Self-supervised adaptation for on-line text recognition

Loïc Oudot, Lionel Prevost, Alvaro Moises
Laboratoire des Instruments et Systèmes d'Ile de France, BP 252
4 Place Jussieu, 75252 Paris Cedex 5
loic.oudot@lis.jussieu.fr

Abstract

We recently developed a handwritten text recognizer for on-line text written on a touch-terminal. This system is based on the activation-verification cognitive model. It is composed of three experts dedicated respectively to signal segmentation in symbols, symbol classification and lexical analysis of the classification results. The baseline system is writer-independent. We present in this paper several strategies of self-supervised writer-adaptation that we compare to the supervised adaptation scheme. The best strategy called "prototype dynamic management" modify the recognizer parameters allowing to get results close to the supervised methods. Results, are presented on a 90 texts (5 400 words) database written by 38 different writers.

keywords: handwriting recognition, self-supervised adaptation, model-based classifier.

1. Introduction

Recently, handheld devices like PDAs, mobiles phones or e-books have became very popular. In opposition to classical personal computers, they are very small, keyboard-less and mouse-less. Therefore, electronic pen is very attractive as pointing and handwriting device. The first application belongs to man-machine interface and the second to handwriting recognition. Here, we focus on the second one.

For such an application, recognition rates should be very high otherwise it should discourage all the possible users. The major problem is the vast variation in personal writing style. This problem can be solved by updating the parameters of a writer-independent recognizer to build a writer dependent recognizer by adapting the system to its user's own writing style. However, it should not be forgotten that the use of a pen as input modality has to be user friendly. So, the training step must be as shorter as possible.

Traditional technics require the writer intervention for this training (the so-called supervised adaptation). We propose in this paper several self-supervised adaptation scheme that we compare to the already existing techniques like supervised adaptation or enrollment.

The paper is organized as follows. In section 2, we present a review of the various techniques of self-supervised adaptation. In section 3, we describe the writer-independents baseline system. In section 4 we describe the different adaptation strategies. Finally, conclusions and prospects are given in section 5.

2. Literature review

The idea of writer adaptation was revealed by the perceptive psychology works. We can notice that in the case of a hardly readable writer, it is easier to read a word if we have already read other words written by the same person. This phenomenon is called the graphemic priming effect. Thus, we learn the user writing characteristics from the words we can read, and then, we use this new knowledge to read the remaining words.

In the literature, we consider two adaptation strategies: systems where the adaptation takes place once first before use (called off-line or batch) and systems with continuous adaptation (on-line).

The majority of the systems [4, 10, 1, 2] using an off-line adaptation scheme need a labelled database of the writer. These examples are use to make a supervised training of the system. Thus, the system learns the characteristics of this particular writer before being used.

On the other hand, the following systems evolve continuously during use.

The on-line handwriting recognition and adaptation system of [8] uses a supervised incremental adaptation strategy. The baseline system uses a single MLP with 72 outputs (62 letters and 10 punctuation marks). An adaptation module, at the output of the MLP modifies its output vector. This adaptation module is a RBF (*Radial Basis Function*) network.

Proceedings of the 9th Int'l Workshop on Frontiers in Handwriting Recognition (IWFHR-9 2004)
0-7695-2187-8/04 $20.00 © 2004 IEEE

The user informs the system of the classification error, giving the letter label, and the RBF is re-trained (modification of the existing kernels or addition of a new one).

Two other systems use a TDNN (*Time Delay Neural Network*) as classifier instead of the MLP. This TDNN is trained on an omni-writer database and the output layer of this network is replaced either by a k-pvv classifier in [3] or by a discriminating classifier in [5]. During the adaptation step, the TDNN is fixed and the output classifier is trained, in order to learn mis-recognized characters.

The system of [11] is very close to our system but is dedicated to isolated alphanumeric character recognition. The k-ppv classifier uses the Dynamic Time Warping algorithm to compare the unknown characters to a prototype database. The writer adaptation consists in adding the unrecognized characters in this data-base. Moreover, useless prototypes can be removed from the database to avoid an excessive growth of this latter.

3. Writer independent baseline system

For the experiments, we collected a large text database written by 38 different writers. Each writer wrote an average of 150 words for a total of 5 400 words and 26 000 letters. All the texts were labelled by a human expert. We present here iterative adaptation strategies: the performances of the system improve continuously with the amount of data. Thus, we will present the evolution of the recognition rate on three ranges corresponding respectively to 50, 100 and 150 words used for the adaptation.

The k-ppv characters classifier uses an omni-writer strokes prototype data set. This base was created by using an automatic clustering algorithm [9] starting from the 60 000 samples of UNIPEN database (corpus Train-R01/V07). After clustering, this prototype database contains some 3 000 prototypes for the 62 classes (26 uppercase letters, 26 lower-case letters and 10 digits). Each sample represent a different character allograph.

We use for lexical analysis a dictionary containing the 8 000 most frequent words of the French language. Our system is able to manipulate very large lexicon (over 200 000 words) but for these experiments, this large lexicon is right sized. The complete analysis speed is about 6 words per second (P4 1,8GHz Matlab) and a small amount of memory is required (about 500Ko including the system program, the lexicon and the database).

The baseline system is presented in figure 1. It is based on the activation-verification cognitive model of Paap [7]. It consist of a set of encoding experts [6] that extracts geometrical and morphological informations of the input data. All these experts provide probabilistic informations at the stroke level. This local and global informations are merged in order to activate a list of hypothetical word in the lexicon. The meaningful of each hypothesis in the list is then evaluated by a probabilistic engine which finally, give the best retranscription of the input data.

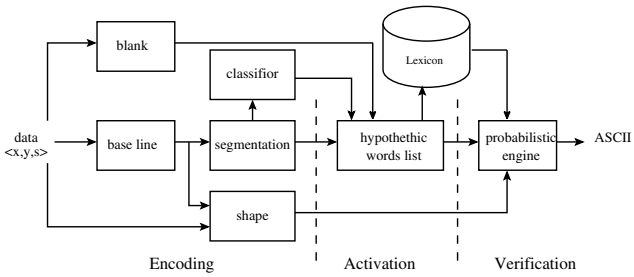

Figure 1. Baseline system.

4. Writer adaptation strategies

The baseline system recognition is writer-independent [6]. Its prototype data-set (the so-called WI database) should cover all the writing styles. Experimental results show that it covers at least the most common writing styles. We also mark that storing characters samples taken from the text database in the prototypes database (multi-writer system) improves greatly the recognition rate. At least, there are two situations which reduce the recognition rate:

- Missing grapheme: the grapheme is missing in the prototype database and it must be stored (added) in this set.
- Confusing grapheme: for a given writer, the prototype is confusing or erroneous and it must be removed from the prototype database.

Prototype-based systems can be adapted very easily and quickly to new writing styles, just by storing new character samples in the writer dependent (WD) database and inactivating existing prototypes. On the contrary, neural or markovian classifiers need more samples to re-estimate their parameters. The system specialization on a given user – by registration of his personal features – makes it writer-dependent and increases its accuracy. The comparison of classification hypothesis with either the labelled data (supervised adaptation) or the lexical hypothesis (self-supervised adaptation) detects classification errors. The mis-recognized characters can be stored in the writer-dependent (WD) database, using the lexical hypothesis as a label.

4.1. Supervised adaptation

Before comparing the accuracy of self-supervised adaptation strategies, we start by studying supervised technics.

We use the labels of the text database and the real text segmentation to carry out supervised adaptation. Characters of the text are classified one after the other. The classification hypothesis (the best answers, top_1, of the characters classifier) are compared with the labels. If they do not match, the mis-recognized character is stored in the user personal database (figure 2). We consider two approaches: the *text* approach where the characters are added at the end of the analysis of the text and the *line* approach where the characters are added at the end of the analysis of each line. The results (table 1) show the improvement of the recognition rate due to the writer adaptation of the handwriting recognition system when the segmentation of the text in words and letters is known.

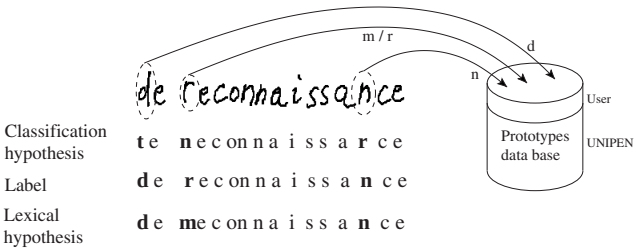

Figure 2. Supervised ans Self-supervised addition of prototypes in the user database.

	WER			WDDBS
Words	50	100	150	
Baseline system	25 %			100 %
Text appr.: min	0 %	0 %	0 %	+3 %
mean	1.3 %	1.1 %	0.6 %	+6 %
max	10 %	5.1 %	4.5 %	+9 %
Line Appr.: min	0 %	0 %	0 %	+2 %
mean	**1.1 %**	**0.7 %**	**0.4 %**	**+4 %**
max	6.2 %	5.2 %	3.7 %	+8 %

Table 1. Supervised adaptation: Word error rate WER and WD database size WDDBS (known segmentation).

The *line* approach allows a faster improvement of the recognition rate and add less prototypes to the user database than the *text* approach. When we add characters after a full text analysis, we can add several similar prototypes (and the average number of added prototypes increases). On the other hand, the *line* approach, adds the first prototype of a mis-recognized character. Thanks to this new sample, the following similar characters are recognized, so they do not need to be stored in the prototypes database. In the rest of this article, we use the *line* approach. The number of added prototypes is smaller. From a perceptive point of view, the prototype addition imitates – at the letter level – the priming repetition effect noticed at the word level: the initial presentation of a word reduces the amount of information necessary to its future identification and makes this identification faster. Nevertheless, activating WD prototypes is not sufficient to perform "perfect" classification, even with a great amount of labelled data. It seems necessary to inactivate – or even delete – some WI prototypes.

4.2. Self-supervised adaptation

Self-supervised adaptation is completely hidden to the writer who is never solicited by the system. Now, classifier hypothesis and lexical hypothesis are compared to find which prototypes must be stored in the user database.

4.2.1 Systematical activation (SA)

In the systematical activation strategy, we considers that the lexical analyzer is perfect. Therefore, when an error (difference between the classification hypothesis and the lexical hypothesis) occurs, the corresponding character is stored in the user personal database. Due to the lexical analyzer errors cumulated with the segmentation errors, some prototypes are stored in bad classes (figure 2). These errors introduce many new classifications errors. The performances of the recognition system decrease (table 2) but remain higher than those of the baseline system (before adaptation).

4.2.2 Conditional activation (CA)

The systematical activation strategy is not really accurate, it seems necessary to study the behavior of the lexical analyzer to store only useful prototypes. Two conclusions are essential at this time of the study:

- The reliability of the lexical analyzer increases with the number of characters composing the word.
- The reliability of the lexical analyzer decreases when the number of classification errors (characters of the mis-recognized word) increases.

So, for a given word, we define a *criterion of lexical reliability (CLR)* to estimate the lexical analyzer probability of error for this word. This criterion is based on two parameters: the length of the word ($nbChar$) and the number of mis-classification (defined as the total number of differences between the classifier hypothesis and the lexical hypothesis: $nbDif$). The conditional activation strategy is described in the following. If, for a given word, the relation (1) is true, then the mis-classified characters of this word are

added to the user database. We determined the K parameter by an exhaustive search. The best result is obtained with $K = 4$ that shows that the longer is the word, the better is the lexical correction.

$$nbDif \leq \mathrm{E}(nbChar/K) \quad K \in \mathbb{N}^{+} \tag{1}$$

	WER			WDDBS
Words	50	100	150	
Baseline system	25 %			100 %
SA strategy: min	0 %	1.9 %	2 %	+2 %
mean	25 %	23 %	23 %	+6 %
max	53 %	73 %	51 %	+14 %
CA strategy: min	0 %	0 %	2 %	+1 %
mean	**22 %**	**20 %**	**17 %**	**+2 %**
max	71 %	58 %	43 %	+3 %

Table 2. Systematic and conditional activation: Word error rate WER and WD database size WDDBS.

The CA strategy is more accurate than the SA strategy as it reduces considerably the false additions of prototypes (see the small growth of the user database). Moreover, the error rate decreases continuously over the time. After 150 words of adaptation, the error rate decrease is about 32 %. However, this method is not yet completely satisfactory: because of the *CLR*, the word with less than 4 characters are never used for the adaptation. And these short words represent 45 % of the text words.

4.2.3 Dynamic management (DM)

This method have two goals. As seen previously, using lexical hypothesis as a reference may add confusing or erroneous prototypes. Dynamic management is used to recover from those prototypes which contribute more often to incorrect than correct classifications. Inactivation methods are also used to prune the prototype set and speed-up the classification [11]. Each prototype (of the WI database as of the WD database) has an initial adequacy ($Q_0 = 1000$). This adequacy is modified during the recognition of the text according to the usefulness of the prototype in the classification process, by comparing the classification hypothesis and the lexical hypothesis. Let us consider the prototype i of the class j, three parameters are necessary for the dynamic management:

- **C** : Reward (+) the prototype i when it perform good classification (classification and lexical hypotheses are the same).
- **I** : Penalize (-) the prototype i when it perform misclassification (classification and lexical hypotheses are different).
- **N** : Penalize (-) for all the useless prototypes of the class j.

$$Q_j^i(n+1) = Q_j^i(n) + [C(t) - I(t) - N(t)]/F_j \tag{2}$$

Where F_j is the frequency of the class j in the French language. These three parameters are mutually exclusive *i.e.* on each occurrence, only one parameter is activated. When $Q_j^i = 0$, the prototype is removed from the database.

After an exhaustive search of the parameters (C, I, N) the optimal triplet for a use with the conditional activation strategy is (30, 200, 8). The dynamic management is very efficient as it greatly reduces the size of the database while preserving the recognition rate of the conditional activation strategy (table 3).

	WER	WDDBS
Baseline system	25 %	100 %
DM strategy	**17 %**	**-40 %**

Table 3. Dynamic management: Word error rate WER and WD database size WDDBS.

Now, let us focus on the evolution of the adequacy for some prototypes (figure 3). For some writers, the WI prototypes are sufficient. For the class 'a', 2 prototypes are used and thus the adequacy of the 45 others decrease. For the class 's', 4 prototypes are useful (the writer has probably an unstable writing, see figure 4) and the 36 others are inactivated. For another writer (class 's' and 'e'), WD prototypes (in bold) are necessary. At the beginning, a WI prototype is used and after some occurrences, a WD prototype is added (the user gets familiar with the handheld device and the pen). After 150 words of adaptation, the size of the prototype database was reduced by 40 %. A linear regression can reasonably bring this reduction to more than 90 % of the initial size.

5. Conclusions & Future works

We have presented several self-supervised adaptation strategies. Train the model-based classifier to be writer-independent was very easy. Thanks to his structure, it can learn new writings styles, by activating new prototypes and inactivating erroneous ones. The prototype dynamic management adaptation scheme, increases the recognition rate (from 75 % to 83 %) and increases the classification

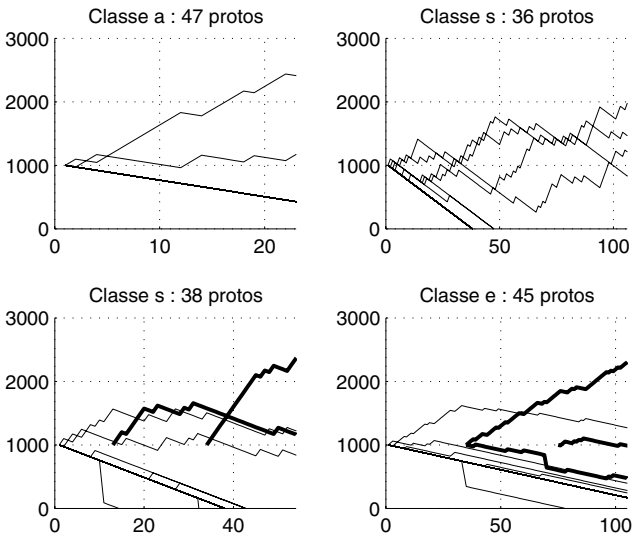

Figure 3. Prototypes adequacy evolution vs. occurrence.

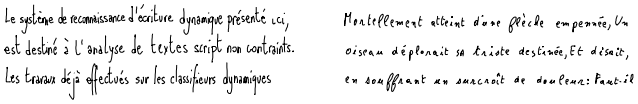

Figure 4. Best recognition rate writer (99 %) and worst writer (70 %).

speed close to twice. We automatically transform a writer-independent database into a writer-dependent database of very great quality and compactness.

It would be interesting to evaluate a semi-supervised strategy where the user is solicited only in the ambiguous cases. We have also to adapt the parameters of the segmentation expert which actually is the biggest error source.

References

[1] A. Brakensiek, A. Kosmala, and G. Rigoll. Comparing adaptation techniques for on-line handwriting recognition. *ICDAR*, 1:486–490, 2001.

[2] S. D. Connel and A. K. Jain. Writer adaptation of online handwriting models. *IEEE Transaction PAMI*, 24(3):329–346, 2002.

[3] I. Guyon, D. Henderson, P. Albrecht, Y. L. Cun, and J. Denker. *Writer independent and writer adaptative neural network for on-line character recognition*. S. Impedovo, From Pixels to Featuress III, Elsevier, 1992.

[4] H. Li. *Traitement de la variabilité et développement de systèmes robustes pour la reconnaissance de l'écriture manuscrite en-ligne*. PhD thesis, UPMC Paris 6, 2002.

[5] N. Matić, I. Guyon, J. Denker, and V. Vapnik. Writer-adaptation for on-line handwritten character recognition. *ICDAR*, 1:187–191, 1993.

[6] L. Oudot, L. Prevost, and A. Moises. An activation-verification model for on-line texts recognition. *IWFHR*, Submited, 2004.

[7] K. Paap, S. L. Newsome, J. E. McDonald, and R. W. Schvaneveldt. An activation-verification model for letter and word recognition: The word superiority effect. *Psychological Review*, 89:573–594, 1982.

[8] J. C. Platt and N. P. Matić. A constructive RBF network for writer adaptation. *Advances in Neural Information Processing Systems, 9*, 1:765–771, 1997.

[9] L. Prevost and M. Milgram. Modelizing character allographs in omni-scriptor frame: a new non-supervised algorithm. *Pattern Recognition Letters*, 21(4):295–302, 2000.

[10] L. Schomaker, E. H. Helsper, H.-L. Teulings, and G. H. Abbink. Adaptive recongnition of online, cursive handwriting. *6th ICOHD*, 1:19–21, 1993.

[11] V. Vuori, J. Laaksonen, and J. Kangas. Influence of erroneous learning samples on adaptation in on-line handwriting recognition. *Pattern Recogntion*, 35(4):915–926, 2002.