A Genetic Algorithm Based Clustering Approach for Improving Off-line Handwritten Digit Classification

Sebastiano Impedovo IAPR Fellow, IEEE S.M., Francesco M. Mangini, Giuseppe Pirlo
{impedovo, pirlo}@di.uniba.it

Dipartimento di Informatica – Università degli Studi di Bari – Via Orabona 4 – Bari – Italy

Abstract

In this paper a new clustering technique for improving off-line handwritten digit recognition is introduced. Clustering design is approached as an optimization problem in which the objective function to be minimized is the cost function associated to the classification, that is here performed by the k-nearest neighbor (k-NN) classifier based on the Sokal and Michener dissimilarity measure. For this purpose, a genetic algorithm is used to determine the best cluster centers to reduce classification time, without suffering a great loss in accuracy. In addition, an effective strategy for generating the initial-population of the genetic algorithm is also presented. The experimental tests carried out using the MNIST database show the effectiveness of this method.

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

1. Introduction

The k-nearest neighbor (k-NN) classifier is one of the most powerful approaches for handwritten digit recognition. 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. There are two key factors in this approach. The first one is extracting discriminant features from the pattern, the second one is selecting an appropriate dissimilarity measure. Furthermore, for large data sets, the k-NN leads to excessive amount of storage and large computation time in the classification stage.

In literature, any procedure for improving k-NN search by reducing the computational load follows one of these two approaches [2, 12]: “editing” or “condensing” the template space. The first one analyses and modifies the training set with the objective of increasing generalization capabilities. This is performed by removing those representatives that lead to the misclassification error. It can be done, for example, by removing “outlier” patterns or those patterns that are surrounded mainly by others from different classes [2]. The second one has the aim to build a small subset of templates that is a part of the training set, leaving the nearest neighbor decision boundary substantially unchanged [12]. This paper presents a new clustering technique for improving handwritten digit recognition using k-NN. Clustering 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 suffering a great loss in accuracy. For this purpose, the Sokal and Michener dissimilarity measure is used. In addition, an effective strategy for generating the initial-population of the genetic algorithm is also adopted. 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 dissimilarity measure adopted. Section 4 introduces the problem of classification by clustering and a genetic algorithm to find the optimal clustering. An effective strategy to generate the initial-population of the genetic algorithm is illustrated in Section 5. The experimental tests and the results are discussed in Section 6. The conclusions are drawn in Section 7.

2. Feature Choice

As discussed before, the feature choice plays a key role independently of the classifier. For this reason, among many other possibilities, a mixed set of features has been used. This set is composed of gradient-based and pixel density features. The features are extracted in two phases: image pre-processing and feature measuring. In the pre-processing phase, each digit
image is binarized, slant corrected and normalized into 35x35 image. Finally the binary image is skeletonized.

![Figure 1. Main steps of the pre-processing phase.](image1.png)

Then, for feature measuring, a 5x5 grid is superimposed on the skeleton of each numeral and thus it is divided into 5 by 5 zones [4, 9].

![Figure 2. Zoning with 5x5 grid.](image2.png)

The gradient-based features are measured by the following steps. First, the Sobel gradient operator is used to compute the gradient components in strength and direction. The Sobel operator has two masks to compute the gradient components in two axes. These masks are shown in Fig. 3

\[
\begin{pmatrix}
-1 & 0 & 1 \\
-2 & 0 & 2 \\
-1 & 0 & 1
\end{pmatrix}
\quad
\begin{pmatrix}
1 & 2 & 1 \\
0 & 0 & 0 \\
-1 & -2 & -1
\end{pmatrix}
\]

![Figure 3. Sobel Masks for gradient.](image3.png)

Second, the range of gradient direction is partitioned into eight chaincode directions and if a gradient direction lies between two standard directions, it is decomposed in the two components of the two standard directions, as shown in Fig. 5.

\[
D_{S-M}(X_1, X_2) = \frac{2x_1^T x_2 + 2x_1^T x_2}{x_1 + x_2}
\]

![Figure 5. Decomposition of gradient direction.](image5.png)

where \(x_1, x_2\) is the inner product and \(0 \leq \beta \leq 1\).

Sokal and Michener dissimilarity measure is always between 0 and 1 (0 \(\leq D_{S-M}(X_1, X_2) \leq 1\)). In particular if two pattern vectors are completely dissimilar, \(D_{S-M}(X_1, X_2)\) is equal to 1, otherwise if they are identically the same, \(D_{S-M}(X_1, X_2)\) is equal to 0.

3. Dissimilarity measure

Dissimilarity measures for binary vectors have been investigated regarding their discriminative capabilities by Tappert et al. [3] who compared many dissimilarity measures. In this paper, Sokal and Michener dissimilarity measure, which demonstrated to achieve high recognition performance, has been chosen. Let \(X_1\) and \(X_2\) be binary feature vectors of fixed length \(l\), Sokal and Michener dissimilarity measure is defined as in the following equation:

\[
D_{S-M}(X_1, X_2) = \frac{2x_1^T x_2 + 2x_1^T x_2}{x_1 + x_2}
\]

4. Clustering Design

In this section, the suggested technique for clustering design is presented. This technique considers clustering design as an optimization problem. If \(M\) is the optimal number of cluster, the corresponding clustering solution \(C^* = \{C^*_1, C^*_2, ..., C^*_M\}\) is computed as the one for which the cost function \(CF(C)\) associated to classification [7, 8] is minimum, with:

\[
CF(C) = \eta \cdot Err(C) + Rej(C)
\]

where \(Err(C), Rej(C)\) and \(\eta\) are respectively the substitution error rate, the rejection rate and the cost value associated to the treatment of an error with respect to a rejection.
If we represent each cluster by its center \( \Phi_i \), it has to be found the set \( S' = \{ S'_i \}_{i=1,...,M} \) which minimize the cost function \( CF(S) \) (2). Now let’s suppose to know the optimal set \( S' \) and to classify a generic pattern \( p \) using k-NN algorithm (where \( k = 1 \) or \( k \) is a small number) applied to \( S' \) dataset. It means that the minimum \( D_{S-M}(X_p, S'_j) \) has to be computed and the pattern \( p \) will be assigned to the class \( c_j \) \( (j=1,...,N) \) corresponding to the cluster center \( S'_j \) for which the following condition is verified:

\[
min \{ D_{S-M}(X_p, S'_j) \}
\]

Therefore the clustering design optimization problem can be written as follows:

\[
\text{Find the set of binary vectors } S' = \{ S'_i \}_{i=1,...,M} \text{ so that: }
\]

\[
CF(S') = \min \{ CF(S) \} \tag{3}
\]

where \( S' \) is the partition clustering related to \( S' \) centers.

For the optimization problem (3) a binary coded genetic algorithm has been considered. At the moment, let’s assume to have the initial–population \( \text{Pop} = \{ \Phi_1, ..., \Phi_{\text{Pop}} \} \) composed by \( N_{\text{Pop}} \) individuals \( (N_{\text{Pop}} \text{ even}) \). Each individual is a binary vector \( \Phi = \{ s_1, ..., s_M \} \) (where each element \( s_k \) is a T bit vector and T is the feature vector dimension) that corresponds to a partition clustering \( S = \{ S_k \}_{k=1,...,M} \). Consequently, the fitness value of the individual \( \Phi \) is taken as the cost function \( CF(S) \) obtained by (2), where \( S = \{ S_k \}_{k=1,...,M} \) and \( S'_k \) is the \( k \)-th cluster center.

The suggested genetic algorithm performs a “naive evolution” (only selection and mutation) as follows:

1. Set \( j=1 \) and evaluate the fitness of each individual \( \Phi_j \) in the initial-population \( P_{1\text{st}} \);
2. Rank the individuals of the population \( P_{j-th} \) \( (j=1,...,N_{\text{iter}}) \) according to their fitness value, and divide them into \( \alpha \) classes assigning a mutation rate \( \mu_{\text{na}} \) to each of these classes;
3. Generate an offspring population as follows:
   a. Select \( N_{\text{Pop}} \) individuals from \( P_{j-th} \) based on their fitness and assign them to \( P_{j+1-th} \) \( (s_r \text{ is the selection rate}) \);
4. Evaluate and assign a fitness value to each individual \( \Phi_j \) in \( U_{j-th} \), according to the objective function value computed for \( \Phi_j \);
5. Select \( s_r N_{\text{Pop}} \) individuals from \( U_{j-th} \) based on their fitness and assign them to \( P_{j+1-th} \) \( (s_r \text{ is the selection rate}) \);
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.

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

\[
\frac{CF(S_{j+1-th})-CF(S_{j-th})}{CF(S_{j-th})} \leq \epsilon
\]

“Naive evolution” is basically based on the mutation operator. This operator introduces random changes into characteristics of chromosomes and, usually, the new individual generated by mutation is not very far from the original one. In other words, the mutation operator provides a small amount of random search in the proximity of the initial-population when it has nearly converged [11]. Therefore it is necessary to produce near optimal solutions and use them as the initial GA population. This problem will be focused in the next section.

5. Initializing the genetic algorithm

In order to be efficient, the genetic clustering algorithm, previously described, needs at least an a priori estimate of the number of clusters, which is mostly hard for domain experts to figure out. On the other hand, the effectiveness of the genetic algorithm largely depends on initialization setting. In this paper, we focus on an effective strategy for generating the initial population of the genetic algorithm. The objective is to improve the GA performance in terms of solution quality and computational time. The initialization approach consists of two distinct phases. First, the training input has been divided into \( N_{\text{Pop}} \) folders obtained by a uniform sampling of the training data. Thus every \( N_{\text{Pop}} \) folder has the same number of representatives from each class. Then, the following algorithm, based on the Adaptive Resonance Theory [1], has been applied to each folder:

**Input:** A folder extracted from the training data.

**Output:** A set \( S = \{ S_k \}_{k=1,...,M} \) which is a chromosome \( \Phi_i \) for the initial-population

**Initialization**

1. Read the first pattern;
2. Set the \( \bar{\delta} = \delta \), where \( \delta \) is a given value;
3. Set the cluster_centers_set = \( \Phi \).

**Algorithm**

1. Read the next pattern;
2. Find the less dissimilar cluster center from the input pattern, among the ones in the cluster_centers_set, with dissimilarity value smaller than \( \bar{\delta} \);
3. If there is a cluster_center, assign the input pattern to that cluster and compute the new cluster center;
   a. 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;
4. 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.

When a new cluster center needs to be computed, it has to be identified the cluster member which is less dissimilar to all the other members. Thus the new cluster center is the most representative prototype of the cluster.

By the implementation of these steps it has been possible to set up the needed promising initial-population.

6. Experimental Results

The experiments have been carried out using the MNIST dataset of handwritten digit images. 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 [10]:

<table>
<thead>
<tr>
<th>Technique</th>
<th>Correct (%)</th>
<th>Error(%)</th>
<th>Reject(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Teow et al.</td>
<td>99.41</td>
<td>0.59</td>
<td>0</td>
</tr>
<tr>
<td>Belongie et al.</td>
<td>99.37</td>
<td>0.63</td>
<td>0</td>
</tr>
<tr>
<td>LeNet-5</td>
<td>99.30</td>
<td>0.70</td>
<td>0</td>
</tr>
</tbody>
</table>

Experiments have been conducted following the initialization approach previously described. All the training patterns have been visited in order to identify candidates for the initial-population.

In particular, the training set has been divided into $N_{pop}$ folder and afterwards, the genetic algorithm has been executed. Obviously, before complete execution, the most suitable values for each parameter have been pre-estimated by performing some pilot tests.

The selected parameters are: $\eta=5$, $\beta=0.35$ (parameter of the dissimilarity measure), $N_{pop}=10$, $\delta=0.55$ (parameter for the algorithm initialization), $N_{iter}^{MAX}=100$, $\epsilon=0.01$, $\alpha=2$, $\mu_{r1}=0.001$, $\mu_{r2}=0.002$, $s_r=0.5$. About the number of nearest neighbors $k$ to be considered, several values have been tested and the best one has been chosen: $k=5$.

Using these parameters, the described approach has lead to a recognition rate up to 98% and an error rate up to 1% when the test set is used. The exhaustive k-NN classification reaches an accuracy rate equal to 98.72 %. The following table shows the number of cluster centers for each class.

<table>
<thead>
<tr>
<th>Numeral</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of centers</td>
<td>25</td>
<td>11</td>
<td>60</td>
<td>93</td>
<td>89</td>
<td>67</td>
<td>85</td>
<td>92</td>
<td>116</td>
<td>71</td>
<td>709</td>
</tr>
</tbody>
</table>

The proposed approach has been able to identify 709 cluster centers, which is a small number compared to the training set dimension of MNIST database (60000). Therefore, the classification time has been greatly reduced, but notwithstanding this an high accuracy has been maintained.

6. Conclusions

This paper addresses the problem of the optimal clustering design in order to improve off-line handwritten digit classification. For this purpose, a new technique for clustering 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. Thus, 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 evaluate the possibility to use the proposed approach not only for binary, but also for real features. In case, a proper distance measure has to be selected or defined.

Open issues also include how to compare the suggested method to machine learning techniques and they will be investigated in future works.

References