Pattern Recognition Letters 51 (2015) 86–93

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

# Window repositioning for printed Arabic recognition ☆

Ihab Khoury, Adrià Giménez, Alfons Juan*, Jesús Andrés-Ferrer

*Universitat Politècnica de València, Camí de Vera s/n, 46022 València, Spain*

ARTICLE INFO

*Article history:*
Received 23 September 2013
Available online 16 September 2014

*Keywords:*
Bernoulli HMMs
Printed Arabic recognition
Sliding window
Repositioning

ABSTRACT

Bernoulli HMMs are conventional HMMs in which the emission probabilities are modeled with Bernoulli mixtures. They have recently been applied, with good results, in off-line text recognition in many languages, in particular, Arabic. A key idea that has proven to be very effective in this application of Bernoulli HMMs is the use of a sliding window of adequate width for feature extraction. This idea has allowed us to obtain very competitive results in the recognition of both Arabic handwriting and printed text. Indeed, a system based on it ranked first at the ICDAR 2011 Arabic recognition competition on the Arabic Printed Text Image (APTI) database. More recently, this idea has been refined by using *repositioning* techniques for extracted windows, leading to further improvements in Arabic handwriting recognition. In the case of printed text, this refinement led to an improved system which ranked second at the ICDAR 2013 second competition on APTI, only at a marginal distance from the best system. In this work, we describe the development of this improved system. Following evaluation protocols similar to those of the competitions on APTI, exhaustive experiments are detailed from which state-of-the-art results are obtained.



## 1. Introduction

Hidden Markov Models (HMMs) are now widely used for off-line text recognition in many languages, in particular, languages with Arabic script [1–5]. Following the conventional approach in speech recognition [6], HMMs at global (line or word) level are built from shared, *embedded*, HMMs at character (subword) level, which are usually simple in terms of number of states and topology. In the common case of real-valued feature vectors, state-conditional probability (density) functions are modeled as Gaussian mixtures since, as with finite mixture models in general, their complexity can be easily adjusted to the available training data by simply varying the number of components.

After decades of research in speech recognition, the use of certain real-valued speech features and embedded Gaussian (mixture) HMMs is a de-facto standard [6]. However, in the case of text recognition there is no such standard. In fact, very different sets of features are in use today. In [7] we proposed to by-pass feature extraction and directly feed columns of raw, binary pixels into *embedded Bernoulli (mixture) HMMs (BHMMs),* that is, embedded HMMs in which the emission probabilities are modeled with Bernoulli mixtures. The basic idea is to ensure that no discriminative information is filtered out during feature extraction, which in some sense is integrated into the recognition model. In [8], we improved our basic approach by using a sliding window of adequate width to better capture image context at each horizontal position of the text image. This improvement, to which we refer as *windowed BHMMs,* achieved very competitive results on the well-known IfN/ENIT database of Arabic town names [9]. More recently, very good results on the Arabic Printed Text Image (APTI) database were also achieved using the same approach, which ranked first in the ICDAR 2011 Arabic recognition competition for printed Arabic text [10].

Although windowed BHMMs achieved good results on IfN/ENIT and APTI, it was clear to us that text distortions are more difficult to model with wide windows than with narrow (e.g. one-column) windows. In order to circumvent this difficulty, we have considered new, adaptive window sampling techniques, as opposed to the conventional, direct strategy in which the sampling window center is applied at a constant height of the text image and moved horizontally one pixel at a time. More precisely, these adaptive techniques can be seen as an application of the direct strategy followed by a *repositioning* step by which the sampling window is repositioned to align its center to the center of gravity of the sampled image. This repositioning step can be done horizontally, vertically or in both directions. Although vertical repositioning is expected to have more influence on recognition results than horizontal repositioning, we have studied both separately and in conjunction, so as to confirm this expectation.

In [11], the repositioning techniques described above are introduced and extensively tested on different databases for off-line

☆ This paper has been recommended for acceptance by Prof. L. Heutte.

* Corresponding author: Tel.: +34 963877007/73516.
*E-mail addresses:* ialkhoury@dsic.upv.es (I. Khoury), agimenez@dsic.upv.es (A. Giménez), ajuan@dsic.upv.es (A. Juan), jandres@dsic.upv.es (J. Andrés-Ferrer).

http://dx.doi.org/10.1016/j.patrec.2014.08.009
0167-8655/

handwriting recognition. As expected, vertical repositioning provides excellent results, not only on IfN/ENIT, but also on other well-known databases such as IAM words and RIMES. In the case of printed text, the use of repositioning techniques has allowed us to significantly improve our system at the ICDAR 2011 first competition on APTI. Indeed, our improved system obtained much better results at the ICDAR 2013 second competition on APTI, in which it ranked second at a marginal distance from the first [12]. In this work, we describe the development of this improved system. Following evaluation protocols similar to those of the competitions on APTI, exhaustive experiments are described from which state-of-the-art results are obtained.

In what follows, we first review BHMMs (Section 2). Then, we describe the approach through which we are achieving the best results: windowed BHMMs with repositioning (Section 3) and its use for printed Arabic recognition by application of the Bayes decision rule (Section 4). In Section 5, we provide the results of a complete series of experiments on APTI as well as a comparison with results from other authors on this database. Finally, concluding remarks are given in Section 6.

## 2. Bernoulli HMMs

Let $O = (\mathbf{o}_1, \ldots, \mathbf{o}_T)$ be a sequence of feature vectors. An HMM is a probability (density) function of the form:

$$P(O \mid \boldsymbol{\Theta}) = \sum_{q_1,\ldots,q_T} \prod_{t=0}^{T} a_{q_t q_{t+1}} \prod_{t=1}^{T} b_{q_t}(\mathbf{o}_t), \tag{1}$$

where the sum is over all possible *paths* (state sequences) $q_0, \ldots, q_{T+1}$, such that $q_0 = I$ (special *initial* or *start* state), $q_{T+1} = F$ (special *final* or *stop* state), and $q_1, \ldots, q_T \in \{1, \ldots, M\}$, being $M$ the number of regular (non-special) states of the HMM. On the other hand, for any regular states $i$ and $j$, $a_{ij}$ denotes the *transition* probability from $i$ to $j$, while $b_j$ is the *observation* probability (density) function at $j$.

A Bernoulli (mixture) HMM (BHMM) is an HMM in which the probability of observing a binary feature vector $\mathbf{o}_t$, when $q_t = j$, follows a Bernoulli mixture distribution for the state $j$

$$b_j(\mathbf{o}_t) = \sum_{k=1}^{K} \pi_{jk} \prod_{d=1}^{D} p_{jkd}^{o_{td}} (1 - p_{jkd})^{1-o_{td}}, \tag{2}$$

where $o_{td}$ is the $d$th bit of $\mathbf{o}_t$, $\pi_{jk}$ is the prior of the $k$th mixture component in state $j$, and $p_{jkd}$ is the probability that this component assigns to $o_{td}$ to be 1.

As discussed in the Introduction, BHMMs at global (line or word) level are built from shared, embedded BHMMs at character level. More precisely, let $C$ be the number of different characters (symbols) from which global BHMMs are built, and assume that each character $c$ is modeled with a different BHMM of parameter vector $\boldsymbol{\Theta}_c$. Let $\boldsymbol{\Theta} = \{\boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_C\}$, and let $O = (\mathbf{o}_1, \ldots, \mathbf{o}_T)$ be a sequence of feature vectors generated from a sequence of symbols $S = (s_1, \ldots, s_L)$, with $L \le T$. The probability of $O$ can be calculated, using embedded HMMs for its symbols, as:

$$P(O \mid S, \boldsymbol{\Theta}) = \sum_{i_1,\ldots,i_{L+1}} \prod_{l=1}^{L} P(\mathbf{o}_{i_l}, \ldots, \mathbf{o}_{i_{l+1}-1} \mid \boldsymbol{\Theta}_{s_l}), \tag{3}$$

where the sum is carried out over all possible segmentations of $O$ into $L$ segments, that is, all sequences of indices $i_1, \ldots, i_{L+1}$ such that

$$1 = i_1 < \cdots < i_L < i_{L+1} = T + 1;$$

and $P(\mathbf{o}_{i_l}, \ldots, \mathbf{o}_{i_{l+1}-1} \mid \boldsymbol{\Theta}_{s_l})$ refers to the probability (density) of the $l$th segment, as given by (1) using the HMM associated with symbol $s_l$.

Maximum likelihood estimation (MLE) of BHMM parameters does not differ significantly from the conventional Gaussian case, and it can be efficiently performed using the well-known EM (Baum-Welch) re-estimation formulae [6,13]. Please see Ref. [11] for more details. Also as in the conventional Gaussian case, BHMM parameters can be estimated by discriminative training [14].

## 3. Windowed BHMMs with repositioning

Given a binary image normalized in height to $H$ pixels, we may think of a feature vector $\mathbf{o}_t$ as its column at position $t$ or, more generally, as a concatenation of columns in a window of $W$ columns in width, centered at position $t$. This generalization has no effect neither on the definition of BHMM nor on its MLE, although it might be very helpful to better capture the image context at each horizontal position of the image. As an example, the first row in Fig. 1 shows a binary image of four columns and five rows, which is transformed into a sequence of four 15-dimensional feature vectors by application of a sliding window of width 3. For clarity, feature vectors are depicted as $3 \times 5$ subimages instead of 15-dimensional column vectors. Note that feature vectors at positions 2 and 4 would be indistinguishable if, as in our previous approach, they were extracted with no context ($W = 1$).

Although one-dimensional, "horizontal" HMMs for image modeling can properly capture non-linear horizontal image distortions, they are somewhat limited when dealing with vertical image distortions, and this limitation might be particularly strong in the case of feature

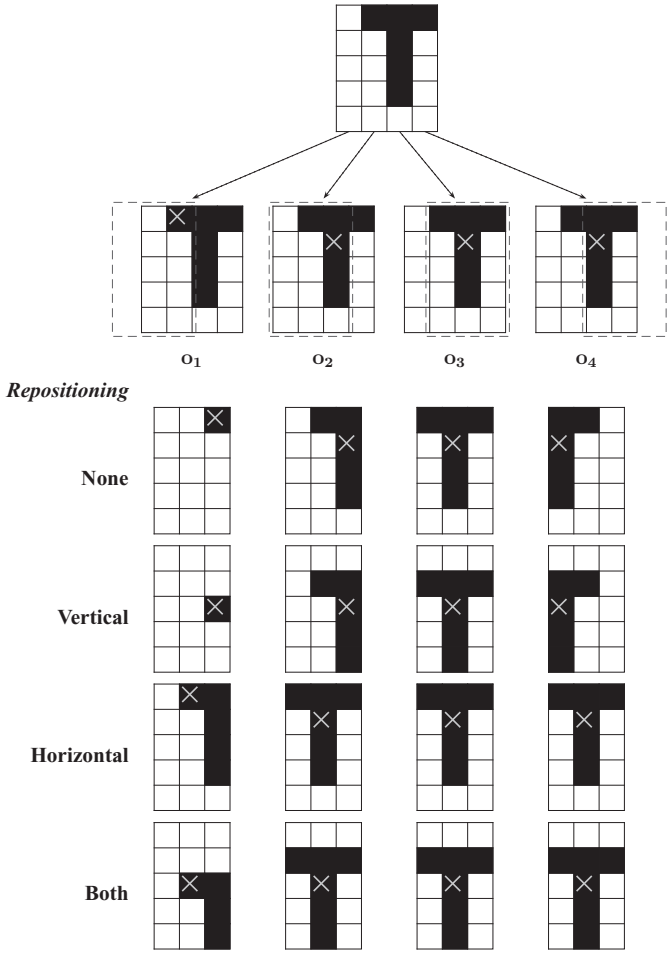


**Fig. 1.** Example of transformation of a $4 \times 5$ binary image (top) into a sequence of four 15-dimensional binary feature vectors $O = (\mathbf{o}_1, \mathbf{o}_2, \mathbf{o}_3, \mathbf{o}_4)$ using a window of width 3. After window extraction (illustrated under the original image), the standard method (no repositioning) is compared with the three repositioning methods considered: vertical, horizontal, and both directions. Mass centers of extracted windows are also indicated.