Expert Systems with Applications 39 (2012) 10564–10570

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

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

Optimized distance metrics for differential evolution based nearest prototype classifier

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

[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

ARTICLE INFO

Keywords:
Classification
Global optimization
Evolutionary algorithm
Differential evolution algorithm
Distance measures

ABSTRACT

In this article, we introduce a differential evolution based classifier with extension for selecting automatically the applied distance measure from a predefined pool of alternative distances measures to suit optimally for classifying the particular data set at hand. The proposed method extends the earlier differential evolution based nearest prototype classifier by extending the optimization process by optimizing not only the required parameters for distance measures, but also optimizing the selection of the distance measure it self in order to find the best possible distance measure for the particular data set at hand. It has been clear for some time that in classification, usual euclidean distance is often not the best choice, and the optimal distance measure depends on the particular properties of the data sets to be classified. So far solving this issue have been subject to a limited attention in the literature. In cases where some consideration to this is problem is given, there has only been testing with couple distance measure to find which one applies best to the data at hand. In this paper we have attempted to take one step further by applying a systematic global optimization approach for selecting the best distance measure from a set of alternative measures for obtaining the highest classification accuracy for the given data. In particular, we have generated pool of distance measures for the purpose and developed a model on how the differential evolution based classifier can be extended to optimize the selection of the distance measure for given data. The obtained results are demonstrating, and also confirming further on the earlier findings reported in the literature, that often some other distance measure than the most commonly used euclidean distance is the best choice. The selection of distance measure is one of the most important factor for obtaining best classification accuracy, and should thereby be emphasized more in future research. The results also indicate that it is possible to build a classifier that is selecting the optimal distance measure for the given data automatically. It is also recommended that the proposed extension the differential evolution based classifier is clearly efficient alternative in solving classification problems.

1. Introduction

One of the most recently emerged methods in evolutionary computation is differential evolution algorithm (Price, Storn, & Lampinen, 2005). Although a significant amount of evolutionary computation research has concerned the theory and practice of classifier systems (Booker, 1985; Fogarty, 1994; Holland, 1985, 1987; Holland, Holyoak, & Thagard, 1987; Robertson, 1987; Wilson, 1987), the possibilities of applying evolutionary algorithms in solving classification tasks has not yet been studied exhaustively, and further research in this area is still required. Differential evolution algorithm is gaining fastly popularity in classification problems. Some include bankruptcy prediction (Chauhan, Ravi, & Chandra, 2009), classification rule discovery (Su, Yang, & Zhao, 2010), feature selection, (Khushaba, Al-Ani, & Al-Jumaily, 2011), edge detection in images (Bastürk & Günay, 2009). Also in some methods classification techniques are used to improve optimization with differential evolution algorithm (Liu & Sun, 2011). Here we tackle classification problems by extending differential evolution (DE) classifier (Luukka & Lampinen, 2011, 2010) further onto cover also optimization of the distance measures selection from a pool of alternative distance measures. The original DE classifier is a nearest prototype classifier applying a single class prototype vector for representing each class. A sample is classified by measuring its distance to each class vector. A sample belongs to the class represented by the nearest class vector. In this type of a problem one must first determine the class vector that are representing each class optimally and also determine the optimal values for the possible distance metrics related parameter i.e. p parameter in Minkowsky distance. Here the optimal results depends heavily

[*] Corresponding author.
E-mail addresses: david.koloseni@lut.fi (D. Koloseni), jouni.lampinen@uwasa.fi (J. Lampinen), pasi.luukka@lut.fi (P. Luukka).

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2012.02.144

on the chosen distance measure.Usually simply euclidean distance or as in Luukka and Lampinen (2011) the Minkowsky distance is used for the purpose. This of course is sometimes a sufficient choice, but there also exists several classification problems where Minkowsky distance is not optimal distance to use or even a sufficient choice. In fact the optimal distance measure is not known a priori. Examples on such situations where a considerable better classification accuracy have been obtained by applying some other distance measure have been reported in several articles i.e. (Chang, Yeung, & Cheung, 2006; Dettmann, Becker, & Schmeiser, 2011; Everson & Penzhorn, 1988; Fernando & Pedro, 2004; Jenicka & Suruliandi, 2011; Kaski, Sinkkonen, & Peltonen, 2001; Madzarov & Gjorgjevikj, 2010; Ogawa & Takahashi, 2008; Shahid, Bertazzon, Knudtson, & Ghali, 2009; Schonfeld & Ashlock, 2006; Yu, Yin, J, & Zhang, 2007). However, typically in these types of studies one has simply tested with a few different distance measure for classifying the data set at hand.

In our approach, instead of taking this type of trial and error approach, we are attempting to select the best distance measure from a predefined pool of alternative measures systematically, automatically and optimally. Furthermore, instead of only selecting the optimal distance measure from a set of alternatives, we also attempt to optimize the values of the possible control parameters related with the selected distance measure.In particular, we first create a pool of alternative distance measure and then apply a global optimization algorithm, differential evolution for selecting the optimal distance measure that yields the highest classification accuracy with the current data. The first advantages of this approach is, that there is no need to carry out tests with different measures manually, since the differential evolution algorithm is searching the optimal distance measure and the optimal parameter values for it in parallel with searching the optimal class prototype vectors for each class. The second advantage is that all these selections will become globally optimized by the differential evolution algorithm for the maximum classification accuracy over the current data set to be classified. As an outcome considerable amount of time, computations and efforts are saved in testing with different distance measures. Also all selections needed will be globally optimized as well as all values of all free parameters of the classier, that should, in principle at least in a better classification accuracy for the given data.

2. Differential evolution based classification with pool of distances

Here first we start with short introductory to differential evolution based classification and then we go into how to extend it to cover optimization of distance measures for the classification problem at hand and possible parameter optimizations which the measures have.

2.1. Differential evolution based classification

The DE algorithm (Price et al., 2005; Storn & Price, 1997) 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 $\vec{v}_{i,G}$ of the population and creates the corresponding trial vector $\vec{u}_{i,G}$ as follows in the most common DE version, DE/rand/1/bin (Price, 1999):

```
$r_1, r_2, r_3 \in \{1, 2, \ldots, NP\}$, (randomly selected, except mutually
  different and different from $i$)
$j_{rand} = \text{floor}\ (rand_i[0,1) \cdot (D) + 1$
for($j = 1;\ j \leqslant D;\ j = j + 1$)
{
    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 $\vec{v}_{i,G}$. 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 rotational 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 $\vec{u}_{i,G}$ is compared to the old vector $\vec{v}_{i,G}$. 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) (Price, 1999):

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

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, $\vec{v}_{i,G}$, as well as each new trial solution, $\vec{u}_{i,G}$, contains the class vectors for all classes and the power value *p*. In other words, DE is seeking the vector $(y(1),\ldots,y(T),p)$ 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 $\vec{v}_{i,G_{max}}$. 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 size of the optimized parameters or if size of the *NP*, also the number of generations used is $G_{max} = 1000$. These values are the same as those used by Luukka and Lampinen (2011), Luukka and Lampinen (2010). However, these selections were mainly based on general recommendations and practical experiences with the usage of DE, (e.g.

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

Luukka & Lampinen, 2011; Luukka & Lampinen, 2010) 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 (Duda & Hart, 1973). 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:

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

Minkowsky metric because is more general than euclidean metric and euclidean metric is included there as a special case when $p = 2$. We also found in Luukka and Lampinen (2011), Luukka and Lampinen (2010) that when p value was optimized it almost never were even near $p = 2$ which corresponds to euclidean metric.

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 parameters are

(1) The class vectors $y_i = (y_i(1),\ldots,y_i(T))$ for each class $i = 1,\ldots,N$
(2) The power value p in (2).

In this study we used differential evolution algorithm (Price et al., 2005) 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 Luukka and Lampinen (2011). The proposed extension to the earlier DE classifier will be described in detail next.

2.2. Extension to optimizing distance measures from pool of distances

Basic underlying idea here is that instead of using Eq. (2) we create a pool of different distance measures and optimize the parameter related to the selection of suitable distance also. For the pool of distances we selected the following distances:

$$d_1(\mathbf{x},\mathbf{y}) = \left(\sum_{j=1}^{T}|x_j - y_j|^{p_1}\right)^{1/p_1}; \quad p_1 \in [1,\infty) \tag{4}$$

(Minkowski metric)

$$d_2(\mathbf{x},\mathbf{y}) = \sum_{j=1}^{T}\sum_{i=1}^{T} c_{ij}(x_i - y_i)(x_j - y_j) = (\mathbf{x}-\mathbf{y})'C(\mathbf{x}-\mathbf{y}) \tag{5}$$

(Mahalanobis-distance; statistical interpretetation: $C = V^{-1}$, where V means an estimation of the covariance matrix of the whole data set)

$$d_3(\mathbf{x},\mathbf{y}) = max_i c_i|x_j - y_j|; \quad c_i \in (0,\infty) \tag{6}$$

(Tschebyscheff metric)

$$d_4(\mathbf{x},\mathbf{y}) = \sum_{j=1}^{T} c_i(x_j - y_j)^2; \quad c_i \in (0,\infty) \tag{7}$$

(Karl–Pearson-distance; statistical interpretation: $c_i = s_i^{-2}$, where s_i^2 means the variance of the i-th coordinate in the whole data set)

$$d_5(\mathbf{x},\mathbf{y}) = \sum_{j=1}^{T}[|x_j - y_j|^{p_2}]/(1 + min\{|x_j|,|y_j|\}) \tag{8}$$

$$d_6(\mathbf{x},\mathbf{y}) = \sum_{j=1}^{T}[|x_j - y_j|^{p_3}]/(max\{|x_j|,|y_j|\}) \tag{9}$$

$$d_7(\mathbf{x},\mathbf{y}) = \sum_{j=1}^{T}|x_j - y_j|/(max\{|x_j|,|y_j|\}) \tag{10}$$

$$d_8(\mathbf{x},\mathbf{y}) = \sum_{j=1}^{T}|x_j - y_j|/(1 + min\{|x_j|,|y_j|\}) \tag{11}$$

$$d_9(\mathbf{x},\mathbf{y}) = \sum_{j=1}^{T}|x_j - y_j|/(1 + max\{|x_j|,|y_j|\}) \tag{12}$$

$$d_{10}(\mathbf{x},\mathbf{y}) = \sum_{j=1}^{T}|x_j - y_j|/(1 + |x_j| + |y_j|) \tag{13}$$

$$d_{11}(\mathbf{x},\mathbf{y}) = \sum_{j=1}^{T}|x_j/(1 + |x_j|) - y_j/(1 + |y_j|)| \tag{14}$$

Collection of distance measures can be found i.e. in Bandemer and Näther (1992). After we had created the pool of distances we needed to find a way to optimize it to the data set at hand. For this we used switch operator which was needed to optimize. For the 11 distances in the pool now we had to add one more parameter to be optimized in order to select the suitable distance. In order to do this properly several issues needed to be considered. In DE we are optimizing a real valued number instead of the integer so now boundaries for this parameter needed to be selected carefully. Here we did it so that for the boundaries we selected interval [0.5, 11.499]. After this what was also needed was to subdivide the value range for each possible choice at hand. So the range for 11 distances was subdivided into 11 equal parts. Then what was used was simply rounding of the real value gained from the DE to nearest integer value. After this the gained integer value was used to select the proper distance function. The applied handling concept for integer valued parameters is explained in detail in Price et al. (2005). In addition to this, as can be noticed from the pool of distances there are different parameter values with different distances which needs to be optimized. What was done in our experiment was that we created additional parameters to be optimized for each of the different parameters in pool of distances. This way we created a situation where it is possible that in the optimization process there is some parameters for which do not need to be optimized, but since there is no way of knowing beforehand which distance would be optimal, this was necessary. Also it showed not to be a problem in the learning process. Basically our vector to be optimized consists now of following components:

$$\vec{v}_{i,G} = \{\{class1, class2 \cdots classN\}, \{switch\}, \{parameters\}\}$$

where $\{class1, class2 \cdots classN\}$ are the class vectors which are to be optimized for data set, $\{switch\}$ is the parameter for finding the optimal distance measure from choices d_1 to d_{11} and $\{parameters\}$ is possible parameters from the distance measures. In this case $\{parameters = \{p_1, p_2, p_3\}\}$. After the vector $\vec{v}_{i,G}$ is divided in its corresponding parts we can calculate the distances between the samples and the class vectors and the classification is done 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).

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 (Newman, Hettich, Blake, & Merz, 1998). Chosen data sets were all such where optimal distance measure was not euclidean distance. The data sets were subjected to 1000 number of generations ($G_{max} = 1000$) and divided into 30 times random splits of testing sets and learning sets ($N = 30$) based on which mean accuracies and variances were then computed. The Fundamental properties of the data sets are summarized in Table 1, also data sets briefly introduced in the following.

3.1. Parkinsons data set

The data sets was created by Max Little of the University of Oxford. Data set is composed of a range of biomedical voice measurements from healthy people and people with Parkinson's disease (PD). Each column in the table is a particular voice measure, and each row corresponds one of 195 voice recording from people who participated in collection of this data. The main aim of the data is to discriminate healthy people from those with PD.

Table 1
Properties of the data sets.

Name	Nb of classes	Nb of attributes	Nb of instances
Parkinsons	2	23	197
Echocardiogram	2	8	132
Horse-colic	2	11	368
Credit approval	2	16	690
Balance scale	3	5	625

3.2. Echocardiogram

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. If this data can be classified accurately enough it would give means for predicting future heart attacks in former heart patients.

3.3. Horse-colic

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.

3.4. Credit approval data

In Australian credit data major concern is in credit card applications. Main decision to be made is whether credit card application should be approved or not. Data set was taken from UCI Machine learning data base (Newman et al., 1998) where it is freely available. Data set consists of 690 instances and has 16 attributes. There are six numerical attributes and eight categorical attributes.

3.5. 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.

4. Results

4.1. The classification results from proposed method with given data sets

The proposed differential evolution classifier was tested with data sets echocardiogram, Horse-colic, credit approval, Parkinsons and Balance scale data set in order to select the optimal distance measure from the defined pool of several different distance measures. 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. Results from the experiments are shown in Table 2.

In Table 3, In each data set we show how many times each optimal distance appeared and its corresponding mean accuracy. The

Table 2
Classification results for the five data sets using $N = 30$ and $G_{max} = 1000$. Mean classification accuracies, variances and optimal distance found are reported in columns 2 to 6. TS is referring to test set and LS to the learning set.

DATA	Mean (TS)	Variance (TS)	Mean (LS)	Variance (LS)	Optimal distance
Echochardiogram	88.36	21.14	96.84	3.29	8, 9, 10
Horse colic	82.55	89.86	89.75	2.21	3, 9, 11
Credit approval data	84.93	3.51	89.92	1.97	10, 11
Parkinsons	84.97	14.21	82.31	5.56	9, 10
Balancescale	90.63	1.68	92.02	0.19	11

Table 3
Data sets results with $N = 30$ and $G_{max} = 1000$ showing convergence to a specific distance.

Data	Distance	No. of times	Mean
Echochardiogram	8	18	90.33
	9	9	84.58
	10	3	89.83
Horse-colic	11	28	84.85
	9	1	83.03
	3	1	59.39
Credit approval data	11	29	84.91
	10	1	85.55
Parkinsons	9	27	84.85
	10	3	86.05
Balancescale	11	30	90.63

Table 4
Data sets results with respect to confidence interval from the proposed method.

DATA	Confidence interval
Echocardiogram	88.36 ± 2.31
Horse-colic	82.55 ± 4.77
Credit approval data	84.93 ± 0.94
Parkinsons	84.97 ± 1.90
Balance scale	90.63 ± 0.65

aim is to show the relationship between the optimized distance and the mean classification accuracy.

4.1.1. The results with echocardiogram data set

Echocardiogram datasets contains 8 attributes. Our proposed classification method gives the mean classification accuracy of 88.36% and variance 21.14 with optimal distances 8, 9 and 10. In the echocardiogram data set, the distance 8 appeared 19 times (out of 30 independent experiment in total) and has the highest mean accuracy. This fact giving preference to distance 8 as an optimal distance as compared to other distances with this data set.

4.1.2. The results with Horse-colic data set

The results with Horse-colic data sets contains 11 attributes. With this data set we got the mean classification accuracy of 82.55% and variance 89.86 with optimal distances 3,9 and 11. In this data set we see that distance 11 gave the optimal mean accuracy of 84.24% compared to other distances. Only twice from 30 runs other distance was found as optimal.

4.1.3. The results with credit approval data set

This data set had 16 attributes. Credit approval data set converged to the optimal distances 10 and 11 with the mean accuracy of 84.93% and variance 3.51. In this category distances 10 and 11 gave the optimal distances with the mean accuracy of 85.55% and 84.91% respectively. With this data set distance 11 was found to be the optimal distance it appears 29 times from 30 runs.

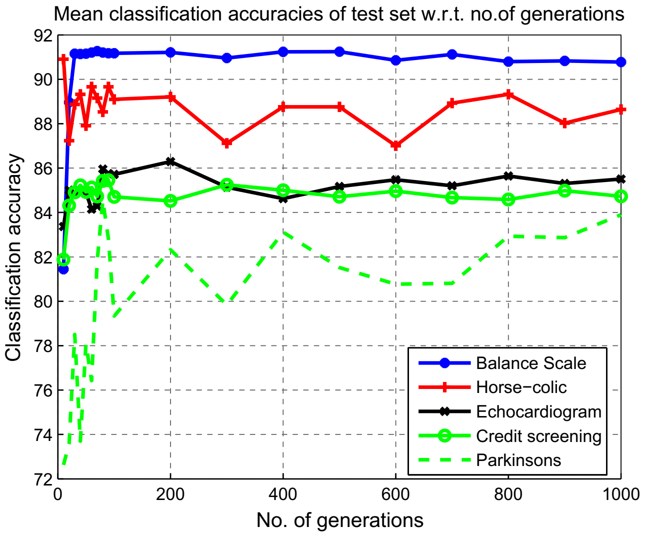

Fig. 1. Mean classification accuracies from data sets studied w.r.t. number of generations computed: Each line represents the mean over all 30 independent experiments with each data.

4.1.4. The results with Parkinsons data set

Parkinsons dataset had 23 attributes. With this data set the mean classification accuracy of 84.97% and variance 14.21 with optimal distances 9 and 10 were received. In the case of Parkinsons data set, the distance 9 showed up 27 times from 30 and had mean accuracy of 84.85%.

For each data set, the mean classification accuracy with 99% confidence interval using student t distribution $\mu \pm t_{1-\alpha} S_\mu / \sqrt{n}$ was also calculated and results are shown in the Table 4.

In Fig. 1 we studied the mean classification accuracies of test data sets with respect to number of iterations and results were plotted from the experiment in which each line represents mean over all 30 independent experiment with each data for the echocardiogram, Credit approval, Parkinson, Horse-colic and Balance scale data sets. With $G_{max} = 1000$ the graph for each data set in Fig. 1 seem to have converged in their respective confidence intervals.

4.2. Comparison of the results with other classifiers

Further analysis of the results were performed using other classifiers, namely k-NN classifer, BackPropagation Neural Network (BPNN) and DE classifer (Luukka & Lampinen, 2011) and compared with the proposed method. The comparison of the proposed method with other classifier is shown in Tables 5–8 as it can be seen in all tables the proposed method seem to perform well by giving slightly higher mean classification accuracy of the chosen data sets compared to other classifiers.

Table 5
Comparison of the results from the proposed method to other classifiers with Parkinsons data.

Method	Mean accuracy	Variance
KNN	82.93	8.33
BPNN	87.50	18.33
DE classifier	83.44	15.34
Proposed method	84.97	14.21

Table 6
Comparison of the results from the proposed method to other classifiers with credit approval data.

Method	Mean accuracy	Variance
KNN	67.77	3.77
BPNN	83.39	8.10
DE classifier	84.19	2.53
Proposed method	84.93	3.51

Table 7
Comparison of the results from the prosed method to other classifiers with Horse-colic data.

Method	Mean accuracy	Variance
KNN	68.53	5.44
BPNN	80.85	19.27
DE classifier	71.76	107.78
Proposed method	82.55	89.86

Table 8
Comparison of the results from proposed method to other classifiers with echocardiogram data.

Method	Mean accuracy	Variance
KNN	90.90	9.80
BPNN	85.80	11.21
DE classifier	86.89	16.43
Proposed method	88.36	21.14

In Table 5, the BPNN classifier seems to have much better mean classification accuracy for the parkinson data sets as compared to other classifiers, though the percentage in the difference with the proposed method is not that large. Second best results came from our proposed method and what current method managed clearly better than its previous version where DE classifier was implemented only using Minkowsky distance.

On another experiment credit approval data were used in doing the comparison of the classifiers. In Table 6,the proposed method performed clearly better than *k*-NN. Also in comparison to BPNN our proposed method gave significantly higher mean accuracy in 0.999 confidence interval. With original DE classifier and the proposed method mean accuracies were not statistically significantly different.

Results from the comparison with the Horse-colic data set can be found in Table 7. With this data set the proposed method achieved again higher mean classification accuracy compared to KNN, BPNN and DE classifier. Improvement with mean accuracy compared to original DE classifier was quite significant, more than 10%.

In Table 8, same comparison as with others is done now for echocardiogram data set. Here proposed method gives second highest mean accuracy where as highest accuracy was achieved with KNN classifier. However is if we again study significance of the difference in mean accuracies we can see that mean accuracies with KNN and proposed method are not significantly different. Also compared to original DE classifier we can see slightly better mean accuracy with proposed method, but it is also not statistically significantly different when performing the mean comparison.

Table 9
Comparison of the results from proposed method to other classifiers with Balance scale data.

Method	Mean accuracy	Variance
KNN	88.06	1.42
BPNN	87.90	2.22
DE classifier	88.66	4.71
Proposed method	90.63	1.68

In Table 9 one can see the results from balance scale data set. As we can see here proposed method clearly outperforms other compared classifiers, now with the mean accuracy of 90.63%.

5. Conclusions and future work

In this article a generalization for differential evolution classifier extended to cover several different types of distance measures is proposed. The results of experiments using the proposed version of DE classifier are indicating that the earlier used Minkowsky distance metrics is often a suboptimal choice, and considerably better classification results can be achieved when also optimizing the distance measure itself to the given data set. This type of generalization lead into even more challenging optimization task since now besides optimizing the class vectors for each class now we also have to optimize the selection of the distance measure and possible extra parameters related with the selected distance measures. Also optimizing the selection of the distance is not a real valued optimization problem, but a mixed discrete one which makes the optimization problem even harder to solve. We selected 11 different distances to our pool of distances to study performance of the method. We selected five different data sets echocardiogram, Horse-colic, credit approval, Parkinsons and Balance scale data sets to study our method. With all of these data sets we found some other than Minkowsky distance to be optimal choice. For all studied data sets the proposed method yielded a higher classification accuracy than the original DE classifier providing further evidences pointing to the direction that also the choice of distance measure itself is important eventhough this issue have been mostly ignored in the previous literature in the field (In almost all classification research papers).

To further study and improve this method we acknowledge the fact that our pool of distances could be extended. Also it can be extended to cover weighted distances. Even further generalization of this would be that instead of optimizing the distance measure to the given data set as what is done over here we can generalize it even further by optimizing the distance measure to the given variable in the data set and using proper aggregation. This will be our future study subject.

Acknowledgments

The work of Jouni Lampinen was supported by the IT4Innovations Centre of Excellence project, reg. No. CZ.1.05/1.1.00/02.0070 funded by the operational programme 'Research and Development for Innovations' of the Structural Funds of the European Union and Czech Republic ministry of education.

References

Bandemer, H., & Näther, W. (1992). *Fuzzy data analysis*. Kluwer Academic Publishers.

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

Booker, L. (1985). Improving the performance of generic algorithms in classifier systems, In J.J. Grefenstette (Eds.), *Proceedings 1st international conference on genetic algorithms* (pp. 80–92) (Pittsburgh, PA).

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), 1905–1917.

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

Dettmann, E., Becker, C., & Schmeiser, C. (2011). Distance functions for matching in small samples. *Computational Statistics and Data Analysis, 55*, 1942–1960.

Duda, R., & Hart, P. (1973). *Pattern classification and scene analysis*. John Wiley & Sons.

Everson, L. R. H., & Penzhorn, W. T. (1988). Experimental comparison on several distance measures for speech processing applications, In *Proceedings southern african conference on communications and signal processing, COMSIG 88* (pp. 12–17).

Fernando, F., & Pedro, I. (2004). Evolutionary design of nearest prototype classifers. *Journal of Heuristics*, 431–545.

Fogarty, T. C. (1994). Co-evolving co-operative populations of rules in learning control systems, Ed T.C. Fogarty, Evolutionary computing AISB Workshop, leeds, selected papers lecture notes in computer science, Vol. 865, pp 195–209.

Holland, J. H. (1985). Properties of the bucket-brigade algorithm, In J.J. Grefenstette (Eds.), *Proceedings 1st international conference on genetic algorithms* (pp. 1–7) (Pittsburgh, PA).

Holland, J. H. (1987). Genetic algorithms and classifier systems: foundations and future directions, In *Proceedings 2nd international conference on genetic algorithms* (pp. 82–89).

Holland, J. H., Holyoak, K. J., Nisbett R. E., & Thagard, P. R. (1987). Classifier systems, Q-morphisms and induction, Ed L. Davis, Genetic algorithms and Simulated Annealing, chapter 9, pp. 116–128.

Jenicka, S., & Suruliandi, A. (2011). Empirical evaluation of distance measures for supervised classification of remotely sensed image with modified multivariate local binary pattern. In *international conference on emerging trends in electrical and computer technology (ICETECT)* (pp. 762–767).

Kaski, S., Sinkkonen, J., & Peltonen, J. (2001). Bankruptcy analysis with self-organizing maps in learning metrics. *IEEE Transactions on Neural Networks, 12*(4), 936–947.

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

Liu, Y., & Sun, F. (2011). A fast differential evolution algorithm using *k*-nearest neighbour predictor. *Expert Systems with Applications, 38*(4), 4254–4258.

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.

Luukka, P., & Lampinen, J. (2011). Differential evolution classifier in noisy settings and with interacting variables. *Applied Soft Computing, 11*, 891–899.

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 (pp. 437–444) Vol. 6113.

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.

Ogawa, A., & Takahashi, S. (2008). Weighted distance measures for efficient reduction of Gaussian mixture components in HMM-based acoustic model. in *IEEE international conference on acoustics, speech and signal processing* (pp. 4173–4176). 2008.

Price, K. V. (1999). *New ideas in optimization*. London: McGraw-Hill. chapter. 6 An Introduction to Differential Evolution, pp. 79–108.

Price, K., Storn, R., & Lampinen, J. (2005). *Differential evolution – A practical approach to global optimization*. Springer.

Robertson, G. (1987). Parallel implementation of genetic algorithms in a classifier system, Ed L. Davis, Genetic algorithms and simulated annealing chapter 10, pp. 129–140.

Schonfeld, J., & Ashlock, D. (2006). Evaluating distance measures for RNA motif search. In *IEEE congress on evolutionary computation CEC 2006* (pp. 2331–2338).

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

Storn, R., & Price, K. V. (1997). Differential evolution – A simple and efficient heuristic for global optimization over continuous space. *Journal of Global Optimization, 11*(4), 341–359.

Su, H., Yang, Y., & Zhao, L. (2010). Classification rule discovery with DE/QDE algorithm. *Expert Systems with Applications, 37*(2), 1216–1222.

Wilson, S. W. (1987). Hierarchical credit allocation in a classifier system, Genetic algorithms and simulated annealing Ed L. Davis, chapter 8, pp. 104–115.

Yu, J., Yin, J, & Zhang, J. (2007). Comparison of distance measures in evolutionary time series segmentation, In *Third international conference on natural computation ICNC 2007* (pp. 456–460).