IEICE TRANS. INF. & SYST., VOL.E90–D, NO.7 JULY 2007

PAPER

Particle Swarms for Feature Extraction of Hyperspectral Data

Sildomar Takahashi MONTEIRO[†a)], ***Student Member and*** **Yukio KOSUGI**[†], ***Member***

SUMMARY This paper presents a novel feature extraction algorithm based on particle swarms for processing hyperspectral imagery data. Particle swarm optimization, originally developed for global optimization over continuous spaces, is extended to deal with the problem of feature extraction. A formulation utilizing two swarms of particles was developed to optimize simultaneously a desired performance criterion and the number of selected features. Candidate feature sets were evaluated on a regression problem. Artificial neural networks were trained to construct linear and nonlinear models of chemical concentration of glucose in soybean crops. Experimental results utilizing real-world hyperspectral datasets demonstrate the viability of the method. The particle swarms-based approach presented superior performance in comparison with conventional feature extraction methods, on both linear and nonlinear models.

key words: *feature extraction, particle swarm optimization, hyperspectral data, neural networks, principal components analysis*

1. Introduction

Advances in optical and computational technology have allowed the acquisition of an ever-increasing amount of information from a scene. However, those huge amounts of data represent a challenge for the algorithms to process and extract the relevant information for the desired applications. Despite the wealth of information, the datasets are commonly plagued by redundant or irrelevant features. In real-world applications, the typical scenario of few data samples in a high-dimensional feature space causes what was termed by Bellman [1] as the curse of dimensionality, referring to the exponential increase in complexity of high-dimensional spaces with the increase in the number of measurements [2].

Hyperspectral imaging systems are able to acquire several hundreds of spectral information from the visible to the infrared region. Nonetheless, neighboring spectral bands are usually highly redundant [3]. To avoid the curse of dimensionality, feature extraction algorithms have been proposed to reduce the amount of data and, at the same time, keep the relevant information necessary for image interpretation or classification [4]. In the field of remote sensing, several methods for dimensionality reduction have been proposed over the years [5]. However, few works have addressed hyperspectral datasets, in which conventional statistics methods have great difficulty. Conventional methods usually aim at improving or preserving classification accuracy (e.g., [6]). Furthermore, the performance of the algorithms on regression problems has seldom been investigated.

Particle swarm optimization (PSO) is a very promising evolutionary computation technique that has been developed recently due to research on bird flock simulation by Kennedy and Eberhart [7]. PSO's main attractiveness is its simplicity and velocity, allied with robustness. The PSO algorithm has similar capabilities as genetic algorithms and, additionally, has the advantage of simpler implementation and fewer parameters to adjust. PSO is able to solve most optimization problems, or problems that can be converted to optimization problems. Different approaches for feature selection using PSO have been reported [8]–[10]. Nevertheless, the search is commonly limited to a predefined number of features, which can be difficult to determine a priori for many problems. In addition, the question of how to define the target functions to be optimized may be highly dependent on the problem at hand, thus warranting further investigation.

PSO has been reported to be impressively resilient and particularly suited to global optimization problems in which the search space potentially contains multiple local minima. One of the characteristics of hyperspectral datasets is the high correlation between neighboring bands. A nonlinear modeling of hyperspectral data may not present monotonic property. Unlike other optimization methods, correlation between search features or nonmonotonic functions does not pose a problem for PSO. The use of PSO to process hyperspectral data is also appealing due to the capability of visualizing the location of particles' positions in the search space. Since each spectral dimension of the hyperspectral dataset represents one band wavelength, the location of the particles' positions during feature extraction may prove useful to identify interesting characteristics of the physical process associated with the induction algorithm. In other words, indicating the corresponding wavelengths of selected features may provide insights about which spectral regions are more significant in modeling the problem.

In this paper, a new approach based on PSO for feature extraction is presented. A multi-criteria optimization technique to perform feature extraction using two particle swarms is investigated. A method to extract optimal spectral bands from hyperspectral data was developed and applied on a regression problem in the field of agriculture. Neural networks were implemented to learn linear and nonlinear models of glucose content in soybeans. Experiments were carried out using real-world hyperspectral datasets from soy-

Manuscript received September 12, 2006.
Manuscript revised January 28, 2007.

[†]The authors are with the Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology, Yokohama 226-8502, Japan.
a) E-mail: monteiro@pms.titech.ac.jp

DOI: 10.1093/ietisy/e90-d.7.1038

bean fields. Our method is innovative in adapting the PSO algorithm to a two-objective feature extraction problem and in proposing its utilization to process remotely sensed hyperspectral imagery data.

2. Background

Feature extraction methods transform the original feature bands into a new feature space. Feature selection is a subtype of feature extraction where the dimensionality reduction is achieved by selecting bands rather than transforming the data [11]. While feature extraction causes loss of interpretability, the feature selection preserves data interpretability [12]. Feature selection methods are advantageous when the user needs to make decisions based on meaningful features of the original data, or if he wants to exclude nonnecessary data components to reduce the cost and labor of data acquisition. Thus, feature selection is highly suitable to hyperspectral imagery, in which the data is intrinsically related to physical wavelengths and not all spectral bands are always necessary for a certain application.

There are two main approaches for feature selection based on whether the algorithm is dependent or independent of the associated induction process using its output [13]. If the algorithm selects features independently of its effect on the performance of the induction algorithm, it follows a filter approach. The drawback of this approach is the difficulty in finding the optimal feature set that improves the performance of the induction algorithm. On the other hand, if the feature selection algorithm is used in conjunction with the induction algorithm evaluating the performance of the selected features, it is classified as wrapper approach. The latter requires the evaluation of each candidate feature set under consideration by the induction algorithm, which may require considerable computational cost.

Assuming that the hyperspectral imagery data matrix I is composed of n spectral images $I(\lambda)$, $(\lambda = 1, \ldots, n)$, at each wavelength λ acquired by the sensor. The aim of the feature selection is to find a set of m features, where $m < n$, to minimize the evaluation criterion. Feature selection can be implemented as an optimization procedure of search for the optimal feature set that better satisfy a desired criterion. The candidate feature set can be coded as a binary vector $b = \{\beta_1, \ldots, \beta_n\}$, where each element β assumes value 1 if the feature is selected or 0 if it is not used.

3. Particle Swarm Optimization

The basic PSO algorithm performs optimization in continuous, multidimensional search spaces. PSO starts with a population of random particles. Each particle is associated with a velocity. Particles' velocities are adjusted according to the historical behavior of each particle and its neighbors while they fly through the search space. Therefore, the particles have a tendency to fly towards the better and better search area over the search process course.

The version of PSO algorithm utilized [14] is described mathematically by the following equations:

$$v_{id}^{t+1} = w v_{id}^{t} + c_1 r_1 (p_{id}^{t} - x_{id}^{t}) + c_2 r_2 (p_{gd}^{t} - x_{id}^{t}) \tag{1}$$

and

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1} \ , \tag{2}$$

where c_1 and c_2 are positive constants, called learning rates; r_1 and r_2 are random functions in the range [0, 1]; w is a inertia weight; $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$ represents the position of the i^{th} particle in a problem space with D dimensions; t indicates the iteration number; $P_i = (p_{i1}, p_{i2}, \ldots, p_{iD})$ represents the best previous position of the swarm; the index g indicates the best particle among all the particles in the population; $V_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$ represents the rate of change of position (velocity). If the sum of the factors in the right side of Eq. (1) exceeds a specified constant value, particles' velocities on each dimension are clamped to a maximum velocity $V_{\max}$.

3.1 Binary PSO

To perform feature selection, the PSO concept needs to be extended in order to deal with binary data [8]. In the binary scheme utilized for feature selection, each feature is represented by one particle of a swarm. The particles' positions are treated as probabilities in order to guide the selection of features.

The candidate feature set is determined using a roulette wheel selection. At each spin of the roulette, the wheel's marker will point to a feature to be selected. The roulette is played until a defined number of selected features is reached. Each feature is assigned with a probability Pr proportional to the real value x_{id} calculated in Eq. (2) limited to the interval [0, 1], according to the equation

$$\Pr(x_{id}) = \frac{{x_{id}}^{\alpha}}{\sum_{d=1}^{n} {x_{id}}^{\alpha}} \ , \tag{3}$$

where α is the selection pressure that controls the probability of selecting highly fit or less fit features. Pr indicates the probability of selecting the feature that is represented by the dimension d of the i^{th} particle.

4. Feature Selection using Two Particle Swarms

By transforming the feature extraction problem into an optimization problem, it is possible to solve it trough the PSO algorithm. The binary PSO presents the limitation of only being able to select a predefined number of features. The optimal size of the feature set could be estimated based on the relation between the number of training samples and number of free parameters in the model. Nevertheless, better models can be obtained by evaluating the candidate feature sets based on their performance and penalizing solutions containing large number of features.

Two particles swarms can be utilized to optimize simultaneously the number of selected features and the error of the model, as shown in Fig. 1. While one particle swarm is sufficient for searching a fixed number of features, two particle swarms can search through the entire space of possible sizes of feature sets. Each candidate feature set is evaluated by observing its performance on a regression problem. The induction algorithm is a neural network utilized to construct regression models. The proposed method falls in the category of wrapper approaches for feature selection. The domain of both particles is the set of all real numbers, since particles' positions in the basic PSO assume continuous values.

The particle swarm on the left of the diagram in Fig. 1 is a "discrete" PSO, configured to search for the number m of features to be selected. To search for the optimal size of the feature set, which assume discrete values, the particles' positions x_{id} real values need to be transformed to integers. The discretization is performed through a nearest integer function $\text{nint}(x) = \lfloor x \rceil$ that is defined as the integer closest to x. To avoid statistical biasing, half-integers are always converted to even numbers, e.g., $\lfloor 1.5 \rceil = 2$, $\lfloor 2.5 \rceil = 2$, $\lfloor 3.5 \rceil = 4$, etc. Each particle of the discrete particle swarm has one dimension and its range is $\{1, \ldots, n\}$, where n is the number of features (hyperspectral bands). In other words, the search space of the discrete particle swarm is feature sets containing 1 to n bands, limited by the number n of dimensions of the original dataset, in the case of hyperspectral imagery data, the maximum number of spectral bands available.

The particle swarm on the right of the diagram in Fig. 1 is a "binary" PSO that effectively searches for the optimal combination of features as described in the previous section. This particle swarm is encoded in n bits, according to the number of dimensions of the dataset. In other words, each particle of the binary particle swarm has n dimensions equal to the number of features (hyperspectral bands) and its range is the binary encoding $\{0, 1\}$.

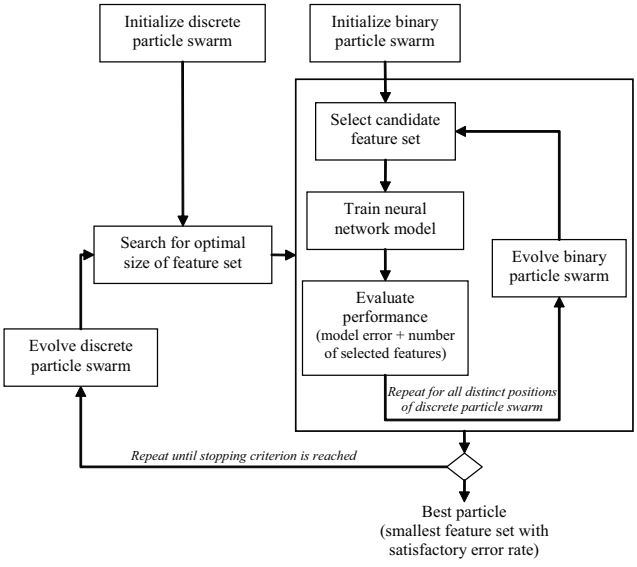


Fig. 1 Diagram of the feature extraction algorithm based on two particle swarms.

The process of feature selection is carried out in cycles called epochs. In the proposed method, each epoch consists of two phases. Firstly, the discrete particle swarm is evolved, letting the particles update their positions. Then, the binary particle swarm is evolved, each step selecting up to the number of features defined by the particles of the discrete particle swarm. The binary particle swarm may be updated several times at each epoch, for the different positions of the discrete particle swarm. However, if two or more particles of the discrete particle swarm are in the same position, then only the first occurrence will result in the evolution of the binary particle swarm.

4.1 Evaluation Function

By simply minimizing the error rate of the induction algorithm, the feature selection algorithm cannot be expected to also minimize the number of selected features [15]. In order to search for the smallest feature set that satisfies a desired level of performance, the feature selection must be treated as a constrained optimization problem. The optimization is constrained by the size of the feature set and by the error rate of the induction algorithm. A formulation was developed in order to provide control on the balance between the two constraints. This may be necessary when dealing with high dimensional data, such as hyperspectral imagery. Otherwise, very small feature sets may be preferred by the algorithm in detriment of possible better performing feature sets with more features.

A performance evaluation function is introduced to accommodate the two constraints, assessing the evolution of the two particle swarms. It can be expressed by the following equation

$$\text{PEF}(b) = k \times l(b) + \text{PF}(e(b)) \,, \tag{4}$$

where b is the candidate feature set selected by the binary particle swarm; $l(b)$ refers to the cost of extracting features, considering that the costs associated with extracting individual hyperspectral bands are equal, the cost function corresponds to the size of the feature set $l(b) = m$, m is the number of selected bands; the constant k is a scaling factor; and PF(e) is a penalty function for the error $e(b)$ of the induction algorithm.

The penalty function defines a region of feasibility of possible solutions in the error space. It can be expressed as

$$\text{PF}(e(b)) = \frac{\exp((e(b) - u)/s) - 1}{\exp(1) - 1} \,, \tag{5}$$

where u is a feasibility threshold, and s is a small scaling constant.

A feature set is considered feasible if the error in the model output is below the feasibility threshold. For other feature sets presenting higher error, the value of the penalty

function grows rapidly.

The evolution of the two particle swarms is induced by the PEF function. Therefore, at each step of the learning process, the best positions of the particles of both swarms, vectors P_i and P_g in Eq. (1), will be determined according to the positions of the particles that were better evaluated by the PEF function.

5. Neural Networks

Artificial neural networks were implemented as the induction algorithm, to provide linear and nonlinear models of the regression problem. The linear model was obtained using a single-layer perceptron network [16]. Its output can be calculated as $y = f(x) = Wx + \theta$, where x is the input vector, y is the output vector, W is the weight vector, and the parameter θ is the bias. The linear networks were trained using the least means squares algorithm [17].

The nonlinear model was constructed using a multilayer perceptron network, composed of input layer, hidden layer, and output layer, sequentially interconnected in a feed-forward way [18]. The output of the multilayer neural network can be expressed as $y = f(x) = W\varphi(Ax + a) + \theta$, where, again, x and y are the input and output vectors, respectively; A and a are, respectively, the weight matrix and the bias vector of the hidden layer; W and θ are, respectively, the weight matrix and the bias vector of the output layer; and φ is the activation function. The activation function for the hidden layer neurons was the hyperbolic tangent sigmoid. The training method was the Levenberg-Marquardt backpropagation [19]. Early stopping was used to improve generalization and avoid overfitting. The parameters of the training are shown in Table 1.

On both neural network architectures, the number of neurons in the input layer is proportional to the number of features of the reduced dataset. The neural networks were trained to minimize the mean of squared errors, $\mathrm{MSE}(y) = \frac{1}{N}\sum_{i=1}^{N}(y^o - y^t)^2$, between the samples' measured values t and the network outputs o.

6. Experiments

6.1 Hyperspectral Imagery Data

To attest the validity of the proposed method, experiments were conducted with hyperspectral imagery data from soybean fields. The experimental data was obtained using a hyperspectral line sensor ImSpector V10 [20], coupled with CCD camera and computer controller. The sensor acquires data in two dimensions, one containing spatial information and, the other, spectral information. In the spatial plane, the hyperspectral camera produces 484 pixels per line. The scene is scanned to produce images at each band wavelength and the limit for the scanning is the amount of memory available in the controller. The spectral range comprises the visible to the near-infrared, from 400 nm to 1000 nm, each band interleaved by approximately 5 nm, thus producing 121 spectral bands.

The hyperspectral data was acquired in middle summer on a sunny day, around noontime. Hyperspectral imagery data is commonly represented as a three-dimensional data cube, as an example the soybean field dataset is displayed in Fig. 2. The data sample consisted of 13 different varieties of green vegetable soybeans cultivated in an experimental field.

The hyperspectral data were firstly preprocessed. The raw radiance acquired by the sensor was converted to reflectance. Then, the hyperspectral dataset was filtered using a spatial-spectral mean filter. Image regions containing vegetation were identified using a normalized difference vegetation index (NDVI). Finally, the reflectance data of vegetated areas were normalized to the interval [-1, 1] to serve as input vectors to the neural networks.

In addition, to provide target data for the supervised training of the neural networks, freeze-dried samples from the soybean fields were analyzed in the laboratory using liquid chromatography. The measured chemical concentrations served as "ground reference" data and constituted the target vectors used as training data for the neural networks.

Table 1 Parameters for the training of the neural networks.

Parameter	Value
Hidden layer size	10
Output layer size	1
Learning rate	0.01
Momentum	0.9
Maximum number of epochs	1000
Maximum validation failures	5

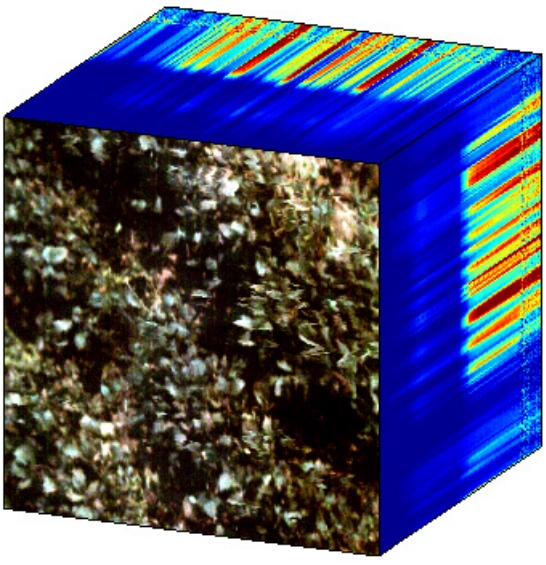


Fig. 2 Hyperspectral data cube of soybean field. Front image is a true-color visualization of the scene. Side images illustrate the spectral information.

The neural networks were trained to model the chemical concentration of glucose in soybeans; the purpose is to predict the sweetness of the soybean crops non-invasively [21].

6.2 Results

The parameters of the particle swarms, shown in Table 2, were chosen through experimentation and based on our previous experience with the binary PSO [22].

To analyze the behavior of the performance evaluation function over the search space, a simple version of the feature selection algorithm was implemented. This version optimizes a predefined number of features using solely the binary PSO in conjunction with the linear networks, which can be trained rapidly and reliably. The simple feature selection algorithm was applied sequentially for all possible sizes of feature sets, starting from 1 to 121.

To define the constants of the penalty function Eq. (5), the error rate of the induction algorithm must be taken into account. The feasibility threshold u must be a value at least slightly greater than the minimum error expected by the best feature set. For the case of the linear model, after preliminary experiments, u was defined as $u = 0.17$. The scaling factor was $s = 5\%$. The constant k, performance evaluation function Eq. (4), is determined considering the dimensionality of the problem and the desired performance. Different values of k were assessed, as shown in Fig. 3. When $k = 0$, the curve is equivalent to that of the penalty function. The curve for $k = 1$ is the original evaluation function without weighting the number of features, which demonstrates the tendency of restricting the search space to very small feature sets. A reasonable search space could be obtained by using $k = 0.05$.

The full version of the feature selection algorithm was then applied to the dataset. To account for the stochastic nature of the PSO algorithm, the experiments were performed over 20 independent runs for each algorithm, every time initializing the swarms with a different random seed. The evolution of the particle swarms is computationally inexpensive, but the overhead of the feature selection process is on the induction algorithm. The performance of the particle swarms in conjunction with the linear networks is presented in Fig. 4.

Table 2 Parameters of the particle swarms.

Parameter	Value
Population size discrete particle swarm	20
Population size binary particle swarm	40
Learning rate $c_1 = c_2$	2
Maximum particle velocity $V_{\max}$	4
Maximum number of epochs	200
Maximum epochs with constant error	30
Initial inertia w_i	0.9
Final inertia w_f	0.2
Epoch of final inertia	190
Selection pressure α †	1

† Utilized by the roulette wheel scheme to turn the PSO into binary.

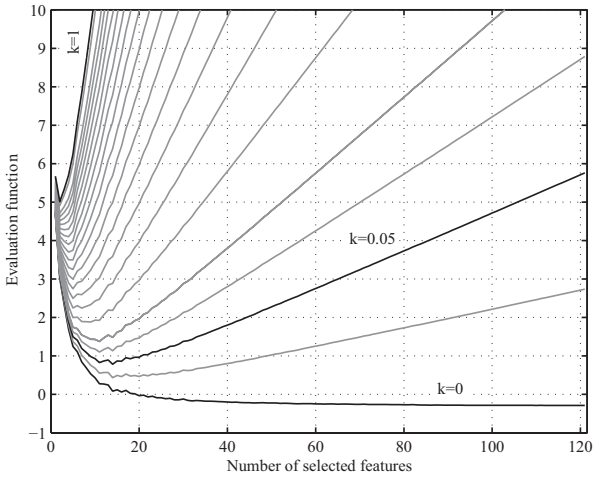


Fig. 3 Curves of the performance evaluation function for various values of k over all possible number of selected features for the linear model case, averages of three runs.

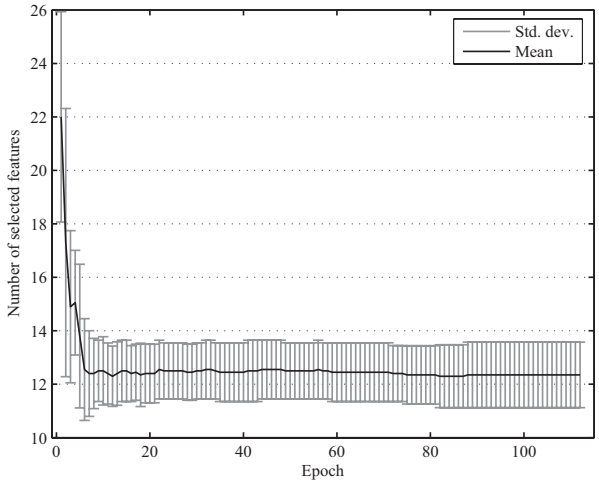
In practice, however, only the best performing feature set selected by the particle swarms is retained, i.e., the feature set presenting the lowest error and highest correlation on the regression problem. In order to verify the accuracy of the method, a linear regression analysis was performed between the best feature set modeled by the neural networks and the ground reference measurements obtained by laboratory analysis. Figure 5 shows the resulting analysis for the best case, glucose content prediction using a reduced feature set modeled by a nonlinear neural network.

The learning process of the particle swarms that selected the best performing feature set is also analyzed. Particles' positions in the beginning, in the middle, and at the end of the feature selection process are shown in Fig. 6. Figure 7 shows the learning curves for each particle swarm: best particles and the averages of all particles.

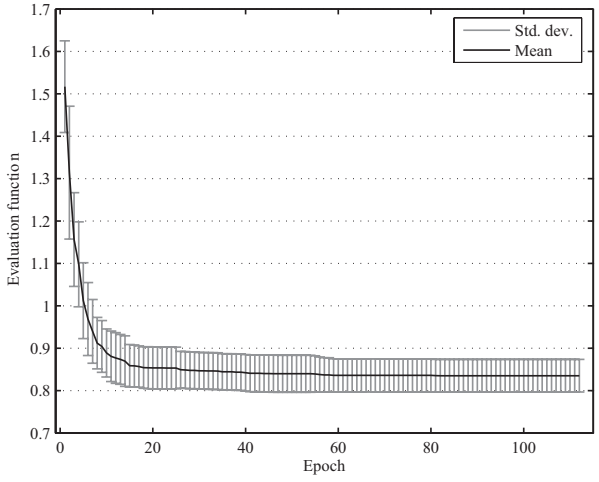
6.3 Comparison with Principal Components Analysis

Principal components analysis (PCA) is a widely used technique to reduce the dimension of hyperspectral datasets. The PCA algorithm identifies and extracts interesting features by retaining only those components that account for a greater part of the variation in the dataset [23]. PCA was implemented through singular value decomposition of the covariance matrix of the dataset. The principal components were ordered according to the magnitude of their variance. The principal components' cumulative contribution for the soybean dataset is shown in Fig. 8. The variance threshold was set to 99.98% in order to retain 11 principal components, the same number of features obtained by the particle swarms.

For the training of the multilayer networks, each candidate feature set was tested in three independent runs and the best networks were retained. The feasibility threshold



(a) Number of selected features optimized by the discrete particle swarm



(b) Performance evaluation function minimized by both swarms

Fig. 4 Graphs of the evolution of the particle swarms with linear models, showing mean value and standard deviation over 20 runs.

of the penalty function Eq. (5) was adjusted to $u = 0.07$, to deal with the lower error presented by the nonlinear network model. The other constants are the same as with the linear model. The particle swarms in conjunction with the multilayer networks were tested over 10 runs.

The feature extraction based on particle swarms is compared with the PCA, for both the linear and nonlinear regression models. To more comprehensibly compare the results between the different methods, the correlation coefficient was calculated as $\mathrm{R}(y) = C(y^o, y^t) \Big/ \sqrt{C(y^o, y^o).C(y^t, y^t)}$, where C is the covariance matrix, and o and t indicate the neural network output and the test dataset measurement, respectively. A summary of the results is presented in Table 3. Additionally, particle's positions, assigned to the correspondent spectral band wavelengths, of the best feature sets selected with both the linear and nonlinear models are shown in Fig. 9.

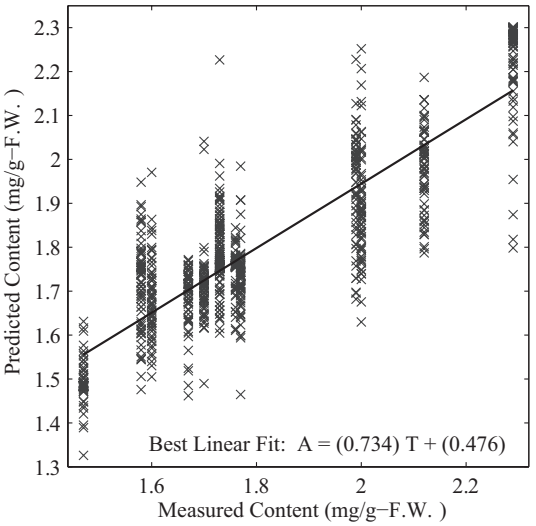


Fig. 5 Linear regression analysis between the ground reference measurements and the predictions given by the nonlinear model with the best feature set.

Table 3 Comparison of particle swarms feature selection (PS-FS) and PCA, applied to model glucose content in soybean crops from hyperspectral data using neural networks.

Algorithm	PEF†	MSE††	R†††
PS-FS linear	0.7542	0.0311	0.6187
PCA linear		0.0401	0.4524
PS-FS nonlinear	0.6382	0.0130	0.8620
PCA nonlinear		0.0149	0.8425

† Performance evaluation function
†† Mean squared error
††† Pearson's correlation coefficient

7. Discussion

The particle swarms were able to optimize the combined criteria efficiently. It is noticeable how fast—less than 20 epochs on average—the particle swarms flocked to the optimal region of the search space, despite the limited size of their populations. This result may have been magnified by the fact that the binary particle swarm can be updated many times at each epoch, which happens frequently at the earlier stages. In the middle of the learning, while the particles were still exploring the search space, the best particles were already close to the optimal region. At the final epoch, nearly all particles' bests were in the same position as the best particles overall. Nevertheless, whereas the fist swarm's particles average and best converged to nearly the same position, the binary particle swarm presented some exploration even close to the final epoch.

The decision on the parameters of the particle swarms affects the exploration-exploitation tradeoff and is highly dependent on the form of the objective function [24]. Successful feature selection was obtained even using conservative values for the PSO basic parameters [25]. Therefore, con-

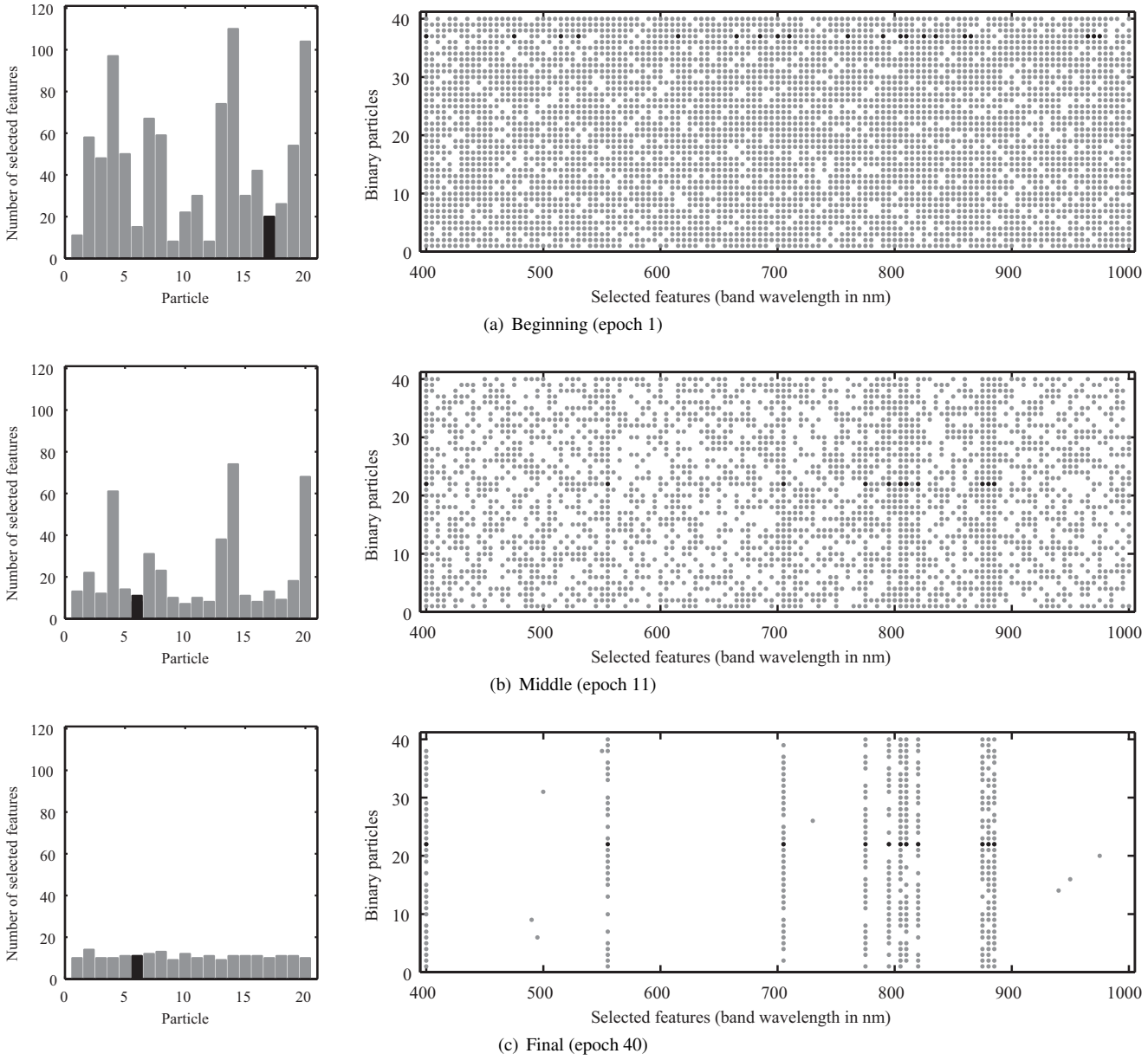
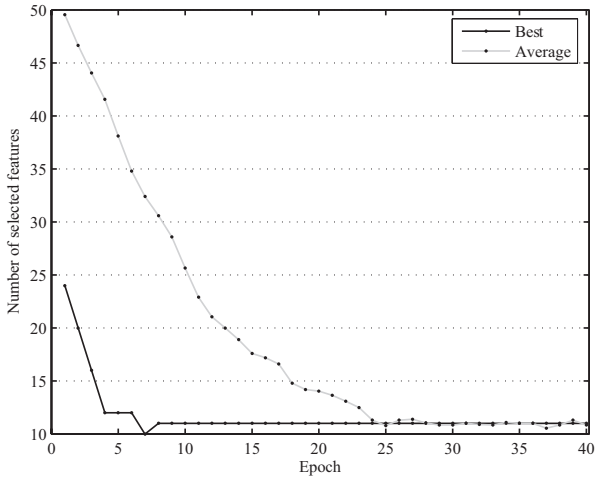


Fig. 6 Location of the particle swarms during the feature selection process while searching for the best performing feature set. Bar plots in the left show the current number of features, particles of the discrete particle swarm. Graphs in the right illustrate the state of the binary particles; the features being selected are represented by dots. Bars/dots in gray represent all particles' current positions. The best particles' positions until that epoch are displayed in dark black.

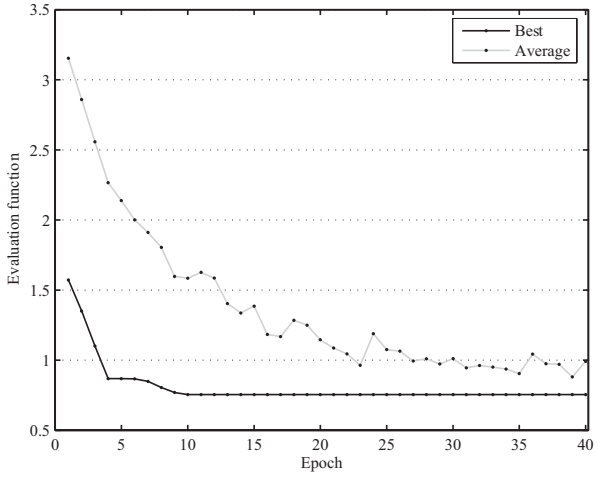
firming the perceived notion of easy implementation and tuning of the PSO algorithm. Furthermore, in the case of feature selection, even if the average performance of the process could be improved, the best performing feature set obtained after running the algorithm a number of times should be nearly the same.

PCA is a filter approach of feature extraction. Even though conceptually different from the proposed wrapper approach, since PCA is a well-studied method for dimensionality reduction, it can be used as a benchmark. Notwithstanding the deceiving impression that only a small number of principal components (about 5) held most of the variability in the soybeans dataset Fig. 8, the rather close results presented by PCA with nonlinear models were only achieved using a higher number of principal components (11). The particle swarms outperformed the PCA in all cases, more distinctively with the linear models. The results confirmed the theoretical superior performance of wrapper approaches over filters, although the latter may present lower overall computational cost.

The performance evaluation function punishes feature sets with high dimensionality. This function may produce



(a) Discrete particle swarm



(b) Combined performance of both particle swarms

Fig. 7 Learning curves of the particle swarms that selected the best performing feature set for the linear model.

excessive punishment, particularly on real-world hyperspectral imagery data, causing the selection of small feature sets that presented undue error. Thus, in order to determine a better compromise between the number of selected features and the induction algorithm's error rate, a constant factor k was introduced in Eq. (4). The usefulness of this constant factor was demonstrated by the improvement of the results in the experiments.

The particle swarms also possess the advantage of permitting the visualization of the selected features in contrast with their spectral locations, providing an appealing analysis tool for the field of remote sensing. The linear models tended to induce the selection of feature sets in which most of the selected features lied in the range between 700 and 900 nm, adjacent to the region known as red-edge, which is reportedly important for vegetation processes [26]. With the nonlinear regression models, on the other hand, the selected features tended to be more spatially distributed over the spectra, resulting in a more efficient use of the available information.

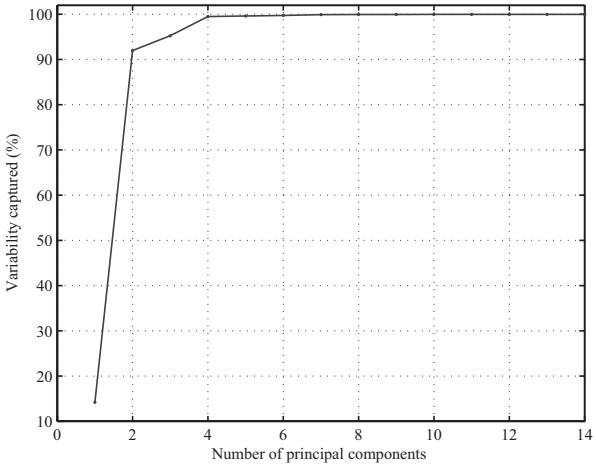


Fig. 8 Cumulative contribution of principal components of the hyperspectral dataset of soybeans.

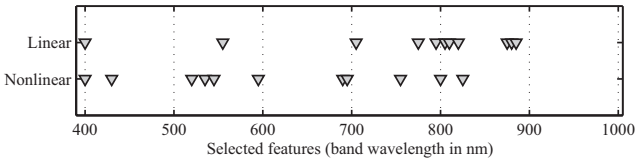


Fig. 9 Spectral location of the best feature sets selected by the particle swarms with the linear and nonlinear models.

8. Conclusions

A new feature extraction method based on two particle swarms, a discrete and a binary, was proposed to concurrently search for the optimal feature set and for the optimal number of features. The applicability of the method to extract information from hyperspectral imagery was demonstrated with experiments utilizing real-world datasets of soybean fields.

The particle swarms were implemented in conjunction with neural networks to model the sweetness in soybean crops, a non-trivial problem. A performance evaluation function was developed to adapt the PSO algorithm for the feature extraction problem. The particle swarms were capable of fast convergence towards the optimal region of the search space. Additionally, a comparison with PCA, a traditional feature extraction technique, showed the competitiveness of the method in extracting better performing features for regression problems.

The proposed method can be a possible alternative for dimensionality reduction problems on search spaces that contain multiple local minima and that require fast convergence and rapid implementation. The parallel nature of the PSO can be useful to circumvent the common drawback of wrapper approaches, the computational load generated by the evaluation of the inducing algorithm, by developing a parallel implementation of the proposed feature selection scheme. The method was proposed not only for dimensionality reduction but also for the spectral analysis of remotely

sensed hyperspectral imagery.

Acknowledgments

The authors are grateful to Mr. Yohei Minekawa of the Tokyo Institute of Technology for providing the datasets for our experiments. This research is part of a joint collaboration with Prof. Tsuneya Akazawa of Yamagata University and Mr. Kunio Oda of the Yamagata General Agricultural Research Center. They also would like to thank Dr. Keisuke Kameyama of the University of Tsukuba for the inspiring discussions about the PSO algorithm.

References

[1] R.E. Bellman, Adaptive Control Processes: A Guided Tour, Princeton University Press, 1961.
[2] P.M. Baggenstoss, "Class-specific classifier: avoiding the curse of dimensionality," IEEE Aerospace and Electronic Systems Magazine, vol.19, no.1–2, pp.37–52, 2004.
[3] P.M. Mather, Computer Processing of Remotely-Sensed Images, An Introduction, John Wiley & Sons, Chichester, 2004.
[4] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," Journal of Machine Learning Research, Special Issue on Variable and Feature Selection, vol.3, pp.1157–1182, 2003.
[5] J.A. Richards and X. Jia, Remote Sensing Digital Image Analysis, An Introduction, third ed., Springer-Verlag, New York, 1999.
[6] S. Yu, S.D. Backer, and P. Scheunders, "Genetic feature selection combined with composite fuzzynearest neighbor classifiers for hyperspectral satellite imagery," Pattern Recognition Letters, vol.23, no.1–3, pp.183–190, 2002.
[7] J. Kennedy and R.C. Eberhart, Swarm Intelligence, Morgan Kaufmann Publishers, San Francisco, 2001.
[8] D.K. Agrafiotis and W. Cedeño, "Feature selection for structure-activity correlation using binary particle swarms," Journal of Medicinal Chemistry, vol.45, pp.1098–1107, 2002.
[9] H.A. Firpi and E. Goodman, "Swarmed feature selection," Proc. 33rd Applied Imagery Pattern Recognition Workshop, pp.112–118, 2004.
[10] Y. Liu, Z. Qin, Z. Xu, and X. He, "Feature selection with particle swarms," Proc. Computational and Information Science, pp.425–430, 2004.
[11] H. Liu and H. Motoda, "Feature transformation and subset selection," IEEE Intelligent Systems, Special Issue on Feature Transformation and Subset Selection, vol.13, no.2, pp.26–28, 1998.
[12] J. Yang and V. Honavar, "Feature subset selection using a genetic algorithm," IEEE Intelligent Systems, vol.13, no.2, pp.44–49, 1998.
[13] R. Kohavi and G.H. John, "Wrappers for feature subset selection," Artificial Intelligence, vol.97, no.1–2, pp.273–324, 1997.
[14] Y. Shi and R.C. Eberhart, "A modified particle swarm optimizer," Proc. IEEE Congress on Evolutionary Computation, pp.69–73, 1998.
[15] W. Siedlecki and J. Sklansky, "A note on genetic algorithms for large-scale feature selection," Pattern Recognition Letters, vol.10, no.5, pp.335–347, 1989.
[16] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, New York, USA, 1995.
[17] B. Widrow and R. Winter, "Neural nets for adaptive filtering and adaptive pattern recognition," IEEE Computer, vol.21, no.3, pp.25–39, 1988.
[18] S. Haykin, Neural Networks: A Comprehensive Foundation, second ed., Prentice-Hall, Englewood Cliffs, 1999.
[19] M.T. Hagan and M.B. Menhaj, "Training feedforward networks with the marquardt algorithm," IEEE Transactions on Neural Networks, vol.5, no.6, pp.989–993, 1994.
[20] Spectral Imaging Ltd., ImSpector Imaging Spectrograph User Manual, 2.21 ed., 2003.
[21] S.T. Monteiro, Y. Minekawa, Y. Kosugi, T. Akazawa, and K. Oda, "Prediction of sweetness and nitrogen content in soybean crops from high resolution hyperspectral imagery," Proc. 2006 IEEE International Geoscience and Remote Sensing Symposium, Denver, Colorado, pp.2263–2266, 2006.
[22] S.T. Monteiro, K. Uto, Y. Kosugi, N. Kobayashi, E. Watanabe, and K. Kameyama, "Feature extraction of hyperspectral data for under spilled blood visualization using particle swarm optimization," International Journal of Bioelectromagnetism, vol.7, no.1, pp.232–235, 2005.
[23] I.T. Jolliffe, Principal Component Analysis, Springer-Verlag, New York, 1988.
[24] I.C. Trelea, "The particle swarm optimization algorithm: convergence analysis and parameter selection," Information Processing Letters, vol.85, pp.317–325, 2003.
[25] J. Kennedy and R.C. Eberhart, "Particle swarm optimization," Proc. 4th IEEE Intl. Conference on Neural Networks, Perth, pp.1942–1948, 1995.
[26] I. Filella and J. Penuelas, "The red edge position and shape as indicators of plant chlorophyll content, biomass and hydric status," International Journal of Remote Sensing, vol.15, no.7, pp.1459–1470, 1994.

Sildomar Takahashi Monteiro was born in Manaus, Brazil, in 1977. He received the B.S. degree in electrical engineering from the Federal University of Amazonas, Manaus, Brazil, in 1999, the M.Sc. degree in electronic engineering and computer science from the Technological Institute of Aeronautics, São José dos Campos, Brazil, in 2002, and the Ph.D. degree in mechano-micro engineering from the Tokyo Institute of Technology, Tokyo, Japan, in 2007. He is currently a Postdoctoral Researcher at the Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology. His research interests include computational intelligence algorithms, especially particle swarm optimization, neural networks and reinforcement learning, with applications in remote sensing, biomedical imaging, and mobile robotics.

Yukio Kosugi was born on June 21, 1947, in Kanagawa Pref., Japan. He received the B.E. degree from Shizuoka University in 1970, and the M.S. and Dr. Eng. degrees from Tokyo Institute of Technology in 1972 and 1975, respectively, both in electronics. From 1975 to 1985 he was with the Research Laboratory of Precision Machinery and Electronics, at Tokyo Institute of Technology. During 1985 to 1999, he served as an associate professor at the Interdisciplinary Graduate School of Science and Engineering, of the same institute. During fiscal years 1998-2000, he conducted the project of "R&D of Brain-like Information Systems" at the Frontier Collaborative Research Center, Tokyo Institute of Technology. He is currently a Professor at the Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology and also serving as a visiting professor at Tsukuba Advanced Research Alliance of Tsukuba University. His research interests include information processing in the nervous system and neural network-aided image processing in medical and remote sensing fields. Dr. Kosugi is a member of IEEE and The Institute of Electronics, Japan Society of Photogrammetry and Remote Sensing.