Differential Evolution Based Nearest Prototype Classifier with Optimized Distance Measures for the Features in the Data Sets

David Koloseni [a,d], Jouni Lampinen[b,c], Pasi Luukka [a,e]

[a] *Laboratory of Applied Mathematics,Lappeenranta University of Technology,P.O. Box 20, FI-53851 Lappeenranta, Finland*

[b] *Department of Computer Science,University of Vaasa,P.O. Box 700, FI-65101 Vaasa, Finland*

[c] *Department of Computer Science, VSB-Technical University of Ostrava,17. listopadu 15,70833 Ostrava-Poruba, Czech Republic.*

[d] *University of Dar es salaam,Department of Mathematics, P.O.Box 35062 Dar es salaam,Tanzania*

[e] *School of business,Lappeenranta University of Technology,P.O. Box 20, FI-53851 Lappeenranta, Finland*

Abstract

In this paper a further generalization of differential evolution based data classification method is proposed, demonstrated and initially evaluated. The differential evolution classifier is a nearest prototype vector based classifier that applies a global optimization algorithm, differential evolution, for determining the optimal values for all free parameters of the classifier model during the training phase of the classifier. The earlier version of differential evolution classifier that applied individually optimized distance measure for each new data set to be classified is generalized here so, that instead of optimizing a single distance measure for the given data set, we take a further step by proposing an approach where distance measures are optimized individually for each feature of the data set to be classified. In particular, distance measures for each feature are selected optimally from a predefined pool of alternative distance measures. The optimal distance measures are determined by differential evolution algorithm, which is also determining the optimal values for all free parameters of the selected distance measures in

Email addresses: David.Koloseni@lut.fi (David Koloseni [a,d]), jouni.lampinen@uwasa.fi (Jouni Lampinen[b,c]), Pasi.Luukka@lut.fi (Pasi Luukka [a,e])

parallel. After determining the optimal distance measures for each feature together with their optimal parameters, we combine all featurewisely determined distance measures to form a single total distance measure, that is to be applied for the final classification decisions. The actual classification process is still based on the nearest prototype vector principle; A sample belongs to the class represented by the nearest prototype vector when measured with the above referred optimized total distance measure. During the training process the differential evolution algorith determines optimally the class vectors, selects optimal distance metrics for each data feature, and determines the optimal values for the free parameters of each selected distance measure. Based on experimental results with nine well known classification benchmark data sets, the proposed approach yield a statistically significant improvement to the classification accuracy of differential evolution classifier.

Keywords:
Differential evolution, Classification, Distance measures, Distance selection for the feature, Pool of distances

---

1. Introduction

Differential evolution algorithm is gaining a fast popularity in classification problems. Some include bankruptcy prediction [1], classification rule discovery [2], feature selection, [3], edge detection in images [4]. Also in some methods classification techniques are used to improve optimization with differential evolution algorithm [5]. The global optimization problems arises in many fields of science, engineering and business [6]. In order to achieve the best available classification performance, we typically need to find the best classification model for the current data set to be classified, and the best opimization algorithm for determining the values of its free parameters during the training phase. One of the emerged global optimization methods is the differential evolution (DE) algorithm [7] that belongs evolutionary computation approaches. DE has also been applied in many areas of pattern recognition, i.e. in remote sensing imagery [8], hybrid evolutionary learning in pattern recognition systems [9], in neural network based learning algorithms [10], [11], in clustering [12],[13], [14], [15], [16] and in classification [17], [18], [19] to mention few. Here we tackle classification problems by extending our Differential Evolution (DE) based classifier [20],[21], [22] further by continuing our generalization on extending it for data featurewise distance measure optimization. In our earlier work [22] we generalized our original DE classifier version [20] by extending the optimization to cover

several distances from the pool of distances. The distances in [22] were vector based distances so that optimal distance found, was always applied to all features of the particular data set which we were classifying. However, in classification of data sets, data set is usually in a form where each sample consists of several measurements, and it is not guaranteed that all the measured values of a sample obey the same optimal distance measure. It is clear that the optimal distance measure for one feature may not be optimal for another feature. This is the underlying motivation for further generalizing our method so, that instead of optimizing the vector based distance measure we concentrate on the problem at the feature level and optimize the suitable distance for the particular feature. We repeat this process of course for all the features in the data set and finally get N different optimal distance measures that are determined individually for each feature of the data set. Also if any of the determined distance measures include one or more free parameters, these values needs to be optimized as well. After determining the optimal distance measures for each feature, to combine the total distance measure we first normalize the values of individual distance measures, and then simply calculate the sum of all normalized distance measures.

The classification model parameterization for featurewise distance measure optimization can be chosen at least in three different ways, in this paper we also investigate the influence of this choise to the classification performance.

Examples on such situations where better classification accuracy have been obtained by applying some other distance measure than the most commonly applied euclidean distance have been reported in several articles i.e. [23], [24], [25], [26], [27], [28], [29], [30], [31], [32],[33]. However, typically in these types of studies one has simply tested with a few different distance measure for classifying the data set at hand, and none of them have concentrated on distances for individual features, but vector based distances for the data set. Instead of testing the method separately with few distance measures, our goal is to concentrate on optimal selection of the distance measure from a predefined pool of distance measures. The data sets for demonstrating the proposed approach and for its empirical evaluation were taken from a UCI-Repository of Machine Learning data set [34] and KEEL data sets [35]. Classifier was implemented with $MATLAB^{TM}$-software.

2. Differential evolution based classifier with optimized distances for the features in the data sets

2.1. Differential Evolution based Classification

The DE algorithm [36], [7] was introduced by Storn and Price in 1995 and it belongs to the family of Evolutionary Algorithms (EAs). The design principles of DE are simplicity, efficiency, and the use of floating-point encoding instead of binary numbers for representing internally the solution candidates for the optimization problem to be solved. As a typical EA, DE starts with a randomly generated initial population of candidate solutions for the optimization problem to be solved that is then improved using selection, mutation and crossover operations. Several ways exist to determine a stopping criterion for EAs but usually a predefined upper limit G_{max} for the number of generations to be computed provides an appropriate stopping condition. Other control parameters for DE are the crossover control parameter CR, the mutation factor F, and the population size NP.

In each generation G, DE goes through each D dimensional decision vector $\mathbf{v}_{i,G}$ of the population and creates the corresponding trial vector $\mathbf{u}_{i,G}$ as follows in the most common DE version, DE/rand/1/bin [37]:

$r_1, r_2, r_3 \in \{1, 2, \ldots, NP\}$, (randomly selected,
  except mutually different and different from i)
$j_{rand} = \text{floor}\,(rand_i[0,1) \cdot D) + 1$
for$(j = 1; j \leq D; j = j + 1)$
{
  if$(rand_j[0,1) < CR \vee j = j_{rand})$
    $u_{j,i,G} = v_{j,r_3,G} + F \cdot (v_{j,r_1,G} - v_{j,r_2,G})$
  else
    $u_{j,i,G} = v_{j,i,G}$
}

In this DE version, NP must be at least four and it remains fixed along CR and F during the whole execution of the algorithm. Parameter $CR \in [0, 1]$, which controls the crossover operation, represents the probability that an element for the trial vector is chosen from a linear combination of three randomly chosen vectors and not from the old vector $\mathbf{v}_{i,G}$. The condition "$j = j_{rand}$" is to make sure that at least one element is different compared to the elements of the old vector. The parameter F is a scaling factor for mutation and its value is typically $(0, 1+]$[1]. In practice, CR controls the ro-

[1]Notation means that the upper limit is about 1 but not strictly defined.

tational invariance of the search, and its small value (*e.g.*, 0.1) is practicable with separable problems while larger values (*e.g.*, 0.9) are for non-separable problems. The control parameter F controls the speed and robustness of the search, *i.e.*, a lower value for F increases the convergence rate but it also adds the risk of getting stuck into a local optimum. Parameters CR and NP have the same kind of effect on the convergence rate as F has.

After the mutation and crossover operations, the trial vector $\mathbf{u}_{i,G}$ is compared to the old vector $\mathbf{v}_{i,G}$. If the trial vector has an equal or better objective value, then it replaces the old vector in the next generation. This can be presented as follows (in this paper minimization of objectives is assumed) [37]:

$$\mathbf{v}_{i,G+1} = \begin{cases} \mathbf{u}_{i,G} & \text{if} \quad f(\mathbf{u}_{i,G}) \leq f(\mathbf{v}_{i,G}) \\ \mathbf{v}_{i,G} & \text{otherwise} \end{cases} .$$

DE is an elitist method since the best population member is always preserved and the average objective value of the population will never get worse. As the objective function, f, to be minimized we applied the number of incorrectly classified learning set samples. Each population member, $\mathbf{v}_{i,G}$, as well as each new trial solution, $\mathbf{u}_{i,G}$, contains the class vectors for all classes and the power value p. In other words, DE is seeking the vector $(y(1), ..., y(T), p)$ that minimizes the objective function f. After the optimization process the final solution, defining the optimized classifier, is the best member of the last generation's, G_{max}, population, the individual $\mathbf{v}_{i,G_{max}}$. The best individual is the one providing the lowest objective function value and therefore the best classification performance for the learning set. The control parameters of DE algorithm were set as follows: CR=0.9 and F=0.5 were applied for all classification problems. NP was chosen so that it was six times the number of optimized parameters, NP=6D. The number of generations used was set to $G_{max} = 1000$. These values are the same as those used by [20], [21]. However, these selections were mainly based on general recommendations and practical experiences with the usage of DE,(e.g. [20] and [21]) and no systematic investigations were performed for finding the optimal control parameter values, therefore further classification performance improvements by finding better control parameter settings in future are within possibilities. Next into the actual classification. The problem of classification is basically one of partitioning the feature space into regions, one region for each category. Ideally, one would like to arrange this partitioning so that none of the decisions is ever wrong [38]. The objective is to classify a set X of objects to N different classes $C_1, \ldots, C_N$ by their attributes. We suppose

that T is the number of different kinds of attributes that we can measure from objects. The key idea is to determine for each class the ideal vector $\mathbf{y}_i$

$$\mathbf{y}_i = (y_{i1}, \ldots, y_{iT}) \tag{1}$$

that represents class i as well as possible. Later on we call these vectors as class vectors. When these class vectors have been determined we have to make the decision to which class the sample $\mathbf{x}$ belongs to according to some criteria. This can be done e.g. by computing the distances d_i between the class vectors and the sample which we want to classify. For computing the distance usual way is to use Minkowsky metric:

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{j=1}^{T} |x_j - y_j|^p \right)^{1/p} \tag{2}$$

After we have the distances between the samples and class vectors then we can make our classification decision according to the shortest distance.

for $\mathbf{x}, \mathbf{y} \in R^n$. We decide that $\mathbf{x} \in C_m$ if

$$d\langle \mathbf{x}, \mathbf{y}_m \rangle = \min_{i=1,\ldots,N} d\langle \mathbf{x}, \mathbf{y}_i \rangle \tag{3}$$

Before doing the actual classification, all the parameters for classifier should be decided. These include

1. The class vectors $\mathbf{y_i} = (y_{i1}, \ldots, y_{iT})$ for each class $i = 1, \ldots, N$
2. The power value p in (2).

In this study we used Differential Evolution algorithm [7] to optimize both the class vectors and p value. For this purpose we split the data into learning set *learn* and testing set *test*. Split was made so that half of the data was used in learning set and half in testing set. We used data available in learning set to find the optimal class vectors $\mathbf{y}_i$ and the data in the testing set *test* was applied for assessing the classification performance of the proposed classifier. In short the procedure for our algorithm is as follows:

1. Divide data into learning set and testing set
2. Create trial vectors to be optimized which consists of classes and parameter p, $\vec{v}_{i,G}$
3. Divide $\vec{v}_{i,G}$ into class vectors and parameter p.

4. Compute distance between samples in the learning set and class vectors
5. Classify samples according to their minimum distance by using (3)
6. Compute classification accuracy (accuracy = no. of correctly classified samples/total number of all samples in learning set)
7. Compute the fitness value for objective function using $f = 1 - accuracy$
8. Create new pool of vectors $\vec{v}_{i,G+1}$ for the next population using selection, mutation and crossover operations of differential evolution algorithm, and goto 3. until stopping criteria is reached. (For example maximum number of iterations reached or 100% accuracy reached)
9. Divide optimal vector $\vec{v}_{i,G_{max}}$ into class vectors and parameter p.
10. Repeat steps 4, 5 and 6, but now with optimal class vectors, p parameter and samples in the testing set.

That far the proposed method is identical with the one reported earlier in [20]. The proposed extension to the earlier DE classifier will be described in detail next.

2.2. Differential Evolution classification with a pool of distances

The idea here is, that instead of using equation (2), a pool of alternative distance measures is created, and the optimization process by differential evolution algorithm is then applied for selecting the distance measure optimally from this predefined pool. The optimization goal is to select from the pool the particular distance measure providing the highest classification accuracy over the current training data set. For constructing a pool of distance measures, a collection of applicable distance measures can be found i.e. in [39]. By using switch operator pointing the currently selected distance measure from the pool, and by optimizing its value, the selection of distance measure can be optimized by DE algorithm for current the data set at hand. In [22] we studied the choice from a pool of eleven distance measures.

In order to do this properly several issues needed to be considered. Vector to be optimized with DE can be divided into subcomponents as follows:

$$\mathbf{v}_{i,G(1)} = \{\{class1, class2 \cdots classN\}, \{switch\}, \{parameters\}\} \quad (4)$$

where first the class vectors are presented. The next in line is the switch operator, which is used to select a distance measure from the pool of alternativedistance measures. In DE algorithm (internally) we are optimizing a

real valued number instead of operating directly with integer numbers, so boundaries for this parameter needs to be selected correspondingly. Here the boundaries of selected interval are [0.5, 11.499]. For 11 distance measures in the pool this range is subdivided into 11 equal parts. Then the transformation from DE algorithm's internal real value representation to integer representation can be carried out by simple rounding operation to the nearest integer value. After this the gained integer value is used to point the corresponding distance function from the pool. The applied handling concept for integer valued parameters is explained in detail in [7]. In addition to this, some distance measures included into pool may have one or more free parameters which values needs to be optimized as well. What was done in [22] was that we created additional parameters vector in which we placed all related free parameters of different distance measures in.

2.3. Differential Evolution extended to optimized distances for the features in the data sets

Underlying idea here compared to previous one is that instead of applying distance measures for vectors like in equation (2), we apply elementwise distance measures like i.e. $d(x,y) = |x-y|^p$ and aim to optimize the selection for proper elementwise distance measures and their possible free parameters. After this we normalize and aggregate the distance vectors gained by this way to the produce overall distance value. In order to do this we can again subdivide the optimization problem into a general vector consisting now the following components:

$$\mathbf{v}_{i,G(2)} = \{\{class1, class2, \cdots, classN\}, \{switch1, \ldots, switchT\}, \{parameters\}\} \tag{5}$$

where $\{class1, class2 \cdots classN\}$ are $T-$dimensional class vectors which are to be optimized for the current data set, $\{switch\}$ is now a vector of integer values which for pointing featurewisely the optimal distance measures from choices d_1 to d_8 (assuming that we happen to have defined eight possible distance measures in our pool of distances). Thereby $\{switch1, \ldots, switchT\}$ is T dimensional vector consisting of integer numbers within $[1, 8]$. The remaining component $\{parameters\}$ is a vector of possible free parameters of the individual distance measures of the pool. In this case the vector $\{parameters\}$ can be created in several different ways and in this paper we have taken a closer examination of three possible cases of parameter optimization.

The pool of distances which we used in this paper is presented in equations (6)-(13).

$$d_1(x,y) = (|x-y|^{r_1}); r_1 \in [1,\infty) \tag{6}$$

$$d_2(x,y) = |x-y|^{r_2}/max\{|x|,|y|\}; r_2 \in [1,\infty) \tag{7}$$

$$d_3(x,y) = |x-y|^{r_3}/\left(1+min\{|x|,|y|\}\right); r_3 \in [1,\infty) \tag{8}$$

$$d_4(x,y) = |x-y|/\left(1+max\{|x|,|y|\}\right); \tag{9}$$

$$d_5(x,y) = |x-y|/[1+|x|+|y|]; \tag{10}$$

$$d_6(x,y) = |x/[1+|x|] - y/[1+|y|]| \tag{11}$$

$$d_7(x,y) = r_4(x-y)^2/(x+y); r_4 \in (0,\infty) \tag{12}$$

$$d_8(x,y) = |x-y|/\left(1+|x|\right)(1+|y|) \tag{13}$$

As can be observed from equations (6)-(13) some of the distance measures includes free parameters which needs to be optimized, while some measures do not include any free parameters. To find suitable value for them we considered three possible arrangements.

In first one we let our parameter vector equal $\{parameters = \{r_1, r_2, r_3, r_4\}\}$ where r_1 to r_4 are simple real valued elements for which proper value needs to be found. In this case our vector to be optimized would be

$$\mathbf{v}_{i,G(3)} = \{\{class1, class2, \cdots, classN\}, \{switch1, \cdots, switchT\}, \{r_1, r_2, r_3, r_4\}\} \tag{14}$$

This is the simplest and fastest, but also the most restrictive case. Namely, for example if the optimal distance measure would be the equation (7) for the first feature of our data set, and also for the third feature, we are restricted to use the very same parameter value r_2 for both of these features (and all other features using the same distance measure). It is likely that often a better classification accuracy could be reached if parameter value r_2 could be optimized individually for all features in which equation (7) is

Algorithm 1 Pseudo code for classification process with optimal parameters r_1, r_2, r_3, r_4 from DE.

Require: $Data[1,\ldots,T]$, $classvec1[1,\ldots,T]$, $classvec2[1,\ldots,T]$,...,$classvecN[1,\ldots,T]$, $switch1,\ldots,switchT$,$parameters$={ r_1, r_1, r_1,r_4} $center$=[$classvec1$; $classvec2$;...$classvecN$]

```
for j = 1 to T do
    for i = 1 to N do
        if switch(j) == 1 then
            d(:,i,j) = dist1(data, repmat(center(i,j),T,1), r1)
        else if switch(j) == 2 then
            d(:,i,j) = dist2(data, repmat(center(i,j),T,1), r2)
        else if switch(j) == 3 then
            d(:,i,j) = dist3(data, repmat(center(i,j),T,1), r3)
        else if switch(j) == 4 then
            d(:,i,j) = dist4(data, repmat(center(i,j),T,1))
        else if switch(j) == 5 then
            d(:,i,j) = dist5(data, repmat(center(i,j),T,1))
        else if switch(j) == 6 then
            d(:,i,j) = dist6(data, repmat(center(i,j),T,1), r4)
        else if switch(j) == 7 then
            d(:,i,j) = dist7(data, repmat(center(i,j),T,1))
        else
            d(:,i,j) = dist8(data, repmat(center(i,j),T,1))
        end if
    end for
end for
D = scale(d)
for i = 1 to T do
    d_total(:,i) = sum(D(:,:,i),2);
end for
for i = 1 to length(d_total) do
    class(:,i) = find(d_total(i,j) == min(d_total(i,:)));
end for
```

applied as a distance measure. Short pseudo algorithm for this case is given in Algorithm 1.

Second possible way to choose parameters and to get rid of restriction from previous case would be to allow parameter vector to be of length T. This reserves individual parameter value for each feature in the data set. In this case the vector to be optimized would be

$$\mathbf{v}_{i,G(4)} = \{\{class1, class2, \cdots, classN\}, \{switch1, \cdots, switchT\}, \{p_1, p_2, \cdots, p_T\}\} \tag{15}$$

In this case $\{parameters\} = \{p_1, \cdots p_T\}$ and this would allow i.e. the case that the optimal distance measure for feature one could be equation (7) and parameter value for it would be now in p_1 and if optimal distance measure for feature three would be also equation (7) then optimal parameter value for it would be in p_3. Pseudo algorithm for this process is in Algorithm

2.

Algorithm 2 Pseudo code for classification process with optimal parameters $p_1, p_2, ..., p_T$ from DE.

Require: $Data[1,\ldots,T]$, $classvec1[1,\ldots,T]$, $classvec2[1,\ldots,T]$,...,$classvecN[1,\ldots,T]$, $switch1,\ldots,switchT$, $parameters=\{p_1,p_2,\ldots,p_T\}$ $center$=[$classvec1$; $classvec2$; ...$classvecN$]

```
for j = 1 to T do
    for i = 1 to N do
        if switch(j) == 1 then
            d(:,i,j) = dist1(data, repmat(center(i,j),T,1), p(j))
        else if switch(j) == 2 then
            d(:,i,j) = dist2(data, repmat(center(i,j),T,1), p(j))
        else if switch(j) == 3 then
            d(:,i,j) = dist3(data, repmat(center(i,j),T,1), p(j))
        else if switch(j) == 4 then
            d(:,i,j) = dist4(data, repmat(center(i,j),T,1))
        else if switch(j) == 5 then
            d(:,i,j) = dist5(data, repmat(center(i,j),T,1))
        else if switch(j) == 6 then
            d(:,i,j) = dist6(data, repmat(center(i,j),T,1), p(j))
        else if switch(j) == 7 then
            d(:,i,j) = dist7(data, repmat(center(i,j),T,1))
        else
            d(:,i,j) = dist8(data, repmat(center(i,j),T,1))
        end if
    end for
end for
D = scale(d)
for i = 1 to T do
    d_total(:,i) = sum(D(:,:,i),2);
end for
for i = 1 to length(d_total) do
    class(:,i) = find(d_total(i,j) == min(d_total(i,:)));
end for
```

A problem with this way of selecting the parameters is the inbuild context dependency of the parameter values. When we optimize the distance measure selection lets say for feature three and it has been in iteration i distance d_2 with parameter p_3 then if we find a better optimum with lets say distance d_1 the optimized parameter p_3 would be for d_2 and that particular value may not suite very well in context of distance measure d_1 and this fact makes it more difficult to change the distance measure during the optimization process for the DE algorithm. Simply, the problem is that the interpretation of the parameter value changes if the distance measure is changed.

The third possible way, which we decided to test, is to create this parameters vector so that we do not restrict the parameter values vectorwise like in previous case, but we allow each of the four parameters $\{r_1, r_2, r_3, r_4\}$ to be vectors of length T, and this way make it more easier for DE to change

the distance measure during the optimization process.

$$\mathbf{v}_{i,G(5)} = \{\{class1, class2, \cdots, classN\}, \{switch1, \ldots, switchT\}, \\ \{p_{1,1}, p_{1,2}, p_{1,3}, p_{1,4}, \cdots, p_{1,T}, p_{2,T}, p_{3,T}, p_{4,T}\}\} \quad (16)$$

Meaning here that i.e. $r_1 = \{p_{1,1}, \cdots, p_{1,T}\}$. The purpose of this arrangement is to remove the potentially harmfull context dependency of the parameter value shown in previous example. Using again our example, it means that if in iteration i optimal distance for feature three would be d_2 with parameter $p_{2,3}$ and if in iteration $i+1$ we would find that d_1 is the best choice for feature three then its parameter would be $p_{1,3}$ this way making this change easier. Pseudo algorithm of this process is given in Algorithm 3. Advantage of the latest arrangement is, that changing the distance measure for a feature is easier and should ease the optimization process in finding globally optimal solution by removing the potentially harmful context dependency of the parameter values. However, on the other hand we are also increasing the length of the optimized vector considerably and enlarge the search space, making the optimization problem harder in this sense. Note here, that despite four distance parameters are optimized for each feature, only one of them (if any) is actually used each time, just depending on the current value of the corresponding switch. Thus, otherwise this approach is enlarging the search space making the optimization problem harder to solve, but otherwise removing the harmfull context dependency from previous case should do the opposite. Therefore both performance improvement and degradation due to latest modification were within possibilities before computational experimentations.

After the vector $\mathbf{v}_{i,G}$ is divided in its corresponding parts we can calculate the distances between the samples and the class vectors and the classification can be carried out according to the minimum distance. Finally we use label information from the samples to calculate the misclassification rate which is to be minimized (our cost function value). The remaining part of the process is basically following the same procedure as our previous papers from the subject [20], [22].

3. The test data sets and experimental arrangements

The data sets for experimentation with the proposed approach were taken from UCI machine learning data repository [34] and KEEL data repository [35]. The stopping criteria for differential evolution algorithm were 1000 generations ($G_{max} = 1000$). Each data set was divided 30 times into

Algorithm 3 Pseudo code for classification process with each optimal parameters is extended to cover the length of the feature from DE.

Require: $Data[1,\ldots,T]$, $classvec1[1,\ldots,T]$, $classvec2[1,\ldots,T]$,...,$classvecN[1,\ldots,T]$,
$switch1,\ldots,switchT$,$parameters$ ={$p1[1,\ldots,T]$, $p2[1,\ldots,T]$, $p3[1,\ldots,T]$,$p4[1,\ldots,T]$}
$center$=[$classvec1$; $classvec2$; ...$classvecN$]
for $j = 1$ to T **do**
  for $i = 1$ to N **do**
    if $switch(j) == 1$ **then**
      $d(:,i,j) = dist1(data, repmat(center(i,j),T,1), p1(j))$
    else if $switch(j) == 2$ **then**
      $d(:,i,j) = dist2(data, repmat(center(i,j),T,1), p2(j))$
    else if $switch(j) == 3$ **then**
      $d(:,i,j) = dist3(data, repmat(center(i,j),T,1), p3(j))$
    else if $switch(j) == 4$ **then**
      $d(:,i,j) = dist4(data, repmat(center(i,j),T,1))$
    else if $switch(j) == 5$ **then**
      $d(:,i,j) = dist5(data, repmat(center(i,j),T,1))$
    else if $switch(j) == 6$ **then**
      $d(:,i,j) = dist6(data, repmat(center(i,j),T,1), p4(j))$
    else if $switch(j) == 7$ **then**
      $d(:,i,j) = dist7(data, repmat(center(i,j),T,1))$
    else
      $d(:,i,j) = dist8(data, repmat(center(i,j),T,1))$
    end if
  end for
end for
$D = scale(d)$
for $i = 1$ to T **do**
  $d_{total}(:,i) = sum(D(:,:,i),2)$;
end for
for $i = 1$ to $length(d_{total})$ **do**
  $class(:,i) = find(d_{total}(i,j) == min(d_{total}(i,:)))$;
end for

new random splits of testing sets and learning sets ($N = 30$). Thereby, the reported mean accuracies and variances have been computed over 30 independent experiments, each with a new random split of data. The Fundamental properties of the data sets are summarized in Table 1 and each data set is briefly introduced in the following.

Hypothyroid data set:

This data contains results from real medical screening test for the thyroid problem. The data discriminates between the patients who have normally functioning thyroid from those who have hypo functioning thyroid. The data offers a good mixture of 18 nominal and 7 continuous features.

Echocardiogram data:

In this data set one sample is results from echocardiogram measurements of one patient who has recently suffered an acute heart attack. Measurements are taken from echocardiograms, which are ultrasound measurements of the heart itself. The goal of physicians using these measurements is to predict a patient's chances of survival. Experimental work is being preformed to determine if an echocardiogram, in conjunction with other measures, could be used to predict whether or not a patient would survive for longer than a certain time period. This data can give means for predicting future heart attacks in former heart patients.

Horse-colic data:

Predictive attribute here is surgical lesion. In other words the purpose is to try to determine the existence or non-existence of the surgical (lesion)problem. All the instances in this data are either operated upon or autopsied so that this value and the lesion type are known and main aim is to find whether the surgical procedure is needed or not. Total of 11 different attributes were measured for this purpose and number of instances were 368.

Wisconsin Diagnostic Breast Cancer data(WBDC):

This data set is a result of the effort made at the University of Wisconsin Hospital for the diagnosis of Breast tumors solely based on Fine Needle Aspirate (FNA)test, this test involves fluid extraction from breast mass using a small gauge needle and then visual inspection of the fluid under a microscope. The Wisconsin Diagnostic Breast Cancer data sets consists of 569 instances of which 357 are benign and 212 are malignant where each one represent FNA test measurement for one diagnostic case.

Balance-scale data:

This data set was generated to model psychological experimental results. Each example is classified as having the balance scale tip to the right, tip to the left, or be balanced. The attributes are the left weight, the left distance, the right weight and the right distance. Total of 5 different attributes were measured for this purpose and number of instances were 625.

Australian Credit Approval data:

This data set concerns credit card applications. It has 2 classes with 690 instances and 14 number of attributes where by 8 are numerical and 6 are categorical features.

German Credit data:

Here, the task is to classify customers as good (1) or bad (2), depending on 20 features about them and their bancary accounts. Number of instances here was 1000 customers.

Redwine data:

The data sets is related to red variants of the Portuguese "Vinho Verde" wine originated from Portugal. This data set has 11 number of classes, 11 features and 1594 number of instances.

Satimage data:

The database consists of the multi-spectral values of pixels in 3x3 neighbourhoods in a satellite image, and the classification associated with the central pixel in each neighbourhood. The aim is to predict this classification, given the multi-spectral values. This data set is seven class classification problem with 36 attributes and 6435 instances.

[Table 1 about here.]

4. Results from proposed methods and comparison

4.1. Analysis of results for proposed methods

The proposed differential evolution classifier with optimal distance measures for each feature was tested with data sets Australian Credit Approval, Balance scale, Echocardiogram, German credit, Horsecolic, Hypothyroid, Red wine, Satimage and Wisconsin Diagnostic Breast Cancer data set. The mean classification accuracies were recorded as well as the corresponding optimal distance measure that was found optimal in each individual experiment. In each case 30 repetitions were performed dividing the data randomly into learning and testing tests. Basically our cross validation technique here is two fold cross validation where samples are divided randomly into folds

30 times and required statistics calculated from there.

As introduced in section 2, we considered three alternative approaches for determining the optimal values for the free parameters of each distances measure in the pool to maximize the classification accuracy over the data set to be classified. The three possibilities on how to define the parameter vector for optimization purposes were

- Parameters=$\{r_1, r_2, r_3, r_4\}$.
- Parameters = $\{p_1, p_2, ..., p_T\}$ where T is the length of the feature in the data set.
- Parameters=$\{ p_{1,j}, p_{2,j}, p_{3,j}, p_{4,j} \}$where $j \in [1, 2, \ldots, T]$, which is also our current choice in this paper selected for comparison with other classifiers.

[Table 2 about here.]

The Table 2 shows the mean accuracies of the data set from the three ways of choosing the parameters. Also the 99% confidence intervals are calculated. The second column shows the mean accuracy for the parameters $\{p_1, p_2, p_3, p_4 \}$which are fixed for each feature in the data set and the third column has the mean accuracy for the parameters $\{p_1, p_2, ..., p_T\}$ whereby these parameters were extended to cover the length of the feature in the data set, while the fourth column show each parameter were extended according to the length of the feature in the data set for the optimized distance.

In Table 2, the mean accuracies for Balance scale, Echocardiogram, and Horsecolic data set in third choice outperformed other results obtained with first and second choice. Second choice gave best results with Redwine data set. The WDBC and Australian data set showed good results with the first choice of parameters.

A more thorough listing of results from third choice of parameters are given in Table 3 mean accuracies and variances are given for both testing set and learning set. In other tables the testing set results are listed and confidence intervals are calculated using variances.

[Table 3 about here.]

4.2. Comparison with other Classifiers

In this subsection we analyze the results obtained from experimental studies in which we are comparing results from the proposed method to the original DE classifier [20], k- Neural Network(k-NN) classifier and Back Propagation Neural Network(BPNN) classifier. For the other classifiers we used freely available toolbox PRtool [40] to calculate the results. With DE classifier we refer to earlier version where there was no pool of distance involved, but just simply Minkowsky distance was used (see [20]). The results from these experiments are collected in Table 4. The proposed method gave good results with the data set Balance scale, German, Horsecolic, Hypothyroid and Red wine and outperform other classifiers though for the case of Hypothyroid data set it can be calculated that the mean accuracy from the proposed method and DE classifier shows no significance difference. For k-NN classifier, Echocardiogram and Satimage data sets have performed very well with the best mean accuracies as compared to other classifiers while BPNN classifier gave good results in Australian, German and WDBC, though the mean accuracy of German data set using BPNN classifier showed no significance difference with BPNN and of the proposed method.

[Table 4 about here.]

5. Conclusions

In this paper we have proposed a further generalization of DE classifier, where the distance measure optimization is extended to features level. The experimental results were computed for nine different benchmarking data sets. In all cases the proposed method provided at least comparable accuracy or outperformed the compared classifiers, including the original version of DE classifier. In particular, the proposed approach statistically significantly outperformed the original DE classifier with most of the studied data sets, while any opposite situations were not observed. Thereby the obtained results are suggesting, that this generalization is useful and clearly of importance. Another importance notation is that the similar generalization can be implemented relatively easily to several other classification methods provided that a suitable efficient global optimization algorithm like, differential evolution algorithm, can be applied for solving the related difficult global optimization problem efficiently.

We examined also the three alternative approaches for handling the free parameters of distance measures in question. Most restrictive was the case where a single parameter value was optimized for each distance measure,

and then applied to all features that the corresponding distance measure were assigned. This approach can work appropriately in some cases. In our experiments this appeared to be the best choice for Australian data set and WDBC data set. The third considered approach was the most general one, where individual distance measures with independent parameters were optimized for each data feature separately. This allows for DE algorithm highest freedom for improving the classification accuracy. Downside of this is, that the number of optimized parameters is growing rapidly with the number of data features, enlarging the optimization search space considerably. Also the increased number of free parameters may promote overlearning to some extend, despite no severe overlearning was not observed in the current experiments. This approach appeared to work well in general, and yielded the best results i.e. in Balance scale and horsecolic data sets. The second alternative we considered is a compromise between the other two cases by being less restrictive than the first case, but also keeping the search space smaller than the third case. This approach also has its merits as the results from Redwine data are showing.

An important related issue that requires attention in future is, that at the moment we are applying simply a sum of featurewise distance measures to calculate the total distance measure. Here also possibilities to apply a suitable aggregation method can be considered. This is one of our future research directions related with the current work.

Acknowledgment

The work of Jouni Lampinen was supported by the IT4Innovations Centre of Excellence project. CZ.1.05/1.1.00/02.0070 funded by the Structural Funds of EU and Czech Republic ministry of education.

References

[1] Chauhan, N., Ravi, V., Chandra, D.K. Differential evolution trained wavelet neural networks: Application to bankruptcy prediction in banks,Expert Systems with Applications, 36, 4, (2009), pp. 7659-7665

[2] Su, H., Yang, Y., Zhao, L. Classification rule discovery with DE/QDE algorithm, Expert Systems with Applications, 37, 2, (2010), pp 1216-1222

[3] Khushaba, R.N., Al-Ani, A., Al-Jumaily, A. Feature subset selection using differential evolution and a statistical repair mechanism,Expert Systems with Applications, 38, 9, (2011), pp 11515-11526

[4] Bastürk, A., Günay, E. Efficient edge detection in digital images using a cellular neural network optimized by differential evolution algorithm,Expert Systems with Applications, 36, 2, Part 2, (2009), pp 2645-2650

[5] Liu, Y., Sun, F. A fast differential evolution algorithm using k-Nearest Neighbour predictor,Expert Systems with Applications, 38, 4,(2011), pp. 4254-4258

[6] Sun, J., Zhang, Q., Tsang, E.P.K. (2005) DE/EDA: A new evolutionary algorithm for global optimization, Information Sciences 169, pp 249-262.

[7] Price, K., Storn, R., Lampinen, J. (2005) Differential Evolution - A Practical Approach to Global Optimization, Springer.

[8] Maulik, U., Saha, I. (2009), Modified differential evolution based fuzzy clustering for pixel classification in remote sensing imagery. Pattern Recognition 42. pp 2135-2149.

[9] Zmudaa, M.A., Rizkib, M.M. (2009) Tamburinoc, L.A. Hybrid evolutionary learning for synthesizing multi-class pattern recognition systems, Applied Soft Computing, Vol 2, Issue 4, pp 269-282

[10] Magoulas, G.D., Plagianakos, V.P., Vrahatis, M.N., (2004), Neural network-based colonoscopic diagnosis using on-line learning and differential evolution, Applied Soft Computing, 4, pp. 369-379.

[11] Fernandez, J.C., Hervas, C., Martinez, J. Gutierrez, P.A., Cruz, M., (2009), Memetic Pareto Differential Evolution for Designing Artificial Neural Networks in Multiclassification Problems Using Cross-Entropy Versus Sensitivity, Hybrid Artificial Intelligence Systems, Lecture Notes in Computer Science, 5572, pp. 433-441.

[12] Bandyopdhyay, S.,Saha, S,(2007) GAPS: A clustering method using a new point symmetry-based distance measure technique. Pattern Recognition, Vol. 40 no.12 pp 3430-3451

[13] Das, S, Abraham,A, Konar,A (2008) Automatic Clustering using an improved Differential Evolution Algorithm. IEEE Transactions on Systems, Man and Cybernetics, Part A 38(1): pp 218-237.

[14] Maulik, U., Saha, I. (2010), Automatic Fuzzy Clustering Using Modified Differential Evolution for Image Classification, IEEE Transactions on geoscience and remote sensing, Vol. 48 no.9, pp 3503-3510.

[15] Omran M.G.H., Engelbrecht, A.P., Salman, A. (2005), Differential Evolution Methods for Unsupervised Image Classification, IEEE Congress on Evolutionary Computation, 2005, pp. 966-973.

[16] Omran M.G.H., Engelbrecht, A.P. (2006), Self-Adaptive Differential Evolution Methods for Unsupervised Image Classification, IEEE Conference on Cybernetics and Intelligent Systems, 2006, pp 1-6

[17] De Falco, I., Della Cioppa, A., Tarantino, E. (2006), Automatic Classification of Handsegmented Image Parts with Differential Evolution, EvoWorkshops, 2006, Lecture notes in computer science, 3907, pp. 403-414

[18] De Falco, I., Differential Evolution for automatic rule extraction from medical databases, Applied Soft Computing, (2012) In Press.

[19] Triguero, I., Garcia, S., Herrera, F. (2011), Differential evolution for optimizing the positioning of prototypes in nearest neighbor classification, Pattern Recognition, 44, pp. 901-916.

[20] Luukka, P., Lampinen, J.,(2011) Differential Evolution Classifier in Noisy Settings and with Interacting Variables, Applied Soft Computing, 11 , pp. 891-899.

[21] Luukka, P., Lampinen, J.,(2010) A Classification method based on principal component analysis differential evolution algorithm applied for predition diagnosis from clinical EMR heart data sets. Computational Intelligence in Optimization: Applications and Implementations. Springer.

[22] Koloseni,D., Lampinen,J., Luukka,P.,(2012) Optimized Distance Metrics for Differential Evolution based Nearest Prototype Classifier, Expert Systems With Applications,39, pp 10564-10570.

[23] Shahid, R., Bertazzon, S., Knudtson, M.L. and Ghali, W.A., (2009) Comparison of distance measures in spatial analytical modeling for health service planning.BMC Health Services Research, 9:200.

[24] Madzarov, G, Gjorgjevikj, D.,(2010) Evaluation of distance measures for multi-class classification in binary SVM decision tree. Artificial intelligence and soft computing, Lecture Notes in Computer Science, vol 6113, pp. 437-444.

[25] Dettmann, E., Becker, C., Schmeiser, C.,(2011) Distance functions for matching in small samples. Computational Statistics and Data Analysis,55, pp. 1942-1960.

[26] Kaski, S., Sinkkonen, J., Peltonen, J.(2001) Bankruptcy analysis with self-organizing maps in learning metrics. IEEE Transactions on Neural Networks 12, number 4, pp. 936-947.

[27] Chang H., Yeung D.Y., Cheung W. K. (2006),Relaxational metric adaptation and its application to semi-supervised clustering and content-based image retrieval, Pattern Recognition, 39,10 pp. 1905-1917

[28] Ogawa, A., Takahashi, S., (2008), Weighted distance measures for efficient reduction of Gaussian mixture components in HMM-based acoustic model. IEEE International Conference on Acoustics, Speech and Signal Processing, 2008 pp. 4173-4176.

[29] Schonfeld, J., Ashlock, D., (2006) Evaluating Distance Measures for RNA Motif Search, IEEE Congress on Evolutionary Computation, 2006. CEC 2006. pp. 2331-2338.

[30] Everson, L.R.H., Penzhorn, W.T., (1988) Experimental comparison on several distance measures for speech processing applications, Southern African Conference on Communications and Signal Processing, 1988. Proceedings., COMSIG 88. pp. 12-17.

[31] Yu, J., Yin, J, Zhang, J.: (2007), Comparison of Distance Measures in Evolutionary Time Series Segmentation, Third International Conference on Natural Computation, 2007. ICNC 2007. pp. 456-460.

[32] Jenicka, S., Suruliandi, A.,(2011) Empirical evaluation of distance measures for supervised classification of remotely sensed image with Modified Multivariate Local Binary Pattern. International Conference on Emerging Trends in Electrical and Computer Technology (ICETECT), pp. 762-767.

[33] Fernando, F and Pedro, I, (2004),Evolutionary Design of Nearest Prototype Classifers, Journal of Heuristics, Kluwer Academic Publishers. pp 431-545.

[34] Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J. (1998).UCI Repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA:University of California, Department of Information and Computer Science.

[35] Alcal-Fdez,J., Snchez,L., Garca, S., del Jesus, M.J., Ventura, S., Garrell,J.M., Otero,J., Romero, C., Bacardit, J., Rivas,V.M., Fernndez, J.C., Herrera,F. KEEL: A Software Tool to Assess Evolutionary Algorithms to Data Mining Problems. Soft Computing 13:3 (2009) 307-318

[36] Storn, R., Price, K.V.,(1997) Differential Evolution - a Simple and Efficient Heuristic for Global Optimization over Continuous Space, Journal of Global Optimization, Vol. 11, no 4, pp.341-359.

[37] Price, K.V.,(1999) New Ideas in Optimization.London: McGraw-Hill, ch.6 An Introduction to Differential Evolution, pp. 79-108.

[38] Duda, R., Hart, P.(1973) Pattern Classification and Scene Analysis. John Wiley & Sons

[39] Bandemer, H. and Näther, W. (1992). Fuzzy Data Analysis, Kluwer Academic Publishers.

[40] R.P.W.Duin, P.Juszczak, P. Paclik, E. Pekalska, D. de Ridder, D,M,J, Tax, S. Verzakov, PRTools4, A matlab toolbox for pattern recognition. August 2007.

List of Tables

1 Properties of the data sets 24
2 Mean accuracies for the three choices of parameters 25
3 Classification results for the nine data sets using $N = 30$ and $G_{max} = 1000$. Mean classification accuracies and variances with parameters=$\{ p_{1,j}, p_{2,j}, p_{3,j}, p_{4,j} \}$where $j \in [1, 2, \ldots, T]$. TS is referring to test set and LS to the learning set. 26
4 Comparison of the results of the data sets from the proposed method to other classifiers. 27

Table 1: Properties of the data sets

Name	Nb of classes	Nb of attributes	Nb of instances
WDBC	2	32	569
Echocardiogram	2	8	132
Horse-colic	2	11	368
Hypothyroid	2	25	3772
Balance scale	3	5	625
Australian Credit	2	14	690
German Credit	2	20	1000
Satimage	7	36	6435
Redwine	11	11	1596

Table 2: Mean accuracies for the three choices of parameters

Data	r_1, r_2, r_3, r_4	$p_1, p_2, ..., p_T$	$p_{1,j}, p_{2,j}, p_{3,j}, p_{4,j}$
Australian	83.52 ± 1.40	83.30 ± 1.66	82.94 ± 1.65
Balance Scale	90.56 ± 0.43	90.32 ± 0.60	91.30 ± 0.12
Echocardiogram	85.54 ∓ 2.85	83.95 ± 3.16	86.55 ± 3.04
German	73.71 ± 0.71	73.30 ± 0.87	73.37 ± 0.86
Horsecolic	69.23 ± 5.48	74.91 ± 4.90	83.35 ± 2.23
Hypothyroid	99.59 ± 0.49	99.25 ± 0.85	99.57 ± 0.63
Redwine	57.18 ± 0.85	57.50 ± 0.72	57.11 ± 0.89
Satimage	80.63 ± 0.45	80.58 ± 0.56	80.44 ± 0.26
WDBC	95.24 ± 0.92	95.16 ± 1.21	93.64 ± 1.29

Table 3: Classification results for the nine data sets using $N = 30$ and $G_{max} = 1000$. Mean classification accuracies and variances with parameters=$\{ p_{1,j}, p_{2,j}, p_{3,j}, p_{4,j} \}$where $j \in [1, 2, \ldots, T]$. TS is referring to test set and LS to the learning set.

Data	Mean (TS)	Variance (TS)	Mean (LS)	Variance (LS)
Australian	82.94	10.71	90,38	1.21
Balance scale	91.30	0.075	92.22	0.028
Echocardiogram	86.55	36.44	95.75	3.84
German	73.37	2.95	80.23	0.91
Horsecolic	83.35	24.67	88.59	2.36
Hypothyroid	99.57	1.98	99.98	0.00092
Redwine	57.11	3.15	60.69	0.66
Satimage	80.44	0.27	81.23	0.34
WDBC	93.64	6.54	97.14	0.48

Table 4: Comparison of the results of the data sets from the proposed method to other classifiers.

Data	KNN	BPNN	DE Classifier	Proposed Method
Australian	65.69 ± 0.95	85.59 ± 0.71	66.78 ± 1.37	82.94 ± 1.65
Balance Scale	88.06 ± 0.53	87.90 ± 0.67	88.66 ± 1.09	91.30 ± 0.12
Echocardiogram	90.90 ± 1.56	85.46 ± 2.66	86.89 ± 2.04	86.55 ± 3.04
German	70.17 ± 0.66	73.07 ± 0.85	71.18 ± 0.90	73.37 ± 0.86
Horsecolic	68.53 ± 1.05	80.85 ± 1.97	71.76 ± 5.22	83.35 ± 2.23
Hypothyroid	98.37 ± 0.03	97.29 ± 0.06	99.95 ± 0.04	99.57 ± 0.63
Redwine	51.88 ± 0.90	41.43 ± 10.10	56.61 ± 0.71	57.11 ± 0.89
Satimage	89.84 ± 0.26	38.59 ± 7.50	67.37 ± 0.81	80.44 ± 0.26
WDBC	92.68 ± 0.63	96.16 ± 0.59	91.78 ± 0.58	93.64 ± 1.29