The Effect of the Inhibition-Compensation Learning Scheme on n-tuple Based Classifier Performance

S. Hoque, M. C. Fairhurst, and R. M. Guest

Department of Electronics, University of Kent,
Canterbury, Kent CT2 7NT, United Kingdom.
E-mail: {msh4,mcf,rmg}@ukc.ac.uk

Abstract

The Inhibition-Compensation Learning Scheme (ICLS) has been proposed as a way of enhancing the performance of the Moving Window Classifier. In this paper, the effect of ICLS on three n-tuple based classification techniques has been investigated. Pre-segmented handwritten characters from the NIST database have been used as the pattern data. Results show that approximately 2–6% gain in classification accuracy can be achieved in the OCR task domain with no adverse effect on the classification throughput.

1. Introduction

Automatic recognition of images using digital computers has been an active research domain for many years. In designing and evaluating pattern classifiers, the two factors of principal concern are the *accuracy* of the classification process and the *processing speed* attainable. Improving either of these characteristics often adversely affects the other and to make a system practically viable, it is often necessary to introduce compromises among the available performance indicators. Therefore, the possibility of enhancing the classification accuracy while minimally affecting the classification speed is very desirable. The ICLS scheme investigated in this paper is such a scheme that can be introduced particularly to the n-tuple based classifiers' training session and a significant improvement in recognition accuracy is likely without affecting the classification speed at all.

The fundamental principles of the n-tuple scheme were introduced by Bledsoe and Browing in the late fifties [3]. Although based on a very simple memory network architecture, it offers a number of advantageous features. This method can be applied to any recognition problem in which patterns can be represented by a fixed format binary (or gray scale) sequence and hence has found applications in many diverse task domains such as speech recognition [15], OCR [5], face recognition [11], medical and chemical fields [16], industrial inspection, edge detection and many others.

This paper first briefly introduces three variants of the n-tuple classification system and also describes the ICLS technique. Then the effect of ICLS on the chosen classifiers' accuracy is investigated.

2. Classification Schemes Investigated

In the basic n-tuple system, the n-tuples are formed by selecting multiple sets of n distinct pixels from the pattern image. Each n-tuple is thus an n-bit number extracted from the pattern. This number and the identity of the n-tuple constitute the features of the pattern. For classification, the system classifies an unknown pattern to that class for which it has most features common. To facilitate this, the system needs to remember all the bit-patterns each n-tuple encountered during a training session. For this, it implements a large memory network and sets a flag (by setting a bit to 1) for all n-bit patterns it sees in the training set for all n-tuples and for all candidate classes. A test pattern gets that class label whose model raises most of the flags.

Many variants of this basic n-tuple scheme have been reported in the literature and, of these, three are described in the following sections.

2.1. Frequency Weighted Scheme (FWS)

The FWS is the simplest enhancement of the basic n-tuple classification system. In the basic scheme, both the common and rare feature occurrences are accorded the same discriminatory weight. Thus, the presence of even one rogue pattern in the training set of a class can reduce the discriminatory power of the n-tuple network significantly. As a remedy, in the FWS, instead of setting the flag to record the occurrence of a certain feature in the training set, the relative frequencies are recorded. The frequency counts need to be normalized when different classes have different numbers of training images. The sum of these frequencies corresponding to a particular test image determine its class label. Details of this technique can be found in [2].

2.2. Scanning n-tuple Classifier (sn-tuple)

The Scanning n-tuple (or simply sn-tuple) classifier has been introduced by Lucas *et al.* as a statistical-cum-syntactic method for high performance OCR applications [12, 13, 14]. This is also a variant of the n-tuple classifier except that instead of using the 2-D raw images directly, the operation is conducted on a 1-D gray scale representation of the bitmap image. Another difference between the n-tuple and the sn-tuple is that, whereas each n-tuple samples a set of fixed points in the input space, each sn-tuple defines a set of relative offsets between its input points. Each sn-tuple is then scanned over the entire input space.

The 1-D representation of the binary pattern image is obtained by tracing the contour edges of the image and representing the path by Freeman chain codes [6]. The sn-tuple algorithm is designed to model only one chain code string per pattern and a difficulty arises for images consisting of more than one contour. This is dealt with by mapping a set of strings to a single string by discarding the positional information (i.e., the start coordinates) and then concatenating the strings together. Besides, the length of a chain coded string is dependent on the character class as well as writing style, degree of slant, image quality etc. Since image classes with short chain codes may be adversely affected, all chains are expanded to a predefined fixed length before training and testing.

2.3. Moving Window Classifier (MWC)

In the Moving Window Classifier, a sub-image isolated by a window is initially classified using a Part-Image Classifier(PIC). The window then gradually shifts over the entire image and all the sub-images thus isolated are classified. A final decision fusion stage then combines these individual decisions and assigns a final class label to the test image. An FWS is used as the PIC but, to limit the RAM network size, the FWS is modified so that all the PIC can share the same memory blocks. Details of the MWC scheme can be found in [4, 8, 9]. Although MWC is originally developed for OCR applications, it has also been successfully used for face recognition [7].

3. The Inhibition-Compensation Learning Scheme

The proposed 'inhibition-compensation learning scheme' (ICLS) is based on a technique, designated simply *inhibition*, originally presented by Jorgensen in 1997 [10]. His one-pass scheme introduced an inhibition factor to a selective set of RAM network addresses when an image from a second training set is misclassified. In the ICLS proposed in this paper, a number of alterations to the above scheme are incorporated.

When testing a pattern, n-tuple based classifiers (like most other classifiers) assign similarity scores for all candidate classes. The class having the highest score is the *winning class* and its label is assigned to the test image. The *true class* (the class to which the image actually belongs) may not be the winning class and there may be more than one class having a higher score than the true class. All classes in this latter category can be termed the *competing classes*. When the true class is the winning class, the class with the second largest score can be called the *succeeding class*.

Two training databases are required. The first database is used to pre-train the classifier in the conventional way. The second image dataset is used for the ICLS phase. All the images from this set that are misclassified or classified with relatively less confidence take part in the inhibition-compensation process. In this process, all the memory cells (of the *winning class*' model if an image is misclassified or of the *succeeding class* if less confidently classified) addressed by the n-tuples are given a small inhibition factor. Therefore, if $M_{c,i}$ denotes the content of class 'c' model for the ith n-tuple, then the inhibition process will set $M_{c,i} \leftarrow M_{c,i} - \alpha_{\text{inhibit}}$. The corresponding cell contents of the *true class* model get a small compensation factor. Therefore, $M_{\text{true},i} \leftarrow M_{\text{true},i} + \alpha_{\text{compensate}}$. The process is repeated until all the images in the second training set are correctly classified or, for practical reasons, a fixed number of iterations have elapsed. A number of schemes are recommended here to compute α_{inhibit} and $\alpha_{\text{compensate}}$.

ICLS-A

A fixed inhibition is applied only to the winning class (when misclassified) or to the succeeding class (for weak classification). The same amount of compensation is applied to the true class.

$$\alpha_{\text{inhibit}} = \alpha_{\text{compensate}} = K/m$$

where, K is a scaling constant, and m is the total number of n-tuples.

ICLS-B

This is a variant of ICLS-A. In a complex classification task domain, multiple classes may show considerable similarity to each other. For example, in alphanumeric character recognition, '0'/'D'/'Q' appear very similar and confusion between these classes may occur. To deal with this phenomenon, in ICLS-B, all the competing classes are inhibited, not just the winning class as in ICLS-A. The inhibition and compensation factors remain the same as in ICLS-A.

ICLS-C

This is a variant of ICLS-B. In the previous two schemes, a constant inhibition and compensation factors are imposed, irrespective of the scores assigned to the individual candidate classes. It seems reasonable to impose higher inhibition and compensation when the classifier assigns a relatively higher score to the wrong class than to the true class. In the light of this, in ICLS-C, a variable factor is introduced. For images that are misclassified,

$$\alpha_{\text{inhibit}} = \alpha_{\text{compensate}} = \frac{|\,\text{score}_{winning} - \text{score}_{true}|}{K \cdot m}$$

For weakly classified images,

$$\alpha_{\text{inhibit}} = \alpha_{\text{compensate}} = \frac{|\,\text{score}_{true} - \text{score}_{succeding}|}{K \cdot m}$$

It should be noted that, in all the schemes mentioned here, only the memory contents of the true and competing or succeeding classes are affected in the ICLS process. The memory contents of all other classes remain unchanged.

4. Experiments And Results

The effect of ICLS is investigated on three n-tuple based classifiers, the FWS, the sn-tuple and the MWC. Pre-segmented handwritten characters from the NIST database [1] have been used for classification. This particular dataset consists only of digits and uppercase letters with no distinctions made between '0'/'O' and '1'/'I' character pairs. There are 30899 binary images for training and 20667 for testing, each of resolution 32×32 pixels. Since the proposed ICLS involves a second training set, the NIST training images have been partitioned into two disjoint training sets. To keep computation to a minimum, all pattern normalization measures such as slant correction, broken image and stroke thickness mending, etc. were bypassed. Experiments were conducted in two task domains, the first involving only the digits and the second involving all 34 alphanumeric characters. All the recognition rates reported here are the arithmetic mean from a number of test runs.

For the sake of comparison, the three classifiers are trained in their classical manner using all available training images as a single training set. The achieved performances are presented in Table 1.

The remaining experiments are focused in determining the effect of ICLS on classifier accuracy. In the light of the previous experimental findings on MWC (see [9] for details), the NIST training images are partitioned in the ratio of 1:9. The smaller set is used for preliminary training using the conventional training methods and then the larger part is used for the ICLS. The scaling factor K is determined empirically. The other parameter that is important is the threshold 'θ' used for identifying low confidence decisions. For a correctly identified image, if $\frac{|\,\text{score}_{true} - \text{score}_{succeeding}|}{|\,\text{score}_{true}|} < \theta$, the decision confidence is treated as low. The value of θ should also determined experimentally for optimal performance achievement. Table 2 and 3 shows the accuracy levels thus achieved using the ICLS for $\theta = 0.3$.

Table 1. Performance with classical training

Classifier type	Recognition Rate (in %)	
	Digit only	Digit+Letter
FWS	85.1	71.3
sn-tuple	89.4	80.5
MWC	89.4	75.5

Table 2. Performance after ICLS training (digits only classification)

Classifier	Recognition Rate (in %)		
	ICLS-A	ICLS-B	ICLS-C
FWS	88.18	88.11	87.77
sn-tuple	92.81	92.88	92.66
MWC	91.05	91.19	91.26

Table 3. Performance after ICLS training (alphanumeric classification)

Classifier	Recognition Rate (in %)		
	ICLS-A	ICLS-B	ICLS-C
FWS	74.33	74.05	73.94
sn-tuple	84.24	84.01	84.22
MWC	81.19	81.43	81.66

It is further possible to investigate the effect of a different ratio for partitioning of the training images. For example, a 3:7 partition for the FWS resulted in 74.94% and 74.99% accuracy with ICLS-B and ICLS-C respectively in the 34 class task domain. In the numeral classification task, these figures are 88.42% and 87.88% respectively. On the other hand, if θ is set to 0.8 for the 1:9 partition, MWC accuracy for the digit only classification becomes 91.58%, 91.60%, and 92.10% respectively for the three ICLS schemes presented.

It is clear from the above results that none of the ICLS schemes showed absolute superiority over others for all three classifiers under investigation. ICLS-C produced better accuracies for MWC whereas ICLS-A appears more appropriate for the FWS and the sn-tuple classifier. But it is very clear that significant gain in accuracy (1.7%–3.5% for digits and 2.6%–6.2% in case of alphanumerics) is achievable by incorporating ICLS compared with the conventional training of the n-tuple based classifiers.

5. Conclusions

The n-tuple classifier and its different variants are favoured in pattern classification especially for their simplicity, high speed and ease of hardware implementation. The generally moderate accuracy achievable with the classical n-tuple scheme could be considered a drawback, but the variants like sn-tuple and MWC are capable of producing higher accuracy levels. As has been shown here, this can be further boosted by incorporating ICLS into the process.

Another major advantage of the ICLS is that it involves only the training phase of the classifier, and as such the classification throughput remains unaffected. Thus the high testing speed available with the n-tuple based classifier is not compromised by introducing ICLS.

References

[1] NIST Special Databases 1-3, 6-8, 19, 20, National Institute of Standards and Technology, Gaithersburg, MD 20899, USA.

[2] H. M. S. S. Abdel-Wahab. *Analysis and Implementation of Multi-level Cellular Architectures for Pattern Recognition.* Phd thesis, Electronic Engineering Laboratory, University of Kent, Kent, United Kindom, 1989.

[3] W. Bledsoe and I. Browning. Pattern recognition and reading by machine. pages 225–232, 1959.

[4] M. C. Fairhurst and M. S. Hoque. Moving window classifier: Approach to off-line image recognition. *Electronic Letters*, 36(7):628–630, 2000.

[5] M. C. Fairhurst and T. J. Stonham. A classification system for alphanumeric characters based on learning network technique. *Digital Processes*, 2:321–329, 1976.

[6] H. Freeman. Computer processing of line drawing images. *ACM Computing Surveys*, 6(1):57–98, 1974.

[7] M. S. Hoque and M. C. Fairhurst. Face recognition using the moving window classifier. In *Proceedings of the 11th British Machine Vision Conference (BMVC2000)*, volume 1, pages 312–321, Bristol, UK, 2000.

[8] M. S. Hoque and M. C. Fairhurst. A moving window classifier for off-line character recognition. In *Proceeding of the 7th International Workshop on Frontiers in Handwriting Recognition*, pages 595–600, Amsterdam, The Netherlands, September 2000.

[9] M. S. Hoque and M. C. Fairhurst. An improved learning scheme for the moving window classifier. In *Proceedings of 6th International Conference on Document Analysis and Recognition (ICDAR)*, pages 607–611, Seattle, USA, Spetember 2001.

[10] T. M. Jorgensen. Classification of handwritten digits using a ram neural network. *International Journal of Neural Systems*, 8(1):17–25, 1997.

[11] S. Lin, S. Kung, and L. Lin. Face recognition/detection by probabilistic decision based neural network. *IEEE Transaction on Neural Networks*, 8(1):114–132, 1997.

[12] S. Lucas and A. Amiri. Recognition of chain-coded handwritten character images with scanning n-tuple method. *Electronic Letters*, 31(24):2088–2089, 1995.

[13] S. Lucas and A. Amiri. Statistical syntactic methods for high performance ocr. *IEE Proceedings Vision, Image and Signal Processing*, 143(1):23–30, 1996.

[14] S. M. Lucas. Improving sacnning n-tuple classifiers by pre-transforming training data. In *Proceedings of International Workshop on Frontiers in Handwriting Recognition-V*, pages 143–146, ESSEX, UK, 1996.

[15] M. J. Sixsmith, G. D. Tattersall, and J. M. Rollett. Speech recognition using n-tuple techniques. *British Telecom Technol Journal*, 8(2):50–60, April 1990.

[16] T. J. Stonham, I. Aleksander, M. Camp, J. Pyke, and M. Shaw. Classification of mass spectra using adaptive logic networks. *Analyt. Chem.*, 47:1817–1823, 1975.