Efficient use of training data in the n-tuple recognition method

Article *in* Electronics Letters · December 1993

DOI: 10.1049/el:19931398 · Source: IEEE Xplore

EFFICIENT USE OF TRAINING DATA IN THE N-TUPLE RECOGNITION METHOD

ABSTRACT

A simple technique is presented for improving the robustness of the n-tuple recognition method against inauspicious choices of architectural parameters, guarding against the saturation problem, and improving the utilisation of small data sets. Experiments are reported which confirm that the method significantly improves performance and reduces saturation in character recognition problems.

1 Introduction

The n-tuple recognition method of Bledsoe and Browning [1] often achieves accuracies competetive with the best methods available, while offering an overwhelming advantage in learning speed [2]. A simple technique is suggested for improving the robustness of the method against inauspicious choices of architectural parameters, guarding against the "saturation" problem, and improving the utilisation of small data sets. The technique has been tested in both a near real-time single-user character recognition system implemented on a PC with input via a digitiser [4], and a

freely available multi-writer optical character recognition (OCR) database.

2 The n-tuple method

The simplest variation of the n-tuple method was used. The patterns to be classified are bit strings of a given length. Several (let us say N) sets of n bit locations are selected randomly. These are the n-tuples. The restriction of a pattern to an n-tuple can be regarded as an n-bit number which constitutes a 'feature' of the pattern. A pattern is classified as belonging to the class for which it has the most features in common with at least 1 training pattern of that class.

Precisely, the class assigned to unclassified pattern u is

$$\underset{c}{\operatorname{argmax}} \left(\sum_{i=1}^{N} \Theta \left(\sum_{v \in \mathcal{C}_c} \delta_{\alpha_i(u),\alpha_i(v)} \right) \right) \tag{1}$$

where $\mathcal{C}_c$ is the set of training patterns in class c, $\Theta(x) = 0$ for $x \leq 0$, $\Theta(x) = 1$ for $x > 0$, $\delta_{i,j}$ is the Kronecker delta ($\delta_{i,j} = 1$ if $i = j$ and 0 otherwise.) and $\alpha_i(u)$ is the $i^{\underline{\text{th}}}$ feature of pattern u:

$$\alpha_i(u) = \sum_{j=0}^{n-1} u_{\eta_i(j)} 2^j. \tag{2}$$

Here u_i is the $i^{\underline{\text{th}}}$ bit of u and $\eta_i(j)$ is the $j^{\underline{\text{th}}}$ bit location of the $i^{\underline{\text{th}}}$ n-tuple.

With C classes to distinguish, the system can be implemented as a network of NC nodes, each of which is a RAM. The memory content $m_{ci\alpha}$ at address α of the $i^{\underline{\text{th}}}$

node allocated to class c is

$$m_{ci\alpha} = \Theta\left(\sum_{v \in \mathcal{C}_c} \delta_{\alpha,\alpha_i(v)}\right). \tag{3}$$

Thus $m_{ci\alpha}$ is set if any pattern of $\mathcal{C}_c$ has feature α and unset otherwise. Recognition is accomplished by tallying the set bits in the nodes of each class at the addresses given by the features of the unclassified pattern.

3 The modification

Although it has much in common with other neural and statistical learning methods, this method departs from the common philosophy of fitting data to a model by minimising an error measure with respect to model parameters. Therefore it suffers relatively little from overfitting on small datasets. But whereas a minimisation-based method always benefits from an increased amount of training data, the n-tuple method can suffer from "saturation". The problem is that because bits in memory are set but never reset, the contents can tend toward solid ones, so the capacity for discrimination is lost. This problem will clearly be present if the amount of memory in each node (2^n bits) is too small for the amount and noisyness of the data. Performance can also be adversely affected by setting n too large [3], which may also require an impractical amount of memory.

In order to optimise the amount of training data presented to the system a simple "test before train" heuristic was developed to restrict the sum in (3). A training

pattern was used to adjust RAM contents only if it was not correctly classified by the system in its current state of training. This process was repeated by rescanning the data until no further adjustments were required. (One example from each class is presented, then another, etc.) No more than 3 passes over the data were ever required in practice.

This method cannot be expected to help if n is chosen too large, but can prevent saturation if n is too small. Furthermore, it allows the system scope to adjust more closely to the structure of the data, in which relatively few or many features may distinguish different pairs of classes. More examples of classes having a large variance over the feature set would be selected than would be for classes concentrated on a few features.

4 The data

Two sources of data were used to test the method.

1. The PC/digitiser system was used to collect 125 sets of 10 samples of isolated handwritten digits $\{1, ..., 9\}$ from one writer. These were represented as 8×8 pixel arrays. 90 of the sets were randomly selected for training and the remaining 35 were used for testing.

2. The fl3 subset of NIST Special Database 3 was also used. The full database [1]

[1] For sale on CD-ROM from Standard Reference Data, National Institute of Standards and Technology, 221/A323, Gaithersburg, MD, 20899, USA.

includes 223125 digits handwritten by 2100 US census workers from throughout the USA, represented as 32×32 pixel arrays. The fl3 subset [2] contains a total of at least 140 samples of each digit distributed unevenly over 49 of these writers. 100 samples of each digit were randomly selected from the first 140 for training, and the remaining 40 used for testing.

On the single-writer data, a system of 4096 4-tuples improved insignificantly from 99.2%±0.3% to 99.4%±0.3% recognition accuracy when the test-before-train principle was applied, but saturation (the percentage of ones in memory) was reduced from 48.1%±0.3% to 26.6%±0.7%. These figures are means and standard deviations over 10 runs using different random mappings η in (2). The system required an average of only 10 examples from the 90 presented in order to capture the features of each digit.

On the NIST fl3 data, the same 4-tuple system improved quite significantly from 84.7%±0.9% recognition accuracy to 89.4%±1.6% in one pass of the test-before-train method, and then to 91.9%±0.9% on a second pass, after which no further improvement was possible. Saturation was lowered from 51.39%±0.05% to 38.39%±0.28%, and an average of 44 of the 100 available training patterns were used.

The full NIST dataset was studied by 26 organisations with a total of 118 OCR systems represented at the First Census Optical Character Recognition Systems Conference [5]. Typical recognition rates were around 95% on the digits, with only one system (based on a set of multilayer perceptrons) approaching the human performance

[2] Freely available by anonymous ftp from sequoyah.ncsl.nist.gov.

of 98.5%. For a system using less than 0.05% of the training data, the n-tuple method seems quite respectable. Extension of these preliminary tests to the full dataset would be an interesting project.

Similar tests were run using various tuple sizes from 2 to 10, with qualitatively similar results. Recognition accuracy is poor for 2-tuples, best for 4-tuples, and degrades slightly as the size increases beyond 4.

5 Conclusions

These results demonstrate in character recognition applications that the test-before-train heuristic provides a convenient way to control saturation in an n-tuple recogniser, thereby making efficient use of the available training data.

References

[1] W. W. Bledsoe and I. Browning. Pattern recognition and reading by machine. In *Proceedings of the Eastern Joint Computer Conference*, pages 232–255, Boston, 1959.

[2] R. Rohwer and D. Cressy. Phoneme classification by boolean networks. In *Proceedings of the European Conference on Speech Communication and Technology*, pages 557–560, Paris, 1989.

[3] R. Rohwer and A. Lamb. An exploration of the effect of super large n-tuples on single layer ramnets. In N. Allinson, editor, *Proceedings of the Weightless Neural Network Workshop '93, Computing with Logical Neurons*, pages 33 – 37, University of York, 1993.

[4] Roland Tarling. Computer recognition of hand-printed characters using weightless neural networks. Final year project report, Dept. of Computer Science and Applied Mathematics, Aston University, Birmingham, UK, 1993.

[5] R. Wilkinson, J. Geist, S. Janet, P. Grother, C. Burges, R. Creecy, R. Hammond, J. Hull, N. Larsen, T. Vogl, and C. Wilson. The first census optical character recognition systems conference. Technical Report NISTIR 4912, National Institute of Standards and Technology, Gaithersburg, MD, USA, 1992.

Roland Tarling

92 Elmcroft Ave.

Wanstead

London E11 2DB

Richard Rohwer

Dept. of Computer Science and Applied Mathematics

Aston University

Aston Triangle

Birmingham B4 7ET