Applied Soft Computing 9 (2009) 1008–1015

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

PSOLDA: A particle swarm optimization approach for enhancing classification accuracy rate of linear discriminant analysis

Shih-Wei Lin [a,*], Shih-Chieh Chen [b]

[a] Department of Information Management, Chang Gung University, No. 259, Wen-Hwa 1st Road, Taoyuan 333, Taiwan, ROC
[b] Department of Industrial Management, National Taiwan University of Science and Technology, No. 43, Keelung Road, Sec. 4, Taipei, Taiwan, ROC

ARTICLE INFO

Article history:
Received 7 December 2007
Received in revised form 8 January 2009
Accepted 18 January 2009
Available online 31 January 2009

Keywords:
Linear discriminant analysis
Feature selection
Particle swarm optimization

ABSTRACT

Linear discriminant analysis (LDA) is a commonly used classification method. It can provide important weight information for constructing a classification model. However, real-world data sets generally have many features, not all of which benefit the classification results. If a feature selection algorithm is not employed, unsatisfactory classification will result, due to the high correlation between features and noise. This study points out that the feature selection has influence on the LDA by showing an example. The methods traditionally used for LDA to determine the beneficial feature subset are not easy or cannot guarantee the best results when problems have larger number of features.

The particle swarm optimization (PSO) is a powerful meta-heuristic technique in the artificial intelligence field; therefore, this study proposed a PSO-based approach, called PSOLDA, to specify the beneficial features and to enhance the classification accuracy rate of LDA. To measure the performance of PSOLDA, many public datasets are employed to measure the classification accuracy rate. Comparing the optimal result obtained by the exhaustive enumeration, the PSOLDA approach can obtain the same optimal result. Due to much time required for exhaustive enumeration when problems have larger number of features, exhaustive enumeration cannot be applied. Therefore, many heuristic approaches, such as forward feature selection, backward feature selection, and PCA-based feature selection are used. This study showed that the classification accuracy rates of the PSOLDA were higher than those of these approaches in many public data sets.

1. Introduction

Various methods have been adopted in classification problems. Linear discriminant analysis (LDA) is a popular classification method applied to a variety of fields. For example, diagnosis of heart valve diseases [1], face recognition [40], classification of adolescent psychotic disorders [26], identification of citrus disease [30], electricity loads [25], digital image recognition [31], and emotional speech recognition [7]. LDA is a supervised algorithm that searches for a discriminative subspace, in which patterns belonging to the same class are as grouped as tightly as possible, while patterns belonging to the other classes are more widely separated. The common purposes of LDA are as follows: (1) to examine differences between groups; (2) to distinguish effectively among groups; (3) to identify significant discriminating variables/features; (4) to perform hypothesis testing on the differences among the expected groupings, and (5) to classify new observations into pre-existing groups [10].

Most pattern classification problems involving a large set of potential features must identify a small subset for features to be employed for classification, an act known as feature selection. The data without feature selection may be redundant or noisy, and may degrade the accuracy rate of classification. The main advantages of feature selection are as follows: (1) lowering computational cost and storage requirements, (2) minimizing the degradation of classification accuracy rate because of the finite nature of training sample sets, (3) decreasing training and prediction time and, (4) facilitating understanding and visualization of data [23].

Finding an optimal subset of features in feature selection is inherently combinatory, since the usefulness of each feature needs to be determined. Hence, feature selection is an optimization problem. An optimal approach is necessary to measure all possible subsets. Many researchers have adopted traditional statistical methods for feature selection over past decades, such as forward feature selection, backward feature selection, PCA-based feature selection. Principal component analysis (PCA) is the most widely adopted traditional statistical method [6,8,22]. Features selected using PCA are proved to be variable-independent but may not be the most beneficial for a specific problem. This study proposes a particle swarm optimization approach that identifies the beneficial

* Corresponding author.
E-mail address: swlin@mail.cgu.edu.tw (S.-W. Lin).

1568-4946/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2009.01.001

subsets of features for different problems in order to maximize the classification accuracy rate of LDA.

The remainder of this study is structured as follows. Section 2 reviews previous work on discriminant analysis, principal component analysis, and feature selection. Section 3 elaborates on the proposed PSOLDA approach to identify the beneficial subset of features for LDA. Section 4 presents the experiment results. Conclusions and suggestions for future research are discussed in the final section.

2. Literature review

2.1. Linear discriminant analysis

Linear discriminant analysis is a multivariate statistical method that is commonly utilized to construct a predictive/descriptive group discrimination model according to observed predictor variables. It is a technique for classifying a set of observations into predefined classes. LDA employs multiple attributes/features to distinguish each classification variable. LDA is different from cluster analysis since it requires prior knowledge of the classes, generally in the form of a sample from each class [2,9,10].

The model is constructed according to a set of observations for which the classes are known in advance. This set of observations is called the training set. A set of linear functions of the predictors, known as discriminant functions, is built from the training set. Eq. (1) depicts an example of such a linear function.

$$d_i = w_{i1}x_1 + w_{i2}x_2 + \cdots + w_{ij}x_j + b_i \quad i = 1, \ldots, g \tag{1}$$

In Eq. (1), w_{ij} denotes discriminant coefficients, where x_j represents the input variables or features, b_i indicates a constant for discriminant function i. These discriminant functions are utilized to predict the class of a new observation with unknown class. g discriminant functions are constructed in a g-class problem. All of the g discriminant functions are measured for each new observation. The new observation is then assigned to the class i with the highest-value discriminant function [2,9]. Detailed procedures for constructing the discriminant functions can be found in [17].

In order to illustrate the above concept, an example shown in Table 1 is used. This demonstrative example has 12 instances, and four variables, x_1, x_2, x_3 and x_4, can be used to classify its class. There are three classes, labeled 1, 2, and 3.

If all of four variables (feature selection is not applied) are used to build the classification model, three discriminant functions can be obtained as follows:

d1: $13.884323x_1 + 2.317973x_2 + 11.975050x_3 + 6.546200x_4 - 12.282051$

d2: $4.375346x_1 + 5.463215x_2 + 1.957248x_3 + 0.028840x_4 - 2.639310$

d3: $7.727297x_1 + 7.652076x_2 - 0.052000x_3 - 0.633000x_4 - 4.214748$

According to these discriminant functions, the classification accuracy rate is 66.67% (8/12). Detailed results of the classification process are shown in Table 2.

If certain feature selection method is performed, only three variables, x_1, x_3 and x_4, are necessary to construct the classification model. Then three discriminant functions can be obtained as follows:

d1: $13.716540x_1 + 13.522430x_3 + 7.650988x_4 - 12.128247$

d2: $3.979899x_1 + 5.604261x_3 + 3.632706x_4 - 1.784938$

d3: $7.173414x_1 + 5.056205x_3 + 3.014117x_4 - 2.538613$

Using these discriminant functions, the classification accuracy rate is 83.33% (10/12). Detailed results of the classification process are shown in Table 3.

In addition we test all possible combinations of the selected features (in this case the number of possibilities is $2^4 - 1 = 15$). The results of this test are displayed in Table 4.

This example shows that feature selection can help increase the classification accuracy rate for LDA. Clearly, the number of possible solutions is $2^D - 1$, where D is the number of total features in a given dataset. For example, if the number of features for a dataset is 20, the number of possible solutions is $2^{20} - 1 = 1{,}048{,}575$. Obtaining the optimal subset of features for problems with a larger number features by the exhaustive enumeration requires greater computational costs. That is, trying all possible combinations and choosing the best one is not a realistic possibility if the number of features is large.

LDA often suffers from the small sample size problem when the number of dimensions of the data is much greater than the number of data points. A number of effective approaches to this problem have been proposed, including regularized LDA, PCA + LDA [27], pseudo-inverse LDA, orthogonal LDA [18], LDA/GSVD [24], and LDA/QR [19]. The PCA + LDA method [27], one of the most popular methods, applies PCA to reduce the number of dimensions of the data before performing LDA. Li et al. [38] showed that discriminant analysis is well-known approach to learn the discriminative feature transformations in the statistical pattern recognition literature, providing a fast, efficient solution. Liang et al. [41] pointed out that the two-dimensional linear discriminant analyses (2DLDA) have advantages in handling the singularity problem and in the computational costs. It has been empirically shown that 2DLDA is not stronger than LDA under the same dimensionality [41]. On the other hand, they found the matrix-based methods are not always better than vector-based methods for small sample size problems.

Table 1
Data values of example.

Instance	x_1	x_2	x_3	x_4	Class label
1	0.904762	0.666667	0.142857	1.000000	1
2	0.095238	0.714286	0.214286	0.000000	3
3	0.523810	0.000000	0.071429	0.000000	2
4	0.000000	1.000000	0.214286	1.000000	2
5	0.714286	0.952381	0.214286	1.000000	1
6	0.809524	0.619048	0.642857	1.000000	1
7	0.285714	0.190476	0.071429	0.000000	3
8	0.476190	0.904762	0.214286	1.000000	3
9	1.000000	0.809524	0.285714	0.000000	3
10	0.238095	0.904762	0.000000	1.000000	2
11	1.000000	0.809524	1.000000	0.000000	1
12	0.095238	0.666667	0.357143	0.000000	2

2.2. Principal component analysis

Principal component analysis (PCA) [12] is a well-known data processing and dimension reduction method (feature selection) for re-expressing multivariate data. It is a statistical method that is primarily used to transform the input space into a new lower dimensional space. When the size of the data set is unwieldy, principal components may be useful in reducing its dimensionality [13]. It permits researchers to reorient the data so that the first few dimensions account for as much of the information as possible. If substantial redundancy exists in the data set, then it can be possible to represent most of the original data set with a relatively small number of dimensions [35]. PCA has the advantage that each component is uncorrelated with any others

Table 2
Classification result of example using four features to construct classification model.

Instance	Discriminant function 1	Discriminant function 2	Discriminant function 3	Predict class	Real class	Correct?
1	10.082193	6.269929	7.237576	1	1	Y
2	−6.737954	2.099098	1.975812	2	3	N
3	−4.153937	−0.207656	−0.170827	3	2	N
4	−0.851792	4.272156	2.793185	2	2	Y
5	8.955206	7.137252	7.948302	1	1	Y
6	14.637024	6.571702	6.111259	1	1	Y
7	−7.018221	−0.208797	−0.553129	2	3	N
8	5.539025	5.835347	5.744079	2	3	N
9	6.900167	6.717853	9.692231	3	3	Y
10	−0.332849	4.374188	3.915390	2	2	Y
11	15.453777	8.115889	9.655089	1	1	Y
12	−5.137614	2.118552	1.603999	2	2	Y

Table 3
Classification result of example using four variables to construct classification model.

Instance	Discriminant function 1	Discriminant function 2	Discriminant function 3	Predict class	Real class	Correct?
1	9.864719	6.249238	7.688051	1	1	Y
2	−7.924243	−0.204985	−0.771957	2	3	N
3	−3.977491	0.700081	1.580053	3	2	N
4	−1.579592	3.048683	1.558978	2	2	Y
5	8.217941	5.891470	6.682847	1	1	Y
6	15.319598	8.672332	9.532971	1	1	Y
7	−7.243345	−0.247518	−0.127908	3	3	Y
8	4.952088	4.943871	4.974886	3	3	Y
9	5.451841	3.796178	6.079430	3	3	Y
10	−1.211419	2.795363	2.183458	2	2	Y
11	15.110724	7.799224	9.691006	1	1	Y
12	−5.992469	0.595623	−0.049643	2	2	Y

component, eliminating multicollinearity when using the results in an analysis of dependence (e.g., discriminant analysis) [17].

In the general principal components problem, the objective is to obtain a linear combination of the original variables $[X = x_1, x_2, \ldots, x_p]$ with maximum variance. If we assume that X is standardized (i.e., each variable is normalized to zero mean and unit variance), then the linear combination can be denoted by the vector $(u = u_1, u_2, \ldots, u_p)'$, and the goal is to choose u to maximize the variance of the elements of $z = Xu$, which may be written as follows: $\mathrm{var}(z) = (1/n-1)u'X'Xu$, where n is the number of the data. Due to X being standardized, the term $(1/n-1)X'X$ is identical to the sample correlation matrix R. The variance of the elements of z can be written as: $\mathrm{var}(z) = u'Ru$. Because we can choose the components of u (the length of the vector) to be arbitrarily large, a constraint of unit length on the vector is imposed, $u'u = 1$. The constrained optimization problem can be solved by forming the Lagrangian, taking the first-derivative, setting it equal to zero, and solving. The Lagrangian is given by $L = u'Ru - \lambda(u'u - 1)$ where λ is called the Lagrange multiplier. The λ can be chosen so as to penalize the objective function if the equality constraint $(u'u = 1)$ is not met. The derivative of L with respect to the elements of u yields $\partial L/\partial u = 2Ru - 2\lambda u$. Setting $\partial L/\partial u$ equal to zero, and solving, $Ru = \lambda u$. The vector u is called an eigenvector and the scalar λ is called an eigenvalue. Provided the matrix R is of full rank, then the solution will consist of p positive eigenvalues and associated eigenvectors.

Table 4
Classification result for all possible combination of selected features for example with four features.

Used features	Classification accuracy rate	Used features	Classification accuracy rate
X_1	66.67%	X_2, X_4	50.00%
X_2	25.00%	X_3, X_4	58.33%
X_3	58.33%	X_1, X_2, X_3	66.67%
X_4	50.00%	X_1, X_2, X_4	58.33%
X_1, X_2	66.67%	X_1, X_3, X_4	83.33%
X_1, X_3	58.33%	X_2, X_3, X_4	66.67%
X_1, X_4	66.67%	X_1, X_2, X_3, X_4	66.67%
X_2, X_3	58.33%		

The principal components (PCs) are ordered in descending order of variance, with PC_1 showing the highest variance, and PC_p showing the lowest variance In other words, $\mathrm{var}(PC_1) \geq \mathrm{var}(PC_2) \geq \mathrm{var}(PC_3) \geq \ldots \geq \mathrm{var}(PC_p)$, where $\mathrm{var}(PC_i)$ represents the variance of PC_i in the data set. The eigenvalues of most of the PCs in PCA should ideally be low enough to be virtually negligible. In this case, the variation in the data set can be adequately described using those PCs whose eigenvalues are not negligible. Accordingly, some degree of economy is achieved, because the variation in the original number of variables (X variables) can be described by a smaller number of new variables (PCs) [4].

To apply the PCA researcher must decide how many principal components to retain for subsequent analysis, trading off simplicity (i.e., a small number of dimensions is easier to manage) against completeness. One possible solution is Bartlett's test. If sphericity is rejected, then the largest principal component is extracted and then the residual correction matrix is tested to see if its determinant is different from zero. Iteratively continue extracting components until the residual matrix is not statistically significant. There are other three common methods to determine how many components should be retained: scree plot, Kaiser's rule, and Horrn's procedure [17]. Though the above methods provide different ways to determine how many principal components (features) should be retained, it remains a critical issue. If too few components are retained, the model will not obtain all of the information in the data, leading to a poor result. On the other hand, if too many components are chosen, then the model will include noise [34]. This type of trial-and-error method may not provide the optimal result [33].

2.3. Feature selection

The purpose of feature selection is to identify a transformation from the original high-dimensional space to a low-dimensional space, which retains as much information as possible that is valuable for classification accuracy. Feature selection approaches can be categorized into two models, filter models and wrapper models [11]. Statistical techniques, including principal component analysis, factor analysis, and independent component analysis can be applied in filter-based feature selection approaches to investigate indirect performance measures, which are mostly based on distance and information. Even though the filter model is fast, the resulting feature subset may not be optimal [11].

PCA-based analysis methods have been applied to perform selection without loss of accuracy. Some such methods even improve classification accuracy. Ravi et al. [39] presented pattern classification with principal component analysis feature selection, combined with the fuzzy rule technique, to extract features. Rocchi et al. [22] proposed a feature selection process based on principal component analysis to discover distinct features of postural's way in Parkinson's disease. Garcia-Cuesta et al. [8] proposed a ground-based remote sensing temperature retrieval system based on principal component analysis feature selection, using the multi-layer perceptron technique to extract features.

The wrapper model [29] employs the classifier accuracy rate as the performance measure. Some studies have concluded that if the objective of the model is to minimize the classifier error rate, and the measurement cost for all the features is equal, then the predictive accuracy of the classifier is the most important factor. In other words, the classifier should be constructed to maximize the classification accuracy rate. The features selected by the classifier are then chosen as the optimal features. The wrapper model generally applies meta-heuristic approaches to help search for the best feature subset. Although meta-heuristic approaches are slow, they may obtain the (near) best feature subset. For example, Lin et al. [37] proposed an SA-based approach for parameter determination of support vector machine and feature selection, and showed that the classification accuracy rate is improved by feature selection at the expense of computational time.

Chiang and Pell [21] presented genetic algorithms combined with discriminant analysis to identify key variables. Their analytical results demonstrate that key variables can be identified correctly. Pacheco et al. [16] proposed several meta-heuristics, including tabu search and variable neighborhood search, to choose variables that are subsequently adopted in discriminant analysis. Their approaches achieved a high success rate with small samples. However, since their approaches were only adopted in specific non-public datasets, further comparison cannot be performed.

3. The proposed approach

Particle swarm optimization [5,15] is an emerging population-based meta-heuristic that simulates social behaviors, including birds flocking to a promising position, in order to accomplish precise goals in a multidimensional space. Like evolutionary algorithms, PSO performs searches using a population (called a swarm) of individuals (called particles) that are updated between iterations. To determine the optimal solution, each particle modifies its search direction based on two factors, its own best previous experience and the best experience of all other members. Shi and Eberhart [28] term the former the cognition component, and the later the social component.

Each particle represents a candidate position (i.e., solution). A particle i is treated as a point in a D-dimension space, and its status is characterized according to its position x_{id} and velocity v_{id}. Fig. 1 illustrates the above concept of modulation of searching points

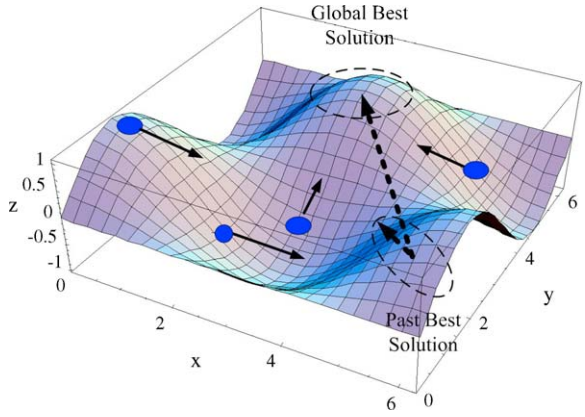

Fig. 1. Search concept of particle swarm optimization.

when attempting to find a solution for maximizing a function $f(z) = \cos(x)\sin(y)$, where $0 \le x \le 2\pi$ and $0 \le y \le 2\pi$.

Let $p_i^t = \{p_{i1}^t, p_{i2}^t, \ldots, p_{iD}^t\}$ denote the best solution that particle i has identified at iteration t, and $p_g^t = \{p_{g1}^t, p_{g2}^t, \ldots, p_{gD}^t\}$ represent the best solution obtained from p_i^t in the population at iteration t. To search for the optimal solution, each particle modifies its velocity according to the cognition and social components as follows:

$$V_{id}^{t+1} = wV_{id}^t + c_1 \cdot rand_1 \cdot (p_{id}^t - x_{id}^t) + c_2 \cdot rand_2 \cdot (p_{gd}^t - x_{id}^t) \quad d = 1, 2, \ldots, D \tag{2}$$

where c_1 represents the cognition learning factor; c_2 denotes the social learning factor, w is inertial weight, $rand_1$ and $rand_2$ indicate random numbers (one value for one dimension) uniformly distributed in $U(0,1)$. Each particle then moves to a new potential solution according to the following equation:

$$X_{id}^{t+1} = X_{id}^t + V_{id}^{t+1} \quad d = 1, 2, \ldots, D \tag{3}$$

It is possible to clamp the velocity vectors by specifying upper and lower bounds on $v_{\max}$ to prevent the particles in the search space from moving too rapidly.

The basic process of the PSO algorithm can be presented as follows.

Step 1: (Initialization) Randomly create initial particles.
Step 2: (Fitness) Measure the fitness of each particle in the population.
Step 3: (Update) Calculate the velocity of each particle using Eq. (2).
Step 4: (Construction) For each particle, move to the next position according to Eq. (3).
Step 5: (Termination) Stop the algorithm if the termination criterion is satisfied; return to Step 2 otherwise.

This study applies PSO to LDA, denoted as PSOLDA, for feature selection in LDA. The following subsections discuss the solution representation, the flowchart and system architecture of PSOLDA.

3.1. Solution representation

The solution representation is shown in Fig. 2. If the data set involves D features, then D variables must be adopted. Each variable is in the range from 0 to 1. That is, the location vectors are limited between a lower bound and an upper bound. If the value of a variable is less than or equal to 0.5, then its corresponding feature

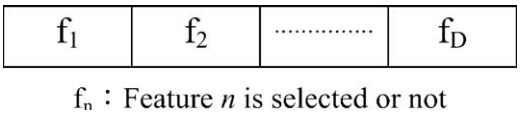

Fig. 2. Solution representation of PSOLDA.

is not selected. Conversely, if the value of a variable is greater than 0.5, then its corresponding feature is selected.

3.2. Flowchart of PSOLDA

Fig. 3 displays a flowchart of PSOLDA. The population of particles is initialized, each particle having a random position within the D-dimensional space and a random velocity for each dimension, where D represents the number of features. Each particle's fitness for the LDA is then evaluated. The higher the classification accuracy rate it is, the higher the fitness that particle has. If the fitness of the ith particle is better than the particle's best fitness, then the position vector is saved for the particle best (p_i). If one of the particle's fitness is better than the global best fitness, then the position vector is saved for the global best (p_g). Finally, the particle's velocity and position are updated until the termination condition is satisfied.

Two termination conditions are used in this study. If the number of iterations reaches the pre-determined maximum number of iterations $I_{\max}$, or p_g is not improved during a maximum allowable number of iterations $I_{\text{non-improving}}$, then PSOLDA is terminated.

In this study, the classification accuracy rates for the datasets were measured by comparing the predicted class and the actual class. For example, in the classification problem with two-classes, positive and negative, a single prediction has four different possible. The true positive (TP) and true negative (TN) are correct classifications. A false positive (FP) occurs when the outcome is incorrectly predicted as positive when it is actually negative. A false negative (FN) occurs when the outcome is incorrectly predicted as negative when it is actually positive. The overall classification accuracy rate is the number of correct classifications divided by the total number of classifications, computed as $(TP + TN)/(TP + TN + FP + FN)$.

In a multi-class prediction, the result on a test set is often displayed as a two-dimensional confusion matrix with a row and column for each class. Each matrix element shows the number of test cases for which the actual class is the row and the predicted class is the column.

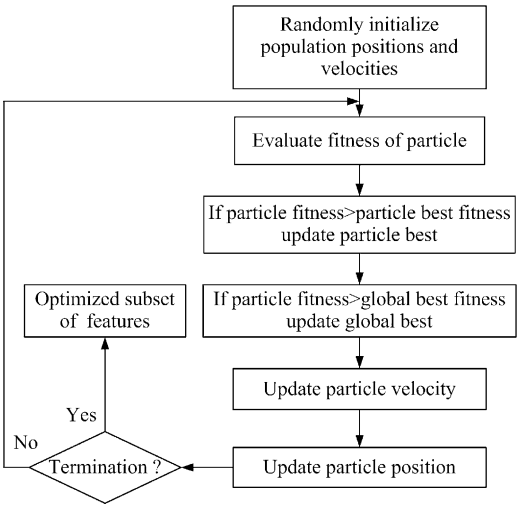

Fig. 3. The flowchart of PSOLDA.

3.3. Demonstration of PSOLDA procedure

To demonstrate the procedures of PSOLDA, the example in Table 1 is used. In the demonstration, w, c_1, and c_2, are set to 0.9, 0.8, and 1.2, respectively. At iteration $t-1$, the location of the global best is supposed to be (0.92, 0.86, 0.55, 0.65), which means all of four features are selected because the corresponding values of the variables are all greater than 0.5. The location of the particle's best of particle i is (0.23, 0.64, 0.87, 0.98), which means features 2, 3, and 4 are used. The current position of particle i is (0.05, 0.40, 0.70, 0.45), which means only the third feature is selected. The current velocity vector of particle i is assumed to (0.45, −0.37, 0.23, −0.20).

The velocity vector for the particle is updated using Eq. (2) as follows.

$$
\begin{aligned}
V_i^t = V_i^t &= \begin{bmatrix} V_{i1}^t \\ V_{i2}^t \\ V_{i3}^t \\ V_{i4}^t \end{bmatrix} \\
&= w \begin{bmatrix} V_{i1}^{t-1} \\ V_{i2}^{t-1} \\ V_{i3}^{t-1} \\ V_{i4}^{t-1} \end{bmatrix} + c_1 \cdot rand_1 \cdot \left(\begin{bmatrix} p_{x1} \\ p_{x2} \\ p_{x3} \\ p_{x4} \end{bmatrix} - \begin{bmatrix} x_{i1}^{t-1} \\ x_{i2}^{t-1} \\ x_{i3}^{t-1} \\ x_{i4}^{t-1} \end{bmatrix} \right) + c_2 \cdot rand_2 \\
&\quad \cdot \left(\begin{bmatrix} g_{x1} \\ g_{x2} \\ g_{x3} \\ g_{x4} \end{bmatrix} - \begin{bmatrix} x_{i1}^{t-1} \\ x_{i2}^{t-1} \\ x_{i3}^{t-1} \\ x_{i4}^{t-1} \end{bmatrix} \right) \\
&= 0.9 \cdot \begin{bmatrix} 0.45 \\ -0.37 \\ 0.23 \\ -0.20 \end{bmatrix} + 0.8 \cdot \begin{bmatrix} 0.61 \\ 0.51 \\ 0.32 \\ 0.54 \end{bmatrix}^T \cdot \left(\begin{bmatrix} 0.23 \\ 0.64 \\ 0.87 \\ 0.98 \end{bmatrix} - \begin{bmatrix} 0.05 \\ 0.40 \\ 0.70 \\ 0.45 \end{bmatrix} \right) + 1.2 \\
&\quad \cdot \begin{bmatrix} 0.24 \\ 0.17 \\ 0.93 \\ 0.42 \end{bmatrix}^T \cdot \left(\begin{bmatrix} 0.92 \\ 0.86 \\ 0.55 \\ 0.65 \end{bmatrix} - \begin{bmatrix} 0.05 \\ 0.40 \\ 0.70 \\ 0.45 \end{bmatrix} \right) \\
&= \begin{bmatrix} 0.74 \\ -0.14 \\ 0.08 \\ 0.15 \end{bmatrix}
\end{aligned}
$$

Then applying these new velocities to a new particle using Eq. (3), the new location of particle i can be obtained as follows.

$$
x^t = \begin{bmatrix} x_{i1}^t \\ x_{i2}^t \\ x_{i3}^t \\ x_{i4}^t \end{bmatrix} = \begin{bmatrix} x_{i1}^{t-1} \\ x_{i2}^{t-1} \\ x_{i3}^{t-1} \\ x_{i4}^{t-1} \end{bmatrix} + \begin{bmatrix} V_{i1}^t \\ V_{i2}^t \\ V_{i3}^t \\ V_{i4}^t \end{bmatrix} = \begin{bmatrix} 0.05 \\ 0.40 \\ 0.70 \\ 0.45 \end{bmatrix} + \begin{bmatrix} 0.74 \\ -0.14 \\ 0.08 \\ 0.15 \end{bmatrix} = \begin{bmatrix} 0.79 \\ 0.26 \\ 0.78 \\ 0.60 \end{bmatrix}
$$

Now the location of the particle is moved to (0.79, 0.26, 0.78, 0.60), meaning that the first, the third and fourth features are selected. The discriminant model is thus constructed using the training data, and the testing data is used to calculated the classification accuracy rates, which is the fitness value of particle i. In each iteration, the same procedure is applied for all particles. If the fitness of the ith particle is better than that particle's best fitness, then the position vector is saved for the particle best (p_i). If one of the particle's fitness is better than the global best fitness, then the position vector is saved for the global best (p_g). The procedure is iterated until termination conditions are met.

3.4. System architecture of PSOLDA

The PSOLDA feature selection system is constructed through the following steps:

(1) *Data preprocessing*: Normalization is applied to prevent feature values in greater numeric ranges from dominating those in

smaller numeric ranges, as well as to prevent numerical difficulties during calculation. The range of each feature value can generally be linearly scaled to the range [0,1] using Eq. (4), where a_i' denotes the scaled value; where a_i is the actual value of attribute i, $\max(a_i)$ represents the maximum value of the feature i in the dataset, and $\min(a_i)$ denotes the minimum value of the feature i in the dataset. If an instance lacks the values of some features (i.e., an instance with a missing value), then it was removed [6].

$$a_i' = \left(\frac{a_i - \min(a_i)}{\max(a_i) - \min(a_i)}\right) \quad (4)$$

(2) *Feature subset selection*: Each particle in the PSO algorithm represents a solution, which denotes the selected subset of features. The selected features and the training dataset are used in building the LDA classifier model.
(3) *Fitness evaluation*: After applying the training dataset to train the discriminant functions, the testing dataset is used to compute the classification accuracy rate. Each particle's fitness is compared with the particle's best fitness and the global best fitness when the classification accuracy rate is obtained. If the ith particle's fitness is better than its best previous experience, its best previous experience is updated accordingly. Furthermore, if the particle's fitness is better than the global best fitness, the global best fitness is also updated.
(4) *Termination criteria*: If the termination criteria are satisfied, then the process ends; otherwise, the next iteration is run. There are two commonly used termination criteria, the maximal number of iterations $I_{\max}$, and allowable number of iterations $I_{\text{non-improving}}$ where the global best is not improved during the PSO procedure. Thus, if the number of iterations reaches $I_{\max}$ or the global best is not improved in $I_{\text{non-improving}}$ successive iterations, the procedure terminates.
(5) *PSO process*: In this step, the system obtains other solutions for particles and then goes to Step (2).

4. Experiment results

The proposed PSOLDA approach was implemented in C language and run on a Windows XP operating system, on a PC equipped with a Pentium IV-3.0 GHz CPU and 512 MB RAM. The proposed PSOLDA approach was evaluated using 23 datasets in the UCI Machine Learning Repository [32]. Table 5 lists the numbers of features, instances, and classes for each UCI dataset used in this study.

The k-fold cross-validation approach proposed by Salzberg [36] was employed in the experiments, with $k = 10$. The dataset was thus split into 10 portions, with each part of the data sharing the same proportion of each class of data. Nine data portions were used in the training process, while the remaining part was used in the testing process [14]. The proposed PSOLDA approach was run 10 times to allow each slice of data to take a turn as the testing data. The classification accuracy rate is calculated by summing the individual accuracy rates for each run of testing, and then dividing the total by 10. Because the numbers of data in each class were not multiples of 10, the dataset could not be partitioned fairly. However, the ratio of the number of data in the training set to the number of data in the testing set was maintained as closely as possible to 9:1.

In order to obtain better parameter values for the PSOLDA, the initial experiment was performed as follows. All datasets are used to test various combinations of parameters. At the beginning, the maximum number of solutions evaluated is set to 50,000 (a large value), while w, c_1, c_2, and p_{size} are set to 0.8, 1.0, 1.0, and 8, respectively. That is, the number of generations is 2500 (50,000/20 = 2500). After several runs, we found that the classification accuracy rates converged at 400 iterations.

Table 5
Dataset from the UCI repository.

Dataset	Number of classes	Number of instances	Number of features
Australian	2	653	15
Bioinformatics	3	391	20
Boston housing	2	1012	13
Breast cancer	2	683	10
Bupa live	2	345	6
Car Evaluation	4	1428	6
Cleveland heart	2	296	13
Dermatology	6	358	34
Ecoli	8	336	7
German	2	1000	30
Glass	6	214	9
Ionosphere structure	2	351	34
Iris	3	150	4
Page	5	5473	10
Pima Indians diabetes	2	768	8
Segment	7	1848	18
Sonar	2	208	60
Teaching Assistant Evaluation	3	151	5
Vehicle	4	846	18
Vowel	11	528	10
Waveform (version 1)	3	5000	21
Wine	3	175	13
Yeast	10	1484	8

After determining the maximum number of generations, the following combinations of parameters were tested.

c_1 = 0.5, 0.8, 1.0, 1.2, 1.5;
c_2 = 0.5, 0.8, 1.0, 1.2, 1.5;
p_{size} = 8, 10, 12, 15, 20;
$w = 0.7, 0.8, 0.9, 1.0$;
$I_{\text{non-improving}}$ = 100, 150, 200.

Setting $c_1 = 0.8$, $c_2 = 1.2$, $w = 0.9$, $p_{size} = 15$, $I_{\text{non-improving}} = 150$, and $I_{\max} = 400$ seemed to give better results; therefore they were used for further computational study. Since the proposed PSOLDA approach is non-deterministic, different runs with the same data may not produce the same solution. Therefore, the proposed PSOLDA approach was executed five times for 10-fold cross-validation in each dataset to calculate the classification accuracy rate.

To verify the proposed approach, the best result obtained by the PSOLDA approach among the five runs is compared with those obtained by LDA without feature selection, LDA with forward feature selection, LDA with backward feature selection, LDA with PCA-based feature selection, and LDA with feature selection by the exhaustive enumeration. As shown in Table 6, the proposed PSOLDA approach performs well in all datasets and the computational time is within an acceptable range. Comparison of the PSOLDA approach and the LDA with feature selection by exhaustive enumeration (which can obtain the optimal solution by selecting the best solution among all solutions) shows that the results obtained by the former are all equal to those of the latter. That is, the proposed PSOLDA approach can obtain the optimal solution when the number of features in the dataset is small. However, because the exhaustive enumeration is time consuming, it is not easily applied to datasets with larger numbers of features. Furthermore, the proposed PSOLDA approach outperforms the LDA without feature selection, LDA with forward feature selection, LDA with backward feature selection, and LDA with PCA-based feature selection for the most of datasets.

Moreover, the results yielded by the proposed PSOLDA approach were compared with LDA and those of Breiman [3,20] who adopted several datasets from UCI [32]. Breiman tested

Table 6
Classification accuracy rates obtained by LDA without feature selection, LDA with forward feature selection, LDA with backward feature selection, LDA with PCA-based feature selection, LDA feature selection by exhaustive enumeration and PSOLDA.

Dataset	LDA without feature selection	LDA with forward feature selection	LDA with backward feature selection	LDA with PCA-based feature selection	LDA with feature selection by exhaustive enumeration	PSOLDA	Time (s)
Australian	83.0%	80.4%	84.2%	82.1%	84.5%	84.5%	19.09
Bioinformatics	80.4%	79.3%	81.6%	80.4%	–	84.4%	18.23
Boston housing	83.8%	82.8%	84.3%	83.8%	85.2%	85.2%	20.10
Breast cancer	96.1%	95.4%	95.8%	91.7%	96.5%	96.5%	10.04
Bupa live	63.5%	61.1%	64.3%	60.5%	65.2%	65.2%	3.48
Car evaluation	79.3%	71.2%	78.0%	81.8%	78.1%	78.1%	29.00
Cleveland heart	74.2%	78.9%	83.3%	77.6%	84.7%	84.7%	7.29
Dermatology	81.6%	96.5%	97.0%	93.3%	–	98.4%	37.09
Ecoli	42.5%	79.7%	79.7%	72.3%	80.1%	80.1%	6.00
German	74.0%	68.7%	74.2%	73.2%	–	75.6%	78.54
Glass	57.8%	57.8%	62.0%	61.0%	64.8%	64.8%	4.82
Ionosphere structure	86.5%	85.3%	90.9%	86.9%	–	92.2%	27.37
Iris	98.0%	96.3%	93.7%	90.0%	97.0%	97.0%	1.37
Page	85.2%	91.3%	91.6%	80.3%	–	91.8%	251.74
Pima Indians diabetes	76.4%	74.8%	76.5%	75.9%	76.7%	76.7%	10.42
Segment	88.6%	85.6%	89.0%	87.1%	–	89.2%	140.91
Sonar	72.5%	76.1%	85.5%	58.7%	–	90.5%	36.51
Teaching assistant evaluation	52.1%	51.4%	52.1%	51.6%	52.5%	52.5%	1.60
Vehicle	77.5%	75.0%	79.0%	77.3%	–	79.4%	45.35
Vowel	63.1%	56.2%	64.2%	49.6%	65.1%	65.1%	22.90
Waveform (version 1)	84.9%	85.7%	80.1%	71.2%	–	86.1%	468.06
Wine	98.8%	96.6%	99.9%	97.2%	100.0%	100.0%	5.61
Yeast	51.1%	35.0%	51.5%	44.6%	51.9%	51.9%	56.67

–, the computation time is over 600 s.

datasets using Bagging and Adaboost without feature selection. Table 7 offers a comparison of Breiman's results with those of PSOLDA. Four of the average accuracy rates of the proposed PSOLDA approach exceeded those obtained by Breiman. That is, the proposed PSOLDA approach achieved the highest classification accuracy rate across different datasets.

The number of the features selected by forward feature selection, backward feature selection, PCA-based feature selection, and PSOLDA are shown in Table 8. Compared with the LDA with forward feature selection, LDA with backward feature selection, and LDA with PCA-based feature selection, the number of selected features obtained by PSOLDA is more appropriate (based on the classification accuracy rates) than those of other feature selection methods. Thus, the PSOLDA approach found the better beneficial subset of features. Analytical results demonstrate that the feature selection did not select all features for use in the LDA classification model. Furthermore, feature selection increased the classification accuracy rates for LDA.

Table 7
Classification accuracy rates obtained by LDA, BG-DA, AB-DA, and PSOLDA approaches.

Dataset	LDA	BG-DA	AB-DA	PSOLDA
Breast cancer	96.1%	96.1%	96.2%	96.5%
Cleveland heart	74.2%	74.2%	73.4%	84.7%
Glass	57.8%	58.5%	59.4%	64.8%
Pima Indians diabetes	76.4%	76.5%	76.1%	76.7%

Table 8
Number of the selected features obtained by LDA with forward feature selection, LDA with backward feature selection, LDA with PCA-based feature selection and PSOLDA.

Dataset	No. of original features	LDA with forward feature selection	LDA with backward feature selection	LDA with PCA-based feature selection	PSOLDA
Australian	15	5.6	13.0	13.2	11.4
Bioinformatics	20	9.5	18.2	19.5	15.7
Boston housing	13	5.1	9.9	13.9	7.7
Breast cancer	10	6.0	6.7	9.7	6.6
Bupa live	6	3.6	4.7	5.6	4.6
Car evaluation	6	4.9	5.6	5.3	5.4
Cleveland heart	13	6.1	11.7	12.7	9.5
Dermatology	34	17.7	26.9	28.4	22.3
Ecoli	7	5.5	5.6	6.2	5.6
German	30	2.2	27.3	26.5	22.4
Glass	9	5.5	7.5	8.1	6.8
Ionosphere structure	34	4.8	30.4	30.3	21.7
Iris	4	2.3	3.9	3.3	3.6
Page	10	6.0	7.1	9.8	6.6
Pima Indians diabetes	8	3.9	5.9	8.8	5.3
Segment	18	7.3	15.9	18.6	14.1
Sonar	60	10.7	56.4	58.9	38.1
Teaching assistant evaluation	5	4.0	3.9	5.1	4.1
Vehicle	18	11.5	16.5	18.8	15.5
Vowel	10	6.7	8.9	10.7	8.4
Waveform (version 1)	21	14.2	18.3	19.8	17.8
Wine	13	7.1	12.8	13.8	12.3
Yeast	8	1.5	7.0	7.9	7.0

5. Conclusions and future research

Linear discriminant analysis is a conventionally adopted classification method. However, data without feature selection may be redundant or noisy, and may degrade the classification accuracy rate. This study proposes a particle swarm optimization-based approach that can search for a subset of beneficial features. This optimal subset of features is then applied in the dataset to obtain the optimal classification outcomes. Comparison of the obtained results with those of other approaches demonstrates that the proposed PSOLDA approach has higher classification accuracy rates than other tested approaches. That is, the PSOLDA approach can be applied to remove unnecessary or insignificant features in LDA, further enhancing the overall classification results. The main contributions of this study include:

(1) This study points out that the feature selection has influence on the LDA by showing an example.
(2) The statistic methods traditionally used for LDA to determine the beneficial feature subset are not easy or cannot guarantee the best results. The proposed PSOLDA approach can be used to perform feature selection for the LDA to archive higher classification accuracy rates.
(3) Comparing the optimal result obtained by the exhaustive enumeration for dataset with small number of features, the PSOLDA approach can obtain the same optimal result. This study showed that the classification accuracy rates of the PSOLDA were higher than those of forward feature selection, backward feature selection, PCA-based feature selection, bagging, and AdaBoost approaches in many public data sets.

Several directions for future studies are suggested. First, the proposed PSO-based meta-heuristic is sensitive to parameter settings. Thus, a more comprehensive study on alternative parameter tuning policies and customization the algorithm for this kind of feature selection problem by developing new parameters should be more deeply investigated. Second, the proposed approach was tested using linear discriminant analysis. However, non-linear discriminant analysis can also be optimized using the same approach. Third, experiments were performed using UCI datasets, but other public datasets and real-world problems should be tested in the future to verify and extend the proposed approach. Fourth, since the proposed PSO-based meta-heuristic is quite versatile, it would be worthwhile to explore the potential of this approach to other classification methods. This is currently being investigated by the authors of this paper.

Acknowledgment

The authors would like to thank the National Science Council of the Republic of China, Taiwan for financially supporting this research under Contract No. NSC97-2410-H-211-001-MY2.

References

[1] A. Sengur, An expert system based on linear discriminant analysis and adaptive neuro-fuzzy inference system to diagnosis heart valve diseases, Expert Systems with Applications 35 (2008) 214–222.
[2] A.C. Rencher, Methods of Multivariate Analysis, Wiley, 2002.
[3] A.J.C. Sharkey, Combining Artificial Neural Nets, Springer, London, 1999.
[4] A.K. Smilde, J.A. Westerhuis, R. Boqué, Multiway multiblock component and covariates regression models, Journal of Chemometrics 14 (2000) 301–331.
[5] C.A.C. Coello, G.T. Pulido, M.S. Lechuga, Handling multiple objectives with particle swarm optimization, IEEE Transactions on Evolutionary Computation 8 (2004) 256–279.
[6] D. Pyle, Data Preparation for Data Mining, Morgan Kaufmann, 1999.
[7] D. Ververidis, C. Kotropoulos, Emotional speech recognition: resources, features, and methods, Speech Communication 48 (2006) 1162–1181.
[8] E. García-Cuesta, I.M. Galván, A.J. de Castro, Multilayer perceptron as inverse model in a ground-based remote sensing temperature retrieval problem, Engineering Applications of Artificial Intelligence 21 (2008) 26–34.
[9] E.T. Lee, J.W. Wang, Statistical Methods for Survival Data Analysis, Wiley, 2003.
[10] G.C.J. Fernandez, Discriminant analysis: a powerful classification technique in data mining, in: Proceedings of the SAS Users International Conference, 2002, pp. 247–256.
[11] H. Liu, H. Motoda, Feature Selection for Knowledge Discovery and Data Mining, Kluwer Academic, Boston, 1998.
[12] I.T. Jolliffe, Principal Component Analysis, Springer, New York, 2002.
[13] J. Cadima, I.T. Jolliffe, Loadings and correlations in the interpretation of principal components, Journal of Applied Statistics 22 (1995) 203–214.
[14] J. Han, M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann, San Francisco, 2005.
[15] J. Kennedy, R. Eberhart, Particle swarm optimization, Proceedings of IEEE Conference on Neural Network 4 (1995) 1942–1948.
[16] J. Pacheco, S. Casado, L. Núñez, O. Gómez, Analysis of new variable selection methods for discriminant analysis, Computational Statistics and Data Analysis 51 (2006) 1463–1478.
[17] J. Lattin, D. Carroll, P. Green, Analyzing Multivariate Data, Duxbury, 2003.
[18] J. Ye, Characterization of a family of algorithms for generalized discriminant analysis on undersampled problems, The Journal of Machine Learning Research (2005) 483–502.
[19] J. Ye, Q. Li, A two-stage discriminant analysis via QR decomposition, IEEE Transactions on Pattern Analysis and Machine Intelligence (2005) 929–941.
[20] L. Breiman, Arcing Classifiers, Technical Report 460, Statistics Department, University of California, Annals of Statistics (available at www.stat.berkeley.edu).
[21] L.H. Chiang, R.J. Pell, Genetic algorithms combined with discriminant analysis for key variable identification, Journal of Process Control 14 (2004) 143–155.
[22] L. Rocchi, L. Chiari, A. Cappello, F.B. Horak, Identification of distinct characteristics of postural sway in Parkinson's disease: a feature selection procedure based on principal component analysis, Neuroscience Letters 394 (2006) 140–145.
[23] N. Abe, M. Kudo, Non-parametric classifier-independent feature selection, Pattern Recognition 39 (2006) 737–746.
[24] P. Howland, H. Park, Generalized discriminant analysis using the generalized singular value decomposition, IEEE Transactions on Pattern Analysis and Machine Intelligence 8 (2004) 995–1006.
[25] P.-F. Pai, T.-C. Chen, Rough set theory with discriminant analysis in analyzing electricity loads, Expert Systems with Applications 36 (2009) 8799–8806.
[26] P.J. Pardo, A.P. Georgopoulos, J.T. Kenny, T.A. Stuve, R.L. Findling, S.C. Schulz, Classification of adolescent psychotic disorders using linear discriminant analysis, Schizophrenia Research 87 (2006) 297–306.
[27] P.N. Belhumeur, J. Hespanda, D. Kriegeman, Eigenfaces vs fisherfaces: recognition using class specific linear projection, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (7) (1997) 711–720.
[28] R.C. Eberhart, Y. Shi, Comparing inertia weights and constriction factors in particle swarm optimization, Proceedings of IEEE on Evolutionary Computation 1 (2000) 84–88.
[29] R. Kohavi, G.H. John, Wrappers for feature subset selection, Artificial Intelligence 97 (1997) 273–324.
[30] R. Pydipati, T.F. Burks, W.S. Lee, Identification of citrus disease using color texture features and discriminant analysis, Computers and Electronics in Agriculture 52 (2006) 49–59.
[31] R. Sabatier, C. Reynès, Extensions of simple component analysis and simple linear discriminant analysis using genetic algorithms, Computational Statistics and Data Analysis 52 (2008) 4779–4789.
[32] S. Hettich, C.L. Blake, C.J. Merz, UCI repository of machine learning databases, Department of Information and Computer Sciences, University of California, Irvine, CA, 1998. http://www.ics.uci.edu/~mlearn/MLRepository.html.
[33] S. Ghosh-Dastidar, H. Adeli, N. Dadmehr, Principal component analysis-enhanced cosine radial basis function neural network for robust epilepsy and seizure detection, IEEE Transactions on Biomedical Engineering 55 (2) (2008) 512–518.
[34] S. Qin, S. Valle, M. Piovoso, On unifying multi-block analysis with applications to decentralized process monitoring, Journal of Chemometrics 15 (2001) 715–742.
[35] S. Wold, N. Kettaneh, K. Tjessem, Hierarchical multiblock PLS and PC models for easier model interpretation and as an alternative to variable selection, Journal of Chemometrics 10 (1996) 463–482.
[36] S.L. Salzberg, On comparing classifiers: pitfalls to avoid and a recommended approach, Data Mining and Knowledge Discovery 1 (1997) 317–328.
[37] S.-W. Lin, Z.-J. Lee, S.-C. Chen, T.-Y. Tseng, Parameter determination of support vector machine and feature selection using simulated annealing approach, Applied Soft Computing 8 (2008) 1505–1512.
[38] T. Li, S. Zhu, M. Ogihara, Using discriminant analysis for multi-class classification: an experimental investigation, Knowledge and Information Systems 10 (4) (2006) 453–472.
[39] V. Ravi, P.J. Reddy, H.-J. Zimmermann, Pattern classification with principal component analysis and fuzzy rule bases, European Journal of Operational Research 126 (2000) 526–533.
[40] X. Zhang, Y. Jia, A linear discriminant analysis framework based on random subspace for face recognition, Pattern Recognition 40 (2007) 2585–2591.
[41] Z. Liang, Y. Li, P. Shi, A note on two-dimensional linear discriminant analysis, Pattern Recognition Letters 29 (2008) 2122–2128.