Feature Reduction of Nearest Neighbor Classifiers using Genetic Algorithm

M. Analoui, and M. Fadavi Amiri

***Abstract*—**The design of a pattern classifier includes an attempt to select, among a set of possible features, a minimum subset of weakly correlated features that better discriminate the pattern classes. This is usually a difficult task in practice, normally requiring the application of heuristic knowledge about the specific problem domain. The selection and quality of the features representing each pattern have a considerable bearing on the success of subsequent pattern classification. Feature extraction is the process of deriving new features from the original features in order to reduce the cost of feature measurement, increase classifier efficiency, and allow higher classification accuracy. Many current feature extraction techniques involve linear transformations of the original pattern vectors to new vectors of lower dimensionality. While this is useful for data visualization and increasing classification efficiency, it does not necessarily reduce the number of features that must be measured since each new feature may be a linear combination of all of the features in the original pattern vector. In this paper a new approach is presented to feature extraction in which feature selection, feature extraction, and classifier training are performed simultaneously using a genetic algorithm. In this approach each feature value is first normalized by a linear equation, then scaled by the associated weight prior to training, testing, and classification. A knn classifier is used to evaluate each set of feature weights. The genetic algorithm optimizes a vector of feature weights, which are used to scale the individual features in the original pattern vectors in either a linear or a nonlinear fashion. By this approach, the number of features used in classifying can be finely reduced.

Keywords—Feature reduction, genetic algorithm, pattern classification, nearest neighbor rule classifiers (k-NNR).

I. Introduction

THE first important fact to be noted is that classification is a general, broad and not completely developed area, in such a way that shape classification is a specific case where the objects to be classified are limited to shapes. It is important to realize that many important contributions to the theory of pattern classification have been made by the most diverse areas, from biology to human sciences, in such a way that the related literature has become particularly vast and eclectic. Since the concepts and results obtained by classification approaches with respect to a specific area can often be immediately translated to other areas, including shape analysis, any reader interested in the fascinating issue of pattern classification should be prepared to consider a broad variety of perspectives.

In past works (e.g.: [1]-[3]), learning time reduction in classifiers was achieved by means of training the classifier with subsets of the initial training set. Many Different methods were used to explore the space of samples, and decide which sample could maximize the recognition rate such as:

1- Genetic Algorithms
2- Principal Component Analysis (PCA)
3- Neural Networks
4- Thresholding method [4]
5- …

Among several classifiers, the Nearest Neighbor Rule is well known [5] and it proves to have successful results even for a small number of training prototypes. However the present paper deals with a different approach which is feature reduction.

In that context, a method is described to implement Nearest Neighbor optimization by Genetic Algorithms (i.e., with feature reduction). The Nearest Neighbor rule (K-NNR) is a simple and powerful classification technique [5]. In the basic Nearest Neighbor rule classifier, each training sample (described by their features) is used as a prototype and a test sample is assigned to the class of the closest prototype [6]. Its asymptotic classification error is bounded by twice of Bayes error.

When the number of prototypes and/or their feature space is large, this method requires a large memory space and long computing time. Because of this reason, the Nearest Neighbor rule has not been wide used to solve Pattern Recognition problems. For example, the Nearest Neighbor Classifier has been shown to have equivalent recognition performance as Radial Basis Function (RBF) and Neural Network based Classifiers [7]. Although it does not need any training to build a Nearest Neighbor Classifier, it is most expensive to implement it in terms of memory storage and computer time.

The approach consists of using Genetic Algorithms based methods for searching the feature space to apply in Nearest Neighbor Rule prototypes, which is the case in the present paper.

M. Analoui, Assistant Professor, Department of Computer Engineering, Iran University of Science & Technology, Tehran 16846, Iran (e-mail: analoui@iust.ac.ir).

M. Fadavi Amiri, Graduate Student, Department of Computer Engineering, Iran University of Science & Technology, Tehran 16846, Iran (corresponding author: phone: +98-911-118-6208; fax: +98-21-2245-2656; e-mail: mfa294@yahoo.com or fadavi@comp.iust.ac.ir).

II. Nearest Neighbors Classifiers

The non-parametric supervised technique known as the nearest neighbor approach constitutes one of the simplest approaches for classification. Assuming that there is a set S of N samples already classified into M classes C_i and that it is needed to classify a new object x, all that is necessary is to

Identify the sample in S that is closest to X and take its class.

Consider, as an example, the situation illustrated in Fig. 1, there are three classes of objects represented in a two-dimensional feature space. In case it is needed to assign a class to the object represented as a question mark in this figure, the nearest neighbor approach consists of taking the class of its nearest neighbor, which is marked by a question mark. Therefore, the new object is Class I. It should be observed that the performance of the nearest neighbor approach is generally inferior to the Bayes decision criterion.

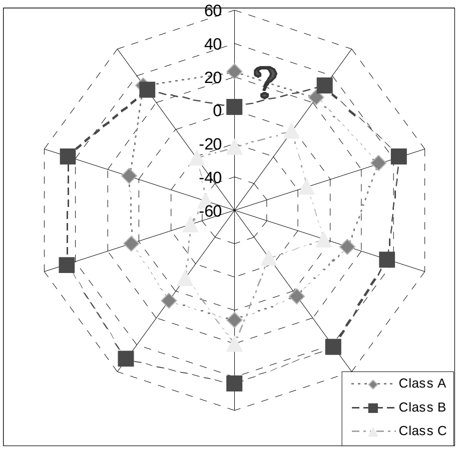


Fig. 1 What is the class of the object identified by the question mark?

The nearest neighbor approach can be immediately extended to the k nearest neighbor's method. In this case, instead of taking the class of the nearest neighbor, k (where k is an integer positive number) nearest neighbors are determined, and the class is taken as that exhibited by the majority of these neighbors (in case of tie, one of the classes can be selected arbitrarily). Theoretically, it can be shown that for a very large number of samples, there are advantages in using large values of k. More specifically, if k tends to infinity, the performance of the k-neighbors method approaches the Bayes rate. However, it is rather difficult to predict the performance in general situations.

III. Genetic Algorithm (GA)

Technically speaking, GAs are probabilistic search algorithms which simulate natural evolution: mechanisms of natural selection and natural genetics are used in order to find solutions to a problem [8].

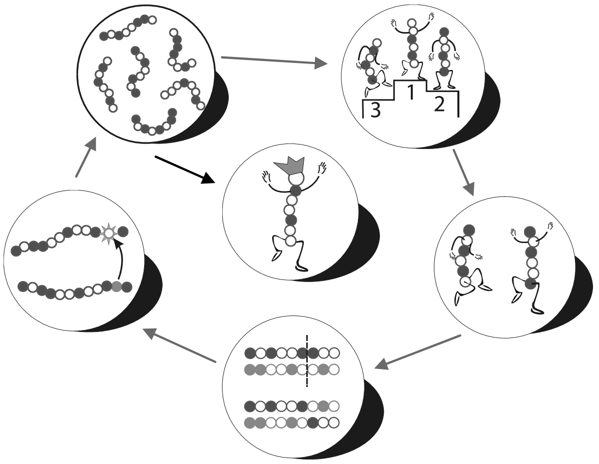
The GA was developed by John H. Holland [9] in the 1960s to allow computers to evolve solutions to difficult search and combinatorial problems, such as function optimization and machine learning. Genetic Algorithms have become a highly effective tool for solving hard optimization problems.

The basic operation of a GA is conceptually simple; the GA Cycle is illustrated in Fig. 2 with following concepts:

1- Maintain a population of solutions to a problem
2- Select the better solutions for recombination with each other, and
3- Use their offspring to replace poorer solutions.

```
{
    1.  Make a set of random individuals
    2.  Until (Best individual found) do
        2.1  Shuffle the population
        2.2  Select 2 parents (P1 & P2) for the mating pool
               from the population
        2.3  For each pair apply crossover with probability $p_c$
        2.4  For each offspring apply mutation (bit-flip with
               probability $p_m$ independently for each bit)
        2.5  Replace P1 & P2 with the resulting offspring in the
               population
    3.  Output the best individual found
}
```

(a)



(b)

Fig. 2 (a) Basic algorithm for a GA (b) The GA Cycle (Selection-Making offspring-Crossover & Mutation)

The combination of selection pressure and innovation (through crossover and mutation - genetic operators) generally leads to improve solutions, often the best found to date by any method [10]. Further details on the Genetic operators, GA codification and GA implementation, can be found in [3].

IV. Genetic Algorithms in Feature Selection and Extraction

Computational studies of Darwinian evolution and natural selection have led to numerous models for solving optimization [11], [12]. GA's comprise a subset of these evolution-based optimization problems techniques focusing on the application of selection, mutation, and recombination to a population of competing problem solutions [9], [13]. GA's are parallel, iterative optimizers, and have been successfully applied to a broad spectrum of optimization problems, including many pattern recognition and classification tasks.

The problem of dimensionality reduction is well suited to formulate as an optimization problem. Given a set of d-dimensional input patterns, the task of the GA is to find a transformed set of patterns in an m-dimensional space ($m < d$) that maximizes a set of optimization criteria. Typically, the transformed patterns are evaluated based upon both their dimensionality, and either class separation or the classification accuracy [14].

Fig. 3 shows the structure of a GA-based feature extractor using classification accuracy as an evaluation criterion. The GA maintains a population of competing feature transformation matrices. To evaluate each matrix in this population, the input patterns are multiplied by the matrix, producing a set of transformed patterns which are then sent to a classifier. The available samples are divided into a training set, used to train the classifier, and a testing set, used to evaluate classification accuracy. The accuracy obtained is then returned to the GA as a measure of the quality of the transformation matrix used to obtain the set of transformed patterns. Using this information the GA searches for a transformation that minimizes the dimensionality of the transformed patterns while maximizing classification accuracy [15].

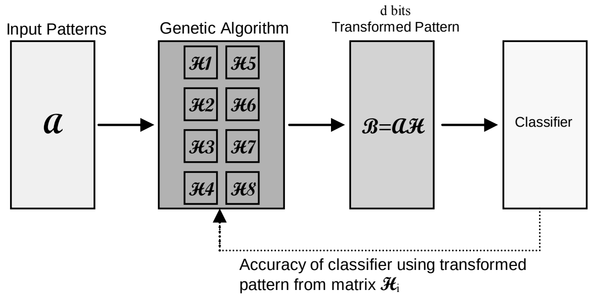


Fig. 3 GA-based feature extractor using an objective function based on classification accuracy. Each transformation matrix from the GA population is used to transform the input patterns, which are then passed to a classifier. The fitness of the matrix is based on the classification accuracy attained on the transformed patterns [15]

A direct approach to using GA's for feature selection was introduced by Siedlecki and Sklansky [14]. In their work, a GA is used to find an optimal binary vector, where each bit is associated with a feature (Fig. 4). If the i_{th} bit of this vector equals 1, then the i_{th} feature is allowed to participate in classification; if the bit is a 0, then the corresponding feature does not participate. Each resulting subset of features is evaluated according to its classification accuracy on a set of testing data using a nearest neighbor classifier.

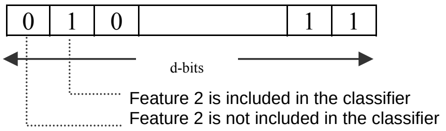

0
1
0
1
1
d-bits
Feature 2 is included in the classifier
Feature 2 is not included in the classifier

Fig. 4 d-dimensional binary vector, comprising a single member of the GA population for GA-based feature selection [15]

The GA feature extraction technique has been expanded to include a binary masking vector along with the feature weight vector on the chromosome [15], [16]. If the mask value for a given feature is zero, the feature is not considered for classification. If the mask value is one, the feature is scaled according to the associated weight value, and is included in the classifier. The inclusion of the mask vector allows the GA to more rapidly sample feature subsets while simultaneously optimizing scale factors for features included in the classifier. An additional improvement specific to the knn classifier was the inclusion of the value of k on the GA chromosome. This modification allows the GA to co-optimize the feature weights and k value for better classification accuracy [15].

V. Proposed Approach

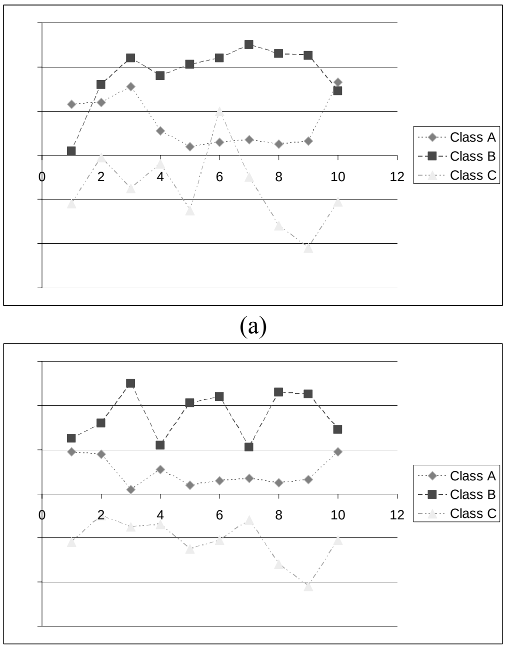
This technique proposed to allow linear feature extraction. The single bit associated with each feature is expanded to a real-valued coefficient, allowing independent linear scaling of each feature, while maintaining the ability to remove features from consideration by assigning a weight of zero. Given a set of feature vectors of the form $X = \{x_1, x_2, \ldots, x_d\}$, the GA produces a transformed set of vectors of the form $X' = \{ w_1 x_1, w_2 x_2 \ldots w_d x_d\}$ where w is a weight associated with feature i.

Each feature value is first normalized by Equation 1, then scaled by the associated weight prior to training, testing, and classification.

$$x'_{i,j} = \left(\frac{x_{i,j} - \min\limits_{k=1\cdots n}(x_{k,j})}{\max\limits_{k=1\cdots n}(x_{k,j}) - \min\limits_{k=1\cdots n}(x_{k,j})} * 9 \right) + 1 \qquad (1)$$

where $x_{i,j}$ is the j_{th} feature of the i_{th} pattern, $x'_{i,j}$ is the corresponding normalized feature, and n is the total number of patterns.

This linear scaling of features prior to classification allows a classifier to discriminate more finely then feature axes with larger scale factors. A knn classifier is used to evaluate each set of feature weights. The effects of linear feature weighting on the knn classification rule are visualized in Fig. 5. Patterns plotted in feature space are spread out along feature axes with higher weight values, and compressed along features with lower weight values. The value of k for the knn classifier is fixed and determined empirically prior to feature extraction.



(a)

(b)

Fig. 5 Effect of scaling feature on k (k = 10) nearest neighbor classification. (a) Original data. (b) Scaled data. It can be seen that after scaling, data can be finely discriminated along this dimension

VI. Conclusion

In this paper a new approach is presented to reduce the dimension of features in a knn classifier which feature selection, feature extraction, and classifier training are performed simultaneously using a genetic algorithm. The genetic algorithm optimizes a vector of feature weights, which are used to scale the individual features in the original pattern vectors. For Better classification accuracy first each future value is normalized with a linear coefficient, and then this normalized feature is used in GA feature reducer that it combines various benefits of feature selection and extraction into a single method. As with feature extraction, the original features are transformed to produce a new set of features. Additionally, since each output feature of the GA feature extraction is based only on a single input feature, the relationships between the original features and the transformed features remain explicit, and easy to identify and analyze.

References

[1] H. Yan, “Building a robust nearest neighbor classifier containing only a small number of prototypes,” *International Journal of Neural Systems*, vol. 3, pp. 361–369, 1992.

[2] V. Ramos, “Using principal component analysis and genetic algorithm techniques to optimize learning time on neural network training for pattern recognition,” in *Proceedings of the RecPad’97*, Coimbra, 1997.

[3] V. Ramos, “Evoluçao e cognicao em an de imagem,” M.Sc. Thesis, IST-UTL, Lisbon, 1998.

[4] N. A. Schmid, “Thresholding method for dimensionality reduction in recognition systems,” *IEEE Trans. Information Theory*, vol. 47, no.7, 2001.

[5] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: John Wiley & Sons, 1973.

[6] B. V. Dasarathy, *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*. IEEE Computer Society Press, 1991.

[7] Y. Lee, “Handwritten digit recognition using k nearest neighbor, radial basis function, and back-propagation neural networks,” *Neural Computation*, vol. 3, pp.440-449, 1991.

[8] J. Hrbrant, "Structure learning of Bayesian networks from databases by genetic algorithms," in *Proceedings of the 1st International Conference on Enterprise Information Systems*, Setubal, Portugal, 27-30 March 1999, pp. 225-231.

[9] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: University of Michigan Press, 1975.

[10] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Mass.: Addison-Wesley Reading, 1989.

[11] A. S. Fraser, “Simulation of genetic systems by automatic digital computers- I: introduction,” *Australian Journal of Biological Sciences*, vol. 10, pp. 484-491, 1957.

[12] D. B. Fogel, *Evolutionary Computation: The Fossil Record*. New York: IEEE Press, 1998.

[13] D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning Reading*. MA: Addison-Wesley, 1989.

[14] W. Siedlecki and J. Sklansky, “A note on genetic algorithms for large-scale feature selection,” *Pattern Recognition Letter*, vol. 10, pp. 335-347, 1989.

[15] M. L. Raymer, W. F. Punch, E. D. Goodman, L. A. Kuhn, and A. K. Jain, “Dimensionality reduction using GA,” *IEEE Trans. Evolutionary Computation*, vol. 4, no. 2, 2000.

[16] M. L. Raymer, W. F. Punch, E. D. Goodman, P. C. Sanschagrin, and L. A. Kuh, “Simultaneous feature scaling and selection using a genetic algorithm,” in *Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA'97)*, San Francisco, 1997, pp. 561-567.