Giménez Pastor, A.; Juan, A. (2009). Bernoulli HMMs at subword level for handwritten word recognition. En Pattern Recognition and Image Analysis. Springer Verlag (Germany). 497-504. doi:10.1007/978-3-642-02172-5_64.

# Bernoulli HMMs at Subword Level for Handwritten Word Recognition*

Adrià Giménez and Alfons Juan

DSIC/ITI, Univ. Politècnica de València,
E-46022 València (Spain)
{agimenez,ajuan}@dsic.upv.es

**Abstract.** This paper presents a handwritten word recogniser based on HMMs at subword level (characters) in which state-emission probabilities are governed by multivariate Bernoulli probability functions. This recogniser works directly with raw binary pixels of the image, instead of conventional, real-valued local features. A detailed experimentation has been carried out by varying the number of states, and comparing the results with those from a conventional system based on continuous (Gaussian) densities. From this experimentation, it becomes clear that the proposed recogniser is much better than the conventional system.

**Key words:** HMM, Subword, Bernoulli, Handwritten word recognition

## 1 Introduction

Hidden Markov models (HMMs) have received significant attention in off-line handwriting recognition during the last few years [2, 3]. As in speech recognition [6], HMMs are used to model the probability (density) of an observation sequence, given its corresponding text transcription or simply its class label.

Observation sequences typically consist of fixed-dimension feature vectors which are computed locally, using a sliding window along the handwritten text image. In [1], we explored the possibility of using raw, binary pixels as feature vectors. This was done with two ideas in mind. On the one hand, this guarantees that no discriminative information is filtered out during feature extraction. On the other hand, this allows us to introduce probabilistic models that deal more directly with the object to be recognised [4, 7]. This lead us to the study of Bernoulli HMMs, that is, HMMs in which the state-conditional probabilities are governed by multivariate Bernoulli probability functions.

The direct method to model handwritten words with Bernoulli HMMs is to use an independent, separate Bernoulli HMM for each word. We did it in [1], where successful results were obtained in a task of word classification with a moderate number of (word) classes. However, this direct approach is no longer

* Work supported by the EC (FEDER) and the Spanish MEC under the MIPRCV "Consolider Ingenio 2010" research programme (CSD2007-00018), the iTransDoc research project (TIN2006-15694-CO2-01), and the FPU grant AP2005-1840.

applicable in the case of classification tasks involving a large number of classes since, typically, most classes do not have enough examples for reliable parameter estimation. As in continuous handwritten text recognition, which can be considered the extreme case of unlimited number of classes, this problem can be alleviated by using subword (character) HMMs; that is, all word classes (sentences) are modelled by concatenation of subword (character) HMMs, and thus only one HMM per character has to be trained. This is precisely what we do in this work. Empirical results are reported in which Bernoulli HMMs at subword level are compared with both, independent Bernoulli HMMs, and conventional (Gaussian) HMMs at subword level [3].

The paper is organised as follows. We first review basic HMM theory in Sections 2 and 3, mainly to fix notation. Then, in Section 4, the previous basic HMM theory is particularised to case of Bernoulli HMMs. Experiments are described in Section 5, while concluding remarks and future work are discussed in Section 6.

## 2 Hidden Markov Models

HMMs are used to model the probability (density) of an observation sequence. In a way similar to [6], we characterise an HMM as follows:

1. $M$, the number of states in the model. Individual states are labelled as $\{1, 2, \ldots, M\}$ and we denote the state at time $t$ as $q_t$. In addition, we define the special states $I$ and $F$ for *start* and *stop.*
2. The state-transition probability distribution, $A = \{a_{ij}\}$, where
$$a_{ij} = P(q_{t+1} = j \mid q_t = i)\,, \qquad 1 \leq i, j \leq M,\ i = I,\ j = F\,. \tag{1}$$
For convenience, we set $a_{IF} = 0$.
3. The observation probability (density) function, $B = \{b_j(o)\}$, in which
$$b_j(o_t) = P(o_t \mid q_t = j)\,, \tag{2}$$
defines the probability (density) function in state $j$, $j = 1, 2, \ldots, M$.

For convenience, the specification of an HMM can be compacted as

$$\lambda = (A, B)\,. \tag{3}$$

Therefore, the probability (density) of an observation sequence $O = o_1, \ldots, o_T$ is given by:

$$P(O \mid \lambda) = \sum_{I, q_1, \ldots, q_T, F} a_{Iq_1} \left[ \prod_{1 \leq t < T} a_{q_t q_{t+1}} \right] a_{q_T F} \prod_{t=1}^{T} b_{q_t}(o_t)\,. \tag{4}$$

Maximum likelihood estimation of the parameters governing an HMM can be carried out using the EM algorithm for HMMs; i.e. using Baum-Welch re-estimation formulae. Assume that the likelihood is calculated with respect to

$N$ observation sequences $O_1, \ldots, O_N$; with $O_n = (o_{n1}, \ldots, o_{nT_n})$. In the E step (at iteration $k$), the forward probability for each sample $n$, state $i$ and time $t$, $\alpha_{nt}(i) = P(o_{n1}, \ldots, o_{nt}, q_{nt} = i \mid \lambda)$, is calculated as:

$$\alpha_{nt+1}^{(k)}(j) = \begin{cases} a_{Ii}^{(k)} b_i^{(k)}(o_{n1}) & 1 \le i \le M, t = 0 \\ \left[\sum_{i=1}^{M} \alpha_{nt}^{(k)}(i) a_{ij}^{(k)}\right] b_j^{(k)}(o_{nt+1}) & \begin{array}{l} 1 \le j \le M \\ 1 \le t < T_n \end{array} \end{cases}, \tag{5}$$

while the backward probability, $\beta_{nt}(i) = P(o_{nt+1}, \ldots, o_{nT_n} \mid q_{nt} = i, \lambda)$, is:

$$\beta_{nt}^{(k)}(i) = \begin{cases} a_{iF}^{(k)} & 1 \le i \le M, t = T_n \\ \sum_{j=1}^{M} a_{ij}^{(k)} b_j^{(k)}(o_{nt+1}) \beta_{nt+1}^{(k)}(j) & \begin{array}{l} 1 \le i \le M \\ 1 \le t < T_n \end{array} \end{cases}. \tag{6}$$

The probability (density) of an observation can be calculated using forward probabilities:

$$P(O_n \mid \lambda) = \sum_{i=1}^{M} \alpha_{nT_n}(i) a_{iF}\,. \tag{7}$$

In the M step (at iteration $k$), the transition parameters are updated as follows:

$$a_{ij}^{(k+1)} = \begin{cases} \dfrac{1}{N} \displaystyle\sum_n \dfrac{\alpha_{n1}(j)^{(k)} \beta_{n1}(j)^{(k)}}{P(O_n \mid \lambda)^{(k)}} & i = 1 \\ & 1 \le j \le M \\ \dfrac{1}{\gamma(i)} \displaystyle\sum_n \dfrac{\sum_{t=1}^{T_n - 1} \alpha_{nt}^{(k)}(i) a_{ij}^{(k)} b_j^{(k)}(o_{nt+1}) \beta_{nt+1}^{(k)}(j)}{P(O_n \mid \lambda)^{(k)}} & 1 \le i, j \le M \\ \dfrac{1}{\gamma(i)} \displaystyle\sum_n \dfrac{\alpha_{nT_n}^{(k)}(i) \beta_{nT_n}^{(k)}(i)}{P(O_n \mid \lambda)^{(k)}} & 1 \le i \le M \\ & j = F \end{cases}, \tag{8}$$

where $\gamma(i)$ is:

$$\gamma(i) = \sum_n \frac{\sum_{t=1}^{T_n} \alpha_{nt}^{(k)}(i) \beta_{nt}^{(k)}(i)}{P(O_n \mid \lambda)^{(k)}}\,. \tag{9}$$

## 3 Subunit Models Based on HMMs

HMMs are often used in classification tasks to model the conditional probability of an observation sequence given a class label. A large number of classes involves a huge number of parameters; more precisely, one independent, complete HMM per class. Nevertheless, if the classes are in fact symbol sequences of a given alphabet $\{1, \ldots, C\}$, this problem can be alleviated by instead defining an HMM for each symbol of the alphabet $\{\lambda_1, \ldots, \lambda_C\}$. Therefore, for each class label we have a virtual HMM by concatenating the HMMs related to the class symbols.

The concatenation is done by joining the state $F$ of an HMM with the state $I$ of the next HMM. Thus, the probability of an observation sequence $o_1, \ldots, o_T$ given a symbol sequence $s_1, \ldots, s_L$, where $L \leq T$, is:

$$P(o_1^T \mid s_1^L, \lambda_1^C) = \sum_{\substack{I, q_1, \ldots, q_{e(1)}, Q_1, \\ q_{b(2)}, \ldots, q_{e(2)}, Q_2 \\ \cdots \\ q_{b(L)}, \ldots, q_T, F}} \prod_{l=1}^{L} P(o_{b(l)}^{e(l)}, q_{b(l)}^{e(l)}, Q_l \mid \lambda_{s_l}, Q_{l-1}), \tag{10}$$

where $e(l)$ and $b(l)$ are the positions of the first and the last observations generated by $\lambda_{s_l}$ respectively, $Q_l = I_{s_{l+1}} = F_{s_l}$, and:

$$P(o_{b(l)}^{e(l)}, q_{b(l)}^{e(l)}, Q_l \mid \lambda_{s_l}, Q_{l-1}) = a_{s_l I q_{b(l)}} \left[ \prod_{t=b(l)}^{e(l)-1} a_{s_l q_t q_{t+1}} \right] a_{s_l q_{e(l)} F} \prod_{t=b(l)}^{e(l)} b_{s_l q_t}(o_t). \tag{11}$$

As in the previous section, the parameters can be estimated using the EM algorithm. Consider the calculation of the likelihood function with respect to $N$ pairs of sequences $(O_1, S_1), \ldots, (O_N, S_N)$; with $O_n = (o_{n1}, \ldots, o_{nT_n})$ and $S_n = (s_{n1}, \ldots, s_{nL_n})$, where $L_n \leq T_n$. In the E step, the forward probabilities are calculated as:

$$\alpha_{nlt+1}^{(k)}(j) = \begin{cases} a_{s_{n1} I j}^{(k)} b_{s_{n1} j}^{(k)}(o_{n1}) & \begin{array}{r} l = 1, t = 0 \\ 1 \leq j \leq M_{s_{n1}} \end{array} \\ \left[ \sum_{\substack{1 \leq i \leq M_{s_{nl}} \\ i = I_{s_{nl}}}} \alpha_{nlt}^{(k)}(i) a_{s_{nl} ij}^{(k)} \right] b_{s_{nl} j}^{(k)}(o_{nt+1}) & \begin{array}{r} 1 \leq l \leq L_n \\ 1 \leq t < T_n \\ 1 \leq j \leq M_{s_{nl}} \end{array} \\ \sum_{i=1}^{M_{s_{nl}}} \alpha_{nlt+1}^{(k)}(i) a_{s_{nl} iF}^{(k)} & \begin{array}{r} 1 \leq l \leq L_n \\ 0 \leq t < T_n \\ j = F_{s_{nl}} \end{array} \\ \alpha_{nl-1t+1}^{(k)}(F_{s_{nl-1}}) & \begin{array}{r} 1 < l \leq L_n \\ 0 \leq t < T_n \\ j = I_{s_{nl}} \end{array} \\ 0 & \text{otherwise} \end{cases}. \tag{12}$$

Similarly, the backward probabilities are given by:

$$\beta_{nlt}^{(k)}(i) = \begin{cases} a_{s_{nL_n} iF}^{(k)} & \begin{array}{r} l = L_n, t = T_n \\ 1 \leq i \leq M_{s_{nL_n}} \end{array} \\ a_{s_{nl} iF}^{(k)} \beta_{nlt}^{(k)}(F_{s_{nl}}) & \begin{array}{r} 1 \leq l \leq L_n \\ 1 \leq t < T_n \end{array} \\ \quad + \sum_{j=1}^{M_{s_{nl}}} a_{s_{nl} ij}^{(k)} b_{s_{nl} j}^{(k)}(o_{nt+1}) \beta_{nlt+1}^{(k)}(j) & \begin{array}{r} 1 \leq i \leq M_{s_{nl}} \\ i = I_{s_{nl}} \end{array} \\ \beta_{nl+1t}^{(k)}(I_{s_{nl+1}}) & \begin{array}{r} 1 \leq l < L_n \\ 1 \leq t < T_n \\ i = F_{s_{nl}} \end{array} \\ 0 & \text{otherwise} \end{cases}. \tag{13}$$

Using the forward probabilities, the probability of an observation can be computed as:

$$P(O \mid S, \lambda_1^C) = \alpha_{LT}(F_{s_L})\,. \tag{14}$$

In the M step, transition parameters of $\lambda_c$ are updated for all $i$ ($1 \le i \le M_c$ and $i = I_c$) as:

$$a_{cij}^{(k+1)} = \frac{1}{\gamma_c(i)} \begin{cases} \displaystyle\sum_n \frac{\sum_{l:s_{nl}=c} \sum_{t=1}^{T_n-1} \alpha_{nlt}^{(k)}(i) a_{cij}^{(k)} b_{cj}^{(k)}(o_{nt+1}) \beta_{nlt+1}^{(k)}(j)}{P(O_n \mid S_n, \lambda_1^C)} & 1 \le j \le M_c \\ \displaystyle\sum_n \frac{\sum_{l:s_{nl}=c} \sum_{t=1}^{T_n} \alpha_{nlt}^{(k)}(i) a_{ciF}^{(k)} \beta_{nlt}^{(k)}(F_c)}{P(O_n \mid S_n, \lambda_1^C)} & j = F_c \end{cases}, \tag{15}$$

where:

$$\gamma_c(i) = \sum_n \frac{\sum_{l:s_{nl}=c} \sum_{t=1}^{T_n} \alpha_{nlt}^{(k)}(i) \beta_{nlt}^{(k)}(i)}{P(O_n \mid S_n, \lambda_1^C)}\,. \tag{16}$$

## 4 Bernoulli HMM

Let $O = (\boldsymbol{o_1}, \ldots, \boldsymbol{o_T})$ be a sequence of $D$-dimensional binary observation vectors. A Bernoulli HMM is an HMM in which the probability of observing $\boldsymbol{o_t}$, when $q_t = j$, is given by multivariate Bernoulli probability function for the state $j$:

$$b_j(o_t) = \prod_{d=1}^{D} p_{jd}^{o_{td}} (1 - p_{jd})^{1-o_{td}}\,, \tag{17}$$

where $p_{jd}$ is the probability for bit $d$ to be 1 when the observation vector is generated in the state $j$. Note that (17) is just the product of state-conditional unidimensional Bernoulli variables. The parameter vector associated with the state $j$, $\boldsymbol{p_j} = (p_{j1}, \ldots, p_{jD})^t$, will be referred to as the prototype of the Bernoulli distribution in the state $j$.

Using the EM algorithm, the Bernoulli prototype corresponding to the state $j$ of $\lambda_c$ has to be updated as:

$$\boldsymbol{p_{cj}}^{(k+1)} = \frac{1}{\gamma_c(j)} \sum_n \frac{\sum_{l:s_{nl}=c} \sum_{t=1}^{T_n} \alpha_{nlt}^{(k)}(j) \boldsymbol{o_{nt}} \beta_{nlt}^{(k)}(j)}{P(O_n \mid S_n, \lambda_1^C)}\,, \tag{18}$$

where $\gamma_c(j)$ is defined in (16).

Note that the time required for an EM iteration over a single sequence is $O(TM^2D)$ ($M = \sum_{l=1}^{L} M_{s_l}$ in the case of subunit HMMs), which reduces to $O(TMD)$ in the usual case of simple, linear HMM topologies. This time cost does not differ from that of continuous (Gaussian) HMMs (with diagonal covariance matrices).

In order to avoid 0 probabilities at Bernoulli prototypes, these are smoothed by a linear interpolation with a flat (uniform) prototype, $\mathbf{0.5}$,

$$\tilde{\boldsymbol{p}} = (1 - \xi)\,\boldsymbol{p} + \xi\,\mathbf{0.5}\,, \tag{19}$$

where typically $\xi = 10^{-6}$.

## 5 Experiments

The experiments have been carried out using the IAM database [5]. This corpus contains forms of unconstrained handwritten English text. All texts were extracted from the LOB corpus. A total of 657 writers contributed. Different datasets were obtained by using segmentations techniques, in particular we have used the handwritten words dataset. More precisely, we have selected those samples in this dataset that are marked as correctly segmented in the corpus, and which belong to a word with at least 10 samples.

All input gray level images were preprocessed before transforming them into sequences of feature vectors. Preprocessing consisted of three steps: gray level normalisation, deslanting, and size normalisation of ascenders and descenders. See [3] for further details.

Selected samples were randomly splitted into 30 80%-20% training-test partitions at the writer level to ensure writer-independent testing. This means about 59000 samples for training and 14000 for testing. The lexicon comprises 1117 different words and the alphabet is composed by 71 characters (upper and lowercase letters, punctuation signs, digits, etc.). This task is similar to that described in [2].

For the Bernoulli system, feature extraction has been carried out by rescaling the image to height 30 while respecting the original aspect ratio, and applying an Otsu binarisation to the resulting image. Therefore, the observation sequence is in fact a binary image of height 30. In the Gaussian case, feature vectors are of dimension 60, where the first 20 values are gray levels, and the other 40 are horizontal and vertical gray level derivatives [3]. In this case, we used the well-known HTK software [8].

Experiments have been carried out by varying number of states, $Q \in \{4, 6, 8, 10, 12\}$, and comparing our Bernoulli system to a conventional system based on Gaussian HMMs. Both systems have been initialised by first segmenting the training set using a "neutral" model, and then using the resulting segments to perform a Viterbi initialisation. The model has been trained with 4 EM iterations, and the recognition has been performed using the Viterbi algorithm. Figure 1 shows the results, where each point is the average of 30 repetitions (30 random splits). Vertical bars denote $\pm$ standard deviation.

The results obtained with the Bernoulli system are much better than those given by the Gaussian system. In particular, the best result for the Bernoulli system is a 44.0% classification error, obtained with $Q = 10$. In contrast, the best result for the Gaussian system is a 64.2% classification error, obtained with $Q = 8$.

We have extended the experiment by using a different number of states for each HMM. For this purpose, the training set was first aligned and segmented. Then, for each HMM, the number of states was calculated as the average length of the segments multiplied by a predefined *load factor,* $f$. This load factor indicates the number of observations that a state generates on average. We have tried several load factors $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$. The results obtained are very similar to those reported above; i.e. the Bernoulli system outperforms the Gaus-

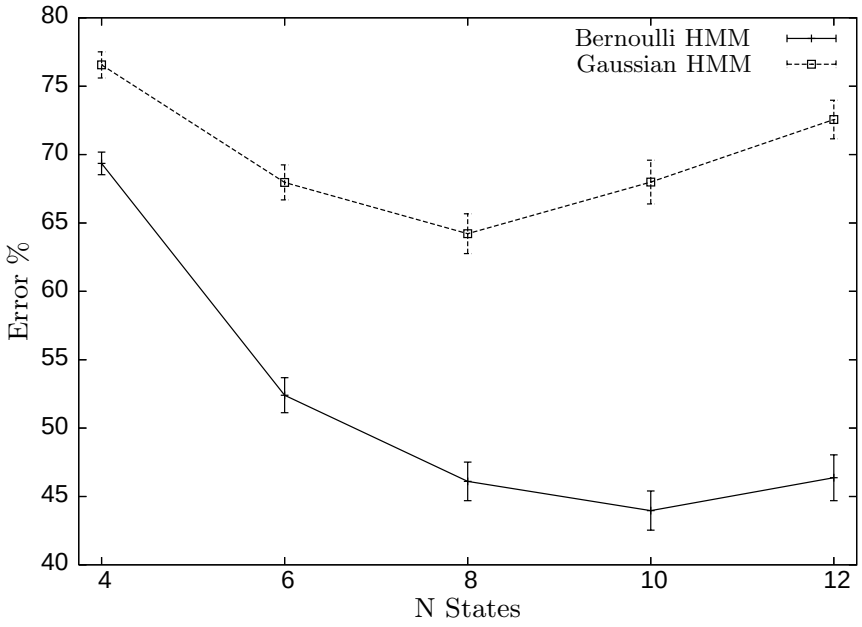


**Fig. 1.** Classification error (in %) as a function of the number of states for the Bernoulli HMM system and the conventional, Gaussian HMM system.

sian system with error rates similar to those in Figure 1. In both systems the best results are obtained with $f = 0.4$.

We concluded the experiments by repeating those shown in Figure 1, but using one Bernoulli HMM per word instead of one Bernoulli HMM per character while (approximately) keeping the same number of parameters. Using Bernoulli HMMs at word level and $Q = 1$ (1117 Bernoulli prototypes) a classification error of 89.3% was achieved, while with Bernoulli HMMs at subword level and $Q = 10$ (710 Bernoulli prototypes) we had a classification error of 44.0%. Moreover, using Bernoulli HMMs at word level and $Q = 10$ (11170 Bernoulli prototypes) the classification error is 64%; that is, it is still not better than that obtained with Bernoulli HMMs at subword level.

## 6 Concluding Remarks and Future Work

Bernoulli HMMs at subword (character) level has been studied and empirically tested on a task of handwritten word classification from the IAM database. We have obtained a classification error of 44.0%, which is 20 points better than the best result obtained with a conventional, Gaussian-based HMM system. It is also worth noting that the proposed system works with less features and parameters than the conventional system (30 vs 60 and half of the parameters). On the other

hand, the proposed system has been also compared with Bernoulli HMM-based classifier at word level. As expected, the advantage of using subword models has been clearly confirmed.

For future work, we plan to try Bernoulli mixtures instead of a single Bernoulli at each state. We also plan to use the ideas reported in [7] for explicitly modelling of invariances in Bernoulli mixtures, and to extend the experiments to general handwritten text recognition.

## References

1. A. Giménez-Pastor and A. Juan-Císcar. Bernoulli HMMs for Off-line Handwriting Recognition. In *Proc. of the 8th Int. Workshop on Pattern Recognition in Information Systems (PRIS 2008)*, pages 86–91, Barcelona (Spain), June 2008.
2. Simon Günter and Horst Bunke. HMM-based handwritten word recognition: on the optimization of the number of states, training iterations and Gaussian components. *Pattern Recognition*, 37:2069–2079, 2004.
3. Moisés Pastor i Gadea. *Aportaciones al reconocimiento automático de texto manuscrito*. PhD thesis, Dep. de Sistemes Informàtics i Computació, València, Spain, Oct 2007. Advisors: E. Vidal and A.H. Tosselli.
4. A. Juan and E. Vidal. Bernoulli mixture models for binary images. In *Proc. of the 17th Int. Conf. on Pattern Recognition (ICPR 2004)*, volume 3, Cambridge (UK), August 2004.
5. U.V. Marti and H. Bunke. The IAM-database: an English sentence database for offline handwriting recognition. 5(1):39–46, 2002.
6. Lawrence Rabiner and Biing-Hwang Juang. *Fundamentals of speech recognition*. Prentice-Hall, 1993.
7. V. Romero, A. Giménez, and A. Juan. Explicit Modelling of Invariances in Bernoulli Mixtures for Binary Images. In *3rd Iberian Conference on Pattern Recognition and Image Analysis*, volume 4477 of *LNCS*, pages 539–546. Springer-Verlag, Girona (Spain), June 2007.
8. S. Young et al. *The HTK Book*. Cambridge University Engineering Department, 1995.