RECOGNITION OF HAND-PRINTED CHARACTERS VIA INDUCT-RDR[1]

Adnan Amin[1] **and Sameer Singh**[2]

[1] School of Computer Science & Engineering
University of New South Wales
Sydney 2052, Australia
(email: amin@cse.unsw.edu.au)

[2] Department of Computer Science
University of Exeter
Exeter EX4 4PT, UK
(email: s.singh@exeter.ac.uk)

Abstract

The goal of character recognition research is to simplify and automate the development of character recognition algorithms. We describe here an approach based on applying pre-processing to data sets of Latin characters and then applying a machine learning approach to the data sets to build a knowledge base able to classify unseen pre-processed characters. The machine learning method, Induct/RDR, has a number of features that make it particularly suitable for character recognition. It has the potential to integrate automatic analysis with a manual knowledge acquisition methodology if further refinement is required. Initial results on hand-printed Latin characters show the recognition accuracy of up to 90.2% on unseen cases for the machine learning system.

Kywords: Handprinted Latin Characters, Structural approaches, Induct Machine Learing, Ripple Down Rules, Character recognition

1. Introduction

For the past three decades there has been an increasing interest amongst researchers in problems related to machine simulation of the human reading process. Intensive research has been carried out in this area with a large number of technical papers and reports in the literature devoted to character recognition. This subject has attracted immense research interest not only because of the very challenging nature of the problem, but also because it provides, the means for automatic processing of large volumes of data in postal code reading, office automation, and other business and scientific applications.

This paper presents a new technique for the recognition of hand-printed Latin characters using Induct/Ripple Down Rules (RDR) [1].

Conventional methods have relied on hand-constructed dictionaries that are tedious to construct and difficult to make tolerant to variation in writing styles. Induct/RDR uses learning by induction to build the knowledge base. An advantage of Induct/RDR methodology over other machine learning methods is that it produces a very compact knowledge representation [2] and allows knowledge bases to be maintained by users without the support of a knowledge engineer[3].

The classification of characters is performed using a two stage classifier; we will call these as primary and exception classifiers. The primary classifier is the backbone of the classification system and is built using the INDUCT/RDR procedure described later. The exception classifier system is built to enhance the results of the overall system. The misclassified and rejected characters from the primary classifier are used in this system. The aim here is to make the system aware of the mistakes that it made earlier and give it manual clues (additional rules) to avoid these mistakes in the future. The generation of new rules in this exception classifier is done in a very simple manner. For each character misclassified, it is displayed to an expert on a computer screen with its individual primitives that were automatically detected during the feature detection phase. For each primitive detected, the computer confirms with the expert that the correct primitive was detected. When the computer makes a mistake in judging the primitive type, the expert corrects the system manually and the computer either modifies an

[1] A. Amin and S. Singh. Recognition of Hand-printed Characters via Induct-RDR, Proc. International Conference on Document Analysis and Recognition, (ICDAR'99), Bangalore, (20-22 September, 1999).

existing rule or creates a new rule to take this exception into account. A user interface was developed to identify such differences graphically and develop the exception classifier's knowledge base. At the end of this dual classification procedure, we are left with rules in the primary classifier, as well as new rules generated by the exception classifier. A final set of rules is now used together for testing unseen data.

2. Definition of the Feature Sets

The characters are digitised using a 300 dpi scanner, pre-processed and thinned. The simple structural information such as lines, curves and loops that describe the characters are extracted by tracing the thinned image with some pre-defined primitives. A moving window of fixed size (3x3 pixels) traces the skeleton of the image by describing the Freeman code. The initial position of the window is dependent on the end points found in a character or if the character has an unbroken loop, it starts from the top left corner. A detailed description of the feature extraction process is given in [4]. The characters are then classified using a primary classifier and an exception classifier. The primary classifier uses a machine learning program (INDUCT/RDR) which uses an induction algorithm for the generation of the classification rules.

3. Classification using Induct/RDR

A data set of pre-processed Latin characters is passed on to the Induct/RDR machine learning algorithm[1]. This produces a knowledge base which is then used with an interpreter to return a classification for an unseen character. The unseen characters go through the same preprocessing as the training data.

The most common approach to machine learning for classification tasks, as exemplified in C4.5 [5], is to attempt to build a decision tree where each node represents an attribute and there is a branch for each value of the attribute. There are a variety of heuristics for deciding on which is the most useful attribute to add as a node at any point. The aim of the attribute selection is to produce a tree which is highly accurate on unseen cases and is fairly compact. C4.5 uses an entropy/information based algorithm which essentially attempts to find the most important attribute in separating the population into different classes at any given point in the decision tree. The information measure used to build trees maximises overall separation of the population into classes.

In inductive machine learning, a set of examples of a concept are presented to the rule- induction program. The program's task is to identify what collection of attributes and values define the general concept that describes the examples. Then the program attempts to build a set of logical rules by examining each of the base examples. This approach can be applied for the automated construction of knowledge base for an expert system. Induct uses induction to build the knowledge base in terms of classes, attributes, attribute values and rules.

An induction task can be stated as: given a set of item descriptions whose classes are known, induce rules to describe the examples. The induction task involves the knowledge engineer drawing up a training set of examples that describe the problem. This is analysed by the computer program that produces induced rules. These induced rules describe the training set, and they can be used to predict results not in the training set. For this reason the selection of training set and features affects the quality of the induced rules. Hence it is critical to have an accurate feature extraction module for high quality classifier. The model should be general so that it can be used to predict classification of cases other than those used for the construction of the knowledge base. The major difficulty for inductive approach is providing well-classified cases. The suitability of training set for inductive method does not just depend on the number of cases but how well classified cases are.

3.1. RDR (exception classifier)

In our study, an expert observes the performance of the KBS-in our case character classifier. If the system provides incorrect classification, the expert is presented with all the features exhibited by the case which was not considered by the system when determining the incorrect classification. The expert is then asked to identify the critical features from the list that may lead to correct classification. Based on this, the system automatically updates the knowledge base. The graphical interface provides substantial help to an expert to refine and correct classification rules the system is currently using. Applied to character recognition, it is possible for an expert to explain why they have classified a given character differently from the automatic analysis. The automatic classifier is updated and refined step

by step by pointing out the differences between what the expert considers crucial and the system's classification rule being used. The overall procedure is conducted using a windowed graphical user interface. The interface has windows that display the misclassified character, and the features that led the automatic classifier to a mistake in recognition.

This study employs manual RDR techniques for knowledge acquisition for the exception classifier which handles rejected and misclassified cases. The user is able to analyse the feature vectors and relationships amongst features and train the system on-line using RDR techniques for knowledge acquisition to develop an expert system. This allows the character recognition system to improve its performance on a continuous basis. This section explains the concepts, advantages and drawbacks of this technique.

Ripple down rule knowledge acquisition methodology resulted from long term maintenance of GARVIN-ESI, a small medical expert system. Ripple down rules methodology allows incremental changes to knowledge base without causing unwanted side effects to the existing knowledge base. Studies have shown that this approach allows users to change the knowledge base without the need of a knowledge engineer.

The root of the tree provides a default conclusion if the rule linked to the child is not satisfied for a particular case. In RDR, rules are only added when the expert system incorrectly classifies a case or fails to classify a case (Rule satisfying the default rule but not any other rule.) The creation of extra rules in the expert system is to cater for the wrongly classified cases. When a rule is added to the expert system, the case that prompted the rule is also stored with the rule.

In RDR, if a case is satisfied by the parent rule which does not have a dependent, i.e. no branches, the conclusion associated with that rule is asserted. On the other hand, if the rule has a TRUE branch then that branch condition is also tested. The conclusion of the parent is returned only if the rule on the branch does not satisfy the case. If a rule is not satisfied for a case, the child rule with an IF FALSE link is tested. If the case satisfies the child rule, its conclusion takes precedence on the conclusion of the IF TRUE link to the parent rule. The new rule to the knowledge base should be such that additions do not cause any misclassification of any previous cases that prompted the additions of the rules. The RDR methodology ensures that such conflicts are effectively managed.

One of the major advantages of RDR in hand written character recognition problem is handling character misclassifications. For example, capital H and capital A may have identical features and relationship among features (i.e. feature vector /,---,\). We can define custom made functions on a particular node with very specific functionality to compute the ratio of the distance between the endpoints of the forward and backslash to correctly identify the character.

3.2. Induction Using Induct/RDR

Induct /RDR takes as input a training set entered as a file of ordered set of primitives values, each terminated by a comma, and uses induction techniques to produce a set of rules in the form of a binary tree with IF TRUE and IF FALSE branches.

For each of the primitives, we can describe their structure. For example, a line can be small, medium or large in size. The same applies to curves and the loop. Hence, there are thirty features in total for lines, curves and the loop described below.

LINES

1. small vertical	4. small horizontal
2. medium vertical	5. medium horizontal
3. large vertical	6. large horizontal
7. small backslash	10. small slash
8. medium backslash	11. medium slash
9. large backslash	12. large slash

CURVES

13. small north	16. small south
14. medium north	17. medium south
15. large north	18. large south
19. small east	22. small west
20. medium east	23. medium west
21. large east	24. large west

LOOPS

25. small preceding	28. small after
26. medium preceding	29. medium after
27. large preceding	30. large loop after

31. None

4. Experimental Results and conclusion

The technique which has been adopted for this study combines a purely structural method (based on structural primitives such as curves, lines, etc. in a similar manner to which human beings describe

characters geometrically) and a classification scheme using Induct machine learning. The total amount of data available is 1040 samples of hand-printed characters. The system has been trained by using 15 samples per alphabet (780 training samples for 52 classes) in the learning stage to generate a decision tree. We then attempted to recognise characters from 5 unseen samples per alphabet using the constructed decision tree and the rate of the recognition achieved is 90.2%. This is a very promising result and clearly shows that Induct machine learning technique is well suited to this type of application.

Table 1 shows the recognition rate for variable sizes of training and test sets. For the first three cases, we randomly select the size of the training and the test set. The results show that as we increase the size of the training set, a larger number of rules are generated by the system. The training and test recognition rates become better with an increase in the training data as better rules are induced by the system for classification. These results compare well with our previous work using neural networks on a larger data set of Latin characters with the same features that resulted in 86% correct classification [4]. One of the key advantages however of using the machine learning approach is that the system is much simpler to implement compared to a neural network. Also, the approach is much faster and computationally cheaper for large data sets. One of the critical criticisms of the machine learning approaches in the past has been their inability to handle non-linear and noisy inputs. In this study, we convincingly demonstrate that well-designed machine learning systems can give high quality performances.

One of the advantages of using the Induct/RDR method is that not only can the performance be improved by adding further cases to the training data, but potentially rules can be added by hand for any cases that have been misclassified. This allows the system to be flexible where continuous improvement is possible. Knowledge-based systems also have the advantage of working with very high speed since the system only executes the *condition satisfied, execute rule* paradigm. A number of advantages of using rule-based systems for practical purposes apply to our Induct/RDR system. More specifically, from the point of view of using this technique for character recognition, the main advantage is that the system is simple to construct and apply, and as we mentioned before, it can be used alongside human interaction for modifying rules to generate more accurate system response. In this paper we have shown that the Induct/RDR methodology coupled with manual guidance on misclassified cases can be a very useful character recognition system. In this paper, we have only conducted a pilot study to realise the potential benefits of the described methodology. As the training data will increase with our further studies, we expect that the results will be compatible and more favourable to the results of this pilot study. We recommend that techniques such as machine learning tree-based approaches are very useful in character recognition and should be compared in larger studies with methods such as neural networks and genetic algorithms.

Size of Training Set	No. of rules generated	Training Recognition Rate	Size of Testing Set	Testing Recognition Rate
252	67	89.6%	84	83.9%
382	88	92.9%	127	86.0%
512	92	95.3%	170	89.1%
780	101	98.1%	260	90.2%

Table1: Recognition rates on training and testing character recognition data using INDUCT/RDR methodology

References

[1] B. R. Gaines and P. Compton, Induction of Ripple Down Rules, *5th Australian Joint Conf. on Artiificial Intell.*, Hobart, Australia, World Scientific, 349-354, 1992.

[2] J. Catlett, Ripple Down Rules as a mediating representation in interactive induction, *Proc. 2nd Japanese knowledge acquisition for knowledge-based system workshop*, Kobe, Japan 155-170, 1992.

[3] P. Compton and R. Janson, A philosophical basis for knowledge acquisition, *Knowledge acquisition* **2** 241-257, 1990.

[4] A. Amin and S. Singh, Optical character recognition: Neural network analysis of hand-printed characters, *Proceedings of SSPR'98*, Lecture Notes in Computer Science -Springer-Verlag, pp. 492-499.

[5] J.R Quinlan C4.5: *Programs for Machine Learning*. San Mateo, CA: Morgan Kauffman (1993).