Discrimination Between Digits and Outliers in Handwritten Documents Applied to the Extraction of Numerical Fields

Clément Chatelain, Laurent Heutte, Thierry Paquet

Discrimination Between Digits and Outliers in Handwritten Documents Applied to the Extraction of Numerical Fields

Clément Chatelain, Laurent Heutte, Thierry Paquet
Laboratoire PSI, CNRS FRE 2645,
Université de Rouen, 76800 Saint Etienne du Rouvray, FRANCE
{clement.chatelain, laurent.heutte, thierry.paquet}@univ-rouen.fr

Abstract

In this article, we propose a numerical field extraction system from unconstrained handwritten documents. The system is based on a segmentation driven by recognition stage followed by a syntactical analysis which detects the sequences that may compose a numerical field. We focus here on the design of a digit classifier embedded in the segmentation/recognition process able to discriminate digits from outliers such as words, fragment of words, noise, etc. For that, we have developed a light classifier used as prior to a standard digit classifier in order to reject "obvious outliers". Several classifiers have been compared in terms of ROC curve and processing time.

1 Introduction

During the last years, many systems have been designed to perform an automatic processing of handwritten documents. However, the well-known variability of handwriting has restricted the researches to specific and very constrained documents such as bank checks, mail address on envelopes or handwritten fields on printed forms. Nowadays, a new challenging problem is the automatic processing of unconstrained handwritten documents with free layout and cursive handwriting. Hence, we are faced with the lack of *a priori* knowledge, which forbids an integral reading of a whole page of handwriting with reasonable reliability and processing time.

Nevertheless, it is now possible to consider information extraction applications, where the *a priori* knowledge supplied to the system concerns the researched information instead of the entire document. The extraction of numerical data (file number, customer reference, phone number, ZIP code, ...) in an incoming mail document (see figure 1) is one particular example of such a realistic problem. It is a very challenging problem since the numerical fields may be situated either in the header or in the body of the text. Furthermore, numerical fields have no linguistic constraints: any digit can follow an other (see figure 2). Thus, our approach cannot be lexicon-directed as in many classical word recognition systems [5].

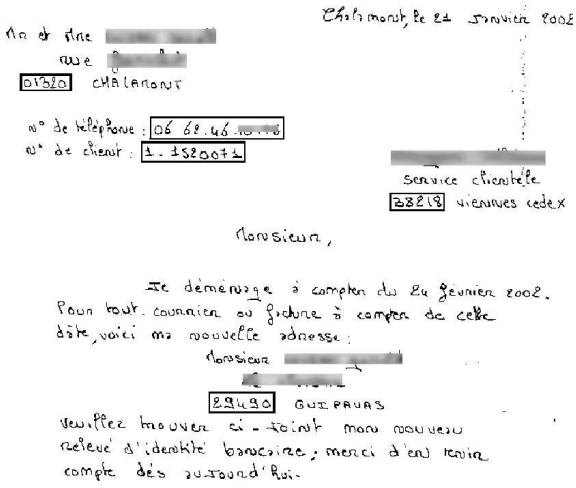

Figure 1. Incoming mail document

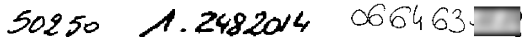

Figure 2. Examples of numerical fields.

The main idea of our approach is to exploit the known syntax of a numerical field to locate it in a text line [2]. For example, a french phone number is always made of ten digits, with optional separators between each pair of digits. Thus, the extraction of a phone number in a text line consists in the detection of a sequence of ten digits with optional separators in the whole line sequence. This is performed by a numeral component recognition stage followed by a syn-

tactical analysis of the recognition hypotheses, which filters the syntactically correct sequences with respect to a particular syntax known by the system. Thus, a crucial point of this system is the ability of a classifier to discriminate numeral patterns from the rest of the document: word, fragment of word, noise, etc. that one can call *outliers*. In this article, we propose a simple two-stage outlier rejection strategy which improves the final system performance.

This paper is organized as follows. In section 2 we present an overview of the numerical field extraction system with a brief description of each processing stage. Section 3 deals with the outlier rejection strategy embedded in the system. We present in section 4 our experimental results on a database of real handwritten incoming mail documents. Conclusion and future works are drawn in section 5.

2 Numerical field extraction

The numerical field extraction strategy relies on a syntactical analysis of the lines of text in order to filter the syntactically correct sequences with respect to a particular syntax known by the system. Hence, a recognition stage is required to distinguish numerical components (isolated and touching digits) from the rest of the document. The recognition stage must also be able to detect the separators which are important syntactical elements. This is performed thanks to a segmentation driven by recognition stage described afterwards, which provides a three-level recognition trellis with confidence values for each component, concatenated over all the line (see figure 3, where 'X' denotes a confidence value).

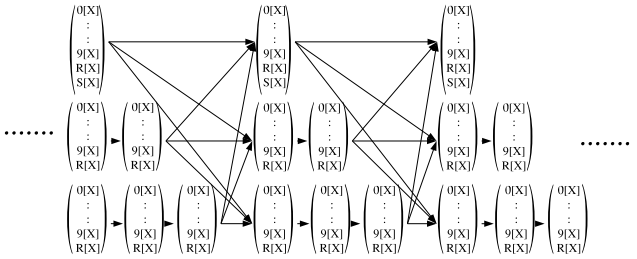

Figure 3. Line trellis obtained by concatenation of the component trellis.

Text line model: a line model is defined for each kind of numerical field, which provides the syntactical constraints of a text line that may contain a numerical field. Models are made of states that may yield 12 symbols : 10 classes of digit + separator (S) + outlier (Reject: R). The authorized transitions between states have been learnt on a handwritten document database containing numerical fields. The resulting text line models are presented in figure 4. The exploration of the trellis is performed according to the confidence values of the recognition hypotheses by dynamic programming [12] under the constraints of the model.

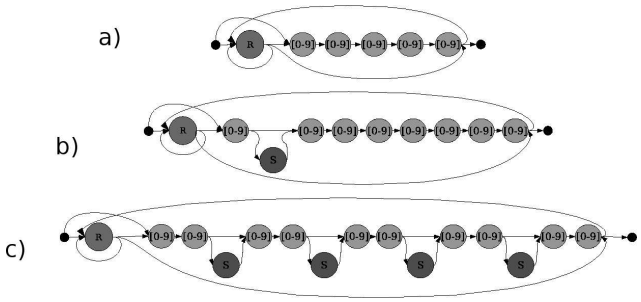

Figure 4. Text line models. a) ZIP code, b) customer code, c) phone number

Line segmentation : the connected components are extracted from the document and grouped into lines, according to a classical method [7]. The three steps for the line segmentation process are (see figure 5): a) the big components are grouped together according to a distance criterion, b) alignments which are too close are merged, c) isolated components are grouped with the nearest line. The handwritten document is thus converted into sequences of connected components.

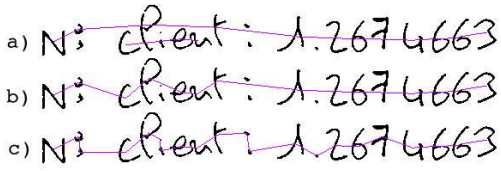

Figure 5. Line segmentation process.

Segmentation driven by recognition: the aim of this recognition stage is to detect the components which belong to a numerical field: single or touching digits, and separators. All the remaining components are outliers and must be rejected. Hence, components are successively considered as: numerical components, separators and outliers, according to three different strategies :

Digits: the single and touching digit recognition is performed thanks to a segmentation/driven recognition stage which successively considers a component as a single, double and triple digit. Figure 6 gives an example of the method for the recognition of a component as a double digit: several cutting paths are generated and are submitted to a digit classifier. The path which maximizes the confidence prod-

uct is retained (in this example, the first path). This stage is iterated for the recognition of triple digits.

Drop fall	ascending left	ascending right	descending left	descending right
cutting path				
digit classifier output	0[98] 8[82]	2[27] 8[35]	0[73] 8[36]	0[92] 8[34]
confidence product	81	09	26	32

Figure 6. Double digit recognition example

Separators: the separator recognition is performed thanks to a small classifier based on contextual features [2]. As separators are always single components, the separator recognition stage is only applied on the first level.

Reject: since most of the components are outliers, the numeral and separator recognition hypotheses must be submitted to an outlier rejection system which provides a confidence value for the reject class. This outlier rejection system is described in section 3.

Finally the outputs of the recognition stage performed on each component are concatenated over all the line to produce a 3-level recognition hypothesis trellis (see figure 3).

3 Outlier rejection strategy applied on a digit classifier

In this section, we focus on the design of an outlier rejection strategy, based on a standard 10 class digit classifier. As seen in the previous section, the digit classifier should be able to output 11 confidence values: ten for digit classes and one for the outlier class.

If the discrimination between handwritten digits is now a quite well-solved problem, the outlier rejection is still a tough problem due to the extreme variability of outlier patterns. The analysis of a database of outliers leads us to consider roughly two kinds of outliers (see figure 7): **(i) Obvious outliers** which have a very different shape from isolated digits like noise, fragment or entire words, stroke, points or dash, etc. **(ii) Ambiguous outliers** which have a similar shape with single digits: letter, group of letters or fragment of word mainly. These outliers are more difficult to distinguish from digits.

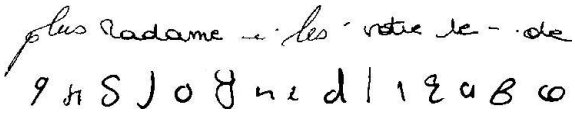

Figure 7. obvious (first line) and ambiguous (second line) outlier examples.

This observation leads us to consider the following two-stage strategy to reject outliers (see figure 8):

The first stage is used to reject obvious outliers. As it seems easy to distinguish obvious outliers from digits, we propose to design a 2-class classifier based on a restricted number of features. The aim is to reject as many outliers as possible, *while accepting all the digits*. Thus, this stage provides a binary decision (accept as a digit or reject).

The second stage aims at discriminating ambiguous outliers from digits among the patterns accepted by the first stage. As it seems to be a tough problem, we propose a soft decision, based on the analysis of the confidences values of a 10-class numeral classifier.

We now detail these two stages.

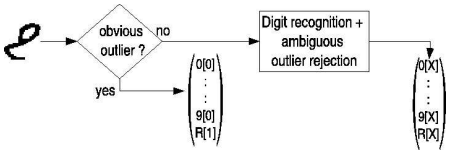

Figure 8. A two-stage outlier rejection strategy for handwritten digit classifier.

3.1 Obvious outlier rejection

This first stage acts as a filter which decides whether a pattern is an obvious outlier or not. The aim is to design a light rejection classifier that rejects as many outliers as possible, while accepting all the digits.

For that, we have designed a 8-feature set. The 8 features are: (f_1) height/width ratio, (f_2) black pixel density, (f_3) number of water reservoirs (metaphor to illustrate a valley in a component, see [10] for more details), (f_{4-6}) number of intersections with two horizontal and one vertical straight lines, (f_7) number of end points, (f_8) number of holes.

To discriminate the patterns, a two-class classifier has to be designed. Several state-of-the-art classifiers have been tested, selected for their performance in classification or/and time processing :

- A rule pruning binary classifier: "RIPPERk", which performs efficiently on noisy datasets and provides a very fast decision [3].
- Two neural networks: a discriminant MultiLayer Perceptron (MLP) and a model-based Radial Basis Function (RBF) [1]. Neural networks are known to provide a very good trade-off between performance and processing time.

- A Support Vector Machine (SVM) classifier [15], known to be very accurate for two-class discrimination problems, but slower than other classifiers.

An important particularity of this classification stage is that a false rejection (FR) leads to more serious consequences than a false acceptance (FA). Indeed, a FR cannot be recovered, whereas a FA can be rejected at higher level, either by the second outlier rejection stage, or during the syntactical analysis. Thus, a trade-off must be found during the learning stage by means of a cost value.

Hence, we decided for each classifier to first find the intrinsic parameters (number of neurons in the hidden layer for the MLP, number of basis functions for the RBF, γ and C for the SVM) and then to tune experimentally the cost parameter. Concerning the intrinsic parameters, the lack of analytic method to find the optimal values for parameters has lead us to an experimental tuning on a test database. Concerning the cost parameter, we will show its influence on the global system performance (see section 4).

All the classifiers have been trained on a database of 7,500 patterns (5,000 outliers, 2,500 digits) with a cost value penalizing the false rejection with respect to the false acceptance. The outlier rejection ability of the classifiers is evaluated by means of the Receiver-Operating Characteristic (ROC) curve which is a graphical representation of the trade-off between the false negative (true digit rejection) and false positive (outlier acceptation) rates, for different cost values. The ROC curve of each classifier is shown on figure 9.

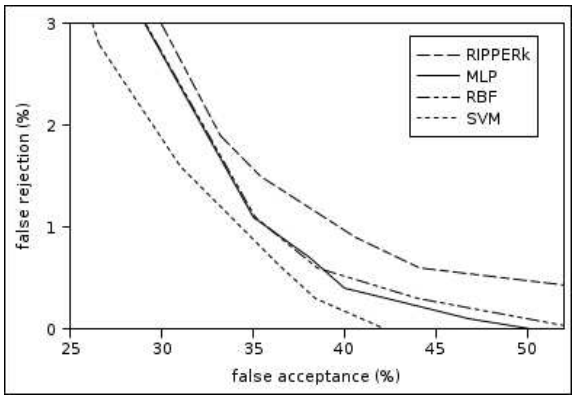

Figure 9. ROC curve for obvious outlier rejection.

The results show that the best ROC curve is obtained with the SVM classifier. The SVM, MLP and RBF reach a zero FR rate while rejecting respectively 57,8%, 49,8% and 47% of the outliers, whereas RIPPERk cannot reach a zero FR rate before 100% FA. Thus, SVM gives the best ROC curve, which is not really surprising. However, note that the decision time of the SVM depends of the number of support vectors (SV) kept by the learning stage. In our case, the average number of SV is about 2500. Hence, one can argue that the SVM cannot be used on all the patterns extracted from a whole page of handwriting. Table 1 shows that even if the SVM is slower than other classifiers, the reduced feature set implies a reasonnable processing time that does not exceed 6 seconds to process 7500 patterns (much more than the number of connected components present in one document). Thus, we retain the SVM classifier to perform the obvious outlier rejection.

Classifier	RIPPERk	MLP	RBF	SVM
processing time	$<$1s	$<$2s	$<$2s	6s

Table 1. Processing time (in seconds) to classify 7500 patterns

3.2 Ambiguous outlier rejection based on a digit classifier

The second stage of our approach aims at discriminating digits from ambiguous outliers. As the remaining outliers (those which have been accepted by the first stage) have a similar shape with the digits, this task is coupled with the digit recognition process. Hence, we need a digit classifier with some outlier rejection ability, i.e. able to output eleven confidence values: 10 digits + Reject. This tough problem requires obviously an important number of features and a large database of examples.

Several techniques have already been designed for the rejection of outliers through a recognition stage: training a classifier with outlier data [8], modeling the target classes and perform a distance rejection strategy [9], use of one class classifiers [14], reject outliers with respect to the outputs of a classical classifier as proposed in [11]. We have chosen this latter solution, applied on a MultiLayer Perceptron (MLP), for the following reasons:

- Even if the obvious outlier rejection stage can generally reject more than half of the patterns in a whole page of handwriting, the remaining patterns, that have to be classified in high dimensional space, still represent an important part of the document. This constraint prohibits therefore multiclass SVM and one-class SVM. Oppositely, MLPs well suit this condition because they have an extremely fast decision processing.
- If the use of model-based classifiers (RBF, one class classifier, etc.) allows a distance-based rejection strategy, these classifiers suffer from a poor discrimination

ability, and the modelisation of classes in high dimensional spaces is still a difficult problem. Oppositely, MLPs have very good discrimination performance and are well adapted to high dimensional spaces [1, 8].

We have thus designed a combination of two MLPs, trained on 130,000 digits, with a 117-structural/statistical feature set developped in our previous work [4], and a 128-feature set extracted from the chaincode [6]. A product rule combination is performed between the two MLPs. On a test database containing 60,000 digits, the classifier combination provides a recognition rate of 98.44%, 99.48% and 99.75% in TOP 1, TOP2 and TOP3 respectively.

From this point, the rejection rule is the following : a confidence value for the reject class is estimated with a Look Up Table (LUT) according to the confidence value of the first proposition of the digit recognizer. The LUT has been generated by considering the behaviour of the digit classifier on a database of 2,300 digits and 4,000 outliers. Statistics on the confidence value of the first proposition provides the LUT shown in figure 10.

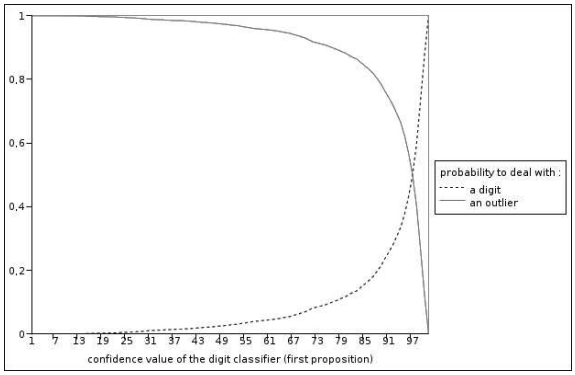

Figure 10. Look up table for the reject confidence value.

Thus, the 10-digits recognition hypothese are submitted to the LUT which estimates a confidence value for the reject class. The softmax function is then applied on the 11 confidence values to output *a posteriori* probability estimates.

3.3 Global outlier rejection performance

To evaluate the outlier rejection performance of our system, a database containing digits and outliers extracted from real documents is submitted to the single (digit classifier+LUT only) and two-stage (digit classifier+LUT preceded by the SVM described above) outlier rejection system. We show on figure 11 the two resulting ROC curves.

The trade-off between false rejection and false acceptance is clearly better with the use of the SVM classifier as prior filter to the digit classifier. The break-even point (BEP) of our two-stage approach is 9%. The BEP is the point where $FA = FR$; it means that the system is able to reject 91% of the outliers, while accepting 91% of the digits.

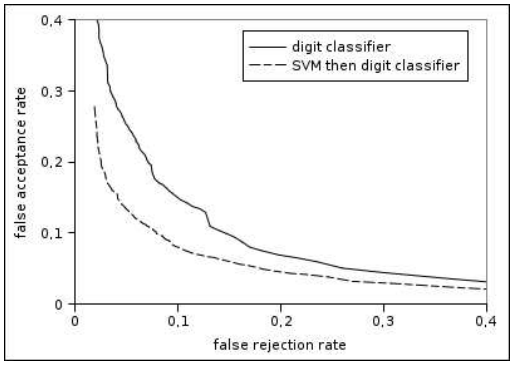

Figure 11. ROC curve for two rejection strategies.

4 Numerical field extraction results

This section presents the results of the numerical field recognition system. We have evaluated our approach on a database of 293 real incoming mail documents provided by a french firm. Note that as these documents contain private information (name, address, phone number, ...), the database can unfortunately not be diffused for result comparison.

As we propose an information extraction system, the performance criterion is the trade-off between recall and precision rates. The recall and precision rates are defined as:

$$recall = \frac{\text{nb of fields well recognized}}{\text{nb of fields to extract}}$$

$$precision = \frac{\text{nb of fields well recognized}}{\text{nb of fields proposed by the system}}$$

A field is considered as "well recognized" if and only if it has been perfectly localized and recognized (i.e. all and only the components that belong to the field have been recognized as the true digit or a separator).

The syntactical analysis is performed thanks to the forward algorithm, which provides the n best alignment paths. A field well detected in *TOP n* means that the right recognition hypothesis for a field stands in the n best propositions of the syntactical analyser.

Figure 12 shows the recall-precision trade-off of the system and the influence of the cost parameter described in section 3.1. Each value of the cost parameter provides a curve

made of 5 points that represent TOP1 to TOP5 performance from upper left corner to lower right corner respectively. The tuning of the cost parameter is made as follows: starting from the value denoted by C_{zeroFR} (that provides a zero-FR as shown in figure 9), we then tune the cost parameter around C_{zeroFR}. Note that when the cost parameter tends to ∞, all the patterns are accepted; this is equivalent to an absence of the obvious outlier rejection stage.

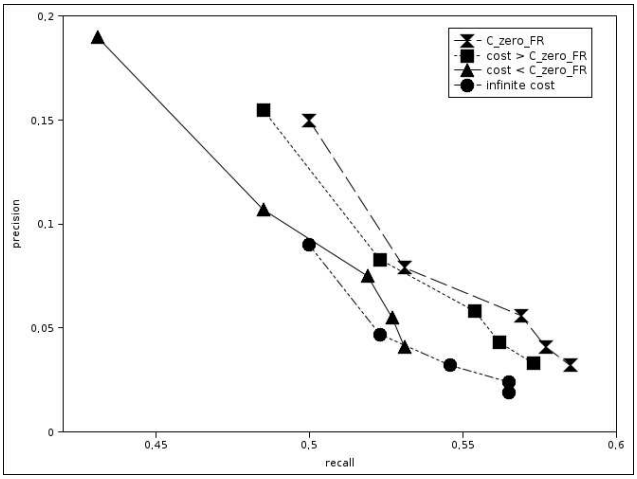

Figure 12. Recall-precision trade-off for different value of the cost parameter.

One can see on figure 12 that the cost parameter value which provides the best recall-precision trade-off is roughly C_{zeroFR}. The interpretation of this result is the following: the cost parameter values above C_{zeroFR} still provide a zero FR rate but with a below obvious rejection rate, whereas the cost parameter values below C_{zeroFR} improve the obvious outlier rejection rate but rejects too many digits that may belong to a numerical field, forbidding their correct localisation.

As a conclusion, the cost parameter is a critical value to improve the recall-precision trade-off of the system.

5 Conclusion and future works

Thanks to a simple feature vector and a SVM classifier used as prior to a digit classifier, we have improved the outlier rejection ability of the digit recognizer. We have shown the influence of such an improvement when the classifier is embedded in a segmentation driven by recognition process. Our future works will focus on the SVM learning : we plan to use an Area Under ROC Curve [13] criterion instead of the recognition rate criterion. This could improve the outlier rejection capacity of the system. Another perspective is the combination of the system described in this paper with an alternative approach developped in our previous work [2] in order to improve the global performance of the system.

References

[1] C. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[2] C. Chatelain, L. Heutte, and T. Paquet. A syntax-directed method for numerical field extraction using classifier combination. *9th International Workshop on Frontiers in Handwriting Recognition, Tokyo, Japan*, pages 93–98, 2004.

[3] W. Cohen. Fast effective rule induction. In *Machine Learning*, pages 115–123, 1995.

[4] L. Heutte, T. Paquet, J. Moreau, Y. Lecourtier, and C. Olivier. A structural/statistical feature based vector for handwritten character recognition. *Pattern Recognition Letters*, 19:629–641, 1998.

[5] G. Kim and V. Govindaraju. A lexicon driven approach to handwritten word recognition for real-time applications. *IEEE Trans. on PAMI*, 19(4):366–378, 1997.

[6] F. Kimura, S. Tsuruoka, Y. Miyake, and M. Shridhar. A lexicon directed algorithm for recognition of unconstrained handwritten words. *IEICE Trans. on Information & Syst.*, E77-D(7):785–793, 1994.

[7] L. Likforman-Sulem and C. Faure. Une methode de resolution des conflits d'alignements pour la segmentation des documents manuscrits. *Traitement du Signal*, 12:541–549, 1995.

[8] J. Liu and P. Gader. Neural networks with enhanced outlier rejection ability for off-line handwritten word recognition pattern recognition. *Pattern Recognition*, 35:2061–2071, 2002.

[9] J. Milgram, R. Sabourin, and M. Cheriet. An hybrid classification system which combines model-based and discriminative approaches. *ICPR'04, Cambridge, UK*, 1:155–162, 2004.

[10] U. Pal, A. Belaïd, and C. Choisy. Touching numeral segmentation using water reservoir concept. *Pattern Recognition Letters*, 24:261–272, 2003.

[11] J. Pitrelli and M. Perrone. Confidence-scoring post-processing for off-line handwritten-character recognition verification. *ICDAR'03*, 1:278–282, 2003.

[12] L. R. Rabiner. A tutorial on hidden markov models and selected apllications in speech recognition. In *Readings in Speech Recognition*, pages 267–296. Kaufmann, 1990.

[13] A. Rakotomamonjy. Optimizing auc with support vector machine. *European Conference on Artificial Intelligence Workshop on ROC Curve and AI*, pages 469–478, 2004.

[14] D. Tax and R. P. W. Duin. Combining one-class classifiers. In *MCS '01*, pages 299–308, 2001.

[15] V. Vapnik. The nature of statistical learning theory. *Springer*, 1995.