An Evolvable-Clustering-Based Algorithm to Learn Distance Function for Supervised Environment

Zeinab Khorshidpour, Sattar Hashemi, Ali Hamzeh
Dept. of Computer Science, Shiraz University, Iran

Abstract

This paper introduces a novel weight-based approach to learn distance function to find the weights that induce a clustering by meeting best objective function. Our method combines clustering and evolutionary algorithms for learning weights of distance function. Evolutionary algorithms, are proved to be good techniques for finding optimal solutions in a large solution space and to be stable in the presence of noise. Experiments with UCI datasets show that employing EA to learn the distance function improves the accuracy of the popular nearest neighbor classifier.

Keywords: *distance function learning; evolutionary algorithm; clustering algorithm; nearest neighbor.*

1 Introduction

Almost all learning tasks, like case-based reasoning [1], cluster analysis and nearest-neighbor classification, mainly depend on assessing the similarity between objects. Unfortunately, however, defining object similarity measures is a difficult and non- trivial task, say, they are often sensitive to irrelevant, redundant, or noisy features. Many proposed methods attemp to reduce this sensitivity by parameterizing K-NN's similarity function using feature weighting.

The idea behind feature weighting is that real-world applications involve with many features; however, the objective function depends on few of them. The presence of noisy objects or irrelevant features in a dataset degrades the performance of machine learning algorithms; for many cases, such in the case of k-nearest neighbor machine learning algorithm (K-NN). Thus feature weighting technique may improve the algorithm's performance. This paper introduces a novel weight-based distance function learning to find the weights that induce a clustering by meeting best objective function.

In the recent years, different approach proposed for learning distance function from training objects. Stein and Niggemann use a neural network approach to learn weights of distance functions based on training objects [2]. Eick et.al introduce an approach to learn distance functions that maximizes the clustering of objects belonging to the same class [3]. Objects belonging to a dataset are clustered with respect to a given distance function and the local class density information of each cluster is then used by a weight adjustment heuristic to modify the distance function. Another approach, introduced by Kira and Rendell and Salzberg , relies on an interactive system architecture in which users are asked to rate a given similarity prediction, and then using a Reinforcement Learning(RL) based-techniques to enhance the distance function based on the user feedback [4], [5]. Kononenko proposes an extension to the work by Kira and Rendell for updating attribute weights based on intracluster weights [6]. Bagherjerian et.al propose a reinforcement learning algorithm that can incorporate feedback and past experience to guide the search toward better cluster [7]. They use an adaptive clustering environment that modifies the weights of a distance function based on a feedback. The adaptive clustering

environment is a non deterministic RL environment. Its state space consist of pairs of cluster representatives and distance function weights. Given a state, the learner can take an action that either increases or decreases the weight of a single attribute. The search process uses the Q-learning algorithm to estimate the value of applying a particular weight change action to a particular state.

Finding the best weight in a weight-based distance function is clearly an optimization problem. This paper presents an alternative search space which is more effective than existing approach. Our method combines clustering algorithm and evolutionary algorithms for learning weights of a distance function. Evolutionary algorithms proved to be good techniques for finding solutions in a large solution space and to be stable in the presence of noise. Central idea of our approach is to use clustering as a tool to evaluate and enhance distance function with respect to an underling class structure. In this work, we address the problem of optimizing a weight-based distance function by means of a clustering algorithm and Evolutionary Strategies (ES). We show that combining CLES-DL method with a K-NN algorithm can improve the predictive accuracy of the resulting classifier.

The paper is organized as follows: Section 2 introduces clustering algorithms. In section 3 we review evolutionary algorithms especially the evolutionary strategies. In section 4, we talk about the proposed algorithm. The experimental methodology and UCI data sets used in this work are covered in section 5. Finally the experimental result is described in section 6 as well as the results. The paper ends with the conclusions.

2 Clustering Algorithm

A clustering algorithm finds groups of objects in a predefined attribute space. Since the objects have no known prior class membership, clustering is an unsupervised learning process that optimizes some explicit or implicit criterion inherent to the data such as the squared summed error [8]. The main objective of clustering algorithm is to divide the data into different groups called clusters in such way that the data within a cluster are closer to each other and data from different clusters are farther from each other. Distance criteria are the principle component of every clustering algorithm to approach its objective. Euclidean distance is a commonly used distance measure in most cases. Which is defined as follow:

$$d(x_i, x_j) = \sqrt{\sum_{a=1}^{m} (x_{ia} - x_{ja})^2}. \quad (1)$$

where x_i and x_j are m-dimensional objects and x_{ia} is the value of attribute "a" for a given object x_i. K-means and other partitioning algorithm typically use Euclidean or Manhattan distance metric [8]. Many extensions to these and other partitioning algorithms attempt to employ more sophisticated distance metrics. On the other hand, another group of approaches attempt to learn the distance metric. Including the approach explored in this paper which tries to learn attribute weights based on the following object distance function *d*:

$$d(x_i, x_j) = \sqrt{\sum_{a=1}^{m} w_a (x_{ia} - x_{ja})^2}. \quad (2)$$

where *w* is weight vector whose components are non-negative and

$\sum_m w_m = 1$. It assumes that objects are described by sets of attributes and the dissimilarity of different attributes is measured independently. The dissimilarity between two objects is measured as a weighted sum of dissimilarities between their attributes. To be able to do that, a weight and a distance measure has to be provided for each attribute. We use clustering algorithm for learning weight of distance function, and modified distance, i.e. Equation 2. In this paper we combine clustering and evolutionary algorithm for selecting best weight of the features and modified distance function. In section 3, we review evolutionary algorithms especially the evolutionary strategies.

3 Evolutionary Algorithms

Two major representative algorithms of the evolutionary algorithms are the Genetic Algorithms (GA) and Evolution Strategies (ES) [9], [10]. We introduce main characteristics of evolutionary algorithm in the next sections.

3.1 Chromosome Representation and Population Initialization

For any GA and ES, a chromosome representation is needed to describe each chromosome in the population. The representation method determines how, a solution is represented in the search space and also determines the type of variation operators such as crossover and mutation. Each chromosome is made up of a sequence of genes from certain alphabet which can consist of binary digits (0 and 1), floating-point numbers, integers, symbols (*i.e., A, B, C, D*), etc. It has been shown that more natural representations can get more efficient and better solutions [11]. In this paper we use ES algorithm because a real-valued representation is utilized to describe the chromosome. ES strategies are typically used for continuous parameter optimization. Standard representation of objective variables $x_1 \ldots x_n$ is very straightforward, where each x_i is represented by a floating-point variable. Chromosome contain some strategy parameters, in particular, parameter of mutation operator. Details of mutation are treat in subsection 3.2. Strategy parameters can be divided into two sets, the α values and the σ values. The σ values represent the mutation step sizes, that can be a vector of size *n* or only a singleton. For any reasonable self-adaptation mechanism at least one σ must be present. The α values which represent interactions between the step sizes used for different variables, are not always used [12].

3.2 Mutation

In typical EAs, Mutation is carried out by randomly changing the value of a single bit (with small probability) to the bit strings.
The mutation operator in ES may be applied independently on each objective variable by adding a normally distributed random variable with expectation zero and standard deviation σ as shown in Equation 4, considering $n_\alpha = 0$. The strategy parameters are mutated using a multiplicative, logarithmic normally distributed process as shown in Equation 3.

$$\sigma_i' = \sigma_i . exp(\tau' N(0,1) + \tau N_i(0,1)). \quad (3)$$

$$x_i' = x_i + \sigma_i' . N_i(0,1). \quad (4)$$

where *N*(0, 1) is a normally distributed random variable with zero expectation and standard deviation of 1, $N_i(0,1)$ indicates that the random variable

is sampled for every objective variable independently. The constant $\tau \propto (\sqrt{2\sqrt{n}})^{-1}$ and $\tau' \propto (\sqrt{2n})^{-1}$ can be interpreted in the sense of "learning rates" as in artificial neural networks [13]. Algorithm 1 shows a basic pseudo code of ES.

Algorithm 1 Basic Evolution Strategy Framework

1: Generate some random chromosomes .
2: Select the i best chromosomes based on some selection algorithm (fitness function) .
3: Use these i chromosomes to generate i children using mutation .
4: Go to step 2, until the desired result is achieved .

4 Combing Clustering Algorithms and ES for Distance Learning

In this section, we will give an overview of CLES-DL approach. The key idea of our approach is to use clustering as a tool to evaluate and enhance distance function with repect to an underlying class structure.

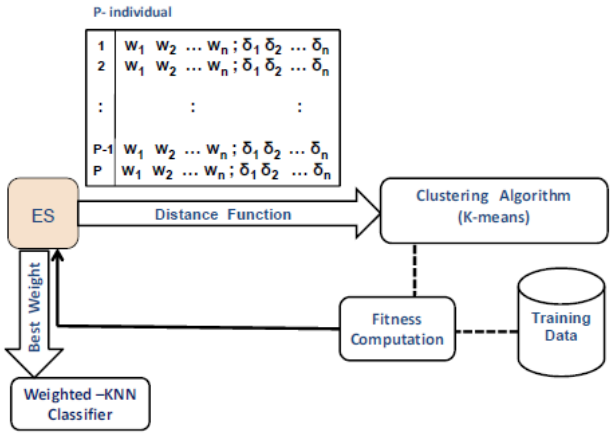

Figure 1: System structure of our distance function learning approach (CLES-DL). Each chromosome contains weights of distance function. ES generates different weights and clustering algorithm used to evaluate the weights.

Our method combines clustering algorithm and evolutionary algorithms for learning weights of a distance function. ES generates different weights and clustering algorithm used to evaluate the chromosomes. Figure 0, shows the overall system architecture of CLES-DL method. Each chromosome contains real-valued genes which are interpreted as the weights of the desired distance function. According to each chromosome, we have a new distance function, and so the clustering algorithm generates new clusters from the data set. In this algorithm, an initial population of size *P* can be randomly generated and lies in the interval [0 , 1].

$$W = [w_1, w_2, ..., w_n; \sigma_1, \sigma_2, ...\sigma_n]. \quad (5)$$

$$w_i = \frac{w_i}{\sum_{l}^{n} w_l}.$$

where *W* is a chromosome and whose ith components are the normalized and non-negative, weight of the ith attribute and σ is the mutation step size according to each attribute.

4.1 Fitness Function

The fitness function is used to define a fitness value for each candidate solution. Our goal is to generate a chromosome with the best (max or min depending on the problem) fitness value. In this problem, the fitness measure is the clustering quality provided by an algorithm which is often a hard and subjective to be measured. We use accuracy criteria as the fitness function to evaluate the results. Assume that the instances in *D* have been already classified in *k* classes $c_1, c_2, ..., c_k$. Consider a clustering algorithm that partitions *D* into *k* clusters $cl_1, cl_2, ..., cl_k$. We refer to a one-to-one mapping, *F*, from classes to clusters, such that each class c_i this mapped to the cluster $cl_j = F(c_i)$. The classification error of the mapping is defined as:

$$E = \sum_{i=1}^{k} | c_i \cap \overline{F(c_i)} |. \quad (6)$$

where $| c_i \cap \overline{F(c_i)} |$ measures the number of objects in class c_i that received the wrong label. The optimal mapping between clusters and classes is the one that minimizes the classification error. We use E_{min} to denote the classification error of the optimal mapping. To obtain the accuracy we compute the following formula:

$$Accuracy = 1 - \frac{E_{min}}{| D |}. \quad (7)$$

5 Experimental Methodology

In order to measure the performance of the proposed algorithms, we employed the 10 fold cross validation methodology. Cross-validation randomly divides the available dataset into *k* mutually exclusive subsets (folds) of approximately equal size, and uses the instances of the *k-1* subsets as the training data and the instances of the remaining subset as the test data. The training dataset is used for determining the classifier's parameters and the test dataset is used to compute the prediction error. The experiment is carried out *i* times to compute the prediction error of the whole dataset. This method is called *k*-fold cross validation. In our experiments we set *k*=10 which is most commonly used value.

5.1 Experimental Setting

Our experiments were run using standard evolutionary strategy with tournament selection. The reported results are based on 10-fold cross validation for each classification task with the following parameter setting.

- Population size: 50
- Number of generation: 50
- Mutation rate: 0.01
- The selection process is ($\mu + \lambda$)

5.2 Description of Datasets

The experimental procedure is to the apply algorithm to several real-world datasets from the UCI machine learning repository [14]. Most of these data sets have been subject of empirical evaluation by other researches. This gives the chance of comparing results among different researches. A summary of characteristics of these data sets are showed in Table 1.

Table 1: Data sets used in the experiments

Data set	Number of Features	Number of Instances	Number of Classes
Glass	9	214	6
Ionosphere	34	351	2
Diabetes	8	868	2
Sonar	60	208	2
Vehicle	18	768	4
Wine	13	178	3
Breast Cancer	32	569	2
Iris	4	150	3

6 Experimental Results

To obtain the best distance function, we run the program 10 times independently. In each run, ES algorithm evolves its population for 50 generation and save the chromosomes with the best fitness value. It is worth mentioning that result of the K-means clustering algorithm based on represented chromosome is used as the fitness value for the evolutionary process.

The whole process of achieving results is depicted in Figure 1. Table 2 shows the results achieved by the algorithm applied to the UCI datasets. CLES-DL method achieve good performance in all datasets. For each dataset, the accuracy of the method with the highest mean accuracy is marked in bold face. The mean accuracy in all UCI data sets using proposed algorithm is 73.87% whereas the original K-means reach to

65.64%. It has about 8% absolute accuracy improvement.

Also to further evaluation, 1-NN classifier algorithm is used [15]. The best chromosome over 10 independent run is passed as a weight vector to weighted 1-NN classifier. Table 3 shows the result of 10 fold cross validation over original 1-NN and 1-NN with CLES-DL. The mean accuracy in 8 UCI data sets by 1-NN with CLES-DL is 80.73% and by original 1-NN is 77.48%. It has about 3% absolute accuracy improvement. When comparing mean accuracies in Table 3, we find that 1-NN with CLES-DL outperforms original 1-NN on 8 datasets (7/8= 0.87), among them. Notably, the accuracy for the wine dataset improved from 75.10% to over 91.50% in contrast accuracy of 1-NN with CLES-DL for breast-cancer dataset is lower than original 1-NN.

Obviously, one may see that our approach outperforms the original K-NN by offering significant improvements.

To explore the advantages of the proposed approach over well known weighting mechanism, we conduct a new series of experiments. In these experiments, we compared our weighting approach against LW1NN algorithm (1-NN classifier with attribute weighting) [3] and bagherjerian method [7]. Bagherjerian approach uses several parameters, according to these parameters four version of this method are represented. Table 4 shows the result of 10 fold cross validation over 1-NN with CLES-DL, LW1NN and bagherjerian's method. The results show the mean accuracy in UCI data sets by 1-NN with CLES-DL is better than LW1NN and bagherjerian's method.

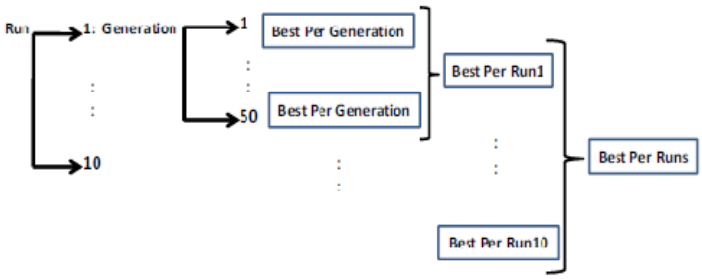

Figure 2: Overall veiw of obtaining result. We run the program 10 times. In each run, ES algorithm evolves population over 50 generation and save chromosome with the best fitness value.

Table 2: Clustering accuracy (and standard deviation), The comparison between K-means with CLES-DL and original K-means on 8 datasets from UCI data repository. All standard deviation less than 0.001 are not denoted.

Data set	K-means_CLES-DL	K-means
Glass	**61.17 ± 0.15**	55.61
Ionosphere	**83.32 ± 0.34**	72.15
Diabetes	**74.96 ± 0.18**	66.02
Sonar	**70.48 ± 0.16**	56.25
Vehicle	**57.23 ± 0.16**	52.23 ± 0.03
Wine	**82.47 ± 0.61**	70.23
Breast-cancer	**64.71**	63.31 ± 0.21
Iris	**96.67**	89.33

Table 3: Classification accuracy (and standard deviation) on 8 datasets selected from the UCI data repository. For each dataset, the accuracy of the method with the highest mean accuracy is marked in bold face.

Data set	1-NN_CLES-DL	1-NN
Glass	70.52 ±1.096	**72.84 ± 0.83**
Ionosphere	**88.90 ± 0.53**	78.46 ± 0.91
Diabetes	**70.58 ± 0.39**	67.30 ± 0.47
Sonar	**85.071 ± 0.53**	82.67 ± 0.65
Vehicle	57.67 ±1.22	**63.67 ± 0.79**
Wine	**91.50 ± 0.62**	75.10 ±1.19
Breast-cancer	**61.39 ± 0.77**	61.10 ± 0.60
Iris	**95.33 ± 0.25**	95.31 ± 0.32

Table 4: Classification accuracy (and standard deviation) on 5 datasets selected from the UCI data repository

Data set	1NN_CLES-DL	Bagherjerian's method				LWINN
		Version_(1)	Version_(2)	Version_(3)	Version_(4)	
Glass	72.39 ± 1.096	68.89 ± 2.00	72.20 ± 1.89	72.01 ± 2.56	**76.26 ± 2.18**	69.95
Ionosphere	88.91 ± 0.53	87.24 ± 0.88	88.24 ± 0.88	87.21 ± 0.91	88.52 ± 1.12	**91.73**
Diabetes	**71.85 ± 0.39**	69.29 ± 0.97	69.75 ± 1.18	70.12 ± 1.11	69.91 ± 1.28	68.89
Sonar	**87.57 ± 0.53**	85.79 ± 1.93	86.12 ± 1.85	86.07 ± 1.48	86.22 ± 1.05	86.05
Vehicle	**70.88 ± 1.22**	69.14 ± 0.89	68.83 ± 1.09	68.59 ± 1.25	70.55 ± 1.12	69.86
Averages	**78.32**	76.07	77.02	76.80	78.29	77.29

7 Conclusions

We present a evolutionary method for learning distance function with the main objective of increasing the predictive accuracy of K-NN classifier. The method is a combination of a clustering algorithm and ES in order to find the best weight of each attribute to be used in a K-NN classifier. The proposed method in this paper overcomes the disadvantages of the K-NN algorithm i.e. its sensitivity to the presence of irrelevant features. As a future work we plan to extend the current work so as it can deal with noise as well.

Acknowledgements

This work was supported by the Iran Tele Communication Research Center.

References

[1] P. perner. Case-based reasoning. Data mininh on multimedia data, Lecture Notes in Articial Intellegence, vol.2558. Springer,Berlin, pp.51-62. 2002.
[2] B. Stein, O. Niggemann. Generation of similarity measures from different sources. 14th International conference on Industrial Engineering Application of Artificial Inteligence and Expert Systems, Budapest, Hungary, Speringer, Berlin, pp. 197-206. 2001.
[3] C. Eick, A. Rouhana, A. Bagherjeiran, R. Vilalta. "Using Clustering to Learn Distance Functions for Supervised Similarity Assessment. Journal of Engineering Applications of Artificial Intelligence", Vol. 19, Issue 4, pp. 395-401. 2006.
[4] S. Salzberg. "A nearest hyperrectangle learning method", Machine Learning 6, pp. 251276. 1991.
[5] K. Kira, L. A. Rendell. "A practical approach to feature selection", Proceedings of the Ninth International Workshop on Machine Learning, Aberdeen, Scotland, UK, Morgan Kaufmann, Los Altos, CA, pp. 249256. 1992.
[6] I. Kononenko. "Estimating attributes: analysis and extensions of RELIEF. Proceedings European Conference on Machine Learning", pp. 171182. 1994.
[7] A. Bagherjeiran, R. Vilalta, C. Eick. "Adaptive Clustering: Obtaining Beter Clusters Using Feedback and Past Experience", Proceedings of the Fifth IEEE International Conference On Data Mining (ICDM'05), 2005.
[8] J. M. McQueen. "Some methods of classification and analysis of multivariate observations", Proc. 5th Berkeley Symp. On Mathematical Statistics and Probability, pp. 281297, 1967.
[9] D. Golberg. "Genetic algorithms in search, optimization and machine learning", Addison-Wesley Publishing Company, 1989.
[10] H. P. Schwefel. "Evolution and Optimum Seeking", Wiley,New York, 1995.
[11] Z. Michalewicz. "Genetic Algorithms+Data Structures=Evolution Programs". AI Series, Springer, New York 1994.
[12] A. E. Eiben, J. E. Smith. "Introduction to Evolutionary Computing", Springer, Natural Computing Series 1st edition 2003.
[13] K. Mehrotra, C. K. Mohan, S. Ranka. Elements of artificial neural networks, MIT Press 1997.
[14] C. L. Blake, C. J. Merz. UCI repository of machine learning database, 1992.
[15] S. Russel, P. Norvig. "Artificial Intelligence: A Modern Approach," Prentice Hall, 2nd edition, 2003.

Zeinab Khorshidpour was born in bandar-lengh, Iran . She received her B.Sc. Degree in Computer Engineering from shahid-beheshti university in 2008. She is currently an M.Sc. student in Artificial Intelligence at Shiraz University. Her research interests include distance function learning for nominal attribute, clustering for categorical data, bio-inspired algorithms.

Sattar Hashemi received the PhD degree in Computer Engineering from the Iran University of Science and Technology, in conjunction with Monash University, Australia, in 2008. He is currently a

lecturer in the Electrical and Computer Engineering School, Shiraz University, Shiraz, Iran. His research interests include data stream mining, database intrusion detection, dimension reduction, and adversarial learning.

Ali Hamzeh received his Ph.D. in artificial intelligence from Iran University of Science and Technology (IUST) in 2007. Since then, he has been working as assistant professor in CSE and IT Department of Shiraz University. There, he is one of the founders of local CERT center which serves as the security and protection service provider in its local area. As one of his research interests, he recently focuses on cryptography and steganography area and works as a team leader in CERT center to develop and break steganography method, especially in image spatial domain. Also, he works as one of team leaders of Soft Computing group of shiraz university working on bio-inspired optimization algorithms. He is co-author of several articles in security and optimization.