduction via sequential semidefinite programming, *Pattern Recognit.* **41**(12) (2008) 3644–3652.
9. G. Rodríguez-Bermúdez, P. J. García-Laencina and J. Roca-Dorda, Efficient automatic selection and combination of EEG features in least squares classifiers for motor imagery brain-computer interfaces, *Int. J. Neural Syst.* **23**(4) (2013) 1350015–1350017.

10. F. Nie, S. Xiang, Y. Jia, C. Zhang and S. Yan, Trace ratio criterion for feature selection, in *Proc. 23rd AAAI Conf. Artif. Intell.*, July 2008, pp. 671–676.
11. L. Wang, Feature selection with kernel class separability, *IEEE Trans. Pattern Anal. Mach. Intell.* **30**(9) (2008) 1534–1546.
12. D. Chakraborty and N. R. Pal, Selecting useful groups of features in a connectionist framework, *IEEE Trans. Neural Netw.* **19**(3) (2008) 381–396.
13. D. Chakraborty and N. R. Pal, Integrated feature analysis and fuzzy rule based system identification in a neuro-fuzzy paradigm, *IEEE Trans. Syst. Man Cybern. B* **31**(3) (2001) 391–400.
14. W. Siedlick and J. Sklansky, A note on genetic algorithms for large-scale feature selection, *Pattern Recognit. Lett.* **10** (1989) 335–347.
15. D. W. Ruck, S. K. Rogers and M. Kabrisky, Feature selection using a multilayer perceptron, *J. Neural Netw.* **2** (1990) 40–48.
16. M. Last, A. Kandel and O. Maimon, Information-theoretic algorithm for feature selection, *Pattern Recognit. Lett.* **22** (2001) 799–811.
17. C. Ding and H. Peng, Minimum redundancy feature selection from microarray gene expression data, *J. Bioinform. Comput. Biol.* **3**(2) (2005) 185–205.
18. T. R. Golub, Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring, *Science* **286** (1999) 531–537.
19. N. R. Pal, K. Aguan, A. Sharma and S. Amari, Discovering biomarkers from gene expression data for predicting cancer subgroups using neural networks and relational fuzzy clustering, *BMC Bioinformatics* **8.5** (2007) 1–8.
20. N. R. Pal, A fuzzy rule based approach to identify biomarkers for diagnostic classification of cancers, FUZZ-IEEE 2007, *IEEE Int. Conf. Fuzzy Systems*, Imperial College, London, UK, 23–26 July 2007, Proceedings, pp. 1–6.
21. Y.-S. Tsai, C.-T. Lin, G. C. Tseng, I.-F. Chung and N. R. Pal, Discovery of dominant and dormant genes from expression data using a novel generalization of SNR for multi-class problems", *BMC Bioinformatics* **9.425** (2008) 1–33.
22. Y.-S. Tsai, K. Aguan, N. R. Pal and I.-F. Chung, Identification of single- and multiple-class specific signature genes from gene expression profiles by group marker index, *PLoS one* **6** (2011).
23. I. Guyon and A. Elisseeff, An introduction to variable and feature selection, *J. Mach. Learn. Res.* **3** (2003) 1157–1182.
24. I. Guyon, J. Weston, S. Barnhill and V. Vapnik, Gene selection for cancer classification using support vector machines, *Mach. Learn.* **46** (2002) 389–422.
25. X. Zhou and D. P. Tuck, MSVM-RFE: Extensions of SVM-RFE for multiclass gene selection on DNA microarray data, *Bioinformatics* **23**(9) (2007) 1106–1114.
26. W. Dinkelbach, On nonlinear fractional programming, *Manage. Sci.* **13**(7) (1967) 492–498.
27. H. Peng, F. Long and C. Ding, Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy, *IEEE Trans. PAMI* **27**(8) (2005) 1226–1238.
28. L. Yu and H. Liu, Efficient feature selection via analysis of relevance and redundancy, *J. Mach. Learn. Res.* **5**(10) (2004) 1205–1224.
29. Y. Hong, S. Kwong, Y. Chang and Q. Ren, Unsupervised feature selection using clustering ensembles and population based incremental learning algorithm, *Pattern Recognit.* **41**(9) (2008) 2742–2756.
30. S. Chatterjee and A. S. Hadi, Influential observations, high leverage points, and outliers in linear regression, *Statistical Science* **1**(3) (1986) 379–393.
31. K. Z. Mao, Identifying critical variables of principal components for unsupervised feature selection, *IEEE Trans. Syst. Man Cybern. Part B* **35**(2) (2005) 339–344.
32. J. G. Dy, C. E. Brodle, A. C. Kak, L. S. Broderick and A. M. Aisen, Unsupervised feature selection applied to content-based retrieval of lung images, *IEEE Trans. Patt. Anal. Mach. Intell.* **25**(3) (2003) 373–378.
33. P. A. Estévez, M. Tesmer, C. A. Perez and J. M. Zurada, Normalized mutual information feature selection, *IEEE Trans. Neural Netw.* **20**(2) (2009) 189–201.
34. G. Qu, S. Hariri and M. Yousif, A new dependency and correlation analysis for features, *IEEE Trans. Knowl. Data Eng.* **17**(9) (2005) 1199–1207.
35. H. Lee, C. Chen, J. Chen and Y. Jou, An efficient fuzzy classifier with feature selection based on fuzzy entropy, *IEEE Trans. Syst. Man Cybern. B* **31**(3) (2001) 426–432.
36. R. Liu, N. Yang, X. Ding and L. Ma, An unsupervised feature selection algorithm: Laplacian score combined with distance-based entropy measure, *Third Int. Symp. Intelligent Information Technology Application* (2009), pp. 65–68.
37. D. Koller and M. Sahami, Toward optimal feature selection, in *Proc. 13th Int. Conf. Machine Learning* (1996), pp. 284–292.
38. X. He, D. Cai and P. Niyogi, Laplacian score for feature selection, in *NIPS* (MIT Press, 2005), pp. 507–514.
39. C. Boutsidis, M. W. Mahoney and P. Drineas, Unsupervised feature selection for the k-means clustering problem, in *NIPS* (MIT Press, 2009), pp. 153–161.
40. M. W. Mahoney and P. Drineas, CUR matrix decompositions for improved data analysis, *Proc. Natl. Acad. Sci.* **106**(3) (2009) 697–702.
41. K. Kira and L. A. Rendell, A practical approach to feature selection, in *Machine Learning*: *Proc. 9th Int. Conf.* (1992), pp. 249–256.

42. M. A. Hall, Correlation-based feature selection for machine learning, The University of Waikato (1999).
43. M. Banerjee and N. R. Pal, Feature selection with SVD Entropy: Some modification and extension, *Inform. Sci.* **264** (2014) 118–134.
44. J. Huang, T. Zhang and D. Metaxas, Learning with structured sparsity, *J. Mach. Learn. Res.* **12** (2011) 3371–3412.
45. M. Yuan and Y. Lin, Model selection and estimation in regression with grouped variables, *J. R. Stat. Soc. B Stat. Methodol.* **68**(1) (2006) 49–67.
46. L. Meier, S. V. D. Geer and P. Bhlmann, The group lasso for logistic regression, *J. R. Stat. Soc. B Stat. Methodol.* **70**(1) (2008) 53–71.
47. J. Liang, F. Wang, C. Dang and Y. Qian, A group incremental approach to feature selection applying rough set technique, *IEEE Trans. Knowl. Data Eng.* **26**(2) 2014.
48. S. Maldonado, R. Weber and J. Basak, Simultaneous feature selection and classification using kernel-penalized support vector machines, *Inform. Sci.* **181**(1) (2011) 115–128.
49. H. Feng *et al.*, Incremental attribute reduction based on elementary sets, in *Rough Sets*, *Fuzzy Sets*, *Data Mining*, *and Granular Computing* (Springer, Berlin Heidelberg, 2005), pp. 185–193.
50. D. P. Muni, N. R. Pal and J. Das, Genetic programming for simultaneous feature selection and classifier design, *IEEE Trans. Syst. Man Cybern. B* **36**(1) (2006) 106–117.
51. S. Nijima and Y. Okuno, Laplacian linear discriminant analysis approach to unsupervised feature selection, *IEEE/ACM Trans. Comput. Biol. Bioinform.* **6**(4) (2009) 605–614.
52. Q. Song, J. Ni and G. Wang, A fast clustering-based feature subset selection algorithm for high-dimensional data, *IEEE Trans. Knowl. Data Eng.* **25**(1) (2013) 1–14.
53. http://archive.ics.uci.edu/ml/datasets.html.
54. G. Dornhege, B. Blankertz, G. Curio and K. R. Mueller, Boosting bit rates in noninvasive EEG single-trial classifications by feature combination and multiclass paradigms, *IEEE Trans. Biomed. Eng.* **51**(6) (2004) 993–1002.
55. H. Adeli and S. Ghosh-Dastidar, *Automated EEG-based Diagnosis of Neurological Disorders — Inventing the Future of Neurology* (CRC Press, Taylor & Francis, Boca Raton, Florida, 2010).
56. H. Adeli, Z. Zhou and N. Dadmehr, Analysis of EEG records in an epileptic patient using wavelet transform, *J. Neurosci. Meth.* **123**(1) (2003) 69–87.
57. A. Temko, G. Boylan, W. Marnane and G. Lightbody, Robust neonatal EEG seizure detection through adaptive background modeling, *Int. J. Neural Syst.* **23**(4) (2013) 1350018.
58. H. Adeli, S. Ghosh-Dastidar and N. Dadmehr, Alzheimerś disease: Models of computation and analysis of EEGs, *Clin. EEG Neurosci.* **36**(3) (2005) 131–140.
59. S. Ghosh-Dastidar, H. Adeli and N. Dadmehr, Voxel-based morphometry in Alzheimerś patients, *J. Alzheimerś Dis.* **10**(4) (2006) 445–447.
60. R. J. Martis, U. R. Acharya, J. H. Tan, A. Petznick, R. Yanti, K. C. Chua, E. Y. K. Ng and L. Tong, Application of empirical mode decomposition (EMD) for automated detection of epilepsy using EEG signals, *Int. J. Neural Syst.* **22**(6) (2012) 1250027.
61. S. Ghosh-Dastidar and H. Adeli, Improved spiking neural networks for EEG classification and epilepsy and seizure detection, *Integr. Comput.-Aided Eng.* **14**(3) (2007) 187–212.
62. S. Ghosh-Dastidar, H. Adeli and N. Dadmehr, Mixed-band wavelet-chaos-neural network methodology for epilepsy and epileptic seizure detection, *IEEE Trans. Biomed. Eng.* **54**(9) (2007) 1545–1551.
63. S. Ghosh-Dastidar, H. Adeli and N. Dadmehr, Principal component analysis-enhanced cosine radial basis function neural network for robust epilepsy and seizure detection, *IEEE Trans. Biomed. Eng.* **55**(2) (2008) 512–518.
64. S. Ghosh-Dastidar and H. Adeli, Spiking neural networks, *Int. J. Neural Syst.* **19**(4) (2009) 295–308.
65. A. S. Kumar, S. Chowdhury and K. L. Mazumder, Combination of neural and statistical approaches for classifying space-borne multispectral data, in *Proc. Int. Conf. Adv. Pattern Recognit. Digit. Tech.*, Calcutta, India, 1999, pp. 87–91.
66. H. Gebelein, Das statistische problem der correlation also variations und eigenwertproblem, *Zeitschrift Ange. Mathematik und Mechanik* **21** (1941) 364–379.
67. O. Kursun and O. V. Favorov, Feature selection and extraction using an unsupervised biologically-suggested approximation to gebelein's maximal correlation, *Int. J. Pattern Recognit. Artif. Intell.* **24**(3) (2010) 337–358.
68. S. Becker and G. E. Hinton, Self-organizing neural network that discovers surfaces in random-dot stereograms, *Nature* **355**(6356) (1992) 161–163.
69. O. V. Favorov and D. Ryder, SINBAD: A neocortical mechanism for discovering environmental variables and regularities hidden in sensory input, *Biol. Cybern.* **90**(3) (2004) 191–202.
70. T. Joachims, Making large scale SVM learning practical, *Advances in Kernel Methods — Support Vector Learning* (MIT Press Cambridge, MA, USA, 1999).
71. J. C. Bezdek, J. M. Keller, R. Krishnapuram and N. R. Pal, *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing* (Kluwer Academic Publisher, 1999).

72. S. L. Hung and H. Adeli, A parallel genetic/neural network learning algorithm for MIMD shared memory machines, *IEEE Trans. Neural Netw.* **5**(6) (1994) 900–909.
73. A. Alexandridis, Evolving RBF neural networks for adaptive soft-sensor design, *Int. J.Neural Syst.* **23**(6) (2013), 1350029, doi: 10.1142/S0129065713500299.
74. L. R. Zhou, J. P. Ou and G. R. Yan, Response surface method based on radial basis functions for modeling large-scale structures in model updating, *Comput. Aided Civil Infrastruct. Eng.* **28**(3) (2013) 210–226.
75. H. Adeli and A. Karim, Fuzzy-Wavelet RBFNN model for freeway incident detection, *J. Transp. Eng.* **126**(6) (2000) 464–471.
76. A. Karim and H. Adeli, Comparison of the Fuzzy wavelet RBFNN freeway incident detection model with the california algorithm, *J. Transp. Eng.* **128**(1) (2002) 21–30.
77. A. Karim and H. Adeli, Radial basis function neural network for work zone capacity and queue estimation, *J. Transp. Eng.* **129**(5) (2003) 494–503.
78. H. Adeli and A. Panakkat, A probabilistic neural network for earthquake magnitude prediction, *Neural Netw.* **22** (2009) 1018–1024.
79. H. Adeli and C. Yeh, Perceptron learning in engineering design, *Microcomputers Civil Eng.* **4**(4) (1989) 247–256.
80. M. Ahmadlou and H. Adeli, Enhanced probabilistic neural network with local decision circles: A robust classifier, *Integr. Comput. Aided Eng.* **17**(3) (2010) 197–210.
81. Z. Sankari and H. Adeli, Probabilistic neural networks for EEG-based diagnosis of Alzheimerś disease using conventional and wavelet coherence, *J. Neurosci. Meth.* **197**(1) (2011) 165–170.