2012 International Conference on Frontiers in Handwriting Recognition

A Novel Technique for Handwritten Digit Classification using Genetic Clustering

S. Impedovo, IAPR Fellow, IEEE S.M.
Dipartimento di Informatica
Università degli Studi di Bari
Via Orabona 4 – Bari – Italy
e-mail: impedovo@di.uniba.it

F. M. Mangini
Dipartimento di Informatica
Università degli Studi di Bari
Via Orabona 4 – Bari – Italy
e-mail: francescomaurizio.mangini@uniba.it

***Abstract*—The aim of this paper is to introduce a novel technique for handwritten digit recognition based on genetic clustering. Cluster design is proposed as a two-step process. The first step is focused on generating cluster solutions, while the second one involves the construction of the best cluster solution starting from a set of suitable candidates. An approach for achieving these goals is presented. Clustering is considered as an optimization problem in which the objective function to be minimized is the cost function associated to the classification. A genetic algorithm is used to determine the best cluster centers to reduce classification time, without greatly affecting the accuracy. The classification task is performed by k-nearest neighbor classifier. It has also been developed a new feature and a distance measure based on the Sokal-Michener dissimilarity measure to describe and compare handwritten numerals. This technique has been evaluated through experimental testing on MNIST dataset and its effectiveness has been proved.**

Keywords: Handwritten Digit Classification, Genetic Clustering, k-Nearest Neighbor.

I. INTRODUCTION

Handwritten digit recognition has received remarkable attention in the field of character recognition. Currently several approaches are able to reach competitive performance for this task, including the ones based on multi layer neural networks, support vector machines and nearest neighbor methods. Neural networks require huge amounts of training data and time to learn effective models, but their feed-forward nature makes them very efficient during runtime. Support vector machines, using recent progresses in convex optimization theory to train classifiers, show a simpler training phase than neural networks, and in the test phase have a complexity which is only a fraction of a brute force k-NN model (considering a non linear kernel SVM) as the number of support vectors generally is a small fraction of the training data. On the other hand, k-NN algorithms, have zero training time but are usually expensive during runtime.

The traditional k-NN rule requires the storage of the whole training set and performs classification based on the closest training examples in the feature space. In particular, when k-NN is considered, classifying an unknown input vector basically consists in finding the top k similar vectors in the given training set and identifying the predominant class among these k neighbors.

This paper presents a new clustering technique for improving handwritten digit recognition using k-NN. Cluster design is considered as an optimization problem and the optimal clustering is the one for which the cost function associated to the classification is minimum. The clustering algorithm is based on a binary-coded genetic algorithm that determines the best cluster centers to reduce k-NN classification time, without greatly affecting the accuracy. For this purpose, it has been explored the capability of histograms of oriented gradients as image features and a distance measure based on the Sokal and Michener dissimilarity measure has been used.

Therefore, the objective of this approach is twofold:

1) the training data clustering drastically reduces the k-NN classification time since it depends just on the number of cluster centers for each class (and, of course, this number is much smaller than the training data size);

2) after the cluster analysis has been done, k-NN classification algorithm will be able to classify, considering only the previously identified cluster centers. For this reason, they have to be chosen so as not to decrease accuracy.

The experimental tests, that have been performed on MNIST data set, demonstrate the effectiveness of the proposed solution compared to other approaches in literature.

The remaining part of this paper is organized as follows. Section 2 describes the feature extraction phase. Section 3 presents the distance measure adopted. Section 4 formulates the clustering problem. Section 5 illustrates how to generate cluster solutions. The genetic algorithm for building the optimal cluster set is illustrated in Section 6. The experimental tests and the results are discussed in Section 7. The conclusions are drawn in Section 8.

II. FEATURE EXTRACTION

Apart from classifiers, accuracy significantly depends on feature extraction. To increase discriminative power, the idea is to acquire information not only about digit local shape, but also about the spatial layout of shape. Local shape information are extracted by the distribution of gradient orientations within some image zones, while spatial layout

978-0-7695-4774-9/12 $26.00 © 2012 IEEE
DOI 10.1109/ICFHR.2012.167

description is obtained by applying different grids to the image at multiple resolution.

This lead to a scheme in which each resolution corresponds to a level. Therefore pattern representations based on binary histograms of oriented gradients have been used. These features basically consist of calculating gradient values and constructing local histograms along certain orientations. In particular, each pixel of the image is assigned to some gradient orientations and the corresponding magnitudes determine intensity values.

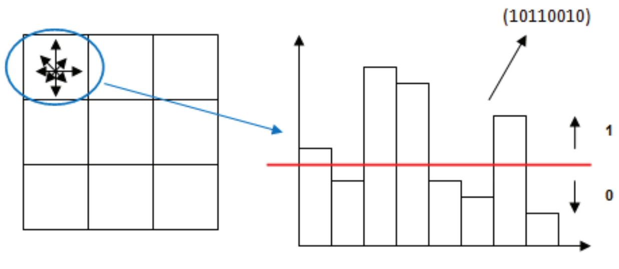

Figure 1. Binary histogram of oriented gradients.

Histograms are built by aggregating these intensity values over the eight chaincode directions for all the pixels belonging to zones of various sizes. Then, a threshold is imposed to histograms and a feature bit is set only if the corresponding measure exceeds the threshold.

It has been constructed histograms at three levels with grids 6x6, 3x3 and 2x2 that cause a partial overlapping of zones. Consequently, the binary histograms of oriented gradient feature provides a binary vector for each level. Level 0 (2x2 grid) is represented by a 32 bit vector, level 1 (3x3 grid) by a 72 bit vector and level 2 (6x6 grid) by a 288 bit vector.

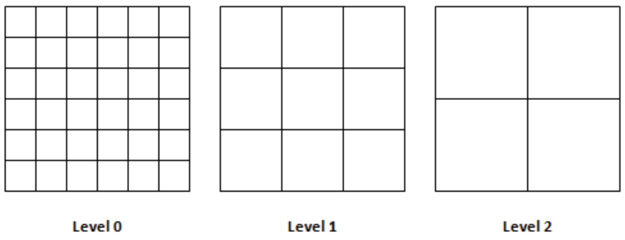

Figure 2. Three level scheme for feature extraction.

These vectors are concatenated together to create a single 392 bit vector. Therefore, the feature vector v is composed of three subvectors $v_{Level\ 0}$, $v_{Level\ 1}$ and $v_{Level\ 2}$:

$$v = (v_{Level\ 0}, v_{Level\ 1}, v_{Level\ 2}) \tag{1}$$

Local gradient magnitude and orientation have been calculated in two steps. First, the Sobel gradient operator has been used to compute the gradient components in strength and direction.

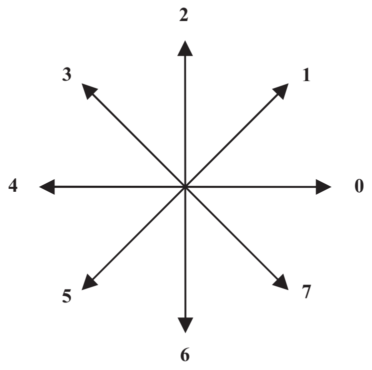

Figure 3. Eight directions of chaincodes.

Second, the range of gradient direction has been partitioned into eight chaincode directions. If a gradient direction has been lied between two standard directions, it has been decomposed in the two components of the two standard directions, as shown in Fig. 4.

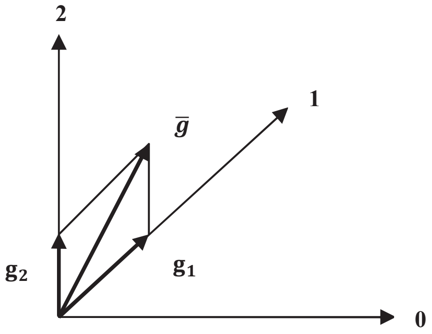

Figure 4. Decomposition of gradient direction.

A preprocessing phase has been performed to generate the input digit images. It has been consisted of binarization using Otsu's algorithm [17], slant correction and size normalization into 36×36 images.

III. DISTANCE MEASURE

In order to make use of spatial information, it is necessary to identify a distance measure able to quantify diversity among binary histograms of oriented gradients at each level of the proposed scheme. For this reason, a weighted sum over histogram differences has been used. It is based on the Sokal-Michener dissimilarity measure [18] which is defined as follows:

$$D_{SM}(X_1, X_2) = \frac{2X_1^T\overline{X_2} + 2\overline{X_1}^T X_2}{{X_1}^T X_2 + \overline{X_1}^T\overline{X_2} + 2X_1^T\overline{X_2} + 2\overline{X_1}^T X_2} \tag{2}$$

where X_1 and X_2 are binary vectors of fixed length l and ${X_1}^T X_2$ is their inner product.

Sokal and Michener dissimilarity measure has been normalized so that its value is always between 0 and 1.

The distance measure adopted in this paper is formalized in the following definition:

$$D_{Measure}(v_1,v_2) = \sum_i w_i D_{SM}(v_{Level\ i_1}, v_{Level\ i_2})/\sum_i w_i \quad (3)$$

where v_1 and v_2 are two feature vectors to be compared:

$$v_1 = (v_{Level\ 0_1}, v_{Level\ 1_1}, v_{Level\ 2_1}) \quad (4)$$

$$v_2 = (v_{Level\ 0_2}, v_{Level\ 1_2}, v_{Level\ 2_2}) \quad (5)$$

and w_0 , w_1 , w_2 are parameters for weighting subvector dissimilarities.

IV. Clustering Problem Formulation

In this paper clustering is formulated as an optimization problem. The optimal clustering is defined as the one that minimizes the following cost function *CF(C)* associated to classification [11, 12]:

$$CF(C) = \eta \cdot Err(C) + Rej(C) \quad (6)$$

where *Err(C)*, *Rej*(C) and η are respectively the misclassification rate, the rejection rate and the parameter for distinguishing errors from rejections.

Being M the number of clusters, the optimal cluster solution C* = $\{C_1^*, C_2^*, ..., C_M^*\}$ can be represented by its cluster centers S_i, and so it has to be found the set $S^* = \{S_i^*\}_{i=1,..,M}$ which minimize the cost function *CF(S)* (6). Supposing to know the optimal set $S^* = \{S_i^*\}_{i=1,..,M}$, in order to classify a generic pattern *p* it has been used the k-NN algorithm (where k = 1 or k is a small number) applied to S^* dataset. It means that the minimum $D_{Measure}(X_{\bar{p}}, S_i^*)$ has been computed and the pattern $\bar{p}$ has been assigned to the class c_j (j=1,…,N) identified by the cluster C^*_i for which the following condition has been verified:

$$\min \{D_{Measure}(X_{\bar{p}}, S_i^*)\}_{i=1,..,M} \quad (7)$$

Therefore the optimization problem can be written as follows:

Find the set of binary vectors $\{S_i^*\}_{i=1,..,M}$ *so that:*

$$CF(S^*) = \min_{\{S^*_i\}} CF(S) \quad (8)$$

where $S^* = \{S_i^*\}_{i=1,..,M}$, is the optimal clustering related to S^*_i, centers i=1,…,M.

V. Generating Cluster Solutions

Cluster solutions have been built in two distinct steps. First, the training input has been divided into N_{pop} folder obtained by a uniform sampling of the training data. Thus every N_{pop} folder has the same number of representatives from each class. Then, the following algorithm, based on a modified version of Carpenter and Grossberg's ART 1 [5], has been applied to each folder:

Input: A folder extracted from the training data.

Output: A set $S = \{S_i\}$ which is a chromosome ϕ_i for the initial population.

Initialization

```
1. Set the $\bar{\delta} = \bar{\bar{\delta}}$, where $\bar{\bar{\delta}}$ is a given value;
2. Set the cluster_centers_set =∅.
```

Algorithm

```
1. Read the next pattern;
2. Find the nearest pattern (applying 1NN rule) and evaluate which class belongs to.
3. If its class is different from the one of the input pattern, put it in the cluster_centers_set.
4. Else find the less dissimilar cluster center among the ones in the current cluster_centers_set, with dissimilarity value smaller than $\bar{\delta}$.
    a. If there is cluster_center, assign the input pattern to that cluster and compute the new cluster center.
    b. Check if this new cluster_center has dissimilarity value smaller than $\bar{\delta}$ to any other cluster center belonging to the cluster_centers_set:
        i. If so, merge the corresponding clusters as one cluster and compute the new cluster center.
        ii. Else nothing.
    c. If there is not cluster_center, build a new cluster and insert the input pattern in the cluster_centers_set as a new cluster center.
5. Repeat steps 1-5 for all the input data.
```

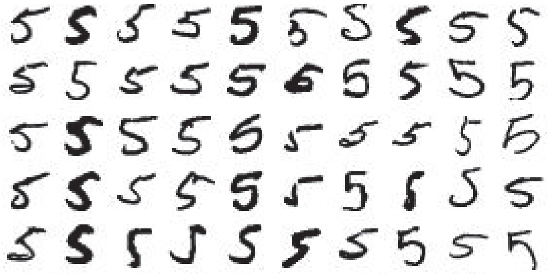

Figure 5. Samples of prototypes of digit 5 in the training set of the MNIST database.

There are some points to be remarked about the proposed approach:

- input patterns are assigned to clusters according to an incremental-update procedure;
- the method is suitable for parallel implementation.

VI. Designing the best cluster solution

To design the optimal cluster set from the ones generated in the previous section, a binary coded genetic algorithm has been considered.

The initial population Pop={$\phi_1, \phi_2,...,\phi_{Npop}$} consists of N_{pop} individuals (N_{pop} even). Each individual is a vector ϕ_i=⟨s_1,...,s_M⟩ that corresponds to a cluster solution $\bar{S} = \{\bar{S_\iota}\}_{i=1,..,M}$. Consequently, the fitness value of the individual ϕ_i =⟨s_1,...,s_M⟩ is taken as the classification cost $CF(\bar{S})$, obtained by (6), where $\bar{S} = \{\bar{S_\iota}\}_{i=1,..,M}$ and $\bar{S_\iota}$ is the center of the i-th cluster. The methodology of the genetic algorithm is as follows:

1. Set j=1;
2. Choose randomly q individuals (q even) from the population P_{j-th} (j=1,..,N_{iter}) and use the best of them as parents;
3. Generate an offspring population as follows: apply crossover operator to these q individuals and mutate the other ones. Then add these new individuals to the previous ones creating the offspring population U_{j-th};
4. Evaluate and assign a fitness value to each individual $\phi_i \in U_{j-th}$, according to the objective function value computed for ϕ_i;
5. Select $s_r N_{pop}$ individuals from U_{j-th} based on their fitness and assign them to P_{j+1-th} (s_r is the selection rate);
6. If the stopping criterion is satisfied, terminate the search and return the current population P_{j+1-th}, else, set j=j+1 and repeat steps 2-6.

Crossover operator works as follows: each parent ϕ_i is divided into p_i+1 points, where p_i is the number of feature vector contained in it. The content inside the j-th and (j+1)-th points is the feature vector corresponding to the cluster center s_j of ϕ_i. An array of random numbers drawn from the range [0, 1] is generated. For each number that is greater than or equal to a given probability p_{cross}, the content inside the corresponding two points is transferred from the first parent to the first child whereas the contents outside is taken from the second parent, maintaining the same order they had before. The second child is obtained by the dual procedure. Therefore crossover operator allows to exchange cluster centers belonging to the same class between a couple of individuals.

Usually mutation operator works on each bit of the chromosome string and flips it with a predefined mutation probability. This needs long random number generations. On the other hand, in order to introduce minor changes to an individual's chromosome, generally, mutation rate has to be very small. This probabilistically means that only a few bits for a population member will be mutated. Hence, to reduce random number generation which is an intensive computation task, for each individual, λ random numbers drawn from the range [0, MxT] are generated, where T is the feature vector dimension. Thus for a given ϕ_i, each random number identifies a bit position to be complemented.

About the stopping criterion, steps from (2) to (6) are repeated until N_{iter} populations of individuals are generated or the cost function satisfies the following condition:

$$\frac{CF\left(S\left(P_{j+1-th}\right)\right)-CF\left(S\left(P_{j-th}\right)\right)}{CF\left(S\left(P_{j-th}\right)\right)} \leq \varepsilon \qquad (9)$$

VII. Experimental Results

The experiments have been carried out using MNIST handwritten digit database provided by LeCun et al. [14]. MNIST training set consists of 60000 samples with a half from NIST's Special Database 3 (SD-3) and another half from Special Database 1 (SD-1). MNIST test set consists of 5000 samples from SD-3 and 5000 samples from SD-1. The best recognition rates achieved using MNIST database are listed in the following table [15]:

TABLE I. State-of-the-art values

Technique	MNIST Test Set		
	Correct (%)	***Error(%)***	***Reject(%)***
Teow et al.	99.41	0.59	0
Belongie et al.	99.37	0.63	0
LeNet-5	99.30	0.70	0

Experiments have been conducted following the approach previously described. Obviously, before complete execution, the most suitable values for each parameter have been pre-estimated by performing some pilot tests.

The selected parameters are: η=5, β=0.35 (parameter of the dissimilarity measure), $w_0 = 1$, $w_1 = 1.3$, $w_2 = 1.8$, N_{pop} =10, $\bar{\bar{\delta}}$ =0.35 (parameter for the algorithm initialization), q=6 (parameter for tournament selection), ${N_{iter}}^{MAX}$=500, ε=0.01, $p_{cross} = 0.75$, $\lambda = 7$ (parameter for mutation operator), s_r=0.5. About the number of nearest neighbors k to be considered, several values have been tested and the best one has been chosen: k = 3.

Using these parameters, the described approach has lead to a recognition rate up to 98.2% and an error rate up to 1% when the test set is used. Recognition rates for each class and overall recognition rate in the MNIST test set are illustrated in the following figure.

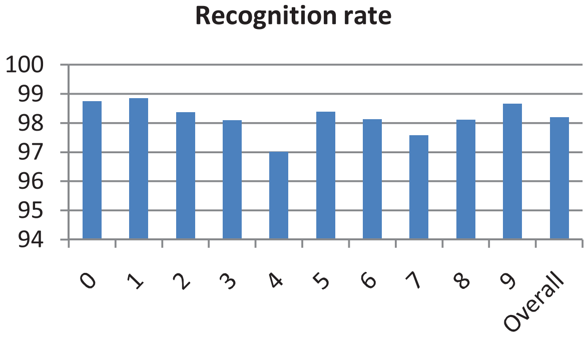

Figure 6. Recognition rates for each numeral class and overall recognition rate in the MNIST test set.

The following table shows the number of cluster centers for each class.

TABLE II. NUMBER OF CLUSTER CENTERS FOR EACH CLASS

Total Number of Centers:701										
Numeral	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
Number of centers	23	12	61	93	89	67	83	92	112	69

The proposed approach has been able to identify 701 cluster centers, which is a small number compared to the training set dimension of MNIST database (60000). On the other hand, it has to be pointed out that the exhaustive k-NN classification reaches an accuration rate equal to 98.83%. Hence, the classification time has been greatly reduced, but notwithstanding this an high accuracy has been maintained.

VIII. CONCLUSION

This paper discusses the problem of the optimal cluster design in order to improve off-line handwritten digit classification with k-NN classifier. To this objective, a new technique for cluster design is introduced that considers clustering as the result of an optimization problem. It is able to automatically discover the optimal number of cluster and their centers reducing the number of dissimilarities to be computed. Hence, it represents a good trade-off between accuracy and speed. The experimental results, obtained using MNIST database, show the effectiveness of the proposed approach. Further research should be done to compare the proposed approach to other clustering techniques, such as hierarchical clustering and k-mean and it will be investigated in future works.

REFERENCES

[1] T. Back, D. Fogel, and Z. Michalewicz (Eds), Handbook of Evolutionary Computation, New York: Institute of Physics Publishing Ltd., Bristol and Oxford University Press, 1997.

[2] T. Baeck, "Evolutionary Algorithms in Theory and Practice: Evolution Strategies", Evolution Programming, Genetic Algorithms, New York: Oxford Univ. Press, 1996.

[3] D. Beasley, D.R. Bull, R.R. Martin, "An Overview of Genetic Algorithms: Part 1, Fundamentals", University Comput. ,Vol 15,n.2,pp. 58-69, 1993.

[4] D. Beasley, D.R. Bull, R.R. Martin, "An Overview of Genetic Algorithms: Part 2, Research Topics", University Computing, Vol. 15, n. 2, pp. 170-181, 1993.

[5] G.A. Carpenter and S. Grossberg, "Adaptive resonance theory", in M.A. Arbib (Ed.), The Handbook of Brain Theory and Neural Networks, Second Edition, Cambridge, MA: MIT Press, 2003.

[6] N. Dalal and B Triggs, "Histogram of oriented gradients for human detection", in Proc. CVPR, 2005.

[7] P.A. Devijver and J. Kittler, "On the edited nearest neighbor rule", in Proc. 5th Int. Conf. Pattern Recognition, Miami, FL, pp.72-80, 1980.

[8] W. T. Freeman and M. Roth, "Orientation histograms for hand gesture recognition" Intl. Workshop on Automatic Face and Gesture-Recognition, IEEE Computer Society, Zurich, Switzerland, pages 296-301, June 1995.

[9] P. E. Hart, "The condensed nearest neighbor rule", IEEE Trans. Inf. Theory, vol. IT-18, no.3, pp.515-516, May 1968.

[10] S. Impedovo, A. Ferrante, R. Modugno, G. Pirlo, "Feature Membership Functions in Voronoi-based Zoning", Emergent Perspectives in Artificial Intelligence, Eds. R. Serra and R. Cucchiara, Lecture Notes in Artificial Intelligence, Vol. 5883, 2009, ISSN: 0302-9743, Springer Publ., pp. 202-211.

[11] S. Impedovo, M.G. Lucchese, G. Pirlo, "Optimal Zoning Design by Genetic Algorithms", IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans, Vol. 36, n. 5, pp. 833-846, Sept. 2006.

[12] S. Impedovo, G. Pirlo, R. Modugno, A. Ferrante, "Zoning Methods for Handwritten Character Recognition: An Overview", Proc. 12th ICFHR, Kolkata (India), 16-18 Nov. 2010, pp. 329-334.

[13] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories", in Proc. CVPR, 2006.

[14] Y. LeCun, MNIST OCR data,
http://www.research.att.com/yann/exdb/mnist/index.html

[15] C.L. Liu, K. Nakashima, H. Sako, and H. Fujisawa, "Handwritten Digit Recognition: Benchmarking of State-the-Art techniques," Pattern Recognition, Vol.36, pp. 2271-2285, 2003.

[16] Z. Michalewicz, "Genetic Algorithms + Data Structure=Evolution Programs", Springer Verlag, Berlin, Germany, 1996.

[17] N. Otsu. "A Threshold Selection Method from Gray-Level Histograms", IEEE Transactions on Systems, Man, and Cybernetics, 1979, 9(1): 62-66

[18] R. R. Sokal and C. D. Michener, "A statistical method for evaluating systematic relationships" , University of Kansas Scientific Bulletin 38, pp. 1409-1438, 1958.

[19] H. Zhang, A. Berg, M. Maire, and J. Malik, "SVM-KNN: Discriminative nearest neighbor classification for visual category recognition", in Proc.CVPR, 2006.