Adaptive Classifier Construction: An Approach to Handwritten Digit Recognition

Tuan Trung Nguyen

Polish-Japanese Institute of Information Technology
ul. Koszykowa 86, 02-008 Warsaw, Poland
nttrung@pjwstk.edu.pl

Abstract. Optical Character Recognition (OCR) is a classic example of decision making problem where class identities of image objects are to be determined. This concerns essentially of finding a decision function that returns the correct classification of input objects. This paper proposes a method of constructing such functions using an adaptive learning framework, which comprises of a multilevel classifier synthesis schema. The schema's structure and the way classifiers on a higher level are synthesized from those on lower levels are subject to an adaptive iterative process that allows to learn from the input training data. Detailed algorithms and classifiers based on similarity and dissimilarity measures are presented. Also, results of computer experiments using described techniques on a large handwritten digit database are included as an illustration of the application of proposed methods.

Keywords: Pattern recognition, handwritten digit recognition,clustering, decision support systems, machine learning

1 Introduction

Pattern Recognition algorithms can be grouped within two major approaches: statistical (or decision theoretic), which assumes an underlying and quantifiable statistical basis for the generation of a set of characteristic measurements from the input data that can be used to assign objects to one of n classes, and syntactic (or structural), which favors the interrelationships or interconnections of features that yield important structural description of the objects concerned. While both approaches seem to be widely used in Pattern Recognition in general, in the particular field of Optical Character Recognition the structural approach, especially methods based on trees an attributed graphs appear to be gaining popularity [7].

Typically, a structural-based OCR system attempts to develop a descriptive language that can be used to reflect the structural characteristics of the input image objects. Once such a language has been established, it is used to describe the characteristic features of the target recognition classes so that new images could be assigned to one of them when checked against those features [2]. Most

J.J. Alpigini et al. (Eds.): RSCTC 2002, LNAI 2475, pp. 578–585, 2002.

existing systems employ some kind of hierarchical descriptions of complex patterns built from *primitives*, elemental blocks that can be extracted directly from input data. (See, e.g., [5],[2]).

Based on the assumption that the construction of a recognition system itself needs to reflect the underlying nature of the input data, we propose a new framework in which the extraction of *primitives*, the development of the descriptive language and the hierarchy of description patterns are all dynamically constructed and improved by an iterative adaptive process driven by the recognition performance achieved on the input data. The framework is essentially based on the granular computing model, in which representational primitives equipped with similar measures play part of information granules, whereas the pattern hierarchy implements the idea of the granular infrastructure comprising interdependencies among information blocks (For a more comprehensive description of granular computing see [6]). This allows for a great flexibility of the system in response to the input data and as a consequence, a gradual improvement of the system's suitability to the underlying object domain.

We later show that the same framework can also be used effectively to generate class dissimilarity functions that can be combined with similarity measures in the final recognition phase of the system, which makes our approach distinct from majority of existing systems, usually employing only class similarity when classifying new, unseen images.

Finally, we present results of experiments on the large NIST 3 handwritten digit database which confirm the effectiveness of the proposed methods.

2 Structural OCR Basics

While both statistical and structural approaches proved to be equally effective in PR in general, the graphical nature of the input data in OCR intuitively favors employing structural methods. A major structural approach is the *relational graph* method, where the image objects from the training data set are first converted to graphs, where specific image features are encoded in *nodes* and relations between them are represented by *edges*. Then, for each target class, a set (library) of protopypical graphs is developed, most often by means of some similarity measures. These prototypes, also called *skeleton* graphs are considered to contain characteristic traits for each target class and, in a way, represent images of that class. Now, given a new image object, a representation graph is extracted and compared with the skeleton graphs from each set. The final class assignment can vary depending on the chosen classification strategy [7].

It is obvious that the successful recognition depends on the choice of:

– the graph model for the image data,
– similarity measures used to build skeleton graphs,
– the distance functions and classification strategy in the final recognition phase

In this paper, we shall show that all three components can be dynamically constructed by an adaptive process based extensively on the actual input image domain.

3 Relational Graph Model for Handwritten Digits

For the researches in this paper we have chosen the Enhanced Loci coding scheme, which assigns to every image's pixel a code reflecting the topology of its neighborhood. The Enhanced Loci algorithm, though simple, has proved to be very successful in digit recognition. For a detailed description of the Loci coding scheme, see [4].

Once the Loci coding is done, the digit image is segmented into regions consisting of pixels with the same code value. These regions then serve as *primitives* to build the graph representation of the image.

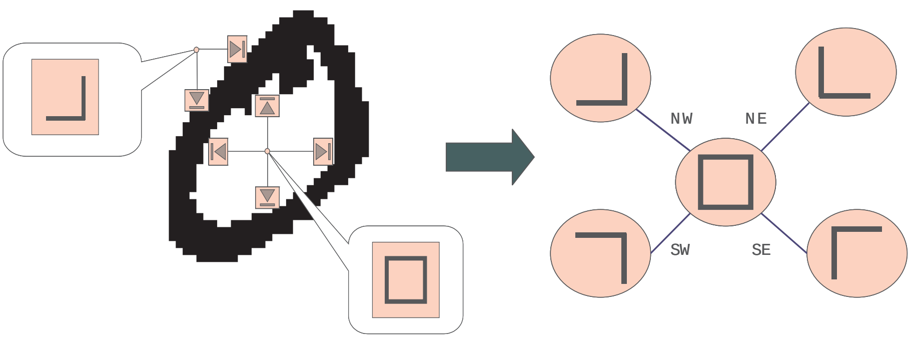

Fig. 1. Graph Model based on Loci Encoding.

Suppose that an image I has been segmented into coded regions $R_1, R_2, ..., R_k$. The graph representation of the image is an attributed labeled graph denoted as:

$$G_I = \{N, P, E\}$$

where $N = \{n_1, n_2, ..., n_k\}$ is a set of *nodes* representing $R_1, R_2, ..., R_k$, P is a set of *properties* of the nodes, containing, among other things, the Loci code, the number of pixels, and the gravity center of the corresponding regions, E is a set of directed labeled edges between pairs of nodes, describing the relative direction between corresponding regions.

One can observe that such a graph G_I will contain local information about black strokes in nodes and non-local features about the shapes of the digit as edges (See also [4]).

3.1 Base Skeleton Graph Construction

Definition 1. *A base segment* $S_b = \{N_b,\ P,\ E_b\}$ *is any 2-node segment of any digit representation graph, i.e.* $|N_b| = 2$ *and* $|E_b| = 1$*. We shall say that a base segment* S_b *matches a graph* G_I*, or* $match(S_b, G_I)$ *if* S_b *is isomorphic to a subgraph of* G_I*.*

Definition 2. *Given a set of base segments with common node and edge sets* $S_1 = \{N, P_1, E\}, S_2 = \{N, P_2, E\}, ..., S_k = \{N, P_k, E\}$*, a base skeleton segment is defined as:*

$$S_{bs} = \{N, P_{bs}, E\}$$

where P_{bs} *is a combined set of properties:*

$$P_{bs} = \{(L_1, f_1), (L_2, f_2), ..., (L_k, f_k)\}$$

with Loci code $L_i \in P_i$*, and the frequency of occurrence of* L_i*:*

$$f_i = \frac{|\{G_m : match(S_i, G_m)\}|}{|\{G_m : \exists 1 \leq j \leq k : match(S_j, G_m)\}|}$$

We shall say that a base skeleton segment $S_{bs} = \{N, \{(L_1, f_1),\ (L_2, f_2), ...,\ (L_k, f_k)\}, E\}$ *matches a graph* G_I*, or* $match(S_{bs}, G_I)$ *if* $\exists 1 \leq j \leq k : match(\{N, L_k, E\}, G_I)$

Having constructed base skeleton segments, we can build *base skeleton graphs*:

Definition 3. *A base skeleton graph (BSG) is any set of base skeleton segments.*

One can look at BSGs as "soft" or "blurred" prototypical graphs that may be used to represent class of digits. By fine-tuning various parameters of the model, e.g. the set of properties or connection labels or by imposing various cut-off thresholds, we can dynamically control the primitives extraction process.

3.2 Graph Similarity Measures

Given a BSG S and a digit representation graph G, the similarity $\tau(S, G)$ is established as follows:

Definition 4. *Suppose that for each node* $n \in S, P(n) = \{(L_1, f_1),\ (L_2, f_2), ..., (L_{k_n}, f_{k_n}\})$ *is the set of* $(code, frequency)$ *pairs at* n*. Then*
if $match(S, G)$ ***then for each*** $n \in S$

$$\tau_n(S, G) = \sum_{i=1}^{k_n} f_i \tau_C(L_i, L^n(G))$$

where $L^n(G)$ is the Loci code found at the node matching n in G, and τ_C is a code-defined similarity function that returns the similarity between two given Loci codes.

$$\tau(S,G) = \sum_{n \in S} w_n \tau_n(S,G)$$

where w_n are connection-defined weight coefficients.

else

$$\tau(S,G) = 0.$$

This definition provides a tolerant matching scheme between representation graphs and skeleton graphs, which allows us to concentrate on the specific aspects of the graph description concepts at each given stage of the learning process. By fine-tuning code-defined and connection-defined weight coefficients, we can achieve a significant flexibility in information granules' construction.

Now let $S = \{S_1, S_2, ..., S_k\}$ be an BSG. The similarity $\tau(S,G)$ can be defined as:

$$\tau(S,G) = \mathbb{F}(\tau_1(S_1,G), \tau_2(S_2,G), ..., \tau_k(S_k,G))$$

where τ_i are single node-defined similarity measures and $\mathbb{F}$ is a synthesis operator. The choice of $\mathbb{F}$ is greatly influenced by the actual structure of the granules' hierarchy, which can either be the interconnections between local patterns in skeleton graphs, or be derived from an interaction with a domain expert. It is noteworthy here that while the expert's domain knowledge can be used to construct the hierarchical infrastructure, we may not rely on the expert's choice of the descriptive language for the primitives. In that way, knowledge passed by the expert will not be blindly used in a stiff manner, but it will rather be refined and combined with other tools on the lower level that we deem more adequate to the problem.

Definition 5. *A distance function between two graphs G_1, G_2 with regard to a skeleton graph S is defined as:*

$$d_S(G_1,G_2) = |\tau_S(G_1) - \tau_S(G_1)|$$

Suppose that for a digit class k, a set (library) of prototypical graphs PG_k has been established. We then can consider different distance functions with regard to that class using various synthesis operators, e.g.

- $d_k(G_1,G_2) = \max_{S \in PG_k} d_S(G_1,G_2)$
- $d_k(G_1,G_2) = \sum_{S \in PG_k} w_S d_S(G_1,G_2)$

where w_S are weight coefficients.

4 Adaptive Construction of Distance Functions

Based on the relational graph, similarity measure and distance function models defined in previous sections, we can construct an iterative process that searches for a optimal classification model as follows:

Algorithm

Step 1 – Initial Skeleton Graph Set

for each digit class k, a set of initial BSGs are constructed based on::

– Heuristic construction of representational BSGs based on class discrimination performance.
– Frequency and histogram analysis.
– Adjustment of connection-defined weights using a **greedy clustering scheme** with recognition rate as quality criteria.
– Manual selection of a number of core initial EBSGs using some domain knowledge.

Step 2 – Distance Function Evaluation

With the established sets of skeleton graphs for each digit class, develop graph similarity measures and distance functions as described in Section 3.3. and perform a k-NN clustering on the input training data collection to obtain class separation.

Evaluate the recognition rate based on developed clusters.

Step 3 – Adjustment of Parameters

Using a **greedy strategy** with regard to the recognition rate, make adjustments to single code similarity function, code-based and connection-based similarity weights and BSG-based distance function weight.
Reconstruct the skeleton graph set as needed. Repeat steps 2-3 until quality criteria are met.

End Algorithm

It can be observed that this is an adaptive iterative process with a two-layered k-NN clustering scheme, aimed at the optimization of three components crucial to the recognition process:

– Primitives extraction process, implemented by Loci coding scheme, code-defined and connection-defined similarity measures.
– Similarity measures model, represented by base skeleton graphs.
– Class distance functions (discriminants), synthesized over extended skeleton graphs.

5 Dissimilarity Measures

So far, similarity measures are used to construct libraries of prototypes so that future input data may be checked against them. Thus, the recognition process relies on how a new object resembles those that had been learnt. However, sometimes it could really help if we knew whether an object u *does not* belong to a class c.

In our relational graph model, dissimilarity to a skeleton graph is defined as similarity to its complementary graph.

Based on the same framework described in Section 4, we can construct complementary skeleton sets for each digit target class or several target classes and use them as discriminants in the recognition process to improve the classification quality.

6 Results of Experiments

In order to verify the developed methods, extensive testing has been conducted. We have chosen the U.S. National Institute of Standards and Technology (NIST) Handwritten Segmented Character Special Database 3, a major reference base within the handwritten character recognition community, as the main data collection. The base contains 223,125 128 $\times$ 128 normalized binary images with isolated handwritten digits from 2,100 different people.(For details see [3])

As a reference experiment's data collection, we have chosen a random portion of the whole base that contained:

– 44,000 digits as a training table, of which 4,000 have been separated for tests during the learning process.
– 4,000 digits for final test table

Table 1. Recognition results with dissimilarity improvement.

Class	No. of skeleton graphs	No. of digits	Misclassified	Reject
0	8	439	0.46 %	0 %
1	7	328	0.61 %	0 %
2	12	417	0.72 %	0 %
3	11	375	2.67 %	0 %
4	14	421	2.85 %	0 %
5	9	389	3.34 %	0 %
6	11	397	1.26 %	0 %
7	8	366	0.00 %	0 %
8	13	432	1.16 %	0 %
9	9	436	0.69 %	0 %
	Total	4000	**1.38** %	0 %

The results obtained qualify our system close to the leading recognition packages tested at NIST, of which the average zero-rejection error rates were 1.70 percent. (See [3])

7 Conclusion

We presented a uniformed framework for the automatic construction of classifiers based on an adaptive scheme. A model for the synthesis of similarity measures from the input data primitives through higher level features has been proposed. The method allows for a flexible learning from the input training data during the construction phase and proved to be effective. The same framework can be used to develop dissimilarity measures that are highly useful in the improvement of the classification quality. Experiments conducted on a large handwritten digit database showed that the method can be applied to practical problems with encouraging results. The framework can easily be adapted to the recognition of other structured objects such as handwritten characters, fingerprints, iris images or human faces.

Acknowledgment. This work has been supported by Grant 8 -T11C02519 from the State Committee for Scientific Researches of the Republic of Poland (KBN).

References

1. Michael R. Anderberg. *Cluster Analysis for Applications.* Academic Press, Inc., 1973.
2. Jan Bazan, Hung Son Nguyen, Tuan Trung Nguyen, Jaroslaw Stepaniuk, and Andrzej Skowron. Application of modal logics and rough sets for classifying objects. In Michel De Glas and Zdzislaw Pawlak, editors, *Proceedings of the Second World Conference on the Fundamentals of Artificial Intelligence*, pages 15–26, Paris, France, 1995. Ankor.
3. J. Geist, R. A. Wilkinson, S. Janet, P. J. Grother, B. Hammond, N. W. Larsen, R. M. Klear, C. J. C. Burges, R. Creecy, J. J. Hull, T. P. Vogl, and C. L. Wilson. The second census optical character recognition systems conference. *NIST Technical Report NISTIR 5452*, pages 1–261, 1994.
4. K. Komori, T. Kawatani, K. Ishii, and Y. Iida. A feature concentrated method for character recognition. *IFIP Proceedings*, pages 29–34, 1977.
5. Z.C. Li, C.Y. Suen, and J. Guo. Hierarchical models for analysis and recognition of handwritten characters. *Annals of Mathematics and Artificial Intelligence*, pages 149–174, 1994.
6. L. Polkowski and A. Skowron. Towards adaptive calculus of granules. In L.A. Zadeh and J. Kacprzyk, editors, *Computing with Words in Information/Intelligent Systems*, pages 201–227, Heidelberg, 1999. Physica-Verlag.
7. Robert J. Schalkoff. *Pattern Recognition: Statistical, Structural and Neural Approaches.* John Wiley & Sons, Inc., 1992.
8. Kodratoff Y. and Michalski R. *Machine Learning: An Artificial Intelligence Approach*, volume 3. Morgan Kaufmann, 1990.