International Journal of Artificial Intelligence and Applications for Smart Devices
Vol.1, No.1 (2013), pp.65-82
http://dx.doi.org/10.14257/ijaiasd.2013.1.1.05

Artificial Immune Linear Discriminant Analysis

Amin Allahyar[1] and Hadi Sadoghi Yazdi[1,2]

[1]Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad, Iran
[2]Center of Excellence on Soft Computing and Intelligent Information Processing, Ferdowsi University of Mashhad

Amin.Allahyar@stu.um.ac.ir, h-sadoghi@um.ac.ir

Abstract

In Linear Discriminant Analysis (LDA), it is assumed that each class has a Gaussian distribution. This assumption rarely holds in the real world problems. However, by removing this assumption, the problem become intractable and cannot be solved in analytic form. Quite recently, a group of evolutionary algorithms is introduced to solve this problem. These algorithms used a combination of fisher criterion and fuzzy membership function as their fitness function. It is widely acknowledged that computing the fitness function in an evolutionary algorithm needs to be very fast. Unfortunately, calculating fisher criterion for each chromosome in iterations of an evolutionary algorithm has a high computational cost. Furthermore it is known that the fuzzy membership function has an assumption of Gaussian distribution, thus using it as a fitness function will have same assumption issue that LDA had previously. In this paper, we suggest a new fisher criterion to incorporate in fitness function and show that it is theoretically faster than previous introduced criterion. In addition we theoretically prove the equality of proposed criterion. Next, in order to eliminate the Gaussian assumption, we offer a substitution for fuzzy membership fitness function which does not have Gaussian assumption. Moreover, the superior speed introduced fitness function theoretically investigated. At last, in order to confirm the effectiveness of proposed fitness functions, comprehensive experiments using twelve UCI repository dataset and two real world problems in face and object recognition is performed and the results is compared in both speed and accuracy.

Keywords: *Feature Extraction; Linear Discriminant Analysis; Gaussian Assumption; Evolutionary Algorithm; Fitness Function*

1. Introduction

Clustering and classification are two main fields of research in machine learning. Due to the so called *curse of dimensionally* problem, a *Dimension Reduction* (DR) preprocessing step found many application in these fields [1, 2]. Many different DR methods have been proposed in the literatures which extract useful features based on the information that might be available beside actual data. In the extreme cases, when no extra information is available, the goal becomes to find directions in which the variance of data is maximized [3]. A popular algorithm to reach such a goal is Principle Component Analysis (PCA) [4]. On the other hand if some information about domain (*e.g.,* label of data) be at hand, it can be exploited to find directions in which the distance between data points in the same class minimized while data in different classes stay far from each other [5]. Fisher formulized this goal from such view which is called *Linear Discriminant Analysis* (LDA) [6]. His well-known cost, tries to find

ISSN: 2288-6710 IJAIASD

directions to maximize the ratio of between-class variances to the within-class variances, thereby maximal separability will be guaranteed [7].

The formulation of LDA assumes that data points is scattered with a unimodal or Gaussian distribution around their corresponding class mean [3]. In the other word, LDA is equivalent to maximum likelihood classification if each class has a Gaussian distribution with common covariance matrix [8]. This assumption is frequently violated in real world problems. Therefore if class distribution follows a multimodal or share the same mean, LDA will gives degenerate result [1]. Thorough the last three decades many different approaches are perused to extract a better feature set. Quadratic Discriminant Analysis (QDA) is proposed by Wald and Kronmal which relaxes the identical covariance assumption and allows for complex discriminant boundaries to be formed [9]. However, in compare to LDA, QDA needs a lot more training samples to reach an admissible solution due to the fact that larger number of parameters needed to be calculated [10].

Quite recently, a group of new approaches is proposed by Mohammadi *et al.*, which use a mixture of evolutionary algorithms and fuzzy membership functions to find a more compact classes with distant mean vector [11]. Evolutionary algorithms have an innate capability in searching space of non-convex problems with many local minimum. Thus, it is a promising methodology to use these kinds of algorithms to finding suitable features. A very essential part in any evolutionary algorithm is fitness function. This part measures the utility of each chromosome to be a solution. In each step of an evolutionary algorithm, many chromosome needs to be evaluated by fitness function. Therefore the fitness function should be very computation efficient. Unfortunately, the fitness function in Mohammadi's evolutionary algorithm is the actual fisher criterion, which has a high computational cost. Hence, this algorithm require a lot of time to reach an appropriate solution. In addition, Mohammadi used a fuzzy membership function to overcome Gaussian distribution assumption issue. But the fuzzy membership function has the same Gaussian distribution assumption. Consequently the assumption issue remained unresolved.

In this paper we aim to propose new fitness function for these algorithms to solve the discussed problems and theoretically show that they can reach the same solution with lower computational cost. In addition, to overcome the assumption issue, we propose a substitution for fuzzy membership function and experimentally show that it can give a better class separability measurement with no Gaussian assumption. This paper is organized as follows: In section 2 we will discuss about preliminaries in discriminant analysis and demonstrate the Gaussian assumption issue in LDA. In section 3 we discuss Mohammadi's evolutionary algorithms and the expressed problem in more detail. Section 4 is dedicated to our proposed algorithm. Experimental result will be demonstrated in section 5 and conclusion is given in section 6.

2. Preliminary

First we define our notation in this paper. Scripted letters such as $\mathcal{C}$ and $\mathcal{X}$ represent sets. Capital letters like X and W are matrixes while bold lower case letters show column vectors *e.g.*, $\boldsymbol{x}$ and $\boldsymbol{u}$. Lower case letters indicate scalars *e.g.* n and d. Similar to popular notation we use subscripts to index elements in matrixes or vectors. For example x_i is i-th element of vector $\boldsymbol{x}$ and $\boldsymbol{w}_j$ is the j-th column vector in matrix W. The vector norm $\|.\|$ is the l_2 norm so by definition $\|\boldsymbol{x}\| = \sqrt{\boldsymbol{x}^T\boldsymbol{x}}$. Let $\mathcal{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n\} \in \mathbb{R}^{d\times n}$ be n given data points in column vector with d dimension. Also there are m classes available. So the whole dataset is divided into m set where $\mathcal{X} = \{\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_m\}$ and depend on their labels l_i, each data point $\boldsymbol{x}_i$

belong to one of these classes. The aim in LDA problem is to find an optimal projection matrix $W^* \in \mathbb{R}^{r \times d}$ where r is the desired dimensionality of data after projection.

In the standard LDA problem label of each data point is given. In the other word, each sample $\boldsymbol{x}_i$ has a corresponding class label l_i. Then the between-class scatter matrix S_b and within-class scatter matrix S_w are determined by (1):

$$S_b = \sum_{k=1}^{m} n_k(\boldsymbol{\mu}_k - \boldsymbol{\mu})(\boldsymbol{\mu}_k - \boldsymbol{\mu})^T \tag{1}$$

$$S_w = \sum_{k=1}^{m} \sum_{\boldsymbol{x}_i \in \mathcal{X}_k} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T$$

Where n_k indicate number of samples that belongs to class $\mathcal{X}_k$. In this formula, $\boldsymbol{\mu}_k$ and $\boldsymbol{\mu}$ indicate mean vector of class k and global mean vector of whole dataset respectively. These two matrixes can be intuitively described as follows. The S_b matrix will measure how far mean vectors corresponding to each class is scattered around input space. The dispersion of class means is quantized by sum of the distances between each class mean vector and the global mean vector. Likewise the S_w matrix measures the spreading of data points around its class mean vector. Mathematically S_w is sum of covariance matrixes calculated from data points belonging to each class. In some literatures [1, 12, 13], another scatter matrix is defined and used which is called *Total Scatter* matrix. It is defined as below and it is mathematically equivalent to covariance of whole dataset.

$$S_t = S_b + S_w = \sum_{\boldsymbol{x}_i \in \mathcal{X}} (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^T = cov(\mathcal{X})$$

We will not use this scatter matrix in our proposed LDA algorithm but as argued later, the definition of proposed algorithm can be easily generalized as any combination of such scatter matrix including S_t. Using these definitions, the standard LDA cost function proposed by Fisher [6] is as follows.

$$J_f(W) = \frac{tr(W^T S_b W)}{tr(W^T S_w W)} \tag{3}$$

We are interested in W such that given S_b and S_w the cost function $J_f(W)$ acquires its maximum value. In the other word, we are looking for a solution to following optimization algorithm.

$$W^* = \underset{W}{\operatorname{argmax}}\, J_f(W) \tag{4}$$

It worth noting that by these expressions it is clear that the LDA will have a suitable result if the distribution of each class in input space follows a uni-modal distribution [1]. This problem is widely known and studied. An example of such situation is demonstrated in Figure 1. This example is given by an inspiration from Fukunaga’s book [1]. In Figure 1.a the mean vector of two classes along with global mean vector are equal. Therefore by the definitions in (1), S_b with be equal to zero and so the numerator of cost function (3) will be zero. In this situation, the cost function will always have the zero value regardless of its denominator. The problem in Figure 1.b is that the data point in each class does not follow a Gaussian distribution *e.g.* the distribution of data points around their corresponding mean vector is not uniform. So there is no suitable projection direction to reach a good class separability.

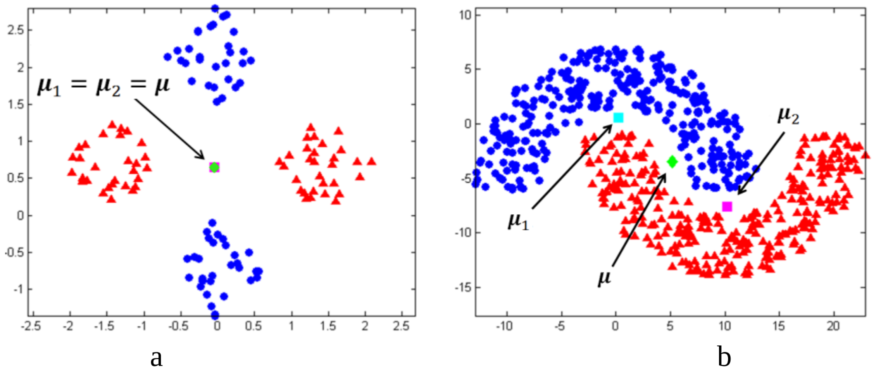


Figure 1. Demonstration of Situation where LDA will Fail to Extract a Good Feature. a. In this Case, Class and Global mean Vector are Equal, thus the between Class Scatter Matrix will be Zero. b. The Data Point in each class does not follow a Gaussian distribution resulting in failure of LDA

Unfortunately the optimization problem (4) is typically non-convex. More generally there is no close form solution for general trace ratio problem. Therefore, these problems are regularly altered into the simpler yet inexact ratio trace problem which is convex and has a close form solution [14]. This transformation can be formulated as (5):

$$W^* = \underset{W}{\operatorname{argmax}}\, tr(\frac{W^T S_b W}{W^T S_w W}) = \underset{W}{\operatorname{argmax}} \frac{|W^T S_b W|}{|W^T S_w W|} \tag{5}$$

Actually Mohammadi *et al.*, [11] used this cost function in his evolutionary fitness function. In the contrary, we use the original fisher criterion (3) as our main cost function and rewrite an equivalent but much computation efficient fitness function to incorporate in the evolutionary algorithm. The cost function in (5) can be easily solved directly by finding the solution to generalized eigenvector problem of the form (6) [1]:

$$S_b U = S_w U \Lambda \tag{6}$$

The eigenvectors corresponding to largest eigenvalues are the transformation matrix that maximize the given cost function in (5). But it should be noted that rank of S_b is at most $m-1$ [15], where m indicate number of classes. So only $m-1$ of these vectors has discriminating information [2]. Furthermore if S_w be non-singular, formula (5) can be rewritten as standard eigenvector problem by multiplying S_w^{-1} from left as follows.

$$S_b U = S_w U \Lambda$$
$$S_w^{-1} S_b U = S_w^{-1} S_w U \Lambda = I U \Lambda$$
$$S_w^{-1} S_b U = U \Lambda$$

The singularity of S_w is the most known difficulty in LDA and is widely studied in the literatures [8, 16, 17]. As our proposed algorithm does not have singularity problem, we will not investigate such difficulties. For comprehensive discussion on these complications please refer to [18]. By these definitions, it is clear that the reason to approximation (3) and conversion to (4) was only to reach a convex problem and so it becomes possible to solve LDA in analytic from. But in evolutionary algorithms the convexity of the cost function is not required. So we are allowed to use the original fisher criterion (3) in fitness function. In addition, we show that (3) can be converted to a very computation efficient fitness function. This is another reason that motivated us to propose our fitness function. To our knowledge, Mohamadi's methods are the first use of evolutionary algorithms to improve LDA result. Thus, in the next section, we will discuss his group of evolutionary algorithms in more detail as related work.

3. Related Work

In this section we provide a detailed description of Mohammadi's algorithms. These groups of algorithms are consist of four evolutionary algorithm including: Genetic based Linear Discriminant Analysis (G-LDA), Artificial Immune System LDA (A-LDA), Fuzzy based Genetic LDA (FG-LDA), Fuzzy based A-LDA (FA-LDA). We explain each of these algorithms in its specific subsection.

3.1. Genetic based Linear Discriminant Analysis (G-LDA)

In this algorithm, similar to conventional genetic algorithms, collections of chromosome are randomly produced and formed an initial *population*. Each chromosome is a *candidate solution* and is an encoded projection matrix W. The encoding is the process of converting matrix W to a vector so that it can be stored in a chromosome. This conversion is demonstrated in Figure 2 and is simply done by sequentially adding each row of matrix W to the end of its previous row.

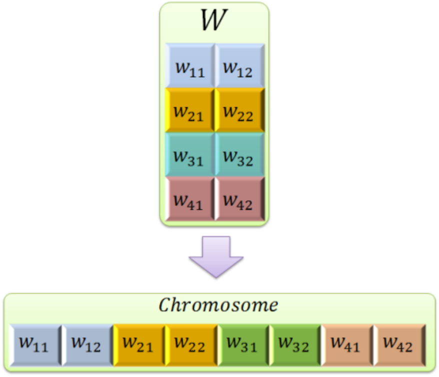


Figure 2. Demonstration of Converting W to a Suitable Chromosome Proposed by Mohammadi

For the selection operator, G-LDA uses *binary tournament selection* [19, 20]. In this selection method, two chromosomes are randomly selected from current population. Next the fitness of each chromosome is evaluated and one with higher fitness will have the permission to be on the next population. The cross over operator is *One Point Crossover*. After selection step, each pair of chromosome will have a constant chance (regularly 0.7) of having a crossover process. In this process, a random number between 1 and chromosome length is generated and considered as crossover point. Then each pair exchanges their values which are located on the left side (or equivalently right side) of crossover point and produce two new children which are regularly called *offspring*. This process is demonstrated in Figure 3.

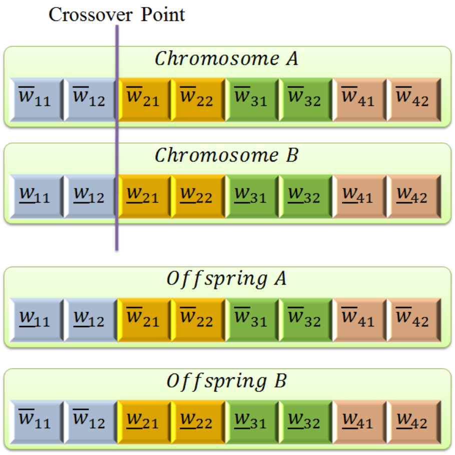

Crossover Point
Chromosome A
$\overline{w}_{11}$ $\overline{w}_{12}$ $\overline{w}_{21}$ $\overline{w}_{22}$ $\overline{w}_{31}$ $\overline{w}_{32}$ $\overline{w}_{41}$ $\overline{w}_{42}$
Chromosome B
$\underline{w}_{11}$ $\underline{w}_{12}$ $\underline{w}_{21}$ $\underline{w}_{22}$ $\underline{w}_{31}$ $\underline{w}_{32}$ $\underline{w}_{41}$ $\underline{w}_{42}$
Offspring A
$\underline{w}_{11}$ $\underline{w}_{12}$ $\overline{w}_{21}$ $\overline{w}_{22}$ $\overline{w}_{31}$ $\overline{w}_{32}$ $\overline{w}_{41}$ $\overline{w}_{42}$
Offspring B
$\overline{w}_{11}$ $\overline{w}_{12}$ $\underline{w}_{21}$ $\underline{w}_{22}$ $\underline{w}_{31}$ $\underline{w}_{32}$ $\underline{w}_{41}$ $\underline{w}_{42}$

Figure 3. Illustration of One point Crossover Operator. Assume that Crossover Point is randomly chosen as 2, then the First and Second Values in Vector A and B will be swapped

For mutation operator Mohammadi reported that four different types of mutations are used including: Random mutation, Swap mutation, Creep mutation and Scramble mutation. The mutation chance factor γ is selected equal to 0.2[1]. A brief description of these mutation operators is as follows:

- **Random Mutation:** One bit of a random chromosome flipped with chance of $25\% \times \gamma$.
- **Swap mutation:** Two random position in chromosome is selected and their value swapped with chance of $25\% \times \gamma$.
- **Creep mutation:** One bit of a random chromosome is changed by a random value between $[-Creepvalue, +Creepvalue]$. The creep value is selected to be 0.2. Chance of running such mutation is set to $30\% \times \gamma$.
- **Scramble mutation:** A random chromosome is selected and its every value reconfigured with chance of $20\% \times \gamma$.

At last the fitness function defined as (7).

$$F_{G-LDA} = \frac{|W^T S_b W|}{|W^T S_w W|} \tag{7}$$

By these definitions a typical genetic algorithms can be constructed. A schematic of such algorithm is given as Figure 4 [11].

[1] This value does not reported in corresponding paper and is acquired after contacting the author.

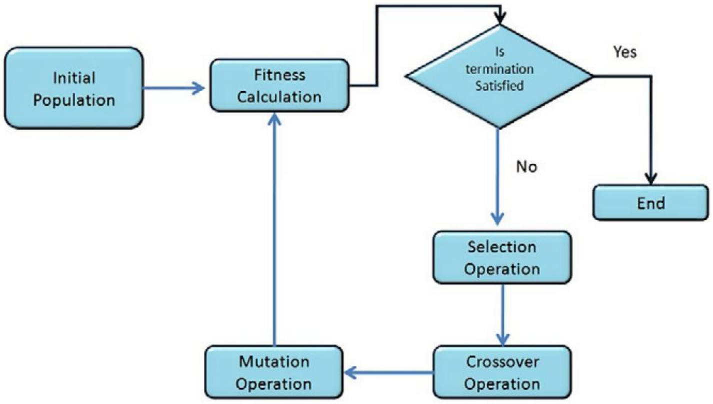


Figure 4. Schematic of G-LDA [11]

3.2. Artificial Immune System based LDA (A-LDA)

The artificial immune system is actually an inspiration from human immunity system. The biological immunity system in human body is a powerful and complicated system. Its main goal is to protect the body from external threats. It is capable of adaptively discriminate body cells from foreign or enemy cells in a distributed manner. The core element in immunity system is the white blood cells which consist of two type B-Cells and T-Cells. Both of these types are produced by bone marrow. In brief (and not exactly true in real immune system), B-Cells are responsible for detection of enemy cells called *Antigens* while T-Cells eliminate the detected antigens. This process is simulated in Artificial Immune System (AIS) algorithms. For more detailed description of real and its replicated artificial immune system please refer to Burke *et al.*, [21]. It has wide variety of applications including intrusion detection [22], image segmentation [11] and optimization [23]. Mohammadi used AIS to find an appropriate transformation matrix where the ratio of between class scatter matrix and within scatter matrix maximized. A schematic of his algorithm is given in Figure 5.

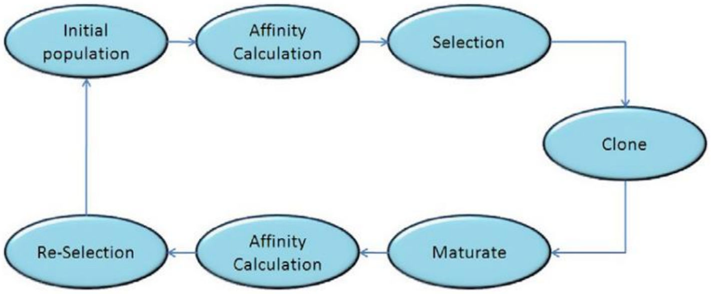


Figure 5. Schematic of A-LDA [11]

The Initial Population and Affinity Calculation step in Mohammadi's AIS algorithm is similar to two first steps in previously described genetic algorithm. In selection step, chromosome with highest affinity value is forwarded to the next step and others are discarded. In the Clone step, the chromosomes are cloned with regards to their affinity value. Chromosome with higher affinity value can produce more of itself. At last in Maturate step, chromosomes go through a mutation operation similar to previous genetic algorithm.

3.3. Fuzzy based Fitness Function for G-LDA and A-LDA

Fuzzy based Fitness Function (FFF) is proposed by Mohammadi with the intention of incorporating ‘‘degrees of truth’’ or ‘‘degrees of membership’’ instead of crisp labels. In the other word, each data point $\boldsymbol{x}_i$ has a degree of membership to every m classes. The degree of membership is defined as (8).

$$u_{ij} = \frac{d_{ij}^{-2/(\theta-1)}}{\sum_{k=1}^{m} d_{ik}^{-2/(\theta-1)}} = \frac{1}{\sum_{k=1}^{m} (\frac{d_{ik}}{d_{ij}})^{-2/(\theta-1)}} \qquad \forall i = 1,2,\dots,n \ and \ \forall j = 1,2\dots,m \tag{8}$$

Where θ is the fuzziness degree with range $[1,\infty)$ which is commonly used in many fuzzy based algorithms i.e. K-means [24]. If $\theta \to 1$, the membership value $u_{ij}\ \forall j = 1,2\dots,m$ corresponding to data point $\boldsymbol{x}_i$ would be more crisp among classes and If $\theta \to \infty$, the u_{ij} gain more fuzziness. Mathematically using this definition u_{ij} has range [0, 1] and determine how much a data point $\boldsymbol{x}_i$ belongs to class j. The key idea in using this fitness function instead of fisher criterion which is used in (7) is that the membership degree of each data point is related to every available class instead of distance from corresponding class only. Hence, increase in membership degree of a data point in specific class means decrease in membership of that data point in other classes. Because we have the following constrain[2]:

$$\sum_{k=1}^{m} u_{ik} = 1$$

After calculation of Fuzzy memberships, a fitness function can be calculated as (9).

$$Fitness_{Fuzzy} = \sum_{k=1}^{m} \sum_{x_i \in \mathcal{X}_k} u_{ik}$$

This fuzzy fitness function is substituted with fisher criterion in described G-LDA and A-LDA algorithms and new fuzzy based evolutionary algorithm called FG-LDA and FA-LDA introduced. Except the fitness function, other parts of algorithms are defined similar to previous ancestors. In the next section, we propose our modifications with aim of increasing accuracy and reducing the computational cost.

4. Proposed Algorithms

In this section we propose our new groups of evolutionary algorithms to find LDA solution. At first we will propose a new representation and discuss the reason why this representation can be a better representation. Next we provide our fitness functions for genetic based, AIS based and fuzzy based fitness function. Finally a nonlinear fitness function is proposed to completely eliminate the unimodal assumption in LDA.

4.1. Chromosome Representation

As discussed in related work section, to generate a chromosome from projection matrix W, Mohammadi used a row by row representation. But the cross over step of evolutionary

[2] Although it was not explicitly reported, but this constrain can be easily deducted by definition in (8). This constrain is added to reach more clarity in the algorithm.

algorithm using this representation can be problematic. By looking at chromosome representation in Figure 2 and cross over method in Figure 3, it can be realized that this operation will break a projection direction and substitute its value with random direction. This is clearly is not ideal because we are looking for some projection direction that maximize the fitness function. Assume that in an iteration of evolutionary algorithm, only one direction $\boldsymbol{w}_p$ needs to be found and every other direction $\boldsymbol{w}_i\ \forall i \in \{1,2,\dots,r\}, i \neq p$ is already has its optimal value $\boldsymbol{w}_i = \boldsymbol{w}_i^*$. Although this chromosome has a good chance to reach the optimal solution, only one cross over operation can completely destroy the optimal directions. This is clearly has contradiction with *Building Block* principle discussed by Goldberg [20] which is the key properties in evolutionary algorithms to reach a good solution in finite time [25]. This problem can be easily avoided if we use a column by column representation as shown in Figure .6. This way, the cross over on two parents will have a more meaningful result as it will nearly substitute two directions in the parents to build the new off springs.

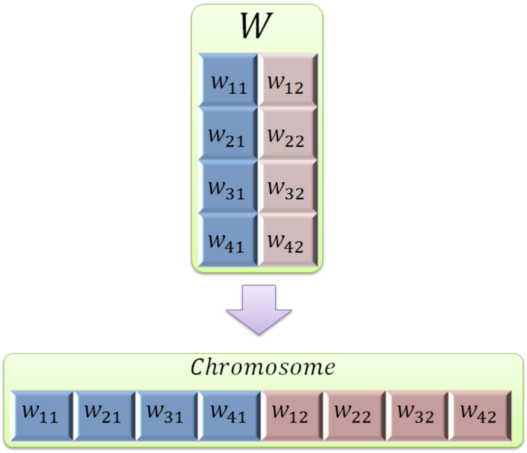


Figure 6. Process of representing W in a Chromosome

4.2. Genetic and Artificial Immune Based Fitness Function

As discussed in the previous section, following Mohammadi's algorithm the transformed fisher criterion (7) is used as fitness function of evolutionary algorithms to reach a better class compactness and separability to other classes. Fortunately because we are not solving this problem in analytic form, we are allowed to use any non-convex fitness function. Thus, we can use the original fisher criterion defined in (10) which is used before and reported a better class density and mean separation [14, 26, 27].

$$J_f(W) = \frac{tr(W^T S_b W)}{tr(W^T S_w W)} \tag{10}$$

Using (10) has some benefits including:

- **Accuracy**: As described in preliminary section, it is the original fisher criterion. Thus instead of approximating (3) and conversion to (5) so that it become possible to solve it using an analytic method we are able to find the global optimal solution of LDA.

- **Speed:** Fastest method to calculate the determinant of a matrix $A \in \mathbb{R}^{d \times d}$ will costs $O(d^3)$ [28]. While the trace is only cost $O(d)$.
- **Null Space Issue:** To describe this issue, we need to provide a Theorem.

Theorem 1: Let $A \in \mathbb{R}^{d \times d}$ be a rank s square matrix where $s < d$. Let $\Psi \in \mathbb{R}^{d \times r}$ where $r > 0$ be an arbitrary matrix. If any column vector $\boldsymbol{\psi}_i$ of Ψ lies in *Null Space* of A. Then we have the following equality.

(11)

$$\det(\Psi^T A \Psi) = 0$$

Proof: Let $\Phi = \{\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \dots, \boldsymbol{\varphi}_d\}$ and $\Lambda = \{\lambda_1, \lambda_2, \dots, \lambda_d\}$ represent the corresponding eigenvectors and eigenvalues of matrix A. Without loss of generality assume that $\lambda_1 > \lambda_2 > \cdots \lambda_d$. We know that last $d - s$ of these eigenvalues are equal to zero. Furthermore we have the following equality:

(12)

$$A\boldsymbol{\varphi}_j = \lambda_j \boldsymbol{\varphi}_j = \mathbf{0}_d^T \quad \forall j \in \{s, s+1, \dots, d\}$$

Wh.ere $\mathbf{0}_d$ represents zero column vector with d elements. Assume that only one column of Ψ is equal to one of these eigenvectors:

(13)

$$\boldsymbol{\psi}_p = \boldsymbol{\varphi}_j$$

$$j \in \{s, s+1, \dots, d\}$$

$$p \in \{1,2, \dots, r\}$$

Substituting (12) to (13) we have:

$$A\boldsymbol{\varphi}_j = A\boldsymbol{\psi}_p = \mathbf{0}_d$$

The last multiplication in (11) $A\Psi$ can be represented as follows:

$$A\Psi = [A\boldsymbol{\psi}_1 \quad A\boldsymbol{\psi}_2 \quad \cdots \quad A\boldsymbol{\psi}_p \quad \cdots \quad A\boldsymbol{\psi}_{\boldsymbol{r}}]$$

(14)

$$A\Psi = [A\boldsymbol{\psi}_1 \quad A\boldsymbol{\psi}_2 \quad \cdots \quad \mathbf{0}_d \quad \cdots \quad A\boldsymbol{\psi}_{\boldsymbol{r}}]$$

It can be easily shown that any arbitrary matrix B that multiplied from left to (14) will result a zero vector $\mathbf{0}_d$ it i-th column. Furthermore if $B = \Psi^T$, the matrix of left multiplication will have i-th column and row equal to $\mathbf{0}_d$ and $\mathbf{0}_d^T$ respectively. Following linear algebra principle [29], any square matrix with a column or row equal to zero will have its determinant equal to zero. More generally any rank deficient matrix has determinant equal to zero. Thus theorem proved.

Following this theorem, in the iterations of evolutionary algorithm, if a column of W become equal to a null space of S_b then the numerator of fitness function in (7) will be equal to zero regardless of its denominator. In this situation, even in semi perfect situation $\boldsymbol{w}_i = \boldsymbol{w}_i^*$ $i \in \{1,2,\dots,r\}, i \neq p$ the cost function will acquire its minimum value irrespective to other columns $\boldsymbol{w}_i$ that are equal to optimal solution. Thus the fitness function would never choose this chromosome to be in the next generation. Similar issue can emerge with S_w. In a case where a column $\boldsymbol{w}_p$ of W is in null space of S_w, the cost function will be equal to infinity. Therefore regardless of other column $\boldsymbol{w}_i$ $i \in \{1,2,\dots,r\}, i \neq p$ that can be a bad solution, the fitness function will choose this chromosome in each iterations and it even can be chosen as the final answer. It should be noted that using (10) would not completely solve null space issue. In a very rare case where every column $\boldsymbol{w}_i$ $i \in \{1,2,\dots,r\}$ be in null space of its corresponding scatter matrix, the cost function in (10) would suffer from the above problem. Fortunately it is an infrequent case and can be disregarded.

In our proposed group of evolutionary algorithms we will use (10) as the fitness function. In addition to theoretic discussion given above, we will show in experimental results section that using this fitness function we can reach a better solution for LDA. Following Mohammadi's algorithm we use (10) as affinity measure in our AIS algorithm. Besides, in Clonal Selection chromosomes with higher affinity is selected and cloned with respect to their corresponding fitness function value.

4.3. Mutation Operator

Four different mutation methods are proposed in Mohammadi's algorithm and they are used simultaneously. Although it is a good approach to increase the mutation power with aim of extending the search area of a chromosome, but this way we will lose the tiny peaks in cost function value which may potentially be the optimal solution. This is a known issue in evolutionary algorithms as there is always a tradeoff between discoverability of new possible solution and extraction of reached optimal solution. We simplify the mutation operator and only use a Modified Creep Mutation. In new mutation, the creep value will decreases over iterations. This way we can have *Simulated Annealing* properties in the mutation operator [30]. The creep value for iteration t can be calculated as follows:

$$creep_t = creep_0 \times \exp\left(\frac{t}{2 \times \sqrt{t_{max}}}\right)$$

Where $creep_t$ indicate the creep value for iteration t and t_{max} is maximum number of iteration. exp is the exponential function where $exp(a) = e^a$.

4.4. Crossover Operator

Because the especial representation of chromosome in our proposed methods, instead on using a regular cross over operator, we propose a new cross over which we call *Local Crossover*. In this method, for each pair of parent chromosome, a random value ϑ where $\vartheta \in \{1,2,\dots,r\}$ is selected and the crossover only applied to that specific column of parents. This way, the features on each direction only substituted with same direction in other chromosomes. We claim that the proposed crossover operator will results in less damage to information that each chromosome carries.

4.5. Fuzzy Based Fitness Function

As discussed in introduction section, while using fuzzy membership function to measure class compactness and mean dispersion is useful but the fuzzy membership function has the same unimodal assumption issue. To solve this problem, we propose a *Supervised Fuzzy C* *means* algorithm. The Fuzzy C-Means (FCM) algorithm is an unsupervised clustering algorithm introduced by Bezdek in 1981 with cost function defined as (15) [31]. The power in FCM is from the fact that rather than assigning each data point to a specific cluster, they are assigned to all clusters with degree of membership.

$$J_{FCM} = \sum_{j=1}^{\pi} \sum_{i=1}^{n} u_{ji}^{2} \|\boldsymbol{x}_i - \boldsymbol{\mu}_j\| \tag{15}$$

Where π is the desired number of clusters. It will result a membership function matrix $U \in \mathbb{R}^{\pi \times n}$ where each column vector $\boldsymbol{u}_i \in \mathbb{R}^{\pi \times 1}$ indicates the degree of membership $\boldsymbol{x}_i$ to each cluster $\boldsymbol{\mu}_j$. In our method, instead of clustering whole dataset, we cluster each class individually. In the other word, FCM is applied to data points of each class. Result of such algorithm for each class $k \in \{1,2, \ldots m\}$ is a class specific membership function matrix $U^{(k)}$. Using FCM we are capable of finding locations with high density distribution and locally measure the class compactness and mean separation. The number of desired cluster π_k where $k \in \{1,2, \ldots m\}$ for each class is a user defined value which is selected by degree of multimodality and nonlinearity of specific class. By default we set it to m^3. By these expressions, the Multi Class Fuzzy fitness function (MCF) for data point $\boldsymbol{x}_i$ can be defined as (15).

$$Fitness_{MCF}(\boldsymbol{x}_i) = \frac{\underset{j}{\operatorname{argmin}}\, u_{ji}^{(l_i)}}{\sum_{k=1}^{\pi_k} \underset{j}{\operatorname{argmin}}\, u_{ji}^{(k)}} \tag{16}$$

Where l_i is the corresponding label of $\boldsymbol{x}_i$. We would like to emphasize that this approach is only a fuzzified enhancement of suggested routine by Fukunaga (Chapter 10, Page 452 of his book) [1]. Using this method, the unimodal assumption is completely removed and LDA works in multimodal distribution like Figure 1.a or nonlinear situations similar to Figure 1.b. In proposed fitness function we followed Mohammadi's idea. But in situations where lower computational cost is needed, instead of FCM, a class specific *Single Linkage* clustering algorithm can be used. This way, instead of number of cluster for each class, a maximum variance threshold can be used to find clusters in each specific class. Then the closeness of each data point $\boldsymbol{x}_i$ with label l_i to its mean cluster $\boldsymbol{\mu}_j^{(l_i)}$ related to other clusters can be used as fitness function. This will reduce the computational cost in a point close to Mohammadi's fitness function.

5. Experimental Result

In this section, we aim to experimentally compare the result of proposed group of algorithms with its parallel Mohammadi's algorithm. To reach a more integrity, we also applied standard LDA to show the effectiveness of evolutionary algorithms.

5.1. Setup

We aimed to provide an experiment environment similar to Mohammadi's experiments as much as possible. Thus, we compared the result with several measurements including: Normalized Mutual Information (MI), Dunn, SD, isolation and Davies–Bouldin (DB) indexes. At last the accuracy improvement for clustering and classification is reported. The result of using these measures will be given in their corresponding subsection. In this experiments, we used an Intel Quad-Core 2.5 GHZ computer with 4GB RAM on windows 7 64bit. Also Matlab 2012a is used as simulation software. These algorithms are applied to ten datasets from UCI repository[3]. In order to show the applicability in real world problems two datasets from face recognition and object recognition is also used in these comparisons. COIL-20 include gray image of 20 objects which is taken from 75 different angles. These samples is reduced to size 32×32. ORL dataset contains gray images of 40 persons. Each person has 10 shots, each with different expressions and facial details. As the source image has dimensionality of 112×92, the input data has 10304 dimension. Properties of these datasets along with desired discriminant set r are given in Table.1. For classifier we used the standard K Nearest Neighbor (KNN) classifier with $k = 1$. To reach more reliable results, we repeat each experiment 25 times and then the mean and variance of experiment is reported. In addition, we used following abbreviations for different method:

- **Normal**: indicates the result obtained from applying KNN to original dataset without any transformation.
- **LDA**: The standard linear discriminant analysis algorithm with (5) as its cost function.
- **G-LDA, A-LDA, FG-LDA, FA-LDA:** which indicates the mohammadi's algorithms including: Genetic based LDA, Artificial Immune based LDA, Fuzzy G-LDA and Fuzzy A-LDA algorithm respectively.
- **G-TL, A-TL, FG-TL, FA-TL:** Our groups of proposed method including: Genetic based Trace ratio LDA, Artificial Immune based Trace ratio LDA, Fuzzy G-TL and Fuzzy A-TL respectively.

Table 1. Properties of data sets used for experiments. As previously defined, n is number of datapoints, d is data dimension, m indicates number of class, r is the desired number of extracted features

	n	d	m	r
Soybeans	47	35	4	4
Iris	150	4	3	2
Wine	178	13	3	3
Sonar	208	60	2	2
Ionosphere	351	34	2	2
Dermatology	366	33	6	4
WDBC	569	30	2	2
Vehicle	864	18	4	4
Vowel	990	10	11	6
Waveform	5000	21	3	3
COIL	1440	256	20	15
ORL	400	10304	40	40

[3] Available at http://www.ics.uci.edu/mlearn/MLRepository.html.

5.2. Normalized Mutual Information Measure

For the first measurement the NMI criterion is used. The un-normalized criterion measures the amount of statistical information shared by each predicted label $\widehat{l_i}$ corresponding to ground truth label l_i. From probability point of view, Let Z be a discreet random variable with $\Omega = \{Z_1, Z_2, \dots, Z_n\}$ alphabet. The Probability Density Function (PDF) of Z can be defined as $p(z) = \Pr\{Z = z, z \in \Omega\}$. The measure of uncertainty or entropy of Z can be defined as follows:

$$H(z) = -\sum_{z\in\Omega} p(z)log(p(x))$$

The join entropy for two random variables Z and Y with alphabet $\Upsilon = \{Y_1, Y_2, \dots, Y_n\}$ can be calculated as follows:

$$H(Z,Y) = \sum_{z\in\Omega}\sum_{y\in\Upsilon} p(z,y)log(p(z,y))$$

In the above formulation $p(z,y)$ indicates the join probability density of Z and Y. The amount of information that is shared between two variable Z and Y is called mutual information and is represented as $I(Z,Y)$. This means that when two variables is closely related, $I(Z,Y)$ has higher values and if they are independent from each other, $I(Z,Y)$ is small. Mathematically $I(Z,Y)$ can be defined as follows:

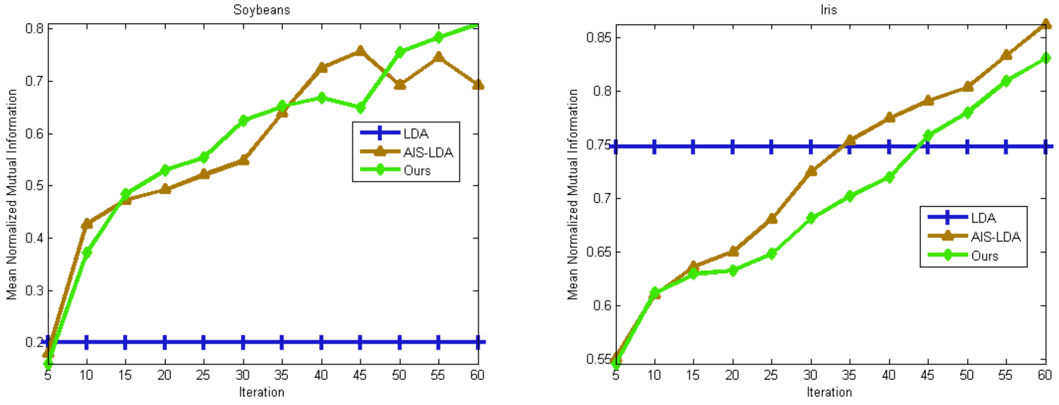
$$I(Z,Y) = H(Z) + H(Y) - H(Z,Y)$$

This measure has a zero lower bound and is called mutual information. The normalized version has a more proper upper bound of 1 which can be defined as follows:

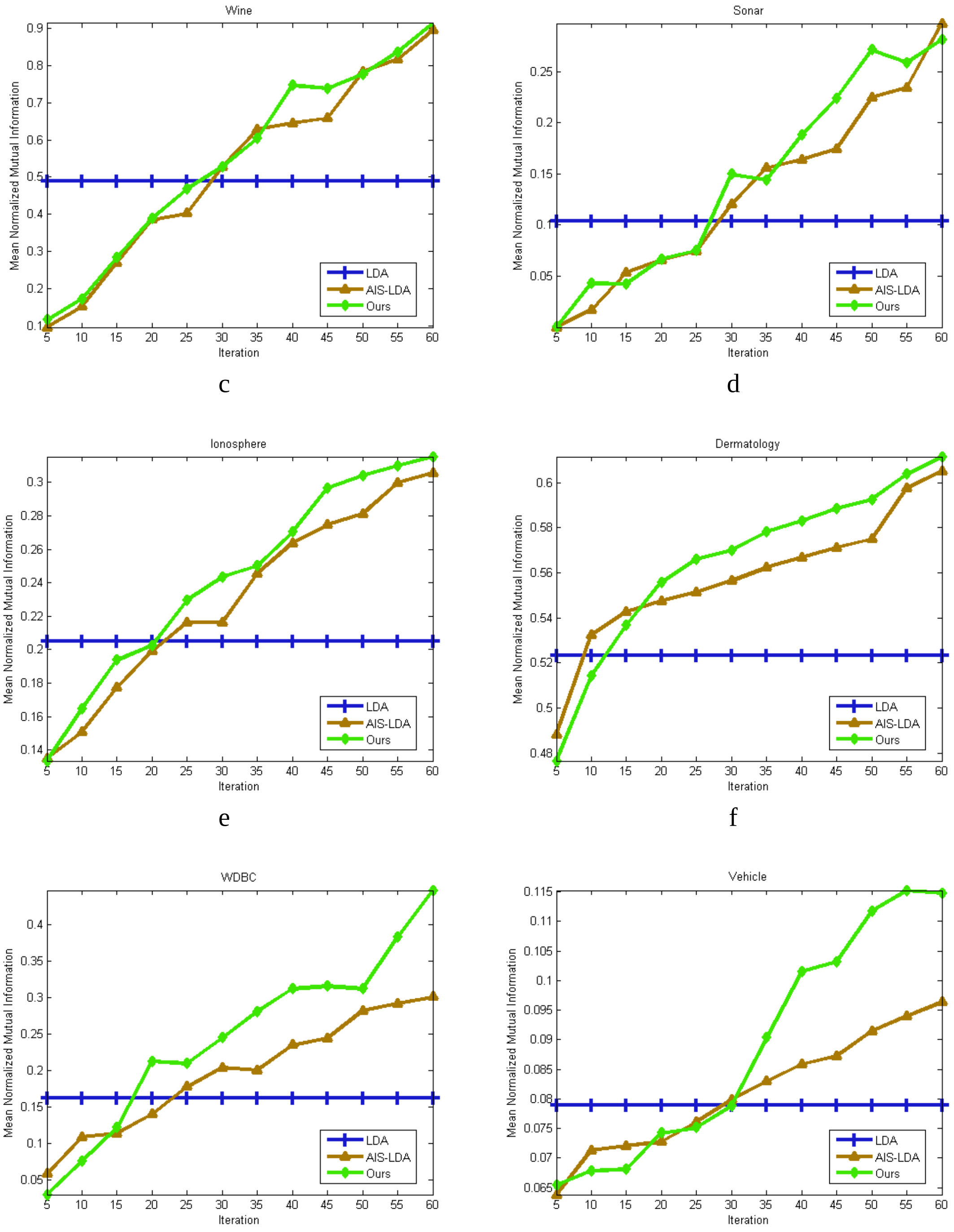
$$NMI(Z,Y) = \frac{I(Z,Y)}{\sqrt{H(Z)H(Y)}}$$

5.3. Result

Result of applying the introduced algorithms to UCI and real world datasets measured by NMI is reported in Figure 2.



a b


Wine
Mean Normalized Mutual Information
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
Iteration
5 10 15 20 25 30 35 40 45 50 55 60
LDA
AIS-LDA
Ours
Sonar
Mean Normalized Mutual Information
0.05
0.1
0.15
0.2
0.25
Iteration
5 10 15 20 25 30 35 40 45 50 55 60
LDA
AIS-LDA
Ours
Ionosphere
Mean Normalized Mutual Information
0.14
0.16
0.18
0.2
0.22
0.24
0.26
0.28
0.3
Iteration
5 10 15 20 25 30 35 40 45 50 55 60
LDA
AIS-LDA
Ours
Dermatology
Mean Normalized Mutual Information
0.48
0.5
0.52
0.54
0.56
0.58
0.6
Iteration
5 10 15 20 25 30 35 40 45 50 55 60
LDA
AIS-LDA
Ours
WDBC
Mean Normalized Mutual Information
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
Iteration
5 10 15 20 25 30 35 40 45 50 55 60
LDA
AIS-LDA
Ours
Vehicle
Mean Normalized Mutual Information
0.065
0.07
0.075
0.08
0.085
0.09
0.095
0.1
0.105
0.11
0.115
Iteration
5 10 15 20 25 30 35 40 45 50 55 60
LDA
AIS-LDA
Ours

c

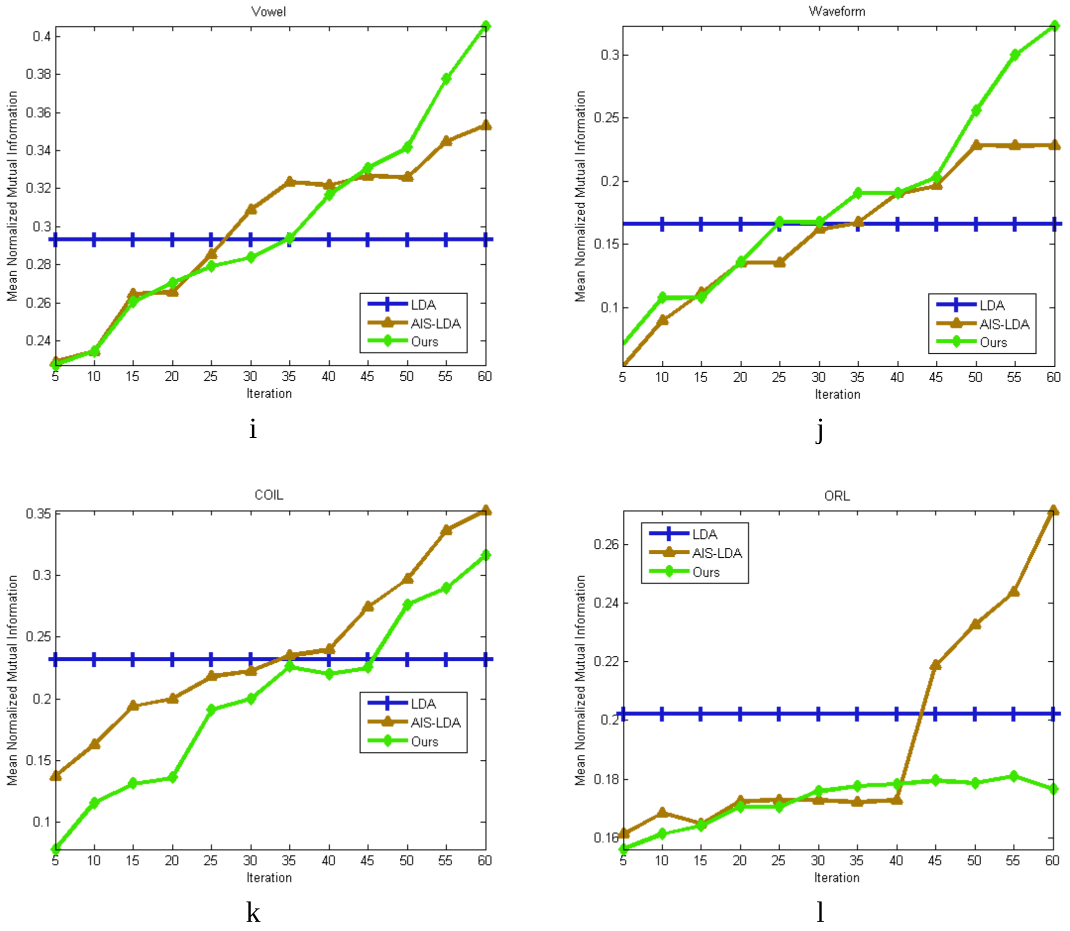
d

e

f

g

h



i

j

k

l

Figure 2. Comparison of running our algorithm compared with LDA and AIS-LDA algorithms. a) Soybeans, b) Iris, c) Wine, d) Sonar, e) Ionosphere, f) Dermatology, g) WDBC, h) Vehicle, i) Vowel, j) Waveform, k) COIL, l) ORL

By these experiments, it can be seen that our proposed method can effectively extract features, which finally provide a better accuracy in KNN classifier. It worth noting that AIS-LDA reached a better set of features in two data sets. These datasets are including ORL and Iris.

6. Conclusion

In this paper we proposed a new artificial immune system based linear discriminant analysis algorithm. This algorithm use trace ratio cost function as AIS fitness function and find the optimal projection matrix where the between class distance maximized while the within class distances minimized. The proposed algorithm is not only have higher accuracy, but it has lower computational cost. Thus it is superior in the speed. The proposed algorithm is then experimentally compared to other recently proposed AIS based LDA. The experimental results show the effectiveness of proposed algorithm.

References

[1] K. Fukunaga, "Introduction to statistical pattern classification", Academic Press, San Diego, California, USA, vol. 1, **(1990)**, pp. 2.

[2] J. Ye and T. Xiong, "Computational and theoretical analysis of null space and orthogonal linear discriminant analysis", The Journal of Machine Learning Research, vol. 7, **(2006)**, pp. 1183-1204.

[3] C. M. Bishop, "Pattern recognition and machine learning", Springer New York, vol. 4, **(2006)**.

[4] I. Jolliffe, Principal component analysis: Wiley Online Library, **(2005)**.

[5] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell, "Distance metric learning, with application to clustering with side-information," Advances in Neural Information Processing Systems, vol. 15, **(2002)**, pp. 505-512.

[6] R. A. Fisher, "The use of multiple measurements in taxonomic problems", Annals of Human Genetics, vol. 7, **(1936)**, pp. 179-188.

[7] S. Balakrishnama and A. Ganapathiraju, "Linear discriminant analysis-a brief tutorial", Institute for Signal and information Processing, **(1998)**.

[8] J. Ye, "Characterization of a family of algorithms for generalized discriminant analysis on undersampled problems", Journal of Machine Learning Research, vol. 6, **(2006)**, pp. 483.

[9] P. W. Wahl and R. A. Kronmal, "Discriminant functions when covariances are unequal and sample sizes are moderate," Biometrics, **(1977)**, pp. 479-484.

[10] J. Lu, K. N. Plataniotis and A. N. Venetsanopoulos, "Regularized discriminant analysis for the small sample size problem in face recognition", Pattern Recognition Letters, vol. 24, **(2003)**, pp. 3079-3087.

[11] M. Mohammadi, B. Raahemi, A. Akbari, B. Nassersharif and H. Moeinzadeh, "Improving linear discriminant analysis with artificial immune system-based evolutionary algorithms", Information Sciences, vol. 189, **(2012)**, pp. 219-232.

[12] Y. F. Guo, S. J. Li, J. Y. Yang, T. T. Shu and L. D. Wu, "A generalized Foley-Sammon transform based on generalized fisher discriminant criterion and its application to face recognition", Pattern Recognition Letters, vol. 24, **(2003)**, pp. 147-158.

[13] J. Mao and A. K. Jain, "Artificial neural networks for feature extraction and multivariate data projection", Neural Networks, IEEE Transactions, vol. 6, **(1995)**, pp. 296-317.

[14] H. Wang, S. Yan, D. Xu, X. Tang and T. Huang, "Trace ratio vs. ratio trace for dimensionality reduction", Computer Vision and Pattern Recognition, CVPR'07. IEEE Conference on, **(2007)**, pp. 1-8.

[15] A. R. Webb, K. D. Copsey and G. Cawley, "Statistical pattern recognition: Wiley", **(2011)**.

[16] L. -F. Chen, H. -Y. M. Liao, M. -T. Ko, J. -C. Lin and G. -J. Yu, "A new LDA-based face recognition system which can solve the small sample size problem", Pattern Recognition, vol. 33, **(2000)**, pp. 1713-1726.

[17] J. Ye, R. Janardan, C. H. Park and H. Park, "An optimization criterion for generalized discriminant analysis on undersampled problems", Pattern Analysis and Machine Intelligence, IEEE Transactions, vol. 26, **(2004)**, pp. 982-994.

[18] D. Chu, S. T. Goh and Y. Hung, "Characterization of all solutions for undersampled uncorrelated linear discriminant analysis problems", SIAM Journal on Matrix Analysis and Applications, vol. 32, **(2011)**, pp. 820-844.

[19] A. E. Eiben and J. E. Smith, "Introduction to evolutionary computing: Springer", **(2008)**.

[20] D. E. Goldberg, "Genetic algorithms in search, optimization, and machine learning", **(1989)**.

[21] E. K. Burke and G. Kendall, "Search methodologies: introductory tutorials in optimization and decision support techniques: Springer", **(2005)**.

[22] S. T. Powers and J. He, "A hybrid artificial immune system and Self Organising Map for network intrusion detection", Information Sciences, vol. 178, **(2008)**, pp. 3024-3042.

[23] M. Gong, L. Jiao and L. Zhang, "Baldwinian learning in clonal selection algorithm for optimization", Information Sciences, vol. 180, **(2010)**, pp. 1218-1236.

[24] J. MacQueen, "Some methods for classification and analysis of multivariate observations", **(1967)**, pp. 14.

[25] R. Poli and C. R. Stephens, "The building block basis for genetic programming and variable-length genetic algorithms", International Journal of Computational Intelligence Research, vol. 1, **(2005)**, pp. 183-197.

[26] Y. Jia, F. Nie and C. Zhang, "Trace ratio problem revisited", Neural Networks, IEEE Transactions, vol. 20, **(2009)**, pp. 729-735.

[27] F. Nie, S. Xiang, Y. Jia, C. Zhang and S. Yan, "Trace ratio criterion for feature selection", Proceedings of the 23rd national conference on Artificial intelligence, **(2008)**, pp. 671-676.

[28] E. Kaltofen and G. Villard, "On the complexity of computing determinants", computational complexity, vol. 13, **(2005)**, pp. 91-130.

[29] G. H. Golub and C. F. Van Loan, "Matrix computations", Johns Hopkins Univ Pr, **(1996)**, vol. 3.

[30] P. J. van Laarhoven and E. H. Aarts, "Simulated annealing: theory and applications", Springer, vol. 37, **(1987)**.

[31] J. Bezdek, "Pattern recognition with fuzzy objective function algorithms", NewYork: Plenum Press, **(1981)**.

Authors

Hadi Sadoghi Yazdi is currently an Associate Professor of Computer Science and Engineering at Ferdowsi University of Mashhad (FUM). He received his B.S. degree in Electrical Engineering from FUM in 1994, and received his M.S. in addition, Ph.D. degrees in Electrical Engineering from Tarbiat Modares University in 1996 and 2005, respectively. Dr. Sadoghi Yazdi has received several awards including Outstanding Faculty Award and Best System Design Award in 2007. His research interests are in the areas of Pattern Recognition, Machine Learning, Machine Vision, Signal Processing, Data Mining and Optimization.