

Dimensionality Reduction Based on Clonal Selection for Hyperspectral Imagery

Liangpei Zhang, *Member, IEEE*, Yanfei Zhong, Bo Huang, Jianya Gong, and Pingxiang Li, *Member, IEEE*

Abstract—A new stochastic search strategy inspired by the clonal selection theory in an artificial immune system is proposed for dimensionality reduction of hyperspectral remote-sensing imagery. The clonal selection theory is employed to describe the basic features of an immune response to an antigenic stimulus in order to meet the requirement of diversity in the antibody population. In our proposed strategy, dimensionality reduction is formulated as an optimization problem that searches an optimum with less number of features in a feature space. In line with this novel strategy, a feature subset search algorithm, clonal selection Feature-Selection (CSFS) algorithm, and a feature-weighting algorithm, Clonal-Selection Feature-Weighting (CSFW) algorithm, have been developed. In the CSFS, each solution is evolved in binary space, and the value of each bit is either 0 or 1, which indicates that the corresponding feature is either removed or selected, respectively. In CSFW, each antibody is directly represented by a string consisting of integer numbers and their corresponding weights. These algorithms are compared with the following four well-known algorithms: sequential forward selection, sequential forward floating selection, genetic-algorithm-based feature selection, and decision-boundary feature extraction using the hyperspectral remote-sensing imagery acquired by the Pushbroom Hyperspectral Imager and the Airborne Visible/Infrared Imaging Spectrometer, respectively. Experimental results demonstrate that CSFS and CSFW outperform other algorithms and hence provide effective new options for dimensionality reduction of hyperspectral remote-sensing imagery.

Index Terms—Artificial immune system (AIS), artificial intelligence, clonal selection, dimensionality reduction, feature selection, remote sensing.

I. INTRODUCTION

ADVANCES in hyperspectral remote sensing have provided an effective alternative for monitoring the Earth's surface. Hyperspectral sensors offer a dense sampling of the spectral range of the sensor, thus facilitating a better discrimination among similar ground-cover classes than traditional multispectral scanners with low spectral resolution [1]. However, the use of hyperspectral images is limited by the lack of reliable and effective data-analysis techniques for the voluminous amount of data. As hyperspectral sensors acquire images in very narrow

spectral channels, the resulting high-dimensional feature sets may contain redundant information. As a result of this, the number of features given as input to a classifier can be reduced without a significant loss of information [2].

Dimensionality reduction in a high-dimensional data space can decrease the computational cost and may also improve the accuracy during the classification process [3], [4]. For instance, when a supervised classifier is applied to classification problems in high-dimensional feature spaces, the Hughes phenomenon [5] can be observed; that is, when the number of input features exceeds a given limit for a fixed training sample size, the classification accuracy will decrease.

Given a set of measurements, dimensionality reduction can be achieved in essentially two ways: feature extraction and feature selection [6]. Feature extraction is to find the transformation from a higher dimension to a lower dimensional feature space with most of the desired information content preserved [7], [8]. This transformation may be a linear or nonlinear combination of the original variables and may be supervised or unsupervised. The commonly used feature-extraction techniques include principal component analysis [6], independent component analysis [9], discriminant analysis feature extraction [2], and decision-boundary feature extraction (DBFE) [8], [10]. Specifically, DBFE is based on the fact that the vector normal to the decision boundary of a classifier for a given pattern classification problem contains information useful for discriminating between classes. The DBFE was extended to multiclass problems by combining the decision-boundary feature matrix of each pair of classes [8], [10].

In contrast to the feature-extraction techniques, feature selection is used to identify the variables that do not contribute to the classification process. In a discrimination problem, those variables that do not contribute to class separability would be neglected. Thus, the task of feature selection is to select a subset from a larger number of features or variables used in classification while maintaining an acceptable classification accuracy [6], [11].

In general, feature selection requires a search strategy and criterion functions [2], [6], [12]. The search algorithm generates and compares possible feature-selection solutions by calculating their criterion function values as a measure of the effectiveness of each considered feature subset. The feature subset with the best criterion function value is given as the output of the feature-selection algorithm. This paper focuses on the search algorithm. More details about criterion functions can be found in [2], [6], and [12].

Different feature-selection techniques including optimal and suboptimal search algorithms have been proposed. Optimal

Manuscript received November 28, 2006; revised April 28, 2007. This work was supported in part by the 973 Program of China under Grant 2006CB701302, by the 863 High Technology Program of China under Grant 2007AA12Z148, and by the National Natural Science Foundation of China under Grants 40471088 and 40771139.

L. Zhang, Y. Zhong, J. Gong, and P. Li are with the State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing, Wuhan University, Wuhan 430079, China (e-mail: zlp62@public.wh.hb.cn; zhongyanfei@lmars.whu.edu.cn).

B. Huang is with the Department of Geography and Resource Management, The Chinese University of Hong Kong, Shatin, Hong Kong.

Digital Object Identifier 10.1109/TGRS.2007.905311

search algorithms, such as an exhaustive search and branch and bound method [13], identify the selected subset that consists of a predefined number of features and evaluates the best subset according to the criterion function. However, when the number of features is larger than a few tens, for example, in hyperspectral space, the optimal search algorithms are not fit for use because of their heavy computational costs. In such a case, suboptimal algorithms will be considered for searching an appropriate feature subset.

The simplest suboptimal search strategy employs the sequential forward-selection (SFS) and sequential backward-selection (SBS) techniques [6]. SFS and SBS achieve the best feature subset with the prefixed number of features by adding to or removing from the current feature subset one feature at a time. However, both algorithms do not allow the features to be reselected once they have been selected. The plus- l -minus- r algorithm [14] applies a more complex sequential search strategy to overcome this problem. This technique is, however, limited by its difficulty in selecting the values of l and r in order to obtain the best feature subset. Built on SFS and SBS, the sequential forward floating selection (SFFS) and the sequential backward floating selection (SBFS) methods [15] have evolved into the two most widely used sequential search methods. These enhance the standard SFS and SBS by dynamically changing the number of features included in SFFS or removed in SBFS at each step and allowing the reselection of the features included or removed at the previous step.

Besides these two methods, many new search algorithms have been devised for dimensionality reduction, in pace with the rapid development in soft computing. Among them, genetic algorithm (GA) is a representative that has been applied to feature selection [16]–[19]. In this algorithm, a feature subset is represented as a “chromosome” in the form of a binary string whose length is equal to the number of features. A “0” bit in the chromosome indicates a discarded feature, while a “1” bit suggests a selected feature. During each iteration of the algorithm, a number of possible solutions are generated by means of genetic operators, such as crossover and mutation, guided by a fitness measure or criterion function. The algorithm seeks to evolve an optimal solution to the feature-selection problem. Another stochastic search algorithm, namely, simulated annealing, was also attempted for feature selection [20].

This paper will add to the literature by proposing a new stochastic search strategy for hyperspectral feature selection. This strategy is based on the clonal selection algorithm (CSA) [21], [22] in artificial immune systems (AISs). AISs, which are inspired by the immune systems, use the immunological properties to support a wide range of applications [23]–[25]. CSA, derived from the clonal selection theory [26], [27], is an effective method of AISs and successfully applies to pattern recognition, multimodal optimization, and classification [21], [22], [24], [28]. Nonetheless, applications of CSA to hyperspectral remote sensing have rarely been reported, owing to the complexity of hyperspectral imaging data. In this paper, two new feature-selection algorithms, namely, a subset search algorithm and a feature-weighting algorithm, have been developed based on the clonal selection theory for feature selection in hyperspectral space. These algorithms are able to find the

optimal solution in feature space using immune operators, such as clone, selection, mutation, and replacement. The algorithms have been tested and compared with the traditional algorithms using the hyperspectral remotely sensed images acquired by the Pushbroom Hyperspectral Imager (PHI) and the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS). Experimental results evince that the proposed algorithms outperform the traditional methods and thus provide an effective option to hyperspectral dimensionality reduction.

The rest of this paper is organized as follows. Section II describes the clonal selection theory that underlies the CSA. Section III reviews the shape-space model in immune systems and the basic CSA. Sections IV and V present the proposed new feature-selection algorithm and the feature-weighting algorithm, respectively. In Section VI, the PHI and AVIRIS hyperspectral data used for experiments are described and the experimental results are provided. Section VII discusses the main features of the proposed methods, and finally, Section VIII concludes this paper.

II. CLONAL SELECTION THEORY

The human immune system, a complex system of cells, molecules, and organs, symbolizes an identification mechanism capable of perceiving and combating dysfunction from our own cells and the action of exogenous infectious microorganisms. This immune system protects the body from infectious agents such as viruses, bacteria, fungi, and other parasites. Any molecule that can be recognized by the adaptive immune system is known as an antigen. The basic component of the immune system is the lymphocytes or the white blood cells. Lymphocytes exist in two forms, B cells and T cells. These two types of cells are rather similar, but they differ in relation to how they recognize antigens and to their functional roles. B cells are capable of recognizing antigens free in solution, while T cells require antigens to be presented by other accessory cells. They have distinct chemical structures and produce many Y-shaped antibodies from their surfaces to kill the antigens. Antibodies are molecules attached primarily to the surface of B cells with an aim to recognize and cope with antigens [29].

In order to clarify how an immune response is mounted when a nonself antigenic pattern is recognized by a B cell, the clonal selection theory has been developed [26], [27]. The main features of the clonal selection theory are concerned with the following: 1) proliferation and differentiation on simulation of cells with antigens; 2) generation of new random genetic changes, expressed subsequently as diverse antibody patterns, by a form of accelerated somatic mutation; and 3) estimation of newly differentiated lymphocytes carrying low-affinity antigen receptors. These will be utilized in this paper.

The principle can be detailed as follows. When a B-cell receptor recognizes a nonself antigen with a certain affinity, it is selected to proliferate and produce antibodies in high volumes. The antibodies are soluble forms of the B-cell receptors that are released from the B-cell surface to cope with the invading nonself antigens. Antibodies bind antigens leading to their eventual elimination by other immune cells. Proliferation in the case of immune cells is asexual, and it is a mitotic process in which the

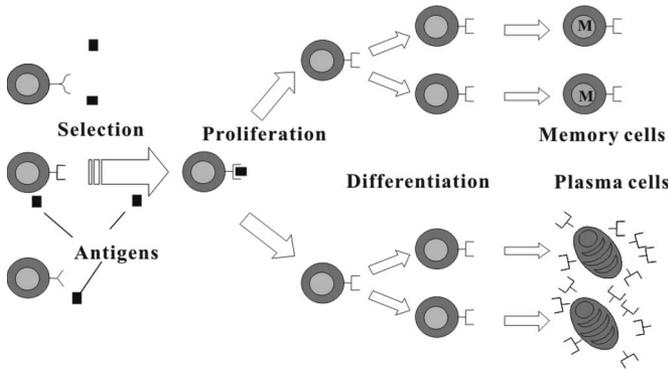


Fig. 1. Clonal selection principle.

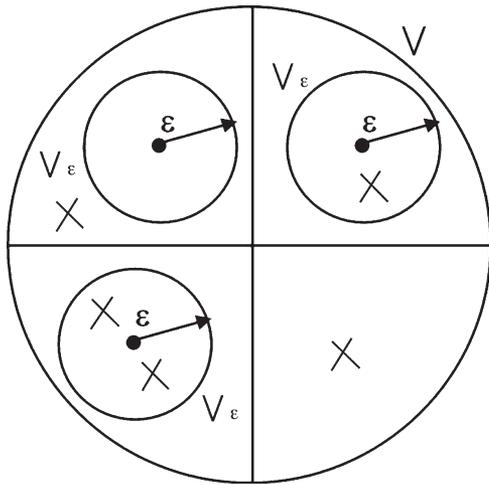


Fig. 2. Diagrammatic representation of shape space.

cells divide themselves. During reproduction, the B-cell clones undergo a hypermutation process where the antigen stimulates the B cell to proliferate and mature into terminal antibody-secreting cells that are named plasma cells. The process of cell division generates a clone. In addition to proliferation and differentiation into plasma cells, the activated B cells with high antigenic affinities are selected to become memory cells with long life spans. These memory cells circulate through the blood, lymph, and tissues. When exposed to a second antigenic stimulus, memory cells commence to differentiate into plasma cells capable of producing high-affinity antibodies, which are preselected for the specific antigen that had stimulated the primary response [22]. Fig. 1 shows the clonal selection, expansion, and affinity maturation processes.

III. CLONAL SELECTION ALGORITHM

In immune systems, to quantitatively describe the interactions between immune cell molecules and antigens, Perelson and Oster [30] introduced the concept of shape space. Based on this, it is then generally agreed that a complete repertoire is attainable within the known parameters of immune recognition [24], [31], [32].

The notion of shape space is that the degree of binding between a receptor and a molecule that it binds, a ligand, generally involves short-range noncovalent interactions based on electro-

static charge, hydrogen binding, and van der Waals interactions. Repertoire completeness is the ability of the immune systems to recognize all antigens and can be represented by the shape-space model [33]. As can be seen from Fig. 2, within the shape space, the immune systems of a given person can be represented by a 2-D circle of volume V . It is assumed that each paratope (\bullet) specifically interacts with all epitopes (\times) that are within a small surrounding region, characterized by the parameter ϵ and called a recognition region of volume V_ϵ . Because each antibody can recognize all epitopes within a recognition region and an antigen might present some different kinds of epitopes, a finite number of antibodies can recognize an almost infinite number of points into the volume V_ϵ [34]. Hence, the repertoire of antibodies can be deemed to be complete if they cover the entire volume of the shape space.

Based on the clonal selection theory and the shape-space model of the immune system, De Castro and Von Zuben [21], [22] developed the CSA. It has been applied to support pattern recognition and solve multimodal optimization problems. The algorithm can be described as follows.

- 1) Randomly initialize a population of individuals M .
- 2) For each input pattern P , present it to the population M and determine its affinity with each element of M .
- 3) Select n of the best highest affinity elements of M and clone these individuals proportionally to their affinity with the antigen. The higher the affinity, the higher the number of copies and vice versa.
- 4) Mutate all these copies with a rate proportional to their affinity with the following input pattern: the higher the affinity, the smaller the mutation rate.
- 5) Add these mutated individuals to the population M and reselect the m of these matured individuals to be kept as memory cells of the systems.
- 6) Repeat steps 2) to 5) until a certain criterion is met.

Similar to CSA, GA is also a heuristic algorithm. However, their underlying mechanisms and methods of evolutionary search significantly differ in terms of inspiration, vocabulary, and fundamentals. While GA uses a vocabulary borrowed from natural genetics and is inspired by the Darwinian evolution theory, CSA makes use of the shape-space formalism, along with immunological terminology, to describe antigen-antibody interactions and cellular evolution in immune systems. GA performs a search through genetic operators including reproduction, crossover, and mutation, while CSA performs its search through the mechanisms of somatic mutation and receptor editing, balancing the exploitation of the best solutions with the exploration of the search space. The CSA maintains a diverse set of local optimal solutions, while the GA tends to polarize the whole population of individuals toward the best one. This occurs mainly owing to the selection and reproduction schemes adopted by the CSA [described in step 3)]. Essentially, their coding schemes and evaluation functions are not different, but their evolutionary search differs from the viewpoint of inspiration, vocabulary, and fundamentals [34]. In addition, CSA inherits the memory property of human immune systems to build a memory-cell population and can recognize the same or similar antigen quickly at different times [24], [35].

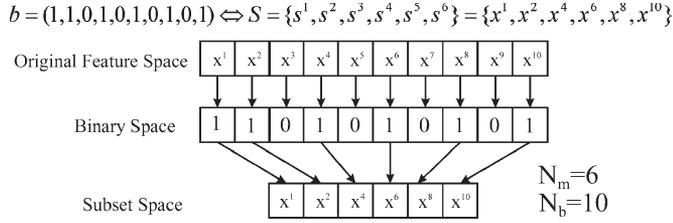


Fig. 3. Ten-dimensional feature space reduced to a 6-D subset space.

IV. PROPOSED CLONAL SELECTION FEATURE SELECTION ALGORITHM

A feature-selection algorithm based on the clonal selection theory, i.e., CSFS, was developed in this paper for dimensionality reduction of hyperspectral imagery. A hyperspectral remote-sensing data set $X = \{x^1, x^2, \dots, x^{N_b}\}^T$ through N_b bands is observed and mapped to a finite rectangular lattice $W = \{(i, j) : 1 \leq i \leq N_{\text{row}}, 1 \leq j \leq N_{\text{col}}\}$, where N_{row} and N_{col} represent the row number and column number, respectively. T denotes the transpose of a matrix. The set $x^t = \{x_{11}^t, \dots, x_{N_{\text{row}}N_{\text{col}}}^t : t = 1, \dots, N_b\}$ denotes the data taken at the t th wavelength, where $x_{ij}^t \in (0, \dots, G - 1)$, and G is the number of observable gray levels. A classified image is denoted as $\omega = \{\omega_{ij} : 1 \leq i \leq N_{\text{row}}, 1 \leq j \leq N_{\text{col}}\}$; each pixel of which will be assigned to one of the c classes. That is, $\omega_{ij} \in (1, 2, \dots, c)$, where c is the number of classes, is assumed to be known.

The objective of feature selection is to reduce the number of features utilized to characterize patterns by selection. This is achieved through optimization in terms of a criterion function F (e.g., maximization of a separability index or minimization of an error bound), a good subset S of N_m features, with $N_m < N_b$, without significantly degrading the performance of the resulting classifier

$$S = \{s^1, s^2, \dots, s^{N_m}\}^T, \quad S \subset X.$$

The criterion function F is computed by using a preclassified reference set of patterns (e.g., training set). The value of F depends on the features included in the subset S , $F = F(S)$.

To apply the CSA to hyperspectral feature selection, the entire set of features is represented by a discrete binary space. In this search space, each point represents an individual band. The value “0” in the i th position indicates that the i th feature is not included in the corresponding feature subset; the value “1” in the j th position indicates that the j th feature is included in the corresponding feature subset. For example, in a simple case with $N_m = 6$ and $N_b = 10$ features, the binary vector $b = (1, 1, 0, 1, 0, 1, 0, 1, 0, 1)$ indicates that the feature subset has the first, second, fourth, sixth, eighth, and tenth features. The process is shown in Fig. 3.

The criterion function F can be viewed as a scalar function defined in the discrete binary space. Without loss of generality, there is a case in which the criterion function has to be maximized. Thus, the dimensionality reduction problem of selecting N_m out of N_b features can be suitably formulated as an optimization problem to find the global maximum of the criterion function.

To describe CSFS, the following notations are used.

- 1) Let AB denote the set of antibodies and ab represent a single antibody where $AB = \{ab_1, ab_2, \dots, ab_N\}$ and N is the number of the antibody population. Each antibody $ab_i = (ab_i^1, ab_i^2, \dots, ab_i^{N_b})$ ($i = 1, 2, \dots, N$) represents a solution to the feature-selection problem in a binary space; N_b is the number of bands.
- 2) Let F denote the criterion function of an antibody, $F = F(S)$. Because each ab_i can describe a binary space or subset space, $F = F(ab_i)$.
- 3) Let mc denote the memory cell. mc indicates the best antibody with the highest criterion function value in each iteration, and mc is a candidate solution.

The CSFS algorithm consists of the following steps.

A. Initialization

The first antibody population AB including N antibodies is generated with the value of each bit in $ab_i \in AB$ assigned with 1 or 0 according to the number of selected subset features N_m . The value 0 in the i th position indicates that the i th feature is not included in the corresponding feature set; the value 1 in the j th position indicates that the j th feature is included in the corresponding feature set

$$ab_i^t = \begin{cases} 1, & \text{if } t = k \\ 0, & \text{otherwise} \end{cases} \quad i = 1, 2, \dots, N; \quad t = 1, 2, \dots, N_b \quad (1)$$

$$k = \text{Irandom}(1, N_b) \quad (2)$$

$$\sum_{t=1}^{N_b} ab_i^t = N_m \quad (3)$$

where N is the number of the initial antibody population, N_b is the number of features or bands, and function $\text{Irandom}(1, N_b)$ returns a random integer value within the range $[1, N_b]$ using a uniform distribution. Equation (3) indicates that the sum of the selected features is equal to N_m , a user-defined feature subset size.

B. Cycle of the Generations

After initialization, the simulation of the clonal selection process begins. One generation after another is created, and each must prove its affinity to the criterion function. In each iteration, a number of possible solutions are generated by means of applying the immune operators such as clone, mutation, and selection in a stochastic process guided by an affinity measure. The algorithm seeks to evolve an optimal solution to the problem.

1) *Calculation of Affinity*: According to the initial antibody population, the affinity of all N ab 's in the antibody population AB is calculated using the criterion function $F = F(ab_i)$. As a criterion function, the proposed algorithm uses the average Jeffries–Matusita (JM) distance [12], [36], [37], which is a common class separability index utilized by the remote-sensing community for feature selection in multiclass problems and is a saturating transform of the Bhattacharyya distance (BD) [38]. Assuming that there exist c classes with Gaussian distributions,

in order to simplify the computation of the BD, the average JM distance of a feature subset is calculated by using the following equation:

$$\text{JM} = \sum_{m=1}^c \sum_{n=1}^c P_m P_n \text{JM}_{mn} \quad (4)$$

$$\text{JM}_{mn} = \sqrt{2(1 - e^{-\text{BD}_{mn}})} \quad (5)$$

$$\text{BD}_{mn} = \frac{1}{8} (M_m - M_n)^T \left(\frac{\Sigma_m + \Sigma_n}{2} \right)^{-1} \times (M_m - M_n) + \frac{1}{2} \ln \left(\frac{|\frac{\Sigma_m + \Sigma_n}{2}|}{\sqrt{|\Sigma_m| |\Sigma_n|}} \right) \quad (6)$$

where c is the number of classes, m and n are the two classes being compared, P_m is an *a priori* probability of the m th class, Σ_m and M_m are the covariance matrix and mean vector of the m th class, respectively, T is the transposition function, and $|\Sigma_m|$ is the determinant of Σ_m .

The JM_{mn} distance between classes m and n is an affinity/distance measure of separability. The smaller the JM_{mn} distance, the more difficult it is to separate the classes and vice versa.

Assuming that $\text{JM}_{mm} = 0$, the average JM distance can be written as (7), and the proposed algorithm uses this equation as the affinity function

$$F(\text{ab}_i) = \text{JM} = \frac{2}{c(c-1)} \sum_{m=1}^{c-1} \sum_{n=m+1}^c P_m P_n \text{JM}_{mn}. \quad (7)$$

2) *Selection*: From AB, the “ n ” highest affinity antibodies are selected to compose a new set $\text{AB}_{\{n\}}$ of high-affinity antibodies, and the highest affinity memory cell (mc) is found.

3) *Clone*: After receiving antibody individuals closer to the solution, the next generation should mainly be derived from the better fitting individuals. Thus, the n selected ab 's are cloned based on their antigenic affinities, generating the clone set C . The total number of clone-generated N_c is defined as follows:

$$N_c = \sum_{i=1}^n \text{round}(\beta \cdot N) \quad (8)$$

where β is a multiplication factor, N is the total number of antibodies, and $\text{round}(\cdot)$ is the operator that rounds its argument.

This step draws the evolutionary process closer to the goal. It raises the average affinity value and gives the following steps a good chance to further move toward the solution.

4) *Mutation*: Provide each ab in the clone set C with the opportunity to produce mutated offspring C^* . The higher the affinity, the smaller the mutation rate. To adaptively determine the mutation rate according to the affinity of each ab , the process is as follows.

First, for each $\text{ab}_i \in \text{AB}$, normalize its affinity $F(\text{ab}_i)$ into the range $[0, 1]$

$$F'(\text{ab}_i) = \frac{F(\text{ab}_i) - \min(F(\text{ab}_i))}{\max(F(\text{ab}_i)) - \min(F(\text{ab}_i))}, \quad i = 1, 2, \dots, N_c. \quad (9)$$

```

mutate(B)
{
  for each(B.v)
  do
    a_i = min v_i
    b_i = max v_i
    rd_mr = random(0,1)
    rd_to = random(-1,1)
    if(rd_mr < p_m)
      if(rd_to >= 0)
        B.v_i = B.v_i + Δ(Ite, b_i - B.v_i)
      else
        B.v_i = B.v_i - Δ(Ite, B.v_i - a_i)
  done
  return B
}

```

Fig. 4. Real-value mutation (p_m represents mutation rate).

Then, let each ab_i have the chance to mutate; the mutation rate is adaptively calculated as

$$p_m = \exp(-2^* F'(\text{ab}_i)) \quad (10)$$

where p_m is the mutation rate of each ab , two is the empirical value to control the decay, and $F'(\text{ab}_i)$ is the affinity according to (9).

In (10), the range of the mutation rate is $[0, 1]$.

Finally, the cloned antibodies are mutated with probability p_m .

The mutation process is as follows.

- 1) An integer string B consisting of integer numbers to represent the corresponding selected features is obtained by decoding the antibody ab_i . For example, as for $N_m = 5$ and $N_b = 10$, a binary string 1011001010 corresponds to the integer string $B = \{1, 3, 4, 7, 9\}$.
- 2) For the integer string B , the mutation procedure and the function mutation (B) with mutation rate p_m are defined in Fig. 4. The maximum and minimum of each element in the integer string B are N_b and one, respectively. In the mutation process, the value of each element is not equal to any other elements existing in the string B . The function random (minimum, maximum) generates a random real value using a uniform distribution in the range from the minimum to the maximum. Function $\Delta(\text{Ite}, u)$ is defined as

$$\Delta(\text{Ite}, u) = u \left(1 - r^{\left(1 - \frac{m}{\text{Ite}}\right)^\lambda} \right) \quad (11)$$

where m is the iteration number, Ite is the maximal iteration number, r is a random value within the range $[0, 1]$, and λ is a parameter to decide the nonconforming degree.

For example, the integer string B in step 1) mutates to the integer string $B' = \{2, 3, 5, 9, 10\}$.

- 3) For the mutated integer string B' , the corresponding binary string is encoded as 0110100011.

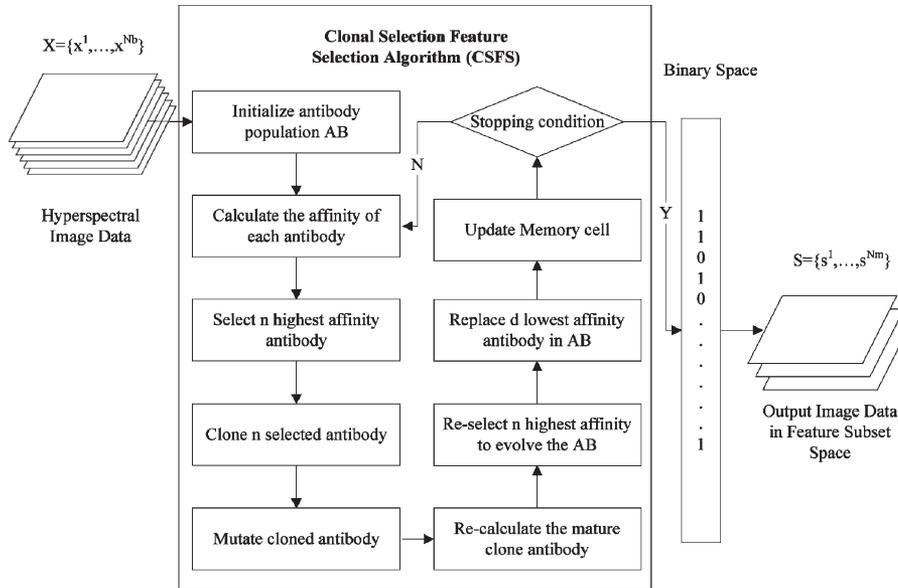


Fig. 5. Process for the CSFS.

This step is crucial in the proposed algorithm. It generates random changes of single features of the individual solutions. The value of these changes can be found at the criterion function calculation within the next-generation cycle. This helps avoid local maximums and produces new properties of mutated antibodies that can remain if they are successful.

To avoid chaotic development and to maintain the best ab's for each clone during evolution, one original ab for each clone without mutation during the maturation process is kept; else, it would destroy the positive development of the previous step and disable any major development toward the solution.

5) *Recalculation of Affinity*: Calculate the affinity $F^*(ab_i)$ of the matured clones C^* .

6) *Reselection*: From the mature clone set C^* , reselect the n ab's with the highest affinity to replace the n ab's with the lowest affinity in AB. Select the highest affinity ab in C^* to be a candidate memory cell $mc_{candidate}$. If the affinity of $mc_{candidate}$ is higher than the memory cell mc , then $mc_{candidate}$ will replace mc and become a new memory cell.

7) *Displace*: In order to replace the d lowest affinity ab's from AB, d new antibodies are produced by a random process. This step may increase the diversity of the antibody population.

C. Stopping Condition

When the number of iterations reaches the user-defined number or the change of memory cell between two consecutive iterations is less than a change threshold, terminate the execution of the algorithm. Otherwise, return to step B until the stop criteria are satisfied.

Finally, the proposed algorithm outputs the value of the memory cell and obtains the subset space through transformation from the binary space.

The flowchart for CSFS is shown in Fig. 5.

V. PROPOSED CLONAL SELECTION FEATURE-WEIGHTING ALGORITHM

In the earlier CSFS algorithm for finding a feature subset, the antibody population is evolved in the binary space, and the value of each bit is 0 or 1. In addition to this, this algorithm is extended to allow linear feature extraction and hence evolves into a feature-weighting algorithm, namely, CSFW. In CSFW, each antibody is directly represented by a string consisting of integer numbers and their corresponding weights, allowing independent linear scaling of each feature. The following is a set of feature vectors with N_b features:

$$X = \{x^1, x^2, \dots, x^{N_b}\}^T.$$

For the defined number of the subset features N_m , the CSFW produces a transformed set of vectors

$$S' = \{w^1 y^1, w^2 y^2, \dots, w^{N_m} y^{N_m}\}^T \quad y^j \in X, j=1, 2, \dots, N_m$$

where w_i is a weight associated with feature i . Each feature value is scaled by the associated weight prior to training, testing, and classification. For example, with $N_m = 5$ and $N_b = 10$, the first, third, fifth, sixth, and tenth features are selected, and the corresponding weights are w^1, w^2, w^3, w^4 , and w^5 , respectively. This process is shown in Fig. 6.

CSFS is modified as follows to support feature weighting.

- 1) In the initialization step, the length of each antibody string is equal to 2^*N_m . The first N_m string represents the selected feature, and the latter string describes the corresponding weight of each selected feature. The value of each bit in $ab_i \in AB$ is generated as follows:

$$ab_i^t = \begin{cases} I\text{randon}(1, N_b), & t = 1, 2, \dots, N_m \\ \text{random}(0, 1), & t = N_m + 1, N_m + 2, \dots, 2^*N_m \end{cases} \quad i = 1, 2, \dots, N$$

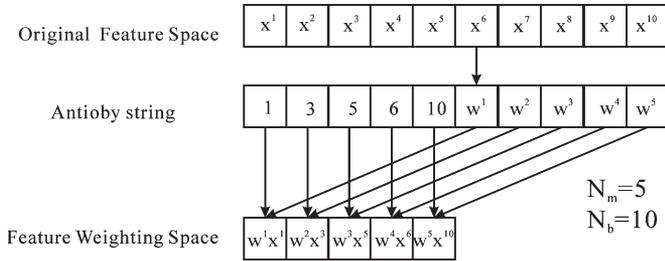


Fig. 6. Ten-dimensional feature space transformed into feature-weighting space.

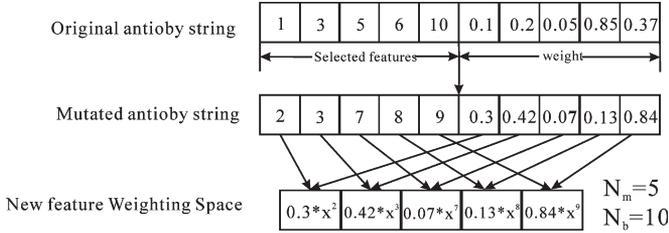


Fig. 7. Mutation sample in CSFW.

where function $Irandom(1, N_b)$ returns a random integer value within the range $[1, N_b]$, and the function $random(0, 1)$ generates a random real value as a feature weight using a uniform distribution in the range $[0, 1]$.

- In the mutation step, the real-value mutation is applied. The mutation procedure is also defined in Fig. 4 as CSFS. Specifically, the maximum and minimum of the weights are zero and one, respectively. A mutation process is shown in Fig. 7. In this figure, the original antibody string represents that the first, third, fifth, sixth, and tenth features are selected, and the corresponding weights are 0.1, 0.2, 0.05, 0.85, and 0.37, respectively. After mutation, the new feature-weighting space is obtained. In the new space, the second, third, seventh, eighth, and ninth features are selected, and the corresponding weights are changed to 0.3, 0.42, 0.07, 0.13, and 0.84, respectively.

VI. EXPERIMENTS AND ANALYSIS

Experiments have been conducted to test the performance of the proposed CSFS and CSFW algorithms by using different hyperspectral remote-sensing imagery.

Consistent comparisons among CSFS, CSFW, and the following traditional dimensionality reduction algorithms: SFS, SFFS, GA (as in [16]), and DBFE, were performed. The optimal feature-selection algorithms, such as exhaustive search and branch and bound method, were not used as they are not considered suitable in terms of computational efficiency for hyperspectral imagery. The estimation of feature selection/extraction quality for these algorithms was done by means of the classification accuracy of the hyperspectral imagery acquired by the PHI and the AVIRIS.

A. Experiment 1: Xiaqiao PHI

The data set used in this experiment was acquired from the Xiaqiao test site, a mixed agricultural area in China, using



Fig. 8. Xiaqiao PHI image RGB (70, 40, 10).

TABLE I
LAND-COVER CLASSES AND ASSOCIATED NUMBERS OF PIXELS USED IN EXPERIMENT 1

Class Name	Number of labeling samples
C1. Corn1	713
C2. Corn2	217
C3. Corn3	322
C4. Vegetable-sweet potato	464
C5. Vegetable-cabbage	253
C6. Soil	1368
C7. Float Grass	220
C8. Road	662
C9. Water	659
Overall	4878

the PHI. PHI image (340×390 pixels) of 80 bands were tested, and their spectral ranges were from 0.417 to $0.854 \mu\text{m}$. Fig. 8 shows the experimental PHI image cube. The ground-truth spectral data were collected by field spectrometer SE590. Nine representative classes, i.e., corn1, corn2, corn3, vegetable-sweet potato, vegetable-cabbage, soil, float grass, road, and water, were considered. The list of classes and the number of labeled samples for each class are given in Table I, and Fig. 9 shows the reflectance curves of the above nine land-cover classes. The field map is shown in Fig. 10 based on the ground-truth data.

The primary running parameters that should be provided by users for feature selection were the number of iterations, the antibody population size N , the number of highest affinity ab , the n [see also step 3)], the clone multiplication factor β [see also (8)], and the number of displace antibody d [see also step 7)]. To facilitate the comparison of the proposed algorithm with the other traditional algorithms, n is set to N and d to zero. The affinity function is determined by the JM distance in (7). The values of the parameters were set as follows: Population size = 50, Maximum iterations = 100, $n = 50$, $\beta = 0.02$, $d = 0$.

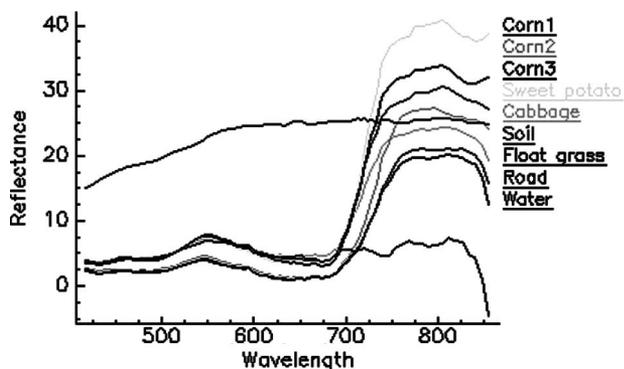


Fig. 9. Reflectance of nine land-cover classes.

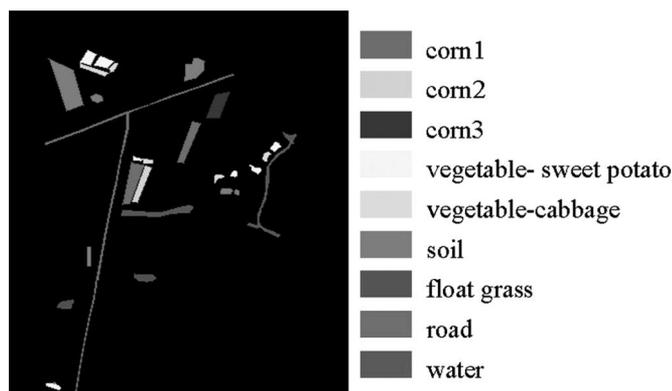


Fig. 10. Field map of the Xiaqiao site.

In order to test these algorithms, approximately half of the available samples were used as the training data set, whereas another half of the available samples served as the test data set.

An important characteristic of the algorithms is their execution time. In our case, it is in the form of processor ticks (1 ms) spent in the user space. Fig. 11 shows the execution times with different feature subset sizes provided by the five algorithms. For every number of selected features from 2 to 79, SFS is the fastest, GA and CSFS cost similar time, and CSFW is slower than CSFS because it adds the process of selecting weights. While the number of selected features is over 67, SFFS is the slowest. Otherwise, CSFW is the slowest.

Fig. 12 shows the values of the criterion function computed on the subsets in the training process provided by CSFS, CSFW, SFS, SFFS, and GA to different numbers of selected features from 2 to 79. All algorithms were described by the relationship curves between the predefined number of features in the subset and the corresponding JM distance. To better present the experiment results, the JM distance of SFS was used as a reference, i.e., the values of the criterion functions provided by SFFS, GA, CSFS, and CSFW are divided by the corresponding values obtained by SFS. For example, if the CSFS and SFS provided the same JM distance values, the value on Fig. 12 is equal to one. DBFE does not have the corresponding JM distance value as it is based on the decision-boundary feature matrix.

For a more detailed comparison among the following different algorithms: SFS, SFFS, GA, CSFS, and CSFW, all the labeled training and test samples were classified using the dimensionality reduction algorithms with the increase of

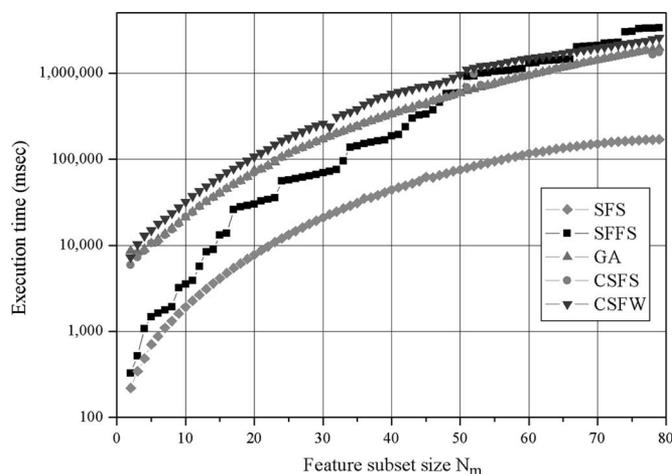


Fig. 11. Execution times required by different algorithms in Experiment 1.

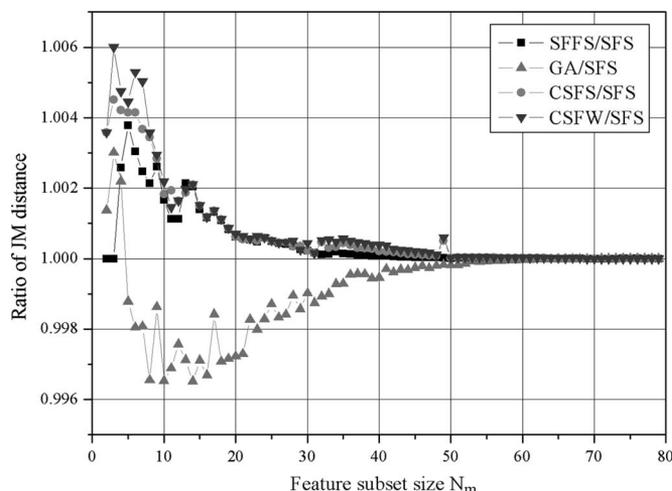


Fig. 12. Values of the criterion function for different feature subset sizes provided by different algorithms in Experiment 1.

selected feature numbers. The maximum-likelihood (ML) classifier was employed in this case. Figs. 13 and 14 show the overall accuracy for training data and test data, respectively. For convenience of description, SFS-ML denotes the ML classifier with the feature subsets provided by the SFS algorithm. Other notations can be inferred by analogy.

As can be found from Figs. 12–14, the SFFS and the proposed CSFS and CSFW algorithms perform better than SFS for all the selected subset features. Comparing the results of CSFS and CSFW with those of SFFS, the two proposed algorithms make some improvements over SFFS. In particular, when the number of selected features is below 50, the improvement is greater, and CSFW for three features achieves the most significant improvement. The highest overall accuracies achieved by CSFS-ML and CSFW-ML using the test data are 93.07% and 93.75%, respectively. A comparison between the two proposed algorithms shows that CSFW usually provides better results than those provided by CSFS; however, the differences can be neglected when the number of selected features is larger than 50. Specifically, the Hughes phenomenon can be observed as follows: when the number of input features exceeds around 50, the classification accuracy decreases.

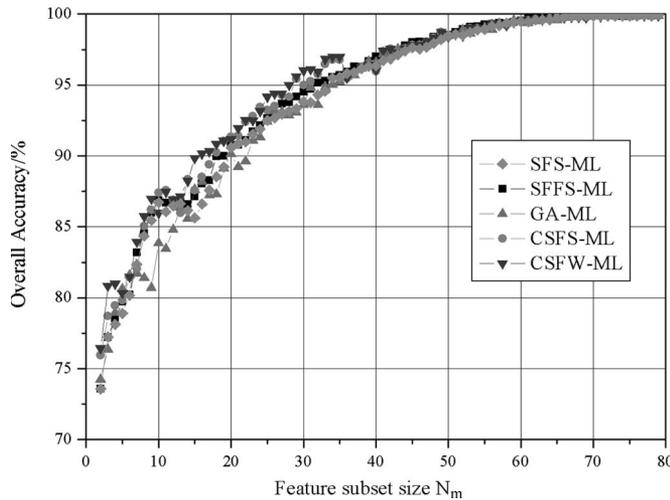


Fig. 13. Classification accuracy achieved by the following five algorithms: SFS, SFFS, GA, CSFS, and CSFW, using the training data in Experiment 1.

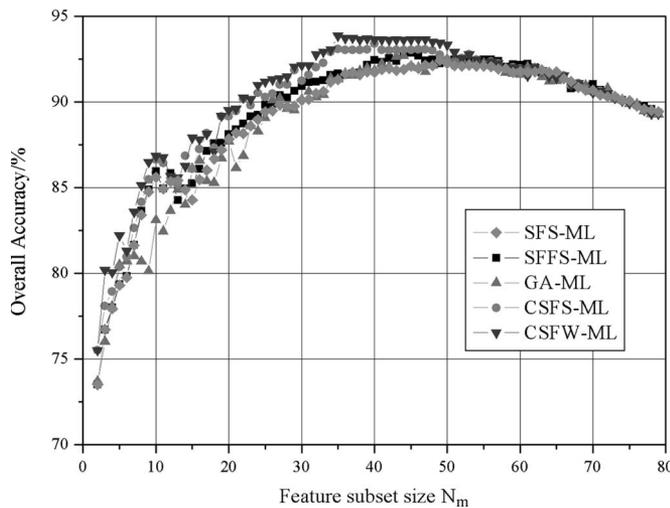


Fig. 14. Classification accuracy achieved by the following five algorithms: SFS, SFFS, GA, CSFS, and CSFW, using the test data in Experiment 1.

Using GA with the population size of 50 for 100 generations, several different settings of the crossover rate (ranging from 0.6 to 0.9) and the mutation rate (ranging from 0.01 to 0.2) were attempted to select the feasible rates. When the crossover and the mutation rates were approximately equal to 0.85 and 0.09, a better result can be obtained. However, the results provided by GA do not outperform those provided by SFS when the number of selected features is larger than two. One reason can be that the simple GA was used in the experiment; it seemed to present a tendency toward premature convergence in the experiment, with most runs reaching their peak by around the tenth generation and failing to make further improvements after that. However, if more values of GA's parameters are tested (e.g., population size) or a more extensive and time-consuming search is used in the solution space, it may lead to better results [18], [19].

CSFS, CSFW, and GA are heuristic feature-selection/extraction algorithms, and they obtain the optimal solution through maximization of the criterion function. However, there

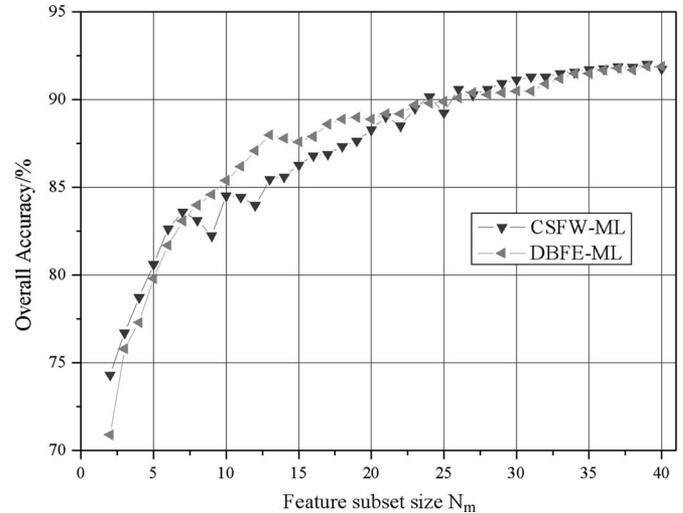


Fig. 15. Test classification accuracy achieved by CSFW and DBFE using the image with 40 bands in Experiment 1.

is an essential difference in the evolutionary process between GA and CSFS or CSFW. Unlike GA, which is simulating a genetic evolution process, CSFS and CSFW, inspired by the immune systems and clonal selection theory, are self-adaptive methods that can adjust themselves to the data without any explicit specification of functional or distributional form for the underlying model. In these two algorithms, the additional clone process may increase the average affinity value, extend the search space, and provide the mutation steps a good chance to move closer to the optimal solution. Therefore, CSFS and CSFW may have the better ability to find the better results. Inspiringly, between the two proposed algorithms, CSFW, as a feature-weighting algorithm, exhibits a better potential than the feature-selection CSFS algorithm when they are applied to the same case in dimensionality reduction.

DBFE is an effective feature-extraction algorithm. To make a better use of such an algorithm, the number of bands was reduced by the simple prereduction method considering one band every two. That is, the number of the selected image's bands used by DBFE is equal to 40 in this experiment. Fig. 15 shows the test accuracy provided by DBFE-ML and CSFW-ML to different numbers of selected features from 2 to 40 in the image with 40 bands.

As shown in Fig. 15, DBFE-ML obtains a better result than CSFW-ML when the number of selected features ranges from 7 to 23. The best accuracies achieved by CSFW-ML and DBFE-ML using the test data are 92.02% and 91.9%, respectively. In addition, the result of DBFE may improve if a more effective prereduction algorithm, such as uniform band combination, is employed. In addition, when the number of selected features is increased from 39 to 40, the test accuracy provided by CSFW-ML decreases from 92.02% to 91.75%.

In order to assess the sensitivity of CSFS and CSFW to the land-cover classes, several experiments were performed with different numbers of classes using the feature subset size of 20. In Figs. 16 and 17, the numbers of classes were two, three, four, five, six, seven, eight, and nine, respectively, i.e., the first two classes, three classes, four classes, five classes, six classes,

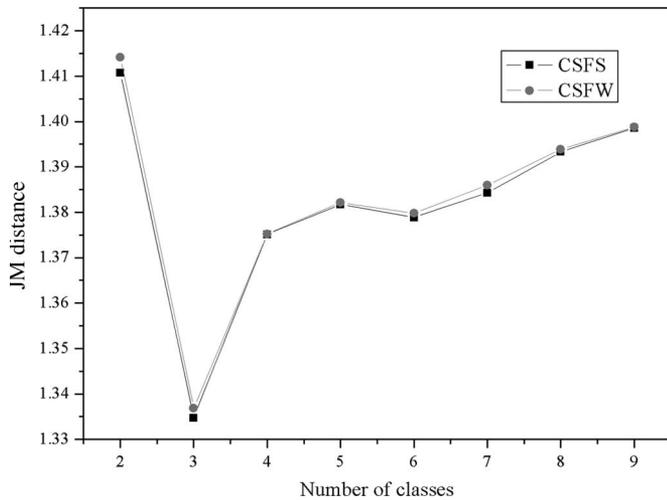


Fig. 16. JM distance versus the number of classes for the two proposed algorithms using 20 features.

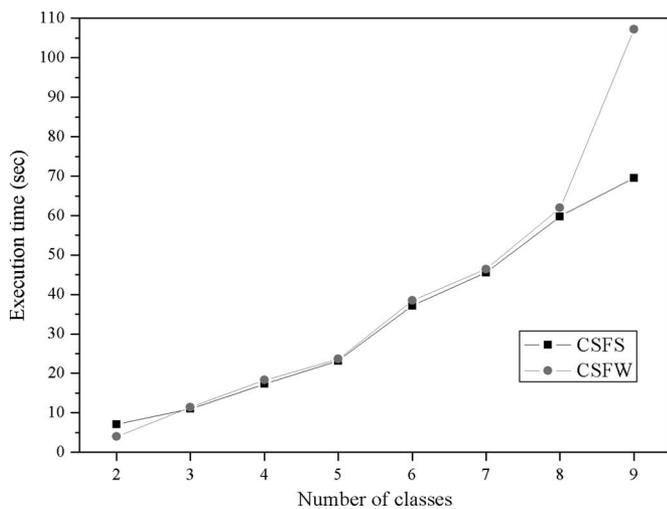


Fig. 17. Execution time (seconds) versus the number of classes for the two proposed algorithms using 20 features.

seven classes, eight classes, and nine classes of the labeled data set, respectively.

As can be observed from Fig. 16, CSFW makes improvements over CSFS for the different number of classes. CSFW can obtain the maximum of the JM distance, 1.41421 [see also (7)] for two classes, whereas the JM distance provided by CSFS is 1.410179. Noteworthy, there is a sharp decrease of JM distance provided by the two algorithms when the number of classes changes from two to three. As can be observed from Fig. 9 by analyzing the reflectance of nine land-cover classes, this is due to the fact that the added new class, corn3, is very similar to corn1, whereas corn1 and corn2 have better classes separability and are easy to be classified when the number of classes is equal to two. With the increase of the number of classes, the class separability becomes more complex. When the number of classes varies from five to nine, the JM distance provided by CSFW makes much more improvement over those provided by CSFS. The reason for this is that CSFW adjusts adaptively to the weights of features in the evolution process to obtain better solutions.



Fig. 18. Indian Pine AVIRIS Image RGB (57, 27, 17).

Fig. 17 shows the execution times for the different classes provided by CSFS and CSFW. Different with Fig. 12, the measurement unit used is second rather than millisecond in Fig. 12 for the sake of displaying the results better. As can be seen from Fig. 17, CSFW has a sharp increase of computation time from eight classes to nine classes. To the other classes, the two proposed algorithms cost similar time.

From the previous discussion, it can be concluded that CSFS and CSFW are both competent algorithms for dimensionality reduction and that CSFW may have better potential than (or at least as well as) CSFS when applied to the same case in dimensionality reduction.

B. Experiment 2: Indian Pine AVIRIS

The image data used in this experiment refer to the agricultural area of Indian Pine in the northern part of Indiana [39], [40]. The image (145 × 145 pixel) was acquired by the AVIRIS in June 1992 and was downloadable from the Web site (<http://dynamo.ecn.purdue.edu/~biehl/MultiSpec/documentation.html>). The data set was composed of 220 spectral channels, and their spectral ranges were from 0.4 to 2.5 μm in approximately 10-nm bandwidths. Fig. 18 shows the experimental AVIRIS image. The ten most representative land-cover classes, i.e., Corn-notill (C1), Corn-min (C2), Grass/Pasture (C3), Grass/Trees (C4), Hay-windrowed (C5), Soybeans-notill (C6), Soybeans-min (C7), Soybeans-clean (C8), Woods (C9), and Bldg-Grass-Tree drives (C10), were considered. The list of classes and the number of labeled samples for each class are given in Table II, and Fig. 19 shows the reflectance of the above ten land-cover classes. The field map is shown in Fig. 20 based on the ground-truth data. The crop canopies were about 5% of the coverage, the rest being soil covered with the residues of the previous year’s crops. The three different levels of tillage indicating large, moderate, and small amounts of residue were referred as no till, minimum till, and clean till, respectively [39].

In this experiment, the method to select the training and test data set is the same with that of Experiment 1: approximately half of the available samples were used as the training data set, whereas another half of the available samples served as

TABLE II
LAND-COVER CLASSES AND ASSOCIATED NUMBERS
OF PIXELS USED IN EXPERIMENT 2

Class Name	Number of labeling samples
C1. Corn-notill	1434
C2. Corn-min	834
C3. Grass/Pasture	497
C4. Grass/Trees	747
C5. Hay-windrowed	489
C6. Soybeans-notil	968
C7. Soybeans-min	2468
C8. Soybeans-clean	614
C9. Woods	1294
C10. Bldg-Grass-Tree drives	380
Overall	9725

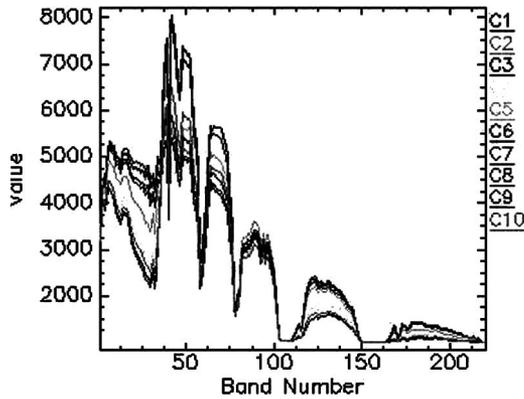


Fig. 19. Reflectance of ten land-cover classes.



Fig. 20. Field map in Experiment 2.

the test data set in the experiment. The values of the primary parameters were set as follows: Population size = 50, Maximum iterations = 100, $n = 50$, $\beta = 0.02$, and $d = 0$.

The execution time is shown in Fig. 21, and SFS runs the fastest. GA, CSFS, and CSFW require similar computation time. SFFS is faster than GA, CSFS, and CSFW for the number of selected features below 25, whereas, for the larger number of features, over 37, SFFS is the slowest. For the DBFE used in the experiment, the most time-consuming operation is related to the calculations of the decision-boundary feature matrix. DBFE

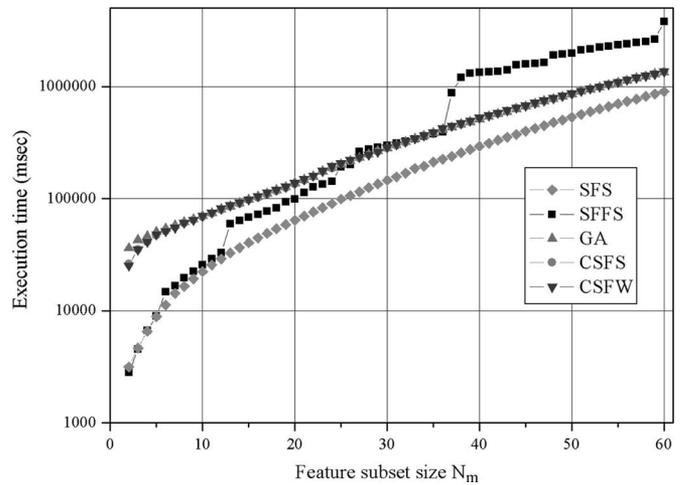


Fig. 21. Execution times required by different algorithms in Experiment 2.

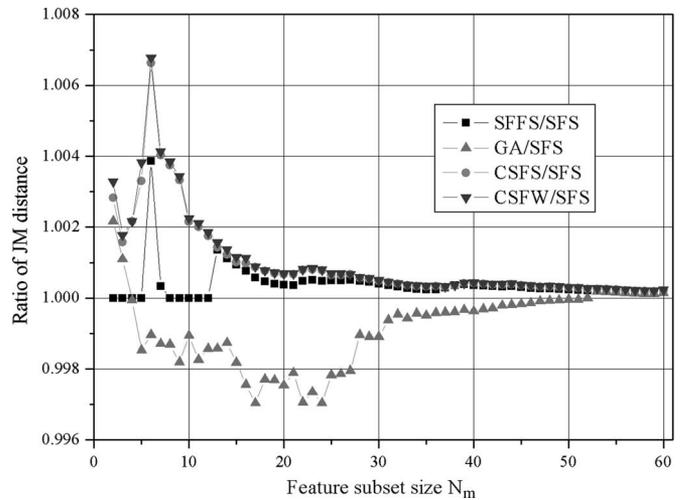


Fig. 22. Values of the criterion function for different feature subset sizes provided by different algorithms in Experiment 2.

took approximately 10 min to obtain the decision-boundary feature matrix.

Fig. 22 shows the values of the criterion function computed on the subsets in the training process provided by SFS, SFFS, GA, CSFS, and CSFW. As in Fig. 12, the ratio of JM distance was used for the y -axis. For a more detailed verification of the above results, the set of the labeled samples was classified using the five algorithms: SFS-ML, SFFS-ML, GA-ML, CSFS-ML, and CSFW-ML. Figs. 23 and 24 show the comparative results using the overall accuracy for training data and test data, respectively.

As can be seen from Fig. 22, SFFS, CSFS, and CSFW make some improvements over SFS for the number of selected features below 24, whereas, for the larger number of features, differences can be ignored. The improvement for six features is the most significant. In addition, CSFS and CSFW have a greater improvement than SFFS when the number of selected features is changed from 2 to 12 and when CSFW exhibits a better potential than CSFS. As for GA, when the crossover and the mutation rates are approximately equal to 0.8 and 0.1, a better result can be obtained. However, the results provided by

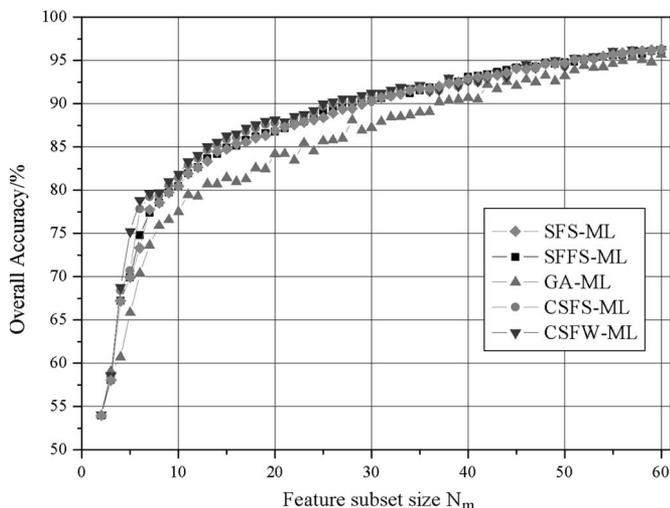


Fig. 23. Classification accuracy achieved by the following five algorithms: SFS, SFFS, GA, CSFS, and CSFW, using the training data in Experiment 2.

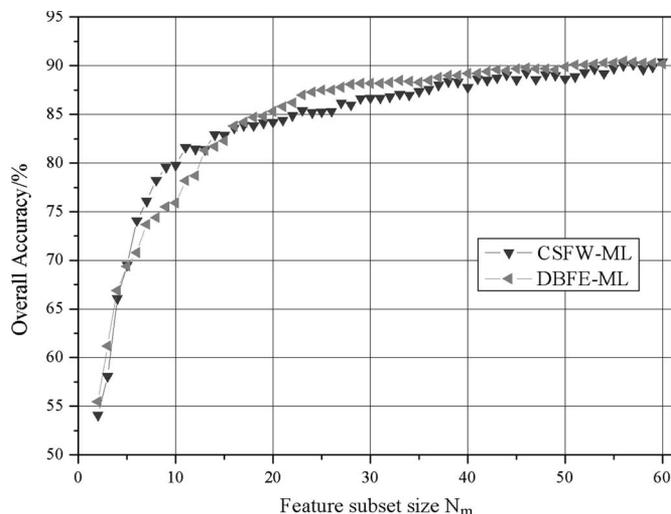


Fig. 25. Test classification accuracy achieved by CSFW and DBFE using the image with 100 bands in Experiment 2.

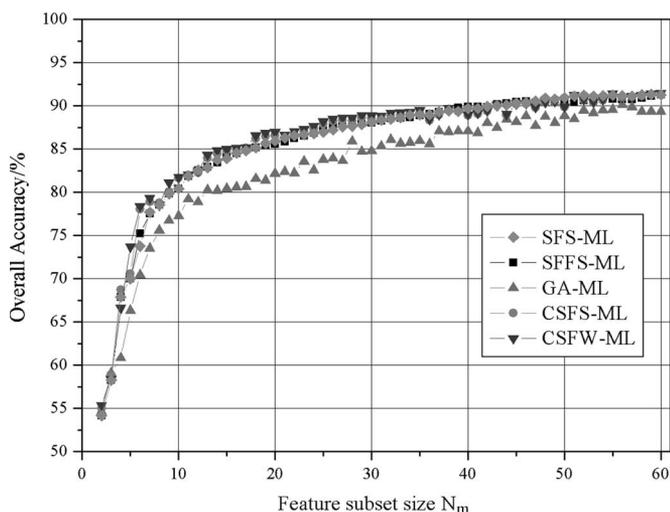


Fig. 24. Classification accuracy achieved by the following five algorithms: SFS, SFFS, GA, CSFS, and CSFW, using the test data in Experiment 2.

GA do not outperform those provided by SFS when the number of selected features is larger than two.

It is observed from Figs. 23 and 24 that the CSFW-ML and CSFS-ML classifiers produce better classification results not only on the training data but also on the test data than traditional algorithms, such as SFS-ML, SFFS-ML, and GA-ML. SFS-ML and SFFS-ML obtain similar results for different numbers of selected features. As can be seen from Fig. 23, the overall accuracy of the training data provided by CSFS-ML and CSFW-ML is over 90%, when the number of selected features is larger than 27 and 26, respectively, whereas SFS-ML, SFFS-ML, and GA-ML require 30, 29, and 37 features to arrive at the same value, respectively. Using the test data in Fig. 24, CSFS-ML and CSFW-ML only require 42 and 41 features to arrive at the 90% accuracy, whereas SFS-ML, SFFS-ML, and GA-ML require 43, 43, and 56 features, respectively.

To utilize DBFE better, the water absorption bands (104–108, 150–163, 220) have been discarded first. Then, the number of bands was reduced by the simple feature-selection method

considering one band every two. That is, the number of the selected image’s bands used by DBFE is equal to 100 in this experiment. Fig. 25 shows the test accuracy provided by DBFE-ML and CSFW-ML to the different numbers of selected features from 2 to 60 in the image with 100 bands.

As shown in Fig. 25, DBFE-ML obtains a better result than CSFW-ML when the number of selected features is larger than 16. The best accuracies achieved by CSFW-ML and DBFE-ML using the test data are 90.44% and 90.5%, respectively. The experiment shows that DBFE-ML slightly outperforms CSFW-ML; however CSFW-ML is an easier way for dimensionality reduction as it does not need the prerduction process, such as from 220 to 100 bands in this experiment.

The above analysis of the experimental results clearly shows that CSFW and CSFS fare the best among all the algorithms tested for dimensionality reduction of hyperspectral remote-sensing imagery.

VII. SENSITIVITY ANALYSIS OF CSFS AND CSFW

CSFS and CSFW have two important parameters that are significantly influenced by the JM distance and the computational complexity. They are as follows:

- 1) d : the number of displace antibodies;
- 2) β : the multiplication factor of clonal antibody.

In order to analyze the effects of these parameters when running CSFS and CSFW, the labeled data set of the Xiaqiao PHI images, with 80 bands and 20 features, was tested with different parametric values. It is noted that the AVIRIS data set can be tested in a similar way.

A. Sensitivity in Relation to Parameter d

To be consistent with other traditional algorithms, d was set to zero in the aforementioned experiments (Section VI). However, when d is larger than zero, the proposed CSFS and CSFW may increase the diversity of the antibody population to improve the performance of JM distance. In order to study

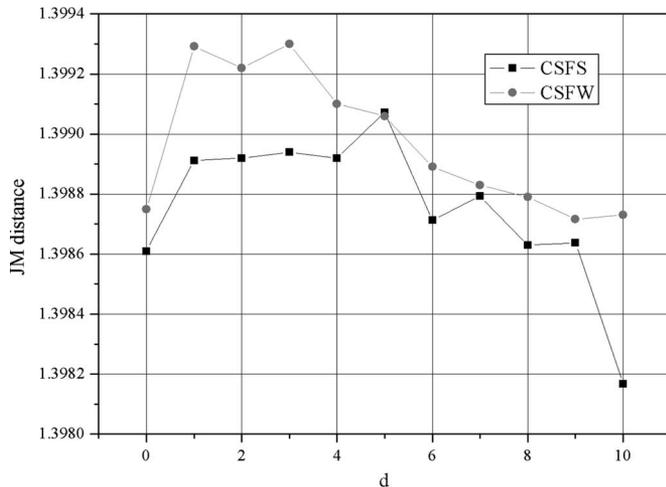


Fig. 26. Sensitivity of CSFS and CSFW in relation to d .

the sensitivity of CSFS and CSFW in relation to d , the parameters other than d were set as the same as Experiment 1. Specifically, the number of selected features is 20 in the experiment, and d is assumed to have the following values: $d = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. Fig. 26 shows the sensitivity of CSFS and CSFW in relation to the parameter d by comparing with the JM distance provided by the two proposed algorithms.

As shown in Fig. 26, the JM distance provided by CSFS increases when the value of d ranges from zero to five, whereas the JM distance decreases when it is larger than five. The JM distance provided by CSFS reaches a minimum, 1.39817, when the value of d is ten. Similarly, CSFW increases the JM distance when the value of d is changed from zero to three, and it can reach a maximum, 1.3993, when the value of d is three, while the JM distance decreases after the value of d increases. These results imply that CSFS and CSFW will achieve a better solution when d is in the appropriate range than when d is equal to zero. However, when d exceeds the range, JM distance may decrease. The reason for this is as follows. The appropriate value of d can increase the diversity of the antibody population to avoid the local optimum, but when the value is too large, d new displace antibodies produced by a random process will destroy the evolution capability of the original antibody population. Based on our experience, d typically ranges between zero and five.

B. Sensitivity in Relation to Parameter β

To evaluate the sensitivity of CSFS and CSFW with respect to β , other parameters were kept the same as Experiment 1, and β was assumed to have the following values: $\beta = \{0.02, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5\}$.

As shown in Fig. 27, the higher the β value, the faster the convergence. However, the computational cost per generation increases linearly with β because the number of antibody population increases linearly as indicated by (8). Furthermore, the value of β is of importance to the CSFS and CSFW algorithms in maintaining diversity of the population because β determines the number of cloning antibody used to extend the search space and gives the mutation a good chance to proceed further

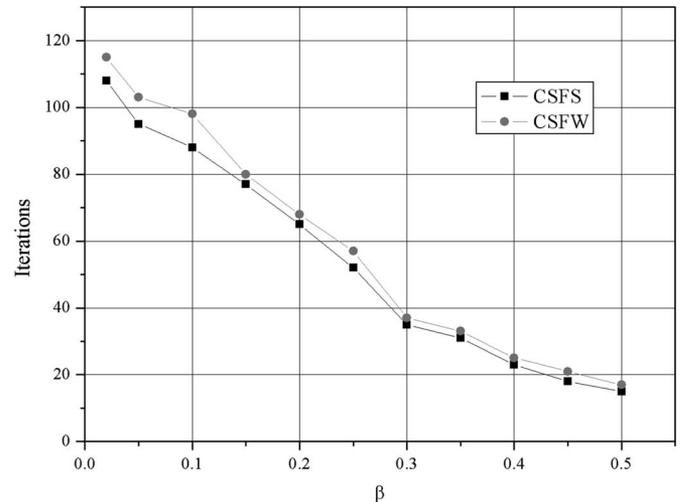


Fig. 27. Sensitivity of CSFS and CSFW in relation to β .

toward the solution. Based on our experience and experiments, β typically ranges between 0.02 and 0.3.

VIII. CONCLUSION

Based on AIS and clonal selection theory, this paper proposed a new search strategy for dimensionality reduction of hyperspectral remote-sensing images. In line with this strategy, a feature-selection algorithm, CSFS, has been developed. Central to CSFS is antibody population evolution, clonal selection, and memory-cell development. In CSFS, the clone process may raise the average affinity value, extend the search space, and provide the mutation a good chance to move closer to optimum. The mutation step is critical to CSFS as it helps escape local optimums and produces new properties of mutated antibodies that can remain if they are successful. In this way, CSFS can obtain an optimum in the feature space. Moreover, a feature-weighting algorithm, CSFW, has also been developed. In this paper, each antibody is directly represented by a string consisting of integer numbers and their corresponding weights, allowing independent linear scaling of each feature. Albeit linear coefficients are obtained by CSFW, the relationship between the input and output features need not necessarily be linear. The transformation of features in CSFW can be used with the feedback from the criterion function in searching the maximal JM distance.

The experimental results in this paper consistently show that the proposed CSFS and CSFW provide better results than the traditional SFS and SFWS algorithms, as well as the latest genetic feature-selection algorithm. In the two experiments using different hyperspectral remotely sensed images, CSFS and CSFW are able to arrive at better results with less number of features and achieve higher classification accuracy of over 90% for the training data set and the test data set in conjunction with the ML classifier. This evinces that CSFS and CSFW are competent for dimensionality reduction of hyperspectral remote-sensing images and that they exhibit a better alternative to the standard SFS and SFWS methods for feature selection of hyperspectral data. Although the computation times of CSFS

and CSFW are, respectively, higher than SFS and SFFS, the execution times of both proposed algorithms are quite acceptable. Moreover, it may be reduced by selecting the appropriate parameters. In the comparison between the two proposed algorithms, CSFW exhibits a better potential than the feature-selection CSFS algorithm when they are applied to the same case in dimensionality reduction. Compared with DBFE, after prereduction, DBFE achieves slightly better results than CSFW in the two experiments, yet CSFW is an easier way as it does not require prereduction. The sensitivity analysis of the parameters in CSFS and CSFW demonstrates that these two algorithms allow users to improve the effectiveness of the algorithms and reduce the computational complexity. In particular, when the value of displace rate d is larger than zero, the two proposed algorithms may make some improvements and obtain better performance.

Our future work will explore further immunological properties and models, and we intend to investigate immune networks to develop a comprehensive immune recognizing system for hyperspectral remote-sensing imagery. In addition, the CSFS and CSFW may be used for preprocessing of feature extraction, e.g., with DBFE, to improve the classification accuracy.

ACKNOWLEDGMENT

The authors would like to thank the research group supervised by Prof. D. Landgrebe, Purdue University, West Lafayette, IN, for providing the free downloads. The AVIRIS image was downloadable from the Web site (<http://dynamo.ecn.purdue.edu/~biehl/MultiSpec/documentation.html>).

REFERENCES

- [1] C. Lee and D. A. Landgrebe, "Analyzing high-dimensional multispectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 31, no. 4, pp. 792–800, Jul. 1993.
- [2] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed. New York: Academic, 1990.
- [3] C.-I. Chang and S. Wang, "Constrained band selection for hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 6, pp. 1575–1585, Jun. 2006.
- [4] A. Plaza, P. Martinez, J. Plaza, and R. Perez, "Dimensionality reduction and classification of hyperspectral image data using sequences of extended morphological transformations," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 466–479, Mar. 2005.
- [5] G. F. Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEE Trans. Inf. Theory*, vol. IT-14, no. 1, pp. 55–63, Jan. 1968.
- [6] A. R. Webb, *Statistical Pattern Recognition*, 2nd ed. Hoboken, NJ: Wiley, 2002.
- [7] A. Ifarraguerri and C.-I. Chang, "Unsupervised hyperspectral image analysis with projection pursuit," *IEEE Trans. Geosci. Remote Sens.*, vol. 38, no. 6, pp. 2529–2538, Nov. 2000.
- [8] C. Lee and D. A. Landgrebe, "Feature extraction based on decision boundaries," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 4, pp. 388–400, Apr. 1993.
- [9] J. Wang and C.-I. Chang, "Independent component analysis-based dimensionality reduction with applications in hyperspectral image analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 6, pp. 1586–1600, Jun. 2006.
- [10] C. Lee and D. A. Landgrebe, "Decision boundary feature extraction for neural networks," *IEEE Trans. Neural Netw.*, vol. 8, no. 1, pp. 75–83, Jan. 1997.
- [11] A. Jain and D. Zongker, "Feature selection: Evaluation, application, and small sample performance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 2, pp. 153–158, Feb. 1997.
- [12] L. Bruzzone, F. Roli, and S. B. Serpico, "An extension of the Jeffreys–Matusita distance to multiclass cases for feature selection," *IEEE Trans. Geosci. Remote Sens.*, vol. 33, no. 6, pp. 1318–1321, Nov. 1995.
- [13] P. M. Narendra and K. Fukunaga, "A branch and bound algorithm for feature subset selection," *IEEE Trans. Comput.*, vol. C-26, no. 9, pp. 917–922, Sep. 1977.
- [14] S. D. Stearns, "On selecting features for pattern classifiers," in *Proc. 3rd Int. Conf. Pattern Recog.*, Coronado, CA, 1976, pp. 71–75.
- [15] P. Pudil, J. Novovicova, and J. Kittler, "Floating search methods in feature selection," *Pattern Recognit. Lett.*, vol. 15, no. 11, pp. 1119–1125, Nov. 1994.
- [16] W. Siedlecki and J. Sklansky, "A note on genetic algorithms for large scale feature selection," *Pattern Recognit. Lett.*, vol. 10, no. 5, pp. 335–347, Nov. 1989.
- [17] M. Mitchell, *An Introduction to Genetic Algorithms*. Cambridge, MA: MIT Press, 1998. Reprint edition.
- [18] T. Mantere and J. T. Alander, "Evolutionary software engineering, a review," *Appl. Soft Comput.*, vol. 5, no. 3, pp. 315–331, Mar. 2005.
- [19] M. L. Raymer, W. F. Punch, E. D. Goodman *et al.*, "Dimensionality reduction using genetic algorithms," *IEEE Trans. Evol. Comput.*, vol. 4, no. 2, pp. 164–171, Jul. 2000.
- [20] W. Siedlecki and J. Sklansky, "On automatic feature selection," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 2, no. 2, pp. 197–220, 1988.
- [21] L. N. De Castro and F. J. Von Zuben, "Clonal selection algorithm with engineering applications," in *Proc. GECCO's*, Las Vegas, NV, 2000, pp. 36–37.
- [22] L. N. De Castro and F. J. Von Zuben, "Learning and optimization using the clonal selection principle," *IEEE Trans. Evol. Comput.*, vol. 6, no. 3, pp. 239–250, Jun. 2002.
- [23] D. Dasgupta, *Artificial Immune Systems and Their Applications*. Berlin, Germany: Springer-Verlag, 1999.
- [24] L. N. de Castro and J. Timmis, *Artificial Immune Systems: A New Computational Intelligence Approach*. London, U. K.: Springer-Verlag, 2002.
- [25] Y. Zhong, L. Zhang, B. Huang, and P. Li, "An unsupervised artificial immune classifier for multi/hyperspectral remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 2, pp. 420–431, Feb. 2006.
- [26] F. M. Burnet, *The Clonal Selection Theory of Acquired Immunity*. Cambridge, U. K.: Cambridge Univ. Press, 1959.
- [27] F. M. Burnet, "Clonal selection and after," in *Theoretical Immunology*, G. I. Bell, A. S. Perelson, and G. H. Pimbley, Eds. New York: Marcel Dekker, 1978, pp. 63–85.
- [28] Y. Zhong, L. Zhang, J. Gong, and P. Li, "Classification of multi-spectral remote sensing image based on clonal selection," *J. Image Graph.*, vol. 10, no. 1, pp. 18–24, 2005. (in Chinese).
- [29] N. K. Jerne, "The immune system," *Sci. Amer.*, vol. 229, no. 1, pp. 52–60, 1973.
- [30] A. S. Perelson and G. F. Oster, "Theoretical studies of clonal selection: Minimal antibody repertoire size and reliability of self-nonself discrimination," *J. Theor. Biol.*, vol. 81, pp. 645–670, 1979.
- [31] L. Segel and A. S. Perelson, "Computations in shape space: A new approach to immune network theory," in *Theoretical Immunology*, A. S. Perelson, Ed. Reading, MA: Addison-Wesley, 1988, pp. 321–343. Part Two.
- [32] A. S. Perelson and G. Weisbuch, "Immunology for physicists," *Rev. Mod. Phys.*, vol. 69, no. 4, pp. 1219–1267, 1997.
- [33] A. S. Perelson, "Immune network theory," *Immunol. Rev.*, vol. 110, pp. 5–36, 1989.
- [34] L. N. De Castro and F. J. Von Zuben, "Artificial immune systems: Part I—Basic theory and application," State Univ. Campinas, Campinas, Brazil, Tech. Rep.-RT DCA 01/99, 1999.
- [35] J. S. Chun, H. K. Jung, and S. Y. Hahn, "A study on comparison of optimization performances between immune algorithm and other heuristic algorithms," *IEEE Trans. Magn.*, vol. 34, no. 5, pp. 2972–2975, Sep. 1998.
- [36] P. H. Swain and S. M. Davis, *Remote Sensing: The Quantitative Approach*. New York: McGraw-Hill, 1978.
- [37] L. Bruzzone and S. B. Serpico, "A technique for feature selection in multiclass problems," *Int. J. Remote Sens.*, vol. 21, no. 3, pp. 549–563, Feb. 2000.
- [38] J. A. Richards, *An Introduction to Remote Sensing Digital Image Analysis*. Berlin, Germany: Springer-Verlag, 1986.
- [39] P. F. Hsieh and D. Landgrebe, "Classification of high dimensional data," Ph.D. dissertation, School Elect. Comput. Eng., Purdue Univ., West Lafayette, IN, 1998.
- [40] S. B. Serpico and L. Bruzzone, "A new search algorithm for feature selection in hyperspectral remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 7, pp. 1360–1367, Jul. 2001.



Liangpei Zhang (M'06) received the B.S. degree in physics from Hunan Normal University, ChangSha, China, in 1982, the M.S. degree in optics from the Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an, China, in 1988, and the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 1998.

From 1997 to 2000, he was a Professor with the School of the Land Sciences, Wuhan University. Since August 2000, he has been with the State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing, Wuhan University as a Professor and Head of the Remote Sensing Section. He has published more than 120 technical papers. His research interests include hyperspectral remote sensing, high-resolution remote sensing, image processing, and artificial intelligence.

Dr. Zhang has served as Cochair of the SPIE Series Conferences on Multispectral Image Processing and Pattern Recognition (MIPPR) and the Conference on Asia Remote Sensing in 1999, Editor of the MIPPR01, MIPPR05, and Geoinformatics Symposiums, Associate Editor of the *Geospatial Information Science Journal* and Chinese National Committee for the International Geosphere-Biosphere Programme, and an executive member of the China Society of Image and Graphics.



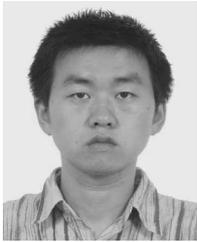
Bo Huang received the Ph.D. degree in geographic information system from the Institute of Remote Sensing Applications, Chinese Academy of Sciences, Beijing, China, in 1997.

He is currently an Associate Professor with the Department of Geography and Resource Management, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong. His current research interests include spatial optimization, spatial statistics for change analysis, and image processing. He has authored or coauthored over 30 papers in refereed international journals.



Jianya Gong received the B.S. degree in surveying and mapping from East China Geology College, Jiangxi, in 1982, and the Ph.D. degree in photogrammetry and remote sensing from Wuhan Technical University of Surveying and Mapping, Wuhan, China, in 1992.

Since 1994, he has been a Professor with the State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing, Wuhan University, Wuhan, China. His research interests include photogrammetry, remote sensing, and geographic information systems.



Yanfei Zhong received the B.S. degree in information engineering from Wuhan University, Wuhan, China, in 2002, and the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2007.

He is with the State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing, Wuhan University. His current research interests focus on multi- and hyperspectral image processing, artificial immune systems, and pattern recognition.



Pingxiang Li (M'06) received the B.S., M.S., and Ph.D. degrees in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 1986, 1994, and 2003, respectively.

Since 2002, he has been a Professor with the State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing, Wuhan University. His research interests include photogrammetry and synthetic aperture radar image processing.