

Detection of erythemato-squamous diseases using AR-CatfishBPSO-KSVM

Davar Giveki¹, Hamid Salimi², Amir Akhavan Bitaraf³, Younes Khademian⁴

¹Department of Computer Engineering University of Ayatollah Boroujerdi,
Boroujerd, Iran

s9dagive@stud.uni-saarland.de

²Department of Computer Science, University of Tehran, Tehran, Iran

salimi.hamid86@gmail.com

³Sama technical and vocational training college, Islamic Azad University, Borujerd
Branch Borujerd, Iran

bitarafaa@gmail.com

⁴Department of Electrical Engineering Iran University of Science and Technology,
Tehran, Iran

khademian2001@yahoo.com

Abstract

Nowadays, one of the most important usages of machine learning is diagnosis of diverse diseases. In this work, we introduces a diagnosis model based on Catfish binary particle swarm optimization (CatfishBPSO), kernelized support vector machines (KSVM) and association rules (AR) as our feature selection method to diagnose erythemato-squamous diseases. The proposed model consisted of two stages. In the first stage, AR is used to select the optimal feature subset from the original feature set. Next, based on the fact that kernel parameter setting in the SVM training procedure significantly influences the classification accuracy and CatfishBPSO is a promising tool for global searching, a CatfishBPSO based approach is employed for parameter determination of KSVM. Experimental results show that the proposed AR-CatfishBPSO-KSVM model achieves 99.09% classification accuracy using 24 features of the erythemato-squamous disease dataset which shows that our proposed method is more accurate compared to other popular methods in this literature like Support vector machines and AR-MLP (association rules - multilayer perceptron). It should be mentioned that we took our dataset from University of California Irvine machine learning database.

Keywords

Support vector machines, catfish particle swarm optimization, association rules, erythemato-squamous.

1. Introduction

The differential diagnosis of erythemato-squamous diseases is a real challenge in dermatology. Despite very little differences, all Erythemato-squamous diseases share the clinical features of erythema and scaling. There are six groups in erythemato-squamous diseases which are psoriasis, seboreic dermatitis, lichen planus, pityriasis rosea, chronic dermatitis and pityriasis rubra pilaris, which can be seen in the outpatient dermatology departments most often [1, 2]. In diagnosing of these diseases, a biopsy is vital. However, unfortunately these diseases share many histopathological features, as well. Another difficulty for the differential diagnosis is that one disease may show the features of another disease at the beginning stage and may have the characteristic features at the following stages. Firstly, patients were examined clinically with 12 features. Afterwards, skin samples were taken in order to evaluate 22 histopathological features. The values of the histopathological features are determined by analysis of the samples using a microscope [1].

Support vector machines (SVM), developed by [3], is a machine learning technique which has been studied increasingly in wide variety of research areas. Due to its many attractive characteristics, it was employed for the problem of diagnosis of erythemato-squamous diseases [4]. In [4] the author showed that SVM has great performance since it can handle a nonlinear classification effectively by mapping samples from low dimensional input space into high dimensional feature space with a nonlinear kernel function. Moreover, she compared SVM with recurrent neural network (RNN) and multilayer perceptrons (MLPs) and showed SVM is more robust and more accurate than other methods.

In SVM, the classification model is generated from the training phase with the training data. Next, classification is done based on the trained model. The largest problems encountered are setting up the SVM such as selecting the kernel function and its parameter values. Former experiments presented radial basis function (RBF) is efficient than polynomial and linear kernel functions for diagnosis of erythemato-squamous diseases [4]. So, we applied RBF as the kernel function. Two major RBF parameters applied in SVM, C and σ , must be set precisely. The parameter C represents the cost of the penalty and the choice of the value for C influences on the classification outcome. If C is too large, the classification accuracy rate is very high in the training phase, but very low in the testing phase. If C is too small, the classification accuracy rate unsatisfactory, making the model useless. Parameter σ has a much greater impact on the classification outcomes than C , because its value affects the partitioning outcome in the feature space. If we choose the value of the parameter σ very large, we will suffer from the danger of over-fitting, while a disproportionately small value, leads to under-fitting [5].

Catfish binary particle swarm optimization (CatfishBPSO), introduced by Li-Yeh Chuang, Sheng-Wei Tsai and Cheng-Hong Yang [6], is suggested to simultaneously optimize the parameter values of SVM. The CatfishBPSO can efficiently find optimal solutions in large search spaces. It is inspired by social behavior among individuals like the birds blocking or the fish grouping.

Feature selection is a preprocessing technique for analysis the data more effectively. The goal of feature selection is the selection of optimal subsets, which are necessary and sufficient for solving the problem. Feature selection enhances the predictive accuracy of algorithms by reducing the dimensionality, removing irrelevant features, and decreasing the amount of data needed for the

learning process. This can be done because not all of the available features are relevant for the classification task. So, for taking the advantages of, we applied a new feature selection technique, the so called Association Rules (AR), as feature selection method for diagnosis of erythematous-squamous diseases [7]. In [7], authors compared AR method with other feature selection algorithms. As a result, they could show that using 24 features, we can achieve high accuracy rate in our classification problem. Hence, in our work we employed AR to reduce the dimension of the input data. So, the novel diagnosis method which is based on Association Rules, CatfishBPSO and KSVM is introduced as AR-CatfishBPSO-KSVM. The proposed method consists of two stages. At first, AR is used as a dimension reduction method, and in the second stage, AR-CatfishBPSO-KSVM is trained as a classification model by new dataset. Experimental results show that our proposed method achieves high classification rate than hybrid AR-NN and stand-alone SVM.

The rest of this paper is organized as follows. Section 2 provides the diagnosis problem for erythematous-squamous diseases. Section 3 introduces AR feature selection method. Section 4 presents a general overview about SVM and multiclass SVM used in present work. In section 5, CatfishBPSO algorithm is described briefly, and section 6 presents our proposed method called AR-CatfishBPSO-KSVM. The experimental results and conclusion are discussed in section 7 and 8, respectively.

2. Erythematous-squamous diseases dataset

In this work, the UCI erythematous-squamous diseases dataset was used and analyzed [8]. There are 366 samples in this database and each sample has 34 attributes. Each sample contains 12 clinical features and 22 histopathological features. These attributes are listed in Table 1. The family history feature has the value '1' if any of these diseases has been observed in the family and '0' otherwise. The age feature represents the age of the patient. It has been missed in some samples, so it is removed in our experiments. Every other feature (clinical and histopathological) was given a degree in the range of '0' to '3' so that '0' indicates that the feature was not present, '3' indicates the largest amount possible, and '1', '2' indicate the relative intermediate values.

Table1. The UCI erythematous-squamous diseases dataset

The erythematous-squamous diseases (number of patients)	Features	
	Clinical	Histopathological
Psoriasis (111)	Feature 1: Erythema	Feature 12: Melanin incontinence
Seboric dermatitis (60)	Feature 2: Scaling	Feature 13: Eosinophils in the infiltrate
	Feature 3: Definite borders	Feature 14: PNL infiltrate
	Feature 4: Itching	Feature 15: Fibrosis of the papillary dermis
Lichen planus (71)	Feature 5: Koebner phenomenon	Feature 16: Exocytosis
	Feature 6: Polygonal papules	Feature 17: Acanthosis
Pityriasis rosea (48)	Feature 7: Follicular papules	Feature 18: Hyperkeratosis
	Feature 8: Oral mucosal involvement	Feature 19: Parakeratosis
		Feature 20: Clubbing of the rete ridges
Cronic dermatitis		Feature 21: Elongation of the rete ridges

(48)		Feature 9: Knee and elbow involvement	Feature 22: Thinning of the suprapapillary epidermis
Pityriasis	rubra	Feature 10: Scalp involvement	Feature 23: Spongiform pustule
pilaris (20)		Feature 11: Family history, (0 or 1)	Feature 24: Munro microabcess
		Feature 34: Age (linear)	Feature 25: Focal hypergranulosis
			Feature 26: Disappearance of the granular layer
			Feature 27: Vacuolisation and damage of basal layer
			Feature 28: Spongiosis
			Feature 29: Saw-tooth appearance of retes
			Feature 30: Follicular horn plug
			Feature 31: Perifollicular parakeratosis
			Feature 32: Inflammatory monoluclear infiltrate
			Feature 33: Band-like infiltrate

3. Feature selection

Feature selection plays an important role in classifying systems such as neural networks (NNs). The goal of this technique is to remove the irrelevant and redundant attributes from a dataset, thus the dataset dimension will be reduced. As the result, the complexity of NNs is reduced and its performance can be increased. There are many popular methods for dealing with this problem such as: principle component analysis (PCA) [9], linear discriminant analysis (LDA) [10], forward feature selection (FFS) and backward feature selection (BFS) [11,12], Individual feature selection (IFS) [13], association rules [14] and etc. based on the result reported by [7], we applied AR method to reduce the dimension of erythemato-squamous diseases dataset.

3.1. Association rules

AR finds interesting associations (relationships) among large set of data items. Actually, it shows which attributes are most likely to occur together in a given dataset. A typical and widely-used example of association rules mining is market basket analysis [15]. Let $I = (i_1, i_2, \dots, i_m)$ be a set of literals called items, and D be a database of transaction, where each transaction T is a set of items such that $T \subseteq I$. For a given itemset $X \subseteq I$ and a given transaction T , we say that T contains X if and only if $X \subseteq T$. The support count of an itemset X is defined to be $\text{sup}_X =$ the number of transactions in D that contain X . We say that an itemset X is *large*, with respect to a support threshold of $s\%$, if $\text{sup}_X \geq |D| \cdot s\%$, where $|D|$ is the number of transactions in the database D . An association rules is an implication of the form $X \Rightarrow Y$, where $X \subseteq I, Y \subseteq I$ and $X \cap Y = \emptyset$. The AR $X \Rightarrow Y$ is said to hold in database D with confidence $c\%$ if no less than $c\%$ of the transactions in D (that contain X) also contain Y . The rule $X \Rightarrow Y$ has support $s\%$ in D if $\text{sup}_{X \cup Y} = |D| \cdot s\%$. Therefore, AR tries to discover the patterns of co-occurrence of attributes in a given database. For instance, an association rule in a supermarket basket data may be in 10% of transactions, 85% of people buying milk also buy yoghurt in that transaction [7,15].

3.2 Apriori algorithm

The most popular and influential AR mining algorithm is Apriori algorithm proposed by Agrawa et al. [14] (figure 1). The first pass of the algorithm simply counts item occurrences to determine the 1-itemsets. A subsequent pass k consist of two phases. First, the large itemsets L_{k-1} found in the $(k-1)$ -th pass are applied to generate the candidate itemsets C_k . Next, the database is scanned and the support of candidate in C_k is counted.

```

(1)   $L_1 = \{\text{large 1-itemsets}\};$ 
(2)  for ( $k = 2; L_{k-1} \neq \emptyset; k++$ ) do begin
(3)     $C_k = \text{apriori-gen}(L_{k-1});$ 
(4)    forall transactions  $t \in D$  do begin
(5)       $C_t = \text{subset}(C_k, t);$ 
(6)      forall candidates  $c \in C_t$  do
(7)         $c.\text{count}++;$ 
(8)      end
(9)       $L_k = \{c \in C_k \mid c.\text{count} \geq \text{min sup}\}$ 
(10) end
(11) return  $\bigcup_k L_k$ 

```

Fig. 1. Apriori Algorithm

The *apriori-gen* function computes the set of potentially large k -itemsets (Fig. 2.) from the set of large $(k-1)$ -itemsets, and *subset* function finds all the candidates contained in transaction T.

```

(1)  procedure LargeItemsets
(2)  begin
(3)    let Large set  $L = \emptyset;$ 
(4)    let Frontier set  $F = \{\emptyset\};$ 
(5)    while  $F \neq \emptyset$  do begin
(6)      -- make a pass over the database
(7)      let Candidate set  $C = \emptyset;$ 
(8)      forall database tuples  $t$  do
(9)        forall itemsets  $f$  in  $F$  do
(10)       if  $t$  contains  $f$  then begin
(11)        let  $C_f$  candidate itemsets that are extensions
                of  $f$  and contained in  $t$ ;
(12)        forall itemsets  $c_f$  in  $C_f$  do
(13)          if  $c_f \in C$  then
(14)             $c_f.\text{count} = c_f.\text{count} + 1;$ 
(15)          else begin
(16)             $c_f.\text{count} = 0;$ 
(17)             $C = C + c_f;$ 
(18)          end
(19)        end

```

```

(20)      end
(21)  -- consolidate
(22)  let  $F = \emptyset$ ;
(23)  forall itemsets  $c$  in  $C$  do begin
(24)    if  $\text{count}(c)/\text{dbsize} > \text{min}$  then
(25)       $L = L + c$ ;
(26)    if  $c$  should be used as a frontier in the next pass then
(27)       $F = F + c$ ;
(28)    end
(29)  end
(30)  end

```

Fig. 2. Algorithm for finding LargeItemsets

4. Support vector machines

SVM classifier is a supervised learning method based on statistical learning theory introduced by [3], whose goal is to give a hyper plane that optimally separates two classes using training dataset. Suppose that $\{x_i, y_i\}_{i=1}^N$ are our training dataset, where x is the input sample, and $y \in \{+1, -1\}$ is class label. The hyper plane is defined as $w \cdot x + b = 0$, where x is a point lying on the hyper plane, w determines the orientation of the hyper plane, and b is the bias of the distance of hyper plane from the origin. As shown in Figure 3, the optimum separating hyper plane can be found by minimizing $\|w\|^2$ under the constraint $y_i(w \cdot x_i + b) \geq 1, i = 1, 2, \dots, n$. Thus, for determining the optimum hyper plane the following optimization problem must be solved:

$$\min \frac{1}{2} \|w\|^2 \quad (1)$$

Now, in order to replace the optimization problem the positive slack variables ξ_i are introduced. So, the method can be extended to allow for nonlinear decision surfaces. The new optimization problem is given as:

$$\min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i \quad (2)$$

where C is a regularization constant or penalty parameter, which controls the tradeoff between the two competing criteria of margin maximization and error minimization. Hence, the classification decision function becomes:

$$f(y) = \text{sgn}\left(\sum_{i=1}^N \alpha_i y_i K(x_i, x_j) + b\right) \quad (3)$$

where $\alpha_i \geq 0, i = 1, 2, \dots, N$ are non-negative Lagrange multipliers that satisfy $\sum_{i=1}^N \alpha_i y_i = 1$.

$K(x_i, x_j)$ represents a symmetric positive definite kernel function that defines an inner product in the feature space [17]. In this work, radial basis function (RBF) (defined

as $K(x_i, x_j) = \exp(-\|x_i - x_j\|^2 / 2\sigma^2)$, σ is a positive real number) was applied as the kernel function.

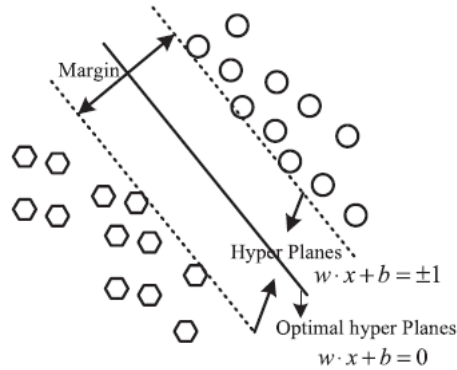


Fig. 3. The classification process of SVM classifier.

As mentioned previously, SVM is naturally a binary classifier. But, the classification of erythemato-squamous diseases includes the simultaneous discrimination of numerous diseases. For coping with this problem, a number of multi-class classification strategies can be employed [6, 15]. One-against-all (OAA) and the one-against-one (OAO) strategies are among the most popular ones. Both strategies lead to similar results in terms of classification accuracy. Here, we consider the OAA strategy. This strategy is based on the following procedure.

Let $\Omega = \{\omega_1, \omega_2, \dots, \omega_6\}$ be the set of diseases that we desire to classify. First, an ensemble of six (parallel) SVM classifiers is trained. Each classifier tries to solve a binary classification problem defined by the discrimination between one disease $\omega_i (i = 1, 2, \dots, 6)$ against all others (i.e., $\Omega - \{\omega_i\}$). Then, in the classification phase, the “winner-takes-all” rule is used to decide which disease is assigned to each patient. This means that the winning disease is the one that corresponds to the SVM classifier of the ensemble that shows the highest output.

5. Particle swarm optimization

5.1. Binary particle swarm optimization (BPSO)

In PSO, each particle is analogous to an individual “fish” in a school of fish. A swarm consists of N particles moving around a D -dimensional search space. The process of PSO is initialized with a population of random particles and the algorithm then searches for optimal solutions by continuously updating generations. Each particle takes advantage of its own memory and knowledge gained by the swarm as a whole to find the best solution. The position of the i -th particle is represented by $x_i = (x_{i1}, x_{i2}, \dots, x_{id})$. The velocity of the i -th particle is written as $v_i = (v_{i1}, v_{i2}, \dots, v_{id})$. Also, it should be mentioned that the positions and velocities of the particles are bounded within $[X_{\min}, X_{\max}]^D$ and $[V_{\min}, V_{\max}]^D$, respectively. We refer to $p_i = (p_{i1}, p_{i2}, \dots, p_{id})$ as the best previously seen position of the i -th particle which its value

called $pbest_i$. The best value of the all individual $pbest_i$ values is denoted the global best position $g_i = (g_{i1}, g_{i2}, \dots, g_{id})$ and called $gbest$. At each generation, the position and velocity of the i -th particle are updated by $pbest_i$ and $gbest$ in the swarm. But, some of the optimization problems occur in a discrete feature space. For this reason, Kennedy and Eberhart introduced binary PSO (BPSO), which can be used to discrete binary variables. In a binary space, a particle may move to near corners of a hypercube by flipping diverse numbers of bits; so, the overall particle velocity may be denoted by the number of bits changed per iteration [22]. In BPSO, each particle is updated based on the following equations:

$$v_{id}^{new} = w \times v_{id}^{old} + c_1 \times r_1 \times (pbest_{id} - x_{id}^{old}) + c_2 \times r_2 \times (gbest_{id} - x_{id}^{old}) \quad (4)$$

$$\text{If } S(v_{id}^{new}) \text{ then } v_{id}^{new} = \max(\min(V_{\min}, v_{id}^{new}), V_{\min}) \quad (5)$$

$$S(v_{id}^{new}) = \frac{1}{1 + e^{-v_{id}^{new}}} \quad (6)$$

$$\text{If } (r_3 < S(v_{id}^{new})) \text{ then } x_{id}^{new} = 1 \text{ else } x_{id}^{new} = 0 \quad (7)$$

In these equations, w is the inertia weight that controls the effect of the previous velocity of a particle on its current one, r_1, r_2 , and r_3 are random numbers between $(0,1)$, and c_1 and c_2 are acceleration constants, which control how far a particle will move in a single generation. Velocities v_{id}^{new} and v_{id}^{old} denote the velocities of the new and old particle, respectively. x_{id}^{old} is the current particle position, and x_{id}^{new} is the new, updated particle position. In Eq. (5), particle velocities of each dimension are tried to a maximum velocity V_{\max} . If the sum of accelerations causes the velocity of that dimension to exceed V_{\max} , then the velocity of that dimension is limited to V_{\max} . V_{\max} and V_{\min} are parameters which determined by users (in our case $V_{\max} = 6$, $V_{\min} = -6$). The position of particles after updating is computed by the function $S(v_{id}^{new})$ (Eq. (6)). If $S(v_{id}^{new})$ is larger than r_3 , then its position value is represented by $\{1\}$. If $S(v_{id}^{new})$ is smaller than r_3 , then its position value is represented by $\{0\}$ (meaning this position is not selected for the next update) [19].

5.2. Catfish binary particle swarm optimization (CatfishBPSO)

CatfishBPSO introduces catfish particles to motivate a renewed search by the ‘‘sardine’’ particles. These catfish particles help particles which are trapped in a local optimum to escape from local optimum and guide them towards a new region of the search space, and thus to potentially better particle solutions. In CatfishBPSO, a particle swarm is randomly initialized in the first step, and the particles are scattered over the D -dimensional search space. The position of each particle is denoted in binary string form; the bit value $\{0\}$ and $\{1\}$ show a non-selected and selected feature, respectively. Each particle is updated using values $pbest$ and $gbest$. The $pbest$

is the best solution (fitness) it has achieved so far and g_{best} is the global best value obtained so far by all particles. The position and velocity of each particle are updated by Eqs. (4)–(7). If the distance between g_{best} and the encompassing particles is small, each particle is considered as a part of the cluster around g_{best} and will only move a very small distance in the next generation. To tackle this problem, catfish particles are applied and substituted the 10% of original particles with the worst fitness value of the swarm. These catfish particles are vital for the success of optimization task. So, catfish particles are able to overcome the intrinsic deficiency (premature convergence) of BPSO by initializing a new search over the entire search space from its farthest points [6, 20].

5.3. K-nearest neighbor

The K-nearest neighbor (K-NN) method is one of the most famous nonparametric techniques [21, 22] used for classification of new objects based on attributes and training samples. K-NN includes a supervised learning algorithm which immediately classifies the result of a query instance based on the majority of the K-nearest neighbor category. Classifiers are based on the minimum distance from the query instance to the training samples. In our work, the feature subset was measured by the leave-one-out cross-validation (LOOCV) of one nearest neighbor (1-NN). Neighbors are determined by their Euclidean distance. The 1-NN classifier does not require any user-specified parameters. In the LOOCV technique, we select a single observation from the original sample as the validation data, and the rest of the observations make up the training data. This process is repeated so that each observation in the sample is used once as the validation data. The procedure is the same as *K-fold* cross validation where K is equal to the number of observations in the original sample. The pseudo-codes for CatfishBPSO and 1-NN are shown below [6].

CatfishBPSO pseudo-code

```

01: Begin
02: Randomly initialize particles swarm
03: while (number of iterations, or the stopping criterion is
    not met)
04: Evaluate fitness of particle swarm by 1-NN()
05: for n = 1 to number of particles
06: Find  $pbest_i$  and  $g_{best}$ 
07: for d = 1 to number of dimension of particle
08: update the position of particles by Eqs. (4)–(7)
09: next d
10: next n
11: if fitness of  $g_{best}$  is the same three times then
12: Sort the particle swarm via fitness from best to
    worst
13: for n = number of Nine-tenth particles to number
    of particles
14: if rand number >0.5 then
15: for d = 1 to number of dimension of particle
16: The position of catfish = 1 (Max of the
    search space)

```

```

17:         next d
18:     else
19:         for d = 1 to number of dimension of particle
20:             The position of catfish = 0 (Min of the
search space)
21:         next d
22:     next n
23: end if
24: next generation until stopping criterion
25: end

```

1-NN pseudo-code

```

01: begin
02:     for i = 1 to sample number of classification problem
03:         for j = 1 to sample number of classification problem
04:             for k = 1 to dimension number of classification
problem
05:                  $dist_i = dist_i + (data_{ik} - data_{jk})^2$ 
06:             next k
07:             if  $dist_i < nearest$  then
08:                  $class_i = class_j$ 
09:                  $nearest = dist_i$ 
10:             end if
11:         next j
12:     next i
13:     for i = 1 to sample number of classification problem
14:         if  $class_i = real\ class\ of\ testing\ data$  then
correct = correct + 1
15:         end if
16:     next i
17:     Fitness value = correct / number of testing data
18: end

```

In our experiment, we set the population size = 20, and the $K=1$ [23] in part of K-Nearest Neighbor. The other parameters of CatfishBPSO are set as follows: generations = 70 (maximum number of iterations), $w=1.0$ and $c_1 = c_2 = 2$ [24].

6. Proposed multiclass AR-CatfishBPSO-KSVM classification system

The process of the classification based on AR-CatfishBPSO-KSVM is shown in Figure 4. The aim of this system is to optimize the accuracy of the multi-class SVM classifier by automatically: 1) Reducing the number of features with association rules. 2) Obtaining the best values of C and σ parameters of SVM. First of all, dataset is divided into training set and test set. Then AR method is applied to extract 24 features from training data. It is obvious that the features with

higher weights are the most important and influential ones among the others and they can play a vital role in the final results of classification. So, in feature selection, 24 features which carry more weights were selected from dataset using AR method. Using this algorithm large itemsets are found for all diseases, as you can see in the following:

psoriasis disease	6-7-8-12-13-15-25-27-28-29-30-31-33
seboric dermatitis disease	5-6-7-8-9-11-12-15-20-22-24-25-26-27-29-30-31-33
lichen planus disease	7-9-10-11-14-15-20-21-22-23-24-30-31
pityriasis rosea disease	6-7-8-9-10-11-12-13-15-20-21-22-23-24-25-27-29-30-31-33
chronic dermatitis disease	5-6-8-9-10-11-12-13-14-20-22-23-24-25-26-27-29-30-31-33
pityriasis rubra pilaris disease	5-6-8-9-10-11-12-13-14-20-22-23-24-25-26-27-29-30-31-33

For instance we can use the features number 7-9-10-11-14-15-20-21-22-23-24-30-31 for diagnosis of the lichen planus disease. As we mentioned earlier in introduction, all Erythemato-squamous diseases share the clinical features of erythema and scaling, with very little differences and there are six groups in erythemato squamous diseases. So, for detection of erythemato-squamous diseases, 5th, 6th, 7th, 8th, 9th, 10th, 11th, 12th, 13th, 14th, 15th, 20th, 21th, 22th, 23th, 24th, 25th, 26th, 27th, 28th, 29th, 30th, 31th and 33th features can be used. So we only used these features in training and testing our proposed model.

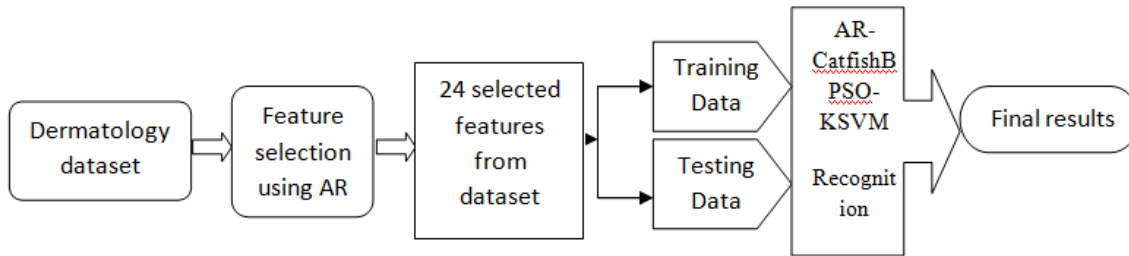


Fig. 4. The model of diagnosis of erythemato-squamous diseases based on AR-CatfishBPSO-KSVM.

And in the last stage the multi-class SVM is chosen to classify patients. The SVM constructed by RBF kernel function only has two parameters, C and σ , which should be determined by the users. Here we use CatfishBPSO for selecting the best initial values for these parameters. Now, we can describe the training process in the following:

- (1) Initialize parameters of CatfishBPSO for six AR-CatfishBPSO-KSVMs. The training dataset of each AR-CatfishBPSO-KSVM is different from the others as described in section 4. For each AR-CatfishBPSO-KSVM classifier, initialize all particles' positions and velocity and set up the learning parameters, the inertia weight and the maximum number of iterations.
- (2) For each AR-CatfishBPSO-KSVM classifier, calculate every particle's fitness. Every particle's fitness is computed by the formula $\frac{classified}{total}$ ($total$ denotes the number of training samples and $classified$ denotes the number of correct classified samples). The particle with low classification error produces a high fitness value.
- (3) Update the velocity and position of the particle. According to Eqs. (4)-(7).

- (4) Stop if the specified number of generations is satisfied; otherwise, go to Step 3.

7. Experimental results

The classifiers proposed for clinical decision-making were implemented by using the MATLAB software package (MATLAB version 7.2 with neural networks toolbox). The features discussed in section 6 were used as the inputs of the classifiers. The dataset (366 patients) was divided into two separate datasets, the training and testing datasets (182 samples as training data and rest of them as testing data). The related parameters of C and σ for RBF kernel function were varied in the arbitrarily fixed ranges $[10^{-3}, 200]$ and $[10^{-3}, 2]$ so as to cover high and small regularization of the classification model, and fat as well as thin kernels, respectively. For comparison, we implemented the SVM classifier without CatfishBPSO with RBF and the polynomial kernels, termed as SVM and SVM-poly, respectively. The degree d of the polynomial kernel was varied in the range $[2, 5]$ in order to span polynomials with low and high flexibility. Classification results of the classifiers were displayed by a confusion matrix. In a confusion matrix, each cell contains the raw number of exemplars classified for the corresponding combination of desired and actual network outputs. The confusion matrices showing the classification results of the classifiers implemented for diagnosis of erythemato-squamous diseases are given in Table 2. From these matrices one can tell the frequency with which a record is misclassified as another.

Table 2. Confusion matrices of the classifiers

Classifier	Desired result	Output result					
		Psoriasis	Seboreic dermatitis	Lichen planus	Pityriasis rosea	Chronic dermatitis	Pityriasis rubra pilaris
AR-CatfishBPSO-SVM	Psoriasis	56	0	0	0	0	0
	Seboreic dermatitis	0	31	0	0	0	0
	Lichen planus	0	0	35	0	0	0
	Pityriasis rosea	0	0	1	25	1	0
	Chronic dermatitis	0	0	0	0	25	0
	Pityriasis rubra pilaris	0	0	0	0	0	10
	AR-MLP	Psoriasis	54	0	0	0	0
	Seboreic dermatitis	1	31	1	0	0	0

SVM	Lichen planus	0	0	35	1	1	0
	Pityriasis rosea	1	0	0	24	0	0
	Chronic dermatitis	0	0	0	0	25	0
	Pityriasis rubra pilaris	0	0	0	0	0	10
	Psoriasis	53	0	0	0	0	0
	Seboreic dermatitis	1	30	1	1	0	0
	Lichen planus	1	1	34	1	1	0
	Pityriasis rosea	1	0	1	23	1	0
	Chronic dermatitis	0	0	0	0	24	0
	Pityriasis rubra pilaris	0	0	0	0	0	10

The test performance of the classifiers can be determined by the computation of specificity, sensitivity and total classification accuracy. The specificity, sensitivity and total classification accuracy are defined as:

Specificity: Number of true negative decisions/number of actually negative cases.

Sensitivity: Number of true positive decisions/number of actually positive cases.

Total classification accuracy: Number of correct decisions/ total number of cases.

A true positive decision occurs when the positive prediction of the network coincided with a positive prediction of the physician. A true negative decision occurs when both the network and the physician suggested the absence of a positive prediction.

Table 3. The values of the statistical parameters of the classifiers

Classifier	Datasets	Statistical parameters (%)		
		Sensitivity	Specificity	Total classification accuracy
AR-CatfishBPSO-SVM	Psoriasis	100	100	99.09
	Seboreic dermatitis	100	100	
	Lichen planus	98.42	100	

	Pityriasis rosea	100	94.56	
	Chronic dermatitis	96.15	100	
	Pityriasis rubra pilaris	100	100	
AR-MLP	Psoriasis	96.42	100	97.28
	Seboreic dermatitis	100	93.93	
	Lichen planus	97.22	94.59	
	Pityriasis rosea	96.00	96.00	
	Chronic dermatitis	96.15	100	
	Pityriasis rubra pilaris	100	100	
SVM	Psoriasis	94.64	100	94.56
	Seboreic dermatitis	96.77	90.90	
	Lichen planus	94.44	89.47	
	Pityriasis rosea	92.00	88.46	
	Chronic dermatitis	92.30	100	
	Pityriasis rubra pilaris	100	100	

8. Conclusion

In this paper a novel approach based on Catfish binary particle swarm optimization-based was proposed in order to search for the optimal parameter values for KSVM for diagnosis of erythemato-squamous diseases. Moreover, AR method was applied to find optimal subset of features in both training and testing phase as a dimensionality reduction. Using the catfish effect the proposed method does not trap in local optimum. In addition, taking RBF kernel into account increased the performance of the classification. Compared to other approaches such as AR-MLP and pure SVM, experimental results show that our proposed method is more accurate and it can achieve a 99.09% of accuracy.

References

- [1] Güvenir, H. A., Demiröz, G., & İter, N. (1998). Learning differential diagnosis of erythematous diseases using voting feature intervals. *Artificial intelligence in medicine*, 13, 147-165.
- [2] Güvenir, H. A., & Emeksiz, N. (2000). An expert system for the differential diagnosis of erythematous diseases. *Expert Systems with Applications*, 18, 43-49.
- [3] Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. New York.
- [4] Übeyli, E. D. (2008). Multiclass support vector machines for diagnosis of erythematous diseases. *Expert Systems with Applications*, 35, 1733-1740.
- [5] Pardo, M., & Sberveglieri, G. (2005). Classification of electronic nose data with support vector machines. *Sensors and Actuators B: Chemical*, 107, 730-737.
- [6] Chuang, L. Y., Tsai, S. W., & Yang, C. H. (2011). Improved binary particle swarm optimization using catfish effect for feature selection. *Expert Systems with Applications*. vol 38, Sep 2011, PP 12699-12707.
- [7] Karabatak, M., & Ince, M. C. (2009). A new feature selection method based on association rules for diagnosis of erythematous diseases. *Expert Systems with Applications*, 36, 12500-12505.
- [8] Blake C. L. , M. C. J. (1998). *UCI repository of machine learning databases*.
- [9] Chatterjee, C., Roychowdhury, V. P., & Chong, E. K. P. (1998). On relative convergence properties of principal component analysis algorithms. *Neural Networks, IEEE Transactions on*, 9, 319-329.
- [10] Duda, R., Hart, P., Stork, D. (2001). *Pattern classification (2nd ed.)*. New York: John Wiley and Sons.
- [11] Hyvärinen, A. (1999). Fast and robust fixed-point algorithms for independent component analysis. *Neural Networks, IEEE Transactions on*, 10, 626-634.
- [12] Hyvärinen, A., & Oja, E. (2000). Independent component analysis: algorithms and applications. *Neural Networks*, 13, 411-430.
- [13] Narendra, P. M., & Fukunaga, K. (1977). A Branch and Bound Algorithm for Feature Subset Selection. *Computers, IEEE Transactions on*, C-26, 917-922.
- [14] Agrawal, R., Imieliski, T., Swami, A., 1993. Mining association rules between sets of items in large databases. *SIGMOD Rec.* 22, 207-216.
- [15] Agrawal, R., Srikant, R. (1994). Fast algorithms for mining association rules in large databases. In *Proceedings of 20th international conference on very large databases*, 487-499, Santiago, Chile.
- [16] Wei, J., Jian-qi, Z., & Xiang, Z. (2011). Face recognition method based on support vector machine and particle swarm optimization. *Expert Systems with Applications*, 38, 4390-4393.
- [17] Chih-Wei, H., & Chih-Jen, L. (2002). A comparison of methods for multiclass support vector machines. *Neural Networks, IEEE Transactions on*, 13, 415-425.
- [18] Melgani, F., & Bruzzone, L. (2004). Classification of hyperspectral remote sensing images with support vector machines. *Geoscience and Remote Sensing, IEEE Transactions on*, 42, 1778-1790.

- [19] Kennedy, J., & Eberhart, R. C. (1997). A discrete binary version of the particle swarm algorithm. In IEEE international conference on systems, man, and cybernetics, computational cybernetics and simulation (Vol. 5, pp. 4104–4108, 12–15).
- [20] Chuang, L. Y., Tsai, S. W., & Yang, C. H. (2008). Catfish particle swarm optimization. In IEEE swarm intelligence symposium 2008 (SIS 2008), St. Louis, Missouri (p. 20).
- [21] Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. Proceedings of the IEEE Transaction on Information Theory, 13(1), 21–27.
- [22] Fix, E., & Hodges, J. L. (1951). Discriminatory analysis-nonparametric discrimination: Consistency properties. Project 21-49-004, Report 4, US Air Force School of Aviation Medicine, Randolph Field (pp. 261–279).
- [23] Oh, I.-S., Lee, J.-S., & Moon, B.-R. (2004). Hybrid genetic algorithms for feature selection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(11), 1424–1437.
- [24] Kennedy, J., Eberhart, R. C., & Shi, Y. (2001). Swarm intelligence. San Francisco: Morgan Kaufmann Publishers.