Giménez Pastor, A.; Andrés Ferrer, J.; Juan, A. (2014). Discriminative Bernoulli HMMs for isolated handwritten word recognition. Pattern Recognition Letters. 35:157-168. doi:10.1016/j.patrec.2013.05.016.

Discriminative Bernoulli HMMs for isolated handwritten word recognition

Adrià Giménez, Jesús Andrés-Ferrer and Alfons Juan

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

Abstract

Bernoulli HMMs (BHMMs) have been successfully applied to handwritten text recognition (HTR) tasks such as continuous and isolated handwritten words. BHMMs belong to the generative model family and, hence, are usually trained by (joint) maximum likelihood estimation (MLE) by means of the Baum-Welch algorithm. Despite the good properties of the MLE criterion, there are better training criteria such as maximum mutual information (MMI). The MMI is the most widespread criterion to train discriminative models such as log-linear (or maximum entropy) models. Inspired by a BHMM classifier, in this work, a log-linear HMM (LLHMM) for binary data is proposed. The proposed model is proved to be equivalent to the BHMM classifier, and, in this way, a discriminative training framework for BHMM classifiers is defined. The behavior of the proposed discriminative training framework is deeply studied in a well known task of isolated word recognition, the RIMES database.

Keywords: HTR, Bernoulli HMM, Log-Linear HMM, MMI, RIMES

Email address: {agimenez,jandres,ajuan}@dsic.upv.es (Adrià Giménez, Jesús Andrés-Ferrer and Alfons Juan)

1. Introduction

In the past few years Bernoulli HMMs (BHMMs) have been proved to be competitive for handwritten text recognition (HTR). Specifically, competitive performance has been reported by BHMMs on handwritten English text (Giménez and Juan, 2009), and Arabic HTR (Giménez et al., 2010; Märgner and El Abed, 2010) [1].

Handwritten word classifiers based on HMMs, and in particular in BHMMs, are *generative models*. Generative models are classifiers based on the optimal Bayes classifier (Duda and Hart, 1973) which classify choosing the class c^* that maximizes the posterior class probability for a given input $\mathbf{x}$ as follows

$$c^\star(\mathbf{x}) = \arg\max_c p(c \mid \mathbf{x}) = \arg\max_c p(c, \mathbf{x}) \tag{1}$$

where instead of directly approximating the posterior class probability $p(c \mid \mathbf{x})$, the joint probability is modelled by a distribution $p_\theta(c, \mathbf{x})$ parameterized with θ. Among many other advantages, the generative models have two outstanding properties. On the one hand, the parameters of the generative models are easily understandable for researchers. For instance, in the BHMMs all parameters can be interpreted as percentages, and in particular, the emission parameters that model the emission probabilities are easily interpreted as grey level images. On the other hand, generative models are mostly trained with maximum likelihood estimation (MLE) criterion for which there are well-known algorithms for training latent variable mod-

[1] the BHMM achieved the first place prize in the Arabic HTR competition organized during the ICFHR 2010 conference.

els (Dempster et al., 1977) in general, and HMMs in particular (Rabiner and Juang, 1993).

Despite the good properties of the MLE criterion, it acknowledges an important drawback in classification problems. The MLE is aimed at explaining the probability distribution that underlies in the training sample by maximizing the likelihood of the joint probability function $p_\theta(c, \mathbf{x})$. However, we are simply interested in classifying inputs, and there is no guarantee that the MLE parameters are the most suitable for classifying, even though they have been proved to be competitive.

An alternative to generative models are the discriminative models. Discriminative models and criteria are aimed at classifying the data without explaining the data distribution itself. These models are also based on the Bayes decision rule in (1) but instead of the joint probability, they directly approximate the posterior class probability by a model $p_\lambda(c\,|\,\mathbf{x})$ parameterized by λ. However, discriminative parameters are difficult to understand provided that they do not explain the input. Discriminative parameters are usually estimated by the *maximum mutual information (MMI)* criterion, which directly maximizes the likelihood of the posterior probability function $p_\lambda(c|\mathbf{x})$. In contrast to MLE, the parameters estimated with MMI maximize the most the differences between classes in order to better classify samples. Unfortunately, there is no closed form solution for the MMI criterion, and few unsatisfactory algorithms are available for finding the optimal parameters. This problem is specially remarkable for discriminative models with hidden variables as HMMs.

In Giménez et al. (2011) a MMI training scheme for Bernoulli mixture

classifiers was proposed and tested in a task of isolated handwritten digit recognition. The proposed approach was based on the idea of finding a similar discriminative classifier to the Bernoulli mixture classifier, and then prove the equivalence between both classifiers. The results analyzed in Giménez et al. (2011) report that discriminatively trained Bernoulli mixture classifier outperforms the generative Bernoulli mixture classifiers. In this paper the work in Giménez et al. (2011) is extended to more complex models, the BHMMs, which are assessed in a complex isolated word recognition task. Specifically, we compared both generative and discriminative approaches in the RIMES database (Grosicki et al., 2009) in which the discriminative models obtained very competitive results surpassing the performance obtained by generative classifiers.

More precisely, the contributions of this work are the following:

1. We propose a particular case of log-linear HMM (LLHMM) classifier, which can also be interpreted as a semi-Markov conditional Markov chain (semi-CRF), for binary data inspired by the BHMM classifier.
2. We prove the equivalence between BHMMs and the proposed discriminative model for binary data.
3. We provide a discriminative training scheme for BHMM classifiers by means of their equivalence with LLHMMs, and analyze several discriminative training criteria such as MMI.
4. We provide the capability to understand discriminative parameters from a generative point of view by means of their equivalence with BHMMs.

The remainder of the paper is organized as follows. A review of BHMMs

is given in Sec. 2. The proposed LLHMM or semi-CRF classifier for binary data is described in Sec. 3. The Sec. 4 proves equivalence between both classifiers, and in Sec. 5 the parameter estimation for the LLHMM is analyzed. The proposed training scheme is deeply analyzed on the RIMES database in Sec. 6. We conclude the paper by summarizing and discussing the most important results and future research directions.

2. Bernoulli HMM

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

$$p_\theta(O) := \sum_{\mathbf{q}} p_\theta(O, \mathbf{q}) = \sum_{\mathbf{q}} \prod_{t=0}^{T} a(q_t, q_{t+1}) \prod_{t=1}^{T} b_{q_t}(o_t)\,, \tag{2}$$

where we have uncovered the latent variables $\mathbf{q} = (q_0, q_1, \ldots, q_{T+1})$ which represent all the possible state sequences (or paths), such that $q_1, \ldots, q_T \in \{1, \ldots, M\}$ are the regular states chosen out of a total of M states, and the first ($q_0 = I$) and last ($q_{T+1} = F$) states are special, the so-called *initial or start state* and the *final or stop* state, respectively. Furthermore, for any regular states i and j, $a(i, j)$ denotes the *transition* probability from i to j, while b_j is the probability distribution for an *observation* at state j.

If we further assume that O is a sequence of binary featured vectors, then a *Bernoulli (mixture) HMM (BHMM)* is an HMM in which the probability of observing $\mathbf{o}_t$ at position t and the state j ($q_t = j$) follows a Bernoulli mixture distribution

$$b_j(\mathbf{o}_t) := \sum_{k=1}^{K} \tau_j(k) \prod_{d=1}^{D} {p_{jkd}}^{o_{td}} \left(1 - p_{jkd}\right)^{1 - o_{td}}\,, \tag{3}$$

where $\tau_j(k)$ and $\mathbf{p}_{jk}$ are, respectively, the prior and prototype of the k-th mixture component in state j. Fig. 1, depicts some prototypes for several states and components.

In isolated handwriting word recognition, BHMMs are used in Bayes' classifiers to approximate the input probability of a binary image, which is represented by an observation sequence of binary feature vectors for a given transcription. More precisely, the most probable transcription $S^\star \in W$ for a given observation sequence O is obtained according to

$$S^\star = \arg\max_S \; p(S, O) = \arg\max_S \; p(S)\, p(O \mid S)\,, \tag{4}$$

where for each possible transcription S, the emission probability, $p(O \mid S)$, is approximated as a BHMM, and $p(S)$ is modelled regarding each probability as a parameter itself, π_S.

The number of possible transcriptions in handwriting recognition is typically large, and consequently, the resulting parameters for a BHMMs are very sparse. In order to amend sparseness problems, BHMMs at 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, we assume that each character c is modeled with a different BHMM with parametric vector θ_c, which is shared among words. Unfortunately, the input featured vectors are not aligned with individual characters, and the character boundaries are, therefore, unknown. For a given feature sequence $O = (\mathbf{o}_1, \ldots, \mathbf{o}_T)$ representing a sequence of symbols $S = (s_1, \ldots, s_L)$, with $L \leq T$; the latent segmentation $\mathbf{i} = (i_1, i_2, \ldots, i_{L+1})$ defined as follows

$$1 = i_1 < \cdots < i_L < i_{L+1} = T + 1; \tag{5}$$

induces the segmentation of O into L segments which monotonically correspond to each symbol. Specifically, the feature segment corresponding to the l-th character, s_l, is denoted by $O(i_l, i_{l+1}) = \mathbf{o}_{i_l}, \ldots, \mathbf{o}_{i_{l+1}-1}$. Finally, the probability of O is determined by

$$p_\theta(O \mid S) = \sum_{\mathbf{i}} p_\theta(O, \mathbf{i} \mid S) = \sum_{\mathbf{i}} \prod_{l=1}^{L} p(O(i_l, i_{l+1}) \mid s_l)\,, \tag{6}$$

where the sum is carried out over all possible segmentations of O into L segments, and $p(O(i_l, i_{l+1}) \mid s_l)$ is the probability of the l-th segment given by a BHMM in (2) with the parameters, θ_{s_l}, associated to the character s_l. Note that θ comprises all the embedded character parameters, i.e. $\theta = \{\theta_1, \ldots, \theta_C\}$. These parameters are commonly estimated by MLE (Giménez et al., 2010).

Many of the parameters of the discussed model are constrained to sum 1, since they directly approximate probabilities. These parameters are the mixture coefficients, the state transitions and the word prior probabilities, which must verify

$$\sum_{w} \pi_w = 1, \qquad \forall_{c,q} : \sum_{q'} a_c(q, q') = 1, \qquad \forall_{c,q} : \sum_{k} \tau_{cq}(k) = 1\,. \tag{7}$$

In Fig. 1, an embedded BHMM for the number 313 is shown. Note that the word model is the result of concatenating character models for the digit 3, blank space, digit 1, and the blank space and digit 3 again, in that order.

3. Log-linear HMM Classifier for Binary Data

In this section, we propose a discriminative classifier inspired by the BHMM classifier for isolated handwritten words (4). The discriminative

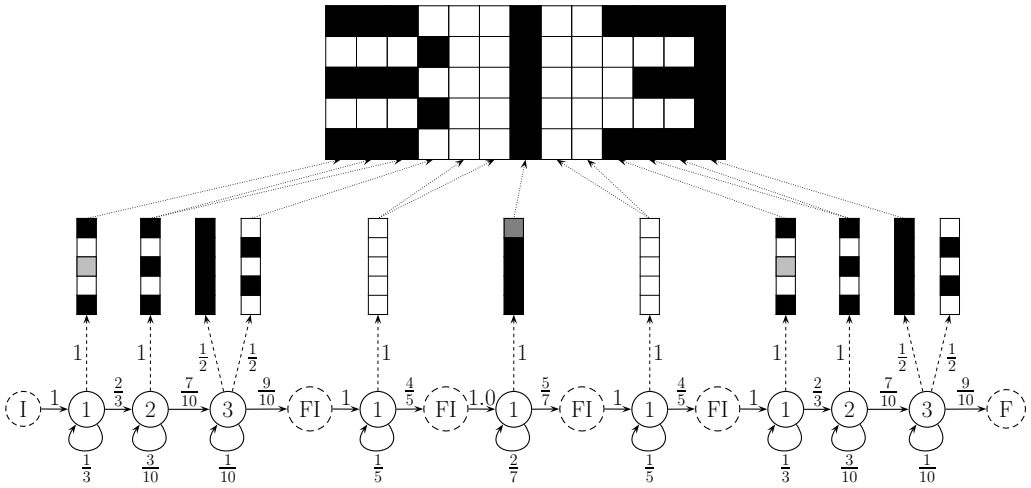


Figure 1: The binary featured vector representing the number 313 and the most probable path for generating it accordingly to a BHMM. Bernoulli prototype probabilities are represented using the following color scheme: black=1, white=0, gray=0.5 and light gray=0.1.

classifier proposed is based on a log-linear model, which is inferred from the parameters of a BHMM classifier. In what follows, we define the log-linear model and how a log-linear HMM (LLHMM) discriminative classifier can be built using it.

3.1. BHMM Inspired Log-linear Model

The BHMM classifier can be expressed by plugging (2), (3), and (6) as follows

$$p_\theta(O, S) = \sum_{\mathbf{i},\mathbf{q},\mathbf{k}} p_\theta(O, S, \mathbf{i}, \mathbf{q}, \mathbf{k}) \,, \tag{8}$$

where by $\mathbf{i}, \mathbf{q}, \mathbf{k}$ we denote the 3 latent variables of the model, namely: the segmentation, $\mathbf{i}$, of O into L segments as defined in (5); the state sequence, $\mathbf{q} = (q_0, q_1, \ldots, q_{T+1})$; and the emission component at each state, $\mathbf{k}$. Accord-

ing to the given segmentation $\mathbf{i}$, the state sequence $\mathbf{q}$ must be valid, which implies that if t belongs to the l-th segment, then the state q_t must be a possible state of the character-level BHMM for the corresponding symbol s_l. Similarly, $\mathbf{k} = (k_1, \ldots, k_T)$ must be a valid integer sequence where k_t denotes the selected mixture component for state q_t, among all the components of the state.

The joint probability in the right-hand-side of previous equation, $p_\theta(O, S, \mathbf{i}, \mathbf{q}, \mathbf{k})$, is decomposed left-to-right as follows

$$p_\theta(O, S, \mathbf{i}, \mathbf{q}, \mathbf{k}) = \pi_S p_\theta(O, \mathbf{i}, \mathbf{q}, \mathbf{k} \mid S) = \pi_S \, p_\theta(\mathbf{i}, \mathbf{q} \mid S) \, p_\theta(O, \mathbf{k} \mid \mathbf{i}, \mathbf{q}, S) \quad (9)$$

where $p_\theta(\mathbf{i}, \mathbf{q} \mid S)$ is the transition probability of the word-level BHMM and $p_\theta(O, \mathbf{k} \mid \mathbf{i}, \mathbf{q}, S)$ the emission probability. The transition probabilities are then decomposed into

$$p_\theta(\mathbf{i}, \mathbf{q} \mid S) := \prod_{l=1}^{L} a_{s_l}(I, q_{i_l}) \cdot a_{s_l}(q_{i_{l+1}-1}, F) \prod_{t=i_l}^{i_{l+1}-2} a_{s_l}(q_t, q_{t+1}) \quad (10)$$

where the first product accounts for the input, $a_{s_l}(I, q_{i_l})$, and output, $a_{s_l}(q_{i_{l+1}-1}, F)$, transitions of the embedded model for the character s_l; and where the second product are the inner transitions within the embedded character model. In the remaining of the paper, we will not differentiate between inner and outer transitions since this is a well-known characteristic of HMM, and by extension to our BHMM model. Furthermore, this significantly simplifies the notation. For instance, previous equation is expressed as

$$p_\theta(\mathbf{i}, \mathbf{q} \mid S) := \prod_{l,t} a_{s_l}(q_t q_{t+1}) \quad (11)$$

where we have also omitted the boundaries of the products, which can always be traced back to (10).

Similarly, the emission probability is decomposed as follows

$$p_\theta(O, \mathbf{k} \mid \mathbf{i}, \mathbf{q}, S) := \prod_{l=1}^{L} \prod_{t=i_l}^{i_{l+1}-1} \tau_{s_l q_t}(k_t) \prod_{d=1}^{D} p_{s_l q_t k_t d}{}^{o_{td}} (1 - p_{s_l q_t k_t d})^{(1-o_{td})} \,. \quad (12)$$

where again by dropping the product boundaries is simplified to

$$p_\theta(O, \mathbf{k} \mid \mathbf{i}, \mathbf{q}, S) := \prod_{l,t} \tau_{s_l q_t}(k_t) \prod_{d} p_{s_l q_t k_t d}{}^{o_{td}} (1 - p_{s_l q_t k_t d})^{(1-o_{td})} \,. \quad (13)$$

with $\tau_{s_l q_t}(k_t)$ and $\mathbf{p}_{s_l q_t k_t}$ being the prior and prototype of the k-th mixture component at state q_t of the character s_l.

Consequently, the model in (9) can be expressed as follows

$$p_\theta(O, S, \mathbf{i}, \mathbf{q}, \mathbf{k}) = \exp\big(\log \pi_S + \log p_\theta(\mathbf{i}, \mathbf{q} \mid S) + \log p_\theta(O, \mathbf{k} \mid \mathbf{i}, \mathbf{q}, S) \big) \quad (14)$$

where the logarithms of the probabilities are given by

$$\log p_\theta(\mathbf{i}, \mathbf{q} \mid S) = \sum_{l,t} \log a_{s_l}(q_t, q_{t+1}) \quad (15)$$

and

$$\log p_\theta(O, \mathbf{k} \mid \mathbf{i}, \mathbf{q}, S) = \sum_{l,t} (\log \tau_{s_l q_t}(k_t) + \xi_{s_l q_t}(k_t) + \sum_{l,t,d} o_{td} \log \frac{p_{s_l q_t k_t d}}{(1 - p_{s_l q_t k_t d})} \,, \quad (16)$$

with $\xi_{cq}(k)$ defined as

$$\xi_{cq}(k) = \sum_{d} \log (1 - p_{cqkd}) \,. \quad (17)$$

Note that the term $\xi_{cq}(k)$ is easily obtained when applying the logarithm to (13) by rearranging terms similarly to Giménez et al. (2011).

At this point, we reparameterize the probabilities in terms of the new parameters, λ, as follow

$$\lambda_S = \log \pi_S \,, \tag{18}$$

$$\lambda_{cqq'} = \log a_c(q, q') \,, \tag{19}$$

$$\lambda_{cqk} = \log \tau_{cq}(k) + \xi_{cq}(k) \,, \tag{20}$$

$$\lambda_{cqkd} = \log \frac{p_{cqkd}}{1 - p_{cqkd}} \,, \tag{21}$$

for each character, c; states, q and q'; mixture component, k; and input dimension, d.

Provided the previous parameterization, the original joint probability in (9) is alternatively expressed as follows

$$p_\lambda(O, S, \mathbf{i}, \mathbf{q}, \mathbf{k}) = \exp(\lambda_S + \sum_{l,t} \lambda_{s_l q_t q_{t+1}} + \sum_{l,t} \lambda_{s_l q_t k_t} + \sum_{l,t,d} o_{td} \lambda_{s_l q_t k_t d}) \tag{22}$$

In order to simplify notation, we adopt here the standard and powerful notation of log-linear models. We define an index m that ranges through all the subindexes of the previous equation, i.e., m ranges from $\{S\}$ over $\{c, q, q'\}$ and $\{c, q, k\}$ to $\{c, q, k, d\}$. We also introduce a function $g_m(O, S, \mathbf{i}, \mathbf{q}, \mathbf{k})$ that equals to the number of times the parameter λ_m is used, except for the parameters $\{\lambda_{cqkd}\}$. In this case, the function $g_m(O, S, \mathbf{i}, \mathbf{q}, \mathbf{k})$ with $m = cqkd$, counts the number of times the d-th bit is *set* and has been generated with the k-th component in the state, q, of the character, c. The simplest case of the function is that of the prior parameters λ_S for which $g_m = 1$ (with $m = S$).

The proposed notation simplifies (22) into

$$p_\lambda(O, S, \mathbf{i}, \mathbf{q}, \mathbf{k}) = \exp(\sum_{m \in \mathcal{M}(O,S,\mathbf{i},\mathbf{q},\mathbf{k})} \lambda_m g_m(O, S, \mathbf{i}, \mathbf{q}, \mathbf{k})) \tag{23}$$

where by $\mathcal{M}(O, S, \mathbf{i}, \mathbf{q}, \mathbf{k})$ we denote the set of values through which the index m ranges. It is important to notice that this set depends on all the variables, namely $O, S, \mathbf{i}, \mathbf{q}, \mathbf{k}$; and changes with them. However, it is simpler to define $\mathcal{M}$ as the union of all the possible indexes that our parameters require and replace the functions g_m by the so-called *feature functions* $f_m(O, S, \mathbf{i}, \mathbf{q}, \mathbf{k})$, which are equal to g_m if m is an index of a required parameter and 0 otherwise. For instance, consider again the word prior example with the new domain $\mathcal{M}$. In this case, the index m can take the value of any word, S' in the vocabulary; and then the feature function is defined as

$$f_{S'}(O, S, \mathbf{i}, \mathbf{q}, \mathbf{k}) = \delta(S, S') \tag{24}$$

where $\delta(a, b)$ is the Kronecker delta function, which equals 1 if both elements are equal, and 0 otherwise. The feature functions for the remaining parameters are detailed in Sec. 3.3.

Finally, equation (22) is expressed as

$$p_\lambda(O, S, \mathbf{i}, \mathbf{q}, \mathbf{k}) = \exp(\sum_{m \in \mathcal{M}} \lambda_m f_m(O, S, \mathbf{i}, \mathbf{q}, \mathbf{k})) = \exp(\lambda' \mathbf{f}(O, S, \mathbf{i}, \mathbf{q}, \mathbf{k}))\,, \tag{25}$$

where we can substitute the sum by its vectorial notation. The model in (25) when plugged into (8) is a log-linear model with binary inputs.

3.2. Discriminative Classifier

Log-linear models are commonly employed to approximate posterior probabilities. From (25), we can approximate the posterior class probability required by the optimal Bayes' classifier in (1) as follows

$$p_\lambda(S \mid O) = \sum_{\mathbf{i},\mathbf{q},\mathbf{k}} p_\lambda(S, \mathbf{i}, \mathbf{q}, \mathbf{k} \mid O)\,, \tag{26}$$

where $p_\lambda(S, \mathbf{i}, \mathbf{q}, \mathbf{k} \mid O)$ is approximated by (25) and the Bayes' theorem as

$$p_\lambda(S, \mathbf{i}, \mathbf{q}, \mathbf{k} \mid O) = \frac{p_\lambda(O, S, \mathbf{i}, \mathbf{q}, \mathbf{k})}{p_\lambda(O)} = \frac{\exp(\sum_{m \in \mathcal{M}} \lambda_m f_m(O, S, \mathbf{i}, \mathbf{q}, \mathbf{k}))}{p_\lambda(O)}. \tag{27}$$

It is worth noting that the denominator is a probability because of the transformation that we have performed in equations (18)-(21). However, we wish to select any arbitrary parametric vector, λ, and in such a case, the denominator also becomes arbitrary, yielding the *LLHMM model*

$$p_\lambda(S, \mathbf{i}, \mathbf{q}, \mathbf{k} \mid O) = \frac{\exp(\sum_{m \in \mathcal{M}} \lambda_m f_m(O, S, \mathbf{i}, \mathbf{q}, \mathbf{k}))}{\mathcal{Z}_\lambda(O)}, \tag{28}$$

where $\mathcal{Z}_\lambda(O)$ is a normalization constant defined as

$$\mathcal{Z}_\lambda(O) = \sum_S \sum_{\mathbf{i},\mathbf{q},\mathbf{k}} \exp(\lambda' \mathbf{f}(O, S, \mathbf{i}, \mathbf{q}, \mathbf{k})), \tag{29}$$

that for the specific parameters in equations (18)-(21) corresponds to the marginal probability $p_\lambda(O)$. The log-linear model in (28) is a log-linear model with hidden variables for the segmentation and for the states which have a first order dependence. This model is variation of a semi-Markov conditional random field.

The previous LLHMM is used in the optimal Bayes' rule to obtain the *LLHMM classifier*

$$S^\star = \arg\max_S \, p_\lambda(S \mid O), \tag{30}$$

3.3. Feature Functions

As discussed before, in order to use the standard notation in log-linear models, we need to define the feature functions for each kind of parameters.

For a given character c out of C different symbols, and for a given pair of state indexes (q, q') of that character, we define the *transition features* $f_{cqq'}(O, S, \mathbf{i}, \mathbf{q}, \mathbf{k}) = f_{cqq'}(S, \mathbf{i}, \mathbf{q})$ as follows

$$f_{cqq'}(S, \mathbf{i}, \mathbf{q}) = \sum_{l=1}^{L} \delta(s_l, c) \begin{cases} \sum_{t=i_l}^{i_{l+1}-2} \delta(q_t, q)\delta(q_{t+1}, q') & 1 \leq q, q' \leq M_c \\ \delta(q_{i_l}, q') & q = I, 1 \leq q' \leq M_c \\ \delta(q_{i_{l+1}-1}, q) & 1 \leq q \leq M_c, q' = F \\ 0 & \text{otherwise} \end{cases} \tag{31}$$

where I and F represent the *initial* and *final* states respectively. Intuitively, this feature counts the number of times the specific transition from q to q' of the character c, is used in the input $S, \mathbf{i}, \mathbf{q}$. Note that it can be 0, if, for instance, the character c is not part of word S.

For the mixture components, we define the *component features* for each character c, state q and component k as follows

$$f_{cqk}(O, S, \mathbf{i}, \mathbf{q}, \mathbf{k}) = f_{cqk}(S, \mathbf{i}, \mathbf{q}, \mathbf{k}) = \sum_{l=1}^{L} \delta(s_l, c) \sum_{t=i_l}^{i_{l+1}-1} \delta(q_t, q)\delta(k_t, k)\,. \tag{32}$$

Intuitively, this feature counts the number of times an emission of O is generated by the k-th component of the state q of the character c.

The final set of features are the *emission features*, which are given as follows for each character c, state q, component k and dimension d

$$f_{cqkd}(O, S, \mathbf{i}, \mathbf{q}, \mathbf{k}) = \sum_{l=1}^{L} \delta(s_l, c) \sum_{t=i_l}^{i_{l+1}-1} \delta(q_t, q)\delta(k_t, k)o_{td}\,. \tag{33}$$

4. Equivalence Between BHMMs and LLHMMs

In this section we prove that the BHMM classifier for isolated words is equivalent to the LLHMM proposed in Sec. 3. A generative classifier is said to be equivalent to a discriminative classifier if for a given set of generative parameters θ, a set of discriminative parameters λ can be found such that

$$\arg\max_{S \in W} p_\theta(O, S) = \arg\max_{S \in W} p_\lambda(S \mid O)\,; \tag{34}$$

and vice-versa. Note that the previous equivalence holds even when any of both probabilities is scaled by a factor that does not depends on S, and consequently, the normalization constant of the LLHMM, $\mathcal{Z}_\lambda(O)$, defined in (29), can be removed from the right-hand side (34) without changing the equivalence. The proof of the equation is done in two steps by proving two implications: left to right, and right to left.

4.1. From Generative to Discriminative Parameters

Unlike the converse direction, it is relatively simple to prove that given a BHMM classifier for isolated word recognition, it can be reparameterized into a LLHMM. Recall that by definition of the LLHMM, if we set the log-linear parameters, λ, using the generative parameters, θ, as defined in (18)-(21), then we have that

$$p_\theta(O, S) = \sum_{\mathbf{i},\mathbf{q},\mathbf{k}} \exp(\lambda' \mathbf{f}(O, S, \mathbf{i}, \mathbf{q}, \mathbf{k})) = \mathcal{Z}_\lambda(O) p_\lambda(S \mid O)\,. \tag{35}$$

Therefore, these two models when inserted into their corresponding classifiers in (34) produce proportional scores and, hence, select the same word or class.

4.2. From Discriminative to Generative Parameters

In this subsection, we prove the converse statement: that given a LLHMM classifier for isolated word recognition as defined in Sec. 3, an equivalent BHMM classifier exists. We begin expressing the right-hand model in (34) as follows

$$p_\lambda(S \mid O) = \sum_{\mathbf{i},\mathbf{q},\mathbf{k}} \frac{h_\lambda(O, S, \mathbf{i}, \mathbf{q}, \mathbf{k})}{\mathcal{Z}_\lambda(O)}\,, \tag{36}$$

where $h_\lambda(O, S, \mathbf{i}, \mathbf{q}, \mathbf{k}) = \exp(\lambda' \mathbf{f}(O, S, \mathbf{i}, \mathbf{q}, \mathbf{k}))$.

We start instantiating the feature $h_\lambda(\cdots)$ in previous equations obtaining

$$h_\lambda(O, S, \mathbf{i}, \mathbf{q}, \mathbf{k}) = \exp(\lambda_S) \cdot h_\lambda(\mathbf{i}, \mathbf{q}; S) \cdot h_\lambda(O, \mathbf{k}; \mathbf{i}, \mathbf{q}, S) \tag{37}$$

with

$$h_\lambda(\mathbf{i}, \mathbf{q}; S) = \exp\Big(\sum_{l,t} \lambda_{s_l q_t q_{t+1}}\Big)\,, \tag{38}$$

and

$$h_\lambda(O, \mathbf{k}; \mathbf{i}, \mathbf{q}, S) = \exp\Big(\sum_{l,t} \lambda_{s_l q_t k_t} + \sum_{l,t,d} o_{td} \lambda_{s_l q_t k_t d}\Big)\,. \tag{39}$$

If we compare (9) expanded accordingly to (11) and (12), with (37) expanded with (38); then it is observed that each of the 3 terms in the right-hand side in (37) must be transformed, independently, into the corresponding term in (9).

Firstly, we transform $h_\lambda(O, \mathbf{k}; \mathbf{i}, \mathbf{q}, S)$ into $p_\theta(O, \mathbf{k} \mid \mathbf{i}, \mathbf{q}, S)$. Therefore, we need to transform the part of the discriminative parameters $\{\lambda_{cqk}\}$ and $\{\lambda_{cqkd}\}$ into the generative parameters $\{\tau; \mathbf{p}\}$, where τ is constrained as shown in (7). For doing that, (37) is multiplied and divided by $\exp(\sum_{l,t} \zeta_{s_l q_t})$,

and then, we rearrange the multiplication into (39) as follows

$$\exp(\sum_{l,t}\zeta_{s_l q_t})h_\lambda(O,\mathbf{k};\mathbf{i},\mathbf{q},S) = \exp\Big(\sum_{l,t}(\lambda_{s_l q_t k_t}+\zeta_{s_l q_t}) + \sum_{l,t,d} o_{td}\lambda_{s_l q_t k_t d}\Big)\,, \tag{40}$$

whereas the division is moved into the second term in the right-hand side of (37), yielding

$$\exp(-\sum_{l,t}\zeta_{s_l q_t})h_\lambda(\mathbf{i},\mathbf{q};S) = \exp\Big(\sum_{l,t}\bar{\lambda}_{s_l q_t q_{t+1}}\Big)\,, \tag{41}$$

where $\bar{\lambda}_{sqq'} = \lambda_{sqq'} - \zeta_{sq}$ will be used afterwards. The unknown parameters $\{\zeta_{sq}\}$ are introduced to force the generative parameters $\{\tau_{cq}(k)\}$ to sum 1 in the transformation.

From (16) and (40), and taking into account the constraints in (7), the solution must fulfill the following 3 constraints

$$\lambda_{cqkd} = \log\frac{p_{cqkd}}{1-p_{cqkd}}\,, \tag{42}$$

$$\lambda_{cqk} + \zeta_{cq} = \log\tau_{cq}(k) + \xi_{cq}(k)\,, \tag{43}$$

$$\sum_{k=1}^{K_{cq}}\tau_{cq}(k) = 1\,. \tag{44}$$

Then, from (42) we work out the value of p_{cqkd}

$$p_{cqkd} = \frac{\exp(\lambda_{cqkd})}{1+\exp(\lambda_{cqkd})}\,, \tag{45}$$

and from (43) the value of $\tau_{cq}(k)$ is expressed as

$$\tau_{cq}(k) = \exp(\lambda_{cqk} - \xi_{cq}(k))\exp(\zeta_{cq})\,, \tag{46}$$

where $\xi_{cq}(k)$ is defined as in (17) using the values of $\{p_{cqkd}\}$ defined in (45). Although $\exp(\zeta_{cq})$ is still unknown, recall that it was introduced to tackle

the normalization constraint in (44), and then its value is worked out by replacing (46) in (44)

$$\exp(\zeta_{cq}) = \frac{1}{\sum_{k'=1}^{K_{cq}} \exp\left(\lambda_{cqk'} - \xi_{cq}(k')\right)}\,. \tag{47}$$

Finally, the exact value of $\tau_{cq}(k)$ is obtained by plugging (47) into (46)

$$\tau_{cq}(k) = \frac{\exp\left(\lambda_{cqk} - \xi_{cq}(k)\right)}{\sum_{k'=1}^{K_{cq}} \exp\left(\lambda_{cqk'} - \xi_{cq}(k')\right)}\,. \tag{48}$$

Now we focus on transforming $\exp(-\sum_{l,t} \zeta_{s_l q_t}) h_\lambda(\mathbf{i}, \mathbf{q}; S)$ from (41) into $p_\theta(\mathbf{i}, \mathbf{q} \mid S)$ as defined in (11). This part of the proof is similar in conception to the proof given in Heigold et al. (2008b). Firstly we define a global transition matrix $\mathcal{Q}$ as follows

$$\mathcal{Q}_{ij} = \begin{cases} \exp\left(\bar{\lambda}_{cqq'}\right) & i = f(c,q) \text{ and } j = f(c,q') \\ 1 & i = f(c,F) \\ 0 & \text{otherwise} \end{cases}\,, \tag{49}$$

where $f : \mathbb{N}^2 \mapsto \mathbb{N}$ is an injective function that maps each pair composed by a character and state, into a global index or state

$$f(c,q) = \begin{cases} B_c & q = I \\ B_c + q & 1 \leq q \leq M_c \\ B_c + M_c + 1 & q = F \end{cases}\,, \tag{50}$$

with M_c being the number of states for the symbol c, and $B_c = 1 + \sum_{n=1}^{c-1}(2 + M_n)$ being the number of preceding states to the first state of symbol c plus 1.

Since all the values of $\mathcal{Q}$ are not negative, accordingly to *Perron-Frobenius theorem*(Rao and Rao, 1998, p.473), the largest eigenvalue of $\mathcal{Q}$, ψ, is positive and unique. Furthermore, the eigenvector associated to the largest eigenvalue, $\mathbf{v}$, has only positive coefficients, and obviously because of the eigenvector definition, $\mathbf{v}$ satisfies

$$\sum_j \mathcal{Q}_{ij} v_j = \psi v_i, \quad \forall i = 1, \dots \tag{51}$$

Now, the transition generative parameters are defined as

$$a_c(q, q') = \frac{\mathcal{Q}_{f(c,q)f(c,q')} v_{f(c,q')}}{\psi v_{f(c,q)}} = \frac{\exp\left(\bar{\lambda}_{cqq'}\right) v_{f(c,q')}}{\psi v_{f(c,q)}}, \tag{52}$$

where $a_c(q, q')$ verifies the normalization constraint (7) because of (51). These parameters yield a probability proportional to that of (41) when used in (11) as the generative parameters of $p_\theta(\mathbf{i}, \mathbf{q} \mid S)$ (see Appendix A),

$$p_\theta(\mathbf{i}, \mathbf{q} \mid S) = \frac{1}{\psi^{T+L}} \Big[\prod_l \frac{v_{f(s_l,F)}}{v_{f(s_l,I)}} \Big] \frac{h_\lambda(\mathbf{i}, \mathbf{q}; S)}{\exp(\sum_{l,t} \zeta_{s_l q_t})}, \tag{53}$$

which is the equivalence we need but for the term $\frac{1}{\psi^{T+L}} \prod_l \frac{v_{f(s_l,F)}}{v_{f(s_l,I)}}$.

We can introduce this constant factor by multiplying and dividing (37) by it. The division is used in this part whereas the multiplication is added to the first term as follows

$$h_\lambda(O, S, \mathbf{i}, \mathbf{q}, \mathbf{k}) = \\ \psi^T \cdot \exp(\bar{\lambda}_S) \cdot \frac{h_\lambda(\mathbf{i}, \mathbf{q}; S)}{\psi^{T+L} \exp(\sum_{l,t} \zeta_{s_l q_t})} \Big[\prod_l \frac{v_{f(s_l,F)}}{v_{f(s_l,I)}} \Big] \cdot \exp(\sum_{l,t} \zeta_{s_l q_t}) h_\lambda(O, \mathbf{k}; \mathbf{i}, \mathbf{q}, S) \tag{54}$$

with $\bar{\lambda}_S = \lambda_S + L \log \psi + \sum_{l=1}^{L} \log \frac{v_{f(s_l,I)}}{v_{f(s_l,F)}}$.

Finally, the last part of the proof consists in the transformation of $\exp(\bar{\lambda}_S)$ into the word prior probabilities π_S. Similarly to the case of mixture coefficients, we multiply and divide the numerator of (37) by an unknown constant, $\exp(\zeta)$. Since the constant $\exp(\zeta)$ is independent of the word S, it can be introduced into the right-hand side of (34). This constant is grouped together with $\exp(\bar{\lambda})$ as follows

$$\exp(\bar{\lambda}_S + \zeta) \tag{55}$$

Thus, taking into account (55) and the constraints (7), we have that following equalities must hold

$$\bar{\lambda}_S + \zeta = \log \pi_S \tag{56}$$

$$\sum_{S \in W} \pi_S = 1 \tag{57}$$

and the solution is found by following a similar procedure to that of the mixture coefficients

$$\pi_S = \frac{\exp(\bar{\lambda}_S)}{\sum_{S' \in W} \exp(\bar{\lambda}_{S'})} \,. \tag{58}$$

In summary, we have proven that for a given set of discriminative parameters λ, a set of generative parameters can be defined, θ, by (45), (48), (52),

and (58); such that

$$
\begin{aligned}
\arg\max_S p_\lambda(S \mid O) &= \arg\max_S \frac{\mathcal{Z}_\lambda(O)}{\exp(\zeta)} \exp(\zeta) p_\lambda(S \mid O) \\
&= \arg\max_S \exp(\zeta) \mathcal{Z}_\lambda(O) p_\lambda(S \mid O) \\
&= \arg\max_S \sum_{\mathbf{q},\mathbf{i},\mathbf{k}} \exp(\zeta) h_\lambda(O, S, \mathbf{i}, \mathbf{q}, \mathbf{k}) \\
&= \arg\max_S \sum_{\mathbf{q},\mathbf{i},\mathbf{k}} \psi^T \exp(\bar{\lambda}_S + \zeta) \cdot \frac{h_\lambda(\mathbf{i}, \mathbf{q}; S)}{\psi^{T+L} \exp(\sum_{l,t} \zeta_{s_l q_t})} \Big[\prod_{l=1}^{L} \frac{v_{f(s_l,F)}}{v_{f(s_l,I)}} \Big] \\
&\qquad \cdot \exp(\sum_{l,t} \zeta_{s_l q_t}) h_\lambda(O, \mathbf{k}; \mathbf{i}, \mathbf{q}, S) \\
&\Rightarrow \arg\max_S \sum_{\mathbf{q},\mathbf{i},\mathbf{k}} \psi^T \cdot \pi_S \cdot p_\theta(\mathbf{i}, \mathbf{q} \mid S) \cdot p_\theta(O, \mathbf{k} \mid \mathbf{i}, \mathbf{q}, S) \\
&= \arg\max_S \psi^T \sum_{\mathbf{q},\mathbf{i},\mathbf{k}} p_\theta(O, S, \mathbf{i}, \mathbf{q}, \mathbf{k}) = \arg\max_S p_\theta(O, S) \,.
\end{aligned}
\tag{59}
$$

where by $\Rightarrow$ we highlight the step of the proof that is not symmetric.

5. LLHMM Parameter Estimation

In contrast to generative models as BHMMs, in which parameter estimation is usually carried out using the MLE criterion, there is not a unique widespread criterion to find the optimal parameters for a class posterior discriminative model such as the LLHMM proposed in this paper. Perhaps, the most well-known criteria for discriminative parameter estimation is the *maximum mutual information (MMI)*. Given a collection of samples $\{(O_1, S_1), \ldots, (O_N, S_N)\}$, the MMI criterion is defined as follows

$$
F_{\mathrm{MMI}}(\lambda) = \sum_{n=1}^{N} \log\left(p_\lambda(S_n \mid O_n)\right). \tag{60}
$$

The optimal discriminative parameters, λ^*, are those that maximize F_{MMI}.

There are several algorithms for obtaining the parameters that maximize (60) (Heigold et al., 2008a), but commonly the *Resilient back-propagation (RPROP) algorithm* (Riedmiller and Braun, 1993) is used (Giménez et al., 2011). The RPROP requires the computation of the gradient sign, for each parameter λ_m. The gradient of F_{MMI} is given by

$$\frac{\partial F_{\text{MMI}}(\lambda)}{\partial \lambda_m} = N_m(\lambda) - Q_m(\lambda) \tag{61}$$

where $N_m(\lambda)$ and $Q_m(\lambda)$ are expected counts defined as follows

$$N_m(\lambda) = \sum_{n=1}^{N} N_{n\,m}(\lambda), \qquad Q_m(\lambda) = \sum_{n=1}^{N} Q_{n\,m}(\lambda)\,, \tag{62}$$

with $N_{n\,m}(\lambda)$ and $Q_{n\,m}(\lambda)$ being the expected latent and class counts respectively. These counts are defined as follows

$$N_{n\,m}(\lambda) = \sum_{\mathbf{i},\mathbf{q},\mathbf{k}} p_\lambda(\mathbf{i},\mathbf{q},\mathbf{k} \mid O_n, S_n) f_m(O_n, S_n, \mathbf{i},\mathbf{q},\mathbf{k})\,, \tag{63}$$

and

$$Q_{n\,m}(\lambda) = \sum_{\mathbf{i},\mathbf{q},\mathbf{k}} \sum_{S} p_\lambda(S,\mathbf{i},\mathbf{q},\mathbf{k} \mid O_n) f_m(O_n, S, \mathbf{i},\mathbf{q},\mathbf{k})\,. \tag{64}$$

The probabilities $p_\lambda(\mathbf{i},\mathbf{q},\mathbf{k} \mid O, S)$ and $p_\lambda(S,\mathbf{i},\mathbf{q},\mathbf{k} \mid O)$ are computed as follows

$$p_\lambda(\mathbf{i},\mathbf{q},\mathbf{k} \mid O, S) = \frac{\exp(\lambda' \mathbf{f}(O, S, \mathbf{i},\mathbf{q},\mathbf{k}))}{\mathcal{Z}_\lambda(O, S)}\,, \tag{65}$$

and

$$p_\lambda(S,\mathbf{i},\mathbf{q},\mathbf{k} \mid O) = p_\lambda(S \mid O) p_\lambda(\mathbf{i},\mathbf{q},\mathbf{k} \mid O, S) = \frac{\exp(\lambda' \mathbf{f}(O, S, \mathbf{i},\mathbf{q},\mathbf{k}))}{\mathcal{Z}_\lambda(O)}\,. \tag{66}$$

Finally, $\mathcal{Z}_\lambda(O)$ is the normalization constant for the model defined in (29) whereas $\mathcal{Z}_\lambda(O,S)$ is a joint normalization constant for the output and the word, which is likewise defined as follows

$$\mathcal{Z}_\lambda(O,S) = \sum_{\mathbf{i},\mathbf{q},\mathbf{k}} \exp(\lambda' \mathbf{f}(O,S,\mathbf{i},\mathbf{q},\mathbf{k}))\,. \tag{67}$$

The RPROP algorithm computes the sign of the gradient with the aid of these expected counts, and then, modifies the current parameters $\lambda^{(k)}$ accordingly, so that a new estimate of the parameters is obtained, $\lambda^{(k+1)}$. The algorithm starts with a rough estimate of the parameters, $\lambda^{(0)}$, and it ends when either a maximum number of iterations have been reached, or the value of the objective function surpass a given threshold.

5.1. γ-MMI Criterion

A modification of the MMI criterion (60), the so-called γ-*MMI* criterion, leads to better performance (Schluter and Macherey, 1998; Povey, 2003). The γ-MMI is defined by introducing a scaling factor γ into the MMI criterion as follows

$$F_{\gamma\text{-MMI}}(\lambda) = \frac{1}{\gamma} \sum_{n=1}^{N} \log(p_{\lambda\,\gamma}(S_n \mid O_n))\,, \tag{68}$$

with $p_{\lambda\,\gamma}(S \mid O)$ defined as follows

$$p_{\lambda\,\gamma}(S \mid O) = \frac{\left[\mathcal{Z}_\lambda(O,S)\right]^\gamma}{\sum_R \left[\mathcal{Z}_\lambda(O,R)\right]^\gamma}\,. \tag{69}$$

The basic idea is to scale the likelihoods for each word in order to make the best words to compete one against the others even if the differences in probability are large. However, the reason why this idea outperforms the standard MMI is unclear. A possible hypothesis that we support is that it addresses some numerical problems related to the machine precision.

The gradient for the γ-MMI criterion in (68) is analogous to (61) but instead of using $Q_{n\,m}(\lambda)$, we now use $Q^{\gamma}_{n\,m}(\lambda)$ which is defined as follows

$$Q^{\gamma}_{n\,m}(\lambda) = \sum_{\mathbf{i},\mathbf{q},\mathbf{k}} \sum_{S} p_{\lambda\,\gamma}(S, \mathbf{i}, \mathbf{q}, \mathbf{k} \mid O_n) f_m(O_n, S, \mathbf{i}, \mathbf{q}, \mathbf{k}) \,, \tag{70}$$

with the probability $p_{\lambda\,\gamma}(S, \mathbf{i}, \mathbf{q}, \mathbf{k} \mid O)$ defined as

$$p_{\lambda\,\gamma}(S, \mathbf{i}, \mathbf{q}, \mathbf{k} \mid O) = p_{\lambda\,\gamma}(S \mid O) p_{\lambda}(\mathbf{i}, \mathbf{q}, \mathbf{k} \mid S, O) \,, \tag{71}$$

where the probabilities $p_{\lambda\,\gamma}(S \mid O)$ and $p_{\lambda}(\mathbf{i}, \mathbf{q}, \mathbf{k} \mid S, O)$ are defined in (69) and (65), respectively.

Fig. 2, summarizes the main idea behind the γ-MMI training criterion. It depicts the differences between the most probable word an the second most probable competitor for a LLHMM model (more details in Sec. 6). It is observed that these differences are of 44 points (in logarithmic scale) at the beginning, which corresponds to MLE. Additionally, the training sample is incorrectly classified at the first training iterations. Although, after 50 iterations all the γ values correctly classify the sample; smaller values of γ induce larger difference between the correct class and its competitors.

5.2. Regularization

A common undesired property of all the proposed discriminative criteria is that they easily overfit the parameters. Even criteria specially designed to avoid outlayers such as the power criterion suffer from overfitting. Since there is no clear way to smooth discriminatively trained models. A typical patch is to add a regularization term to the criterion itself

$$F_{C*}(\lambda) = F_{*}(\lambda) - \frac{C}{2} \sum_{m} (\lambda^{(0)}_{m} - \lambda_m)^2 \,, \tag{72}$$

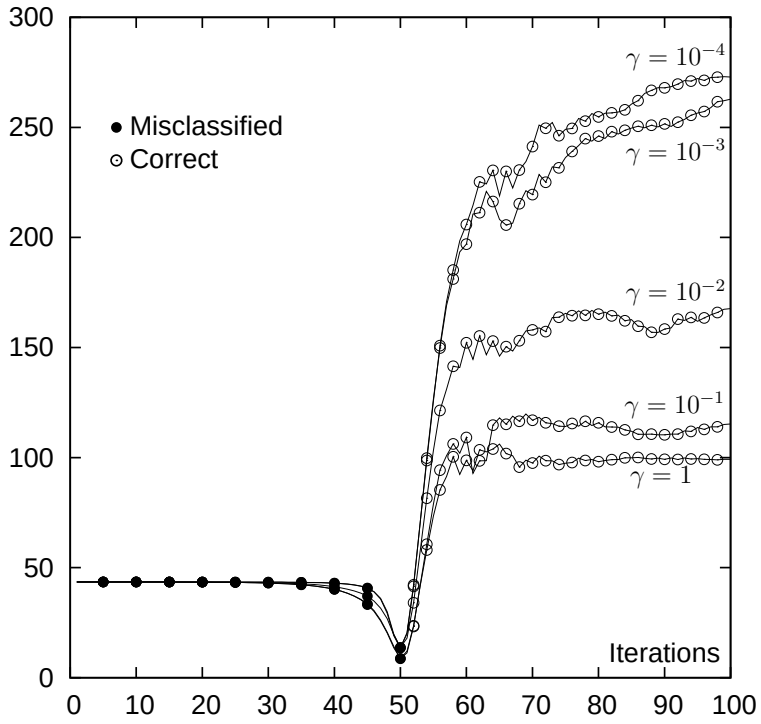


Figure 2: Differences (in logarithmic scale) between the most probable and the second most probable word for a given training sample (*cette*). Several values of the γ-MMI criterion are plotted. The most probable word changes at iteration 50 becoming the correct word.

with $F_*(\lambda)$ denoting the original criterion, namely F_{MMI} or F_γ; and $\lambda^{(0)}$ being either a reliable estimation of the parameters or simply $\mathbf{0}$.

The inclusion of the regularization term, only modifies the gradient in the following form

$$\frac{\partial F_{C*}(\lambda)}{\partial \lambda_m} = \frac{\partial F_*(\lambda)}{\partial \lambda_m} + C(\lambda_m^{(0)} - \lambda_m) = N_m(\lambda) - Q_m(\lambda) + C(\lambda_m^{(0)} - \lambda_m)\,, \quad (73)$$

where the expected counts, N_m and Q_m, are calculated as in the original criterion.

6. Experiments

In this section, we perform several experiments on the RIMES database of handwritten French letters (Grosicki et al., 2009), so that the performance of several discriminative training criteria for BHMM is assessed with respect to the generative training. Furthermore, we visually inspect several discriminative parameters by transforming them into their generative counterpart.

6.1. The RIMES Database

All the experiments were carried out on the protocol WR2 used in the handwritten word recognition competition of the ICDAR 2009. This protocol comprises 51 738 and 7 464 samples for training and testing, respectively. The lexicon used during the recognition comprises the words that occur in the test (1 612 words) and an alphabet of 81 characters. A three step preprocess was applied to all input images (Pastor i Gadea, 2007): gray level normalization, deslanting, and vertical size normalization.

Preprocessed images were first scaled in height to 30 pixels maintaining the aspect ratio, and then binarized with the Otsu's method. A sliding

window of width 9 was applied centered on each column in order to extract a sequence of 270-dimensional binary feature vectors. More precisely, for each column the sliding window was horizontally centered, and then vertically repositioned so that the center of the window is aligned with mass center of the window before repositioning. Once realigned, the 9 binary vectors of the window were concatenated in order to compose a binary feature vector of dimension 270 which is fed into the BHMM or LLHMM as input.

6.2. Experimental Setup

In order to properly initialize the MMI training scheme the LLHMM was initialized with a BHMM classifier trained with the EM (Baum-Welch) algorithm (Rabiner and Juang, 1993), using the training scheme described in Giménez et al. (2010). The best generative BHMM, which is composed by $Q = 8$ states per character and $K = 64$ mixture components per state, obtains an error of 21.2%.

Regarding to the discriminative training, the RPROP algorithm was used for optimizing the criteria. The initial discriminative parameters were obtained transforming the generative parameters of a BHMMs with $Q = 8$ states per character and $K = 26$ mixture components per state. Despite the best generative results is obtained with $K = 64$ mixture components per state, some works reported (Giménez et al., 2011) that the best classifier obtained using MMI training has half (0.4 ratio) the number of mixture component per state than its generative counterpart. Consequently, in preliminary experiments we checked that the results obtained using the conventional MMI criterion with $K = 26$ are similar or better to those obtained increasing the value of K.

Finally, the proposed discriminative training criteria require to compute sums over all the words for calculating several values such as $\mathcal{Z}_\lambda(O)$ in (66). Consequently discriminative training algorithms become unfeasible in a straight implementation. For this reason we have approximated the sums over all the words by a beam pruning strategy together with a histogram pruning up to 100 best hypothesis accordingly to $p(S \mid O)$.

6.3. Experiments

Firstly, we wanted to assess the repercussions of the regularization term in the conventional MMI criterion. For doing so, we scanned several values of the regularization parameter $C = \{0, 0.1, 1, 10, 100\}$ as introduced in (72), where $C = 0$ stands for not using the regularization at all. In Fig. 3, the classification error rate (CER) as a function of the number of RPROP iterations is plotted for different regularization values. In all cases, even with standard MMI, the CER decreases in a similar way, until iteration 60, where the best result is obtained. At this point, the behavior diverges depending on the precise value of C. If no regularization is applied ($C = 0$) the error becomes unstable and increases (overfits) as the training iterates. However, the larger the regularization parameter is, the less overtrained the model becomes, until that for $C = 10$ the error becomes stable while providing similar performance to that at iteration 60. Note that if the regularization parameter is further increased, an slight drop in performance is observed. As expected, the regularization term reduces the overfitting problem.

Fig. 3 also shows that the regularized MMI criterion obtains a CER around 19.5% using only $K = 26$ components per state. If we compare it with the best generative result, which is 21.2% and is obtained using $K = 64$,

we observe not only an improvement of 1.7 absolute points but also a reduction on the number of parameters of 0.4, i.e. less than half the parameters are needed for such improvement.

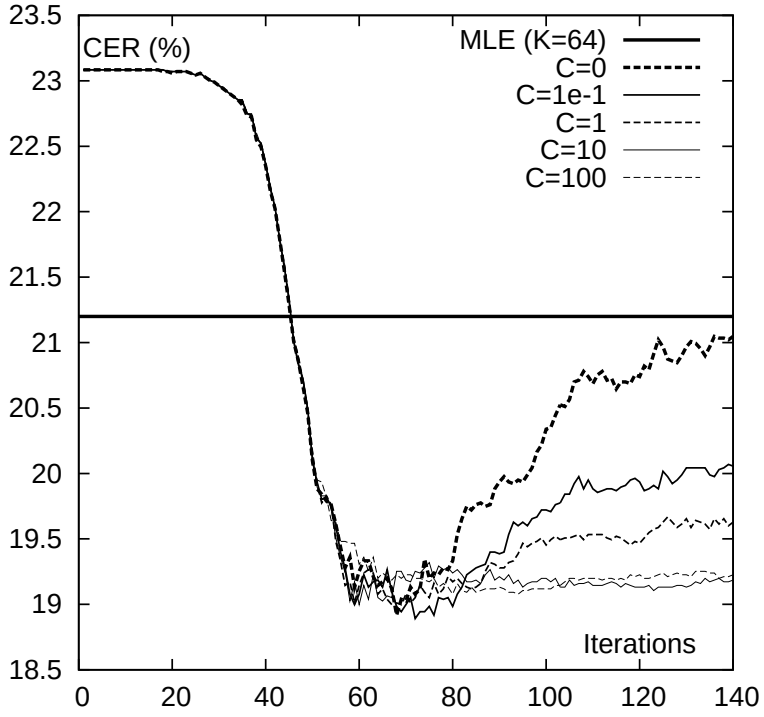


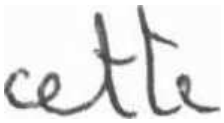
Figure 3: Classification error (in %) as a function of RPROP iterations for the regularized MMI criterion with several values of the regularization parameter C. Note that $C = 0$ stands for the non-regularized MMI.

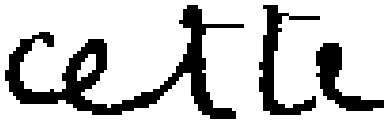
For a deeper understanding of the MMI criterion, in Fig. 4 we depict the top 5 most probable words as a function of the training iterations (0,50,55,60,100) and the γ value of the γ-MMI criterion (1-MMI=MMI). We selected a common example in which the MLE misclassifies the sample and the MMI learns how to discriminate it among the the other competitors. More precisely, for several training iterations the 5 most probable transcriptions are shown. In

addition, for each transcription the difference (in logarithmic scale) between its score and the best score at that iteration is also shown. As expected, the correct transcription (*cette*) gains relevance with the iterations, that is, the training algorithm is modifying the model parameters in order to better classify the sample. In particular, at the beginning there is a separation of 44 points between *cette* and the best transcription (*celle*). However, at some point near to iteration 50 this situation is reverted, and from this point on the score difference keeps increasing (see Fig. 2). A total of 60.3% of the training samples that are misclassified by the MLE, are correctly classified at the end of the last MMI iteration. In contrast, only 1.3% of the correctly classified samples by the MLE are misclassified at the end of the training process.

In Fig. 6, we explored several values of γ, ranging from standard MMI ($\gamma = 2$) to 10^{-4}, using the best regularization term obtained in the previous experiment $C = 10$. In the previous experiment the 100-best words were recalculated every 10 iterations, however, with the γ-MMI we observed a severe overfitting. Consequently, we repeated the experiments recalculating the best words every iteration. It is observed in Fig. 6 that the modified γ-MMI obtains a very competitive performance in terms of CER (15%) if applied properly. If we compare the best result in Fig. 6 with the best generative result, the former obtains an improvement boost of more than 6 absolute points with respect to the latter. In Fig. 4 the behavior of the γ-MMI can be checked for a training sample. As we can see, the use of small values of γ leads to an increase of the separation between classes, which is consistent with the idea that the γ is increasing the competition between classes during

Original image **Preprocessed without sliding window**





	MLE		$\gamma-$**MMI** ($\gamma = 1$)							
Iterations	**1**		**50**		**55**		**60**		**100**	
1-best	-	celle	-	celle	-	**cette**	-	**cette**	-	**cette**
2-best	44	**cette**	6	**cette**	73	celle	72	celle	100	celle
3-best	447	Cette	376	dette	342	dette	359	dette	354	dette
4-best	467	dette	406	Cette	382	Cette	388	Cette	393	Cette
5-best	499	celles	497	celles	564	celles	542	celles	485	geste

	MLE		$\gamma-$**MMI** ($\gamma = 10^{-1}$)							
Iterations	**1**		**50**		**55**		**60**		**100**	
1-best	-	celle	-	celle	-	**cette**	-	**cette**	-	**cette**
2-best	44	**cette**	7	**cette**	79	celle	94	celle	116	celle
3-best	447	Cette	375	dette	340	dette	358	dette	336	dette
4-best	467	dette	409	Cette	385	Cette	403	Cette	393	Cette
5-best	499	celles	497	celles	570	celles	556	geste	459	geste

	MLE		$\gamma-$**MMI** ($\gamma = 10^{-3}$)							
Iterations	**1**		**50**		**55**		**60**		**100**	
1-best	-	celle	-	celle	-	**cette**	-	**cette**	-	**cette**
2-best	44	**cette**	6	**cette**	128	celle	207	celle	260	celle
3-best	447	Cette	403	dette	413	dette	417	dette	440	dette
4-best	467	dette	516	celles	663	Cette	631	Cette	657	Cette
5-best	499	celles	530	Cette	667	celles	784	celles	828	telle

Figure 4: γ-MMI behavior on a training sample for several values of γ. The figures stand for the difference (in logarithmic scale) between each n-best word and the best transcription at each iteration. Bold words highlight the position of the correct word *cette*.

the training process. For example, at iteration 100 the separation between the two best hypothesis using $\gamma = 1$ is 100 points, while using $\gamma = 10^{-3}$ the separation increases up to 260 points.

Fig. 5 depicts a similar experimentation to that of Fig. 3 but for several *test samples*. The first sample (*vous*) is a sample that is misclassified by the MLE model and it is correctly classified using γ-MMI criterion. The remaining two samples are correctly classified by the MLE criterion. However, the first one is finally misclassified by the discriminative model, while the second one remains correctly classified. It is worth noting, that these three cases represent the 10.2%, 2.2% and 74.7% of the test set,respectively.

As discussed before, all experiments were carried out using $K = 26$ components per state. In order to better compare the performance of the MLE and γ-MMI criteria we carried out a final experiment, in which both criteria are tested using several components per state $K \in \{1, 4, 16, 64\}$. For the γ-MMI criteria the best parameters from previous experiments were used ($\gamma = 10^{-3}$ and $C = 10$). Results are shown in Fig. 7.

From the results reported in Fig. 7 it is clear that γ-MMI outperforms MLE in all cases. The improvement of the MMI decreases as the number of components increases. For example, the improvement using $K = 1$ is about 20 points while using $K = 64$ is about 5 points. It is worth noting, that the best result in this figure is 15.2% which is achieved using $K = 16$ components and it is very similar to the best result obtained with $K = 26$, which we chose for all the previous experimentation.

Finally, a visual inspection of some Bernoulli prototypes for several training criteria is given in Fig. 8. The Bernoulli prototypes for letters e and s are

Incorrect → Correct

	MLE		γ−**MMI** ($\gamma = 10^{-3}$)							
Iterations	**0**		**40**		**45**		**50**		**100**	
1-best	-	virus	-	virus	-	virus	-	**vous**	-	**vous**
2-best	26	Vous	35	**vous**	15	**vous**	16	virus	89	virus
3-best	44	**vous**	40	Vous	50	Vous	93	Vous	318	Vous
4-best	82	bruits	118	bruits	165	bruits	268	bruits	350	nous
5-best	231	plus	243	plus	261	plus	323	plus	419	viens

Correct → Incorrect

	MLE		γ−**MMI** ($\gamma = 10^{-3}$)							
Iterations	**0**		**40**		**45**		**50**		**100**	
1-best	-	**Suite**	-	**Suite**	-	**Suite**	-	suite	-	suite
2-best	97	suite	77	suite	50	suite	16	**Suite**	247	**Suite**
3-best	357	Socit	376	Socit	395	seule	336	seule	465	seule
4-best	405	socit	413	socit	402	Socit	457	socit	698	sant
5-best	428	Sant	431	seule	425	socit	470	Socit	702	suis

Correct → Correct

	MLE		γ−**MMI** ($\gamma = 10^{-3}$)							
Iterations	**0**		**40**		**45**		**50**		**100**	
1-best	-	**que**	-	**que**	-	**que**	-	**que**	-	**que**
2-best	239	due	233	due	221	due	188	due	167	due
3-best	350	dire	371	dire	396	dire	438	dire	753	dire
4-best	539	grise	590	grise	620	date	628	date	810	date
5-best	595	avez	611	date	639	avez	659	d'une	930	quel

Figure 5: γ-MMI behavior on three selected *test* samples. The figures stand for the difference (in logarithmic scale) between each n-best word and the best transcription at each iteration. Bold words highlight the position of the correct word *cette*.

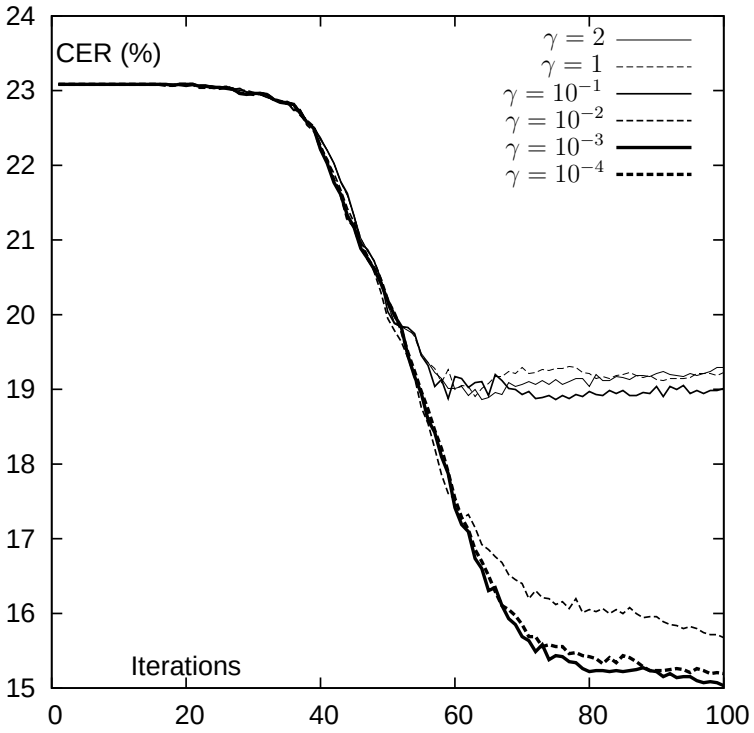


Figure 6: Classification error (in %) as a function of RPROP iterations for the modified γ-MMI criterion with regularization $C = 10$ and several values of γ. Note that $\gamma = 1$ corresponds to the standard MMI criterion.

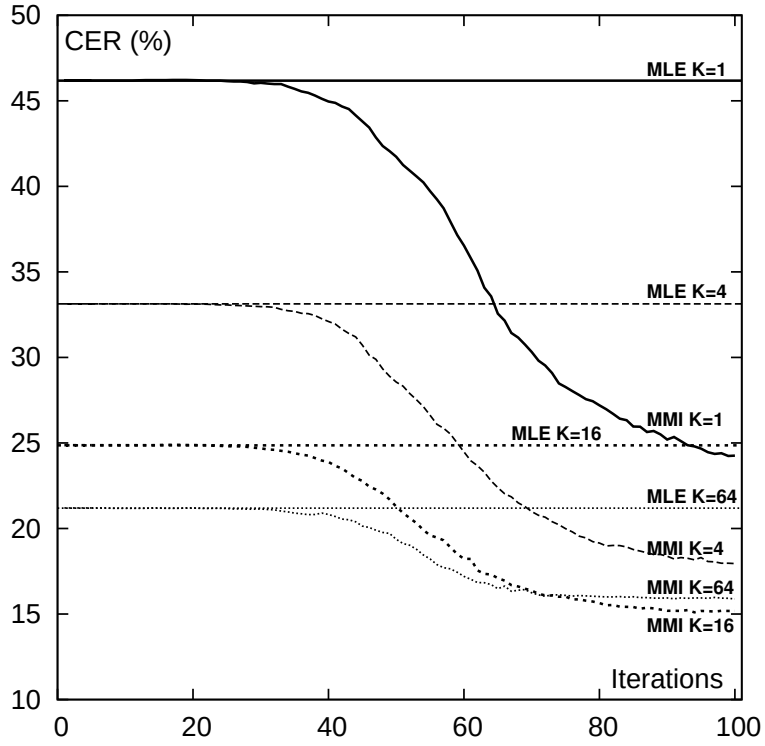


Figure 7: MLE and γ-MMI ($\gamma = 10^{-3}C = 10$) criteria comparison using several components per state.

shown, where the columns represent states, and rows represent mixture components. Provided that the number of mixture components in each state is large ($K = 26$) we have selected the 4 components with the highest mixture coefficients when trained using the MLE criterion. Prototypes are plotted for 3 different training criteria (from left to right): MLE training; the γ-MMI with $\gamma = 10^{-3}$ and regularization $C = 10$; and the conventional MMI training without regularization. It is worth noting that the MLE prototypes are the initial prototypes for both represented discriminative training criteria.

It is observed that the prototypes without regularization are apparently a noise version of the MLE prototypes, however we know that they have a better performance when classifying. A further observation reveals that discriminative training is focused on modifying those pixels that discriminate the most while keeping the remaining pixels unmodified. These unmodified pixels are those that keep the same state (0 or 1) for many words. When no regularization is employed, a pixel that discriminates a single training sample can be set to 1, however, those spurious pixels are eliminated by adding the regularization term.

7. Concluding Remarks and Future Work

In this work, we presented a log-linear HMM (LLHMM) to recognize isolated handwritten words that directly deals with binarized images without the need of a sophisticated feature extraction process. This model has been proved to be equivalent to Bernoulli HMMs (BHMMs), and in this way, we have provided a framework for discriminatively training BHMMs. Furthermore, this allows us to visually inspect and understand discriminative

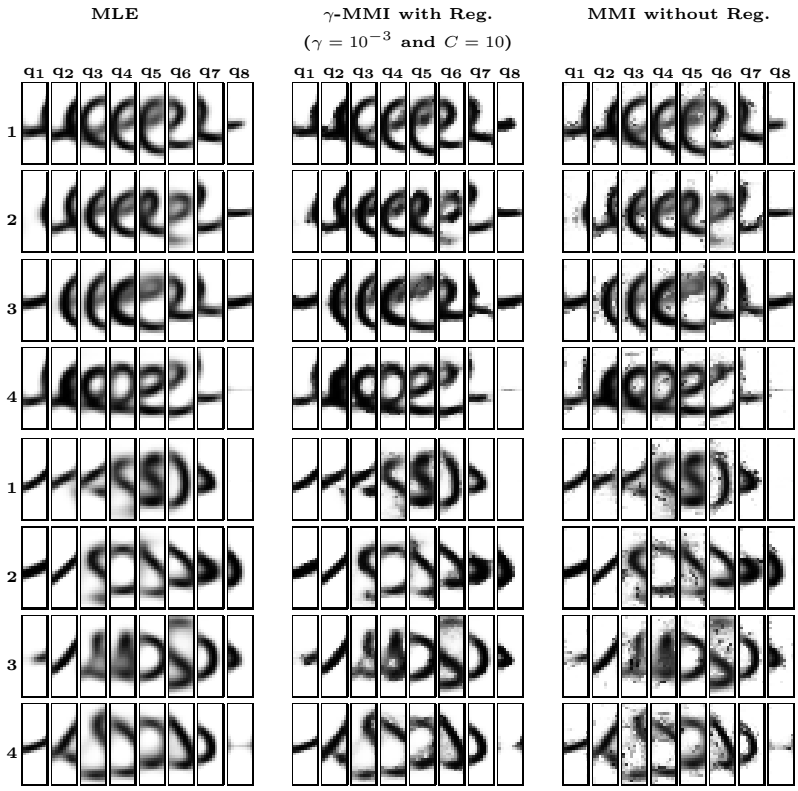


Figure 8: Bernoulli prototypes of letters **e** and **s** using three different training criteria (from left to right): MLE, γ-MMI with regularization and MMI without regularization.

parameters by transforming them into generative ones.

Two discriminative training criteria have also been analyzed for the LLHMM model: conventional MMI and γ-MMI. We tried all of them discussing problems (over-fitting, computational cost) and some typical approximation to those problems (regularization term, pruning techniques). All these methods have been tested over the well-known RIMES database of handwritten French words. Furthermore, in all cases discriminative training clearly outperformed the conventional MLE training. In particular, very competitive results were obtained using the γ-MMI training scheme which obtained nearly 15% of CER, or in other words an improvement of more than 6% of absolute points with respect to the generative counterpart. However, there are many more discriminative training criteria such as margin-based or minimum phoneme error. As future work we plan to implement and adapt these discriminative criteria to the proposed model.

The best result obtained in this work on the considered task of the RIMES database is 15%, which to our knowledge is the best result reported using HMMs and without system combination. If we compare our system with the results of the ICDAR 2009 (Grosicki and El Abed, 2009), our system would be positioned in the third position and very close to the second system (13.9%), which is in fact a combination of hybrid HMM/MLP systems, and far from the first system (6.8), which is a system based on a hierarchy of multidimensional recurrent neural networks, and has shown to be extremely competitive in this task. Moreover, if we compare our result with the results reported on the same task in Bianne-Bernard et al. (2011), it is observed that our system outperforms the results of the three systems presented on that

paper: a dynamic context-independent system based on HMMs (24.5%), a dynamic context-dependent system based on HMMs (19.6%), and a hybrid HMM/neural network system (20.5%). However, when the three systems are combined an error of 10.9% is obtained. Consequently, as future work we plan to combine the proposed discriminative BHMMs system, with the conventional generative BHMMs system and other state of the art systems, as for instance those based on recurrent neural networks (Graves and Schmidhuber), in order to measure the impact of discriminative BHMMs when combined with other systems.

Finally, we intend to extend all the work developed in this paper to continuous HTR, that is, a discriminative BHMM in which the words are replaced by word sequences, and hence, the prior probabilities are replaced by a language model.

Acknowledgment

Work supported by the EC (FEDER/FSE) and the Spanish MEC/MICINN under the MIPRCV "Consolider Ingenio 2010" program (CSD2007-00018), iTrans2 (TIN2009-14511) and MITTRAL (TIN2009-14633-C03-01) projects. Also supported by the IST Programme of the European Community, under the PASCAL2 Network of Excellence, IST-2007-216886, and by the Spanish MITyC under the erudito.com (TSI-020110-2009-439).

Appendix A. Discriminative to generative transition probabilities

In this appendix, we prove that the parameters in (52) yield a probability proportional to that of (41) when used in (11) as the generative parameters

of $p_\theta(\mathbf{q}, \mathbf{i} \mid S)$. In order to clarify this, we plug the parameters as computed in (52) into (11) yielding

$$\prod_{l=1}^{L} \Big[\frac{\exp\left(\bar{\lambda}_{s_l I q_{i_l}}\right) v_{f(s_l,q_{i_l})}}{\psi v_{f(s_l,I)}} \cdot \prod_{t=i_l}^{i_{l+1}-2} \frac{\exp\left(\bar{\lambda}_{s_l q_t q_{t+1}}\right) v_{f(s_l,q_{t+1})}}{\psi v_{f(s_l,q_t)}} \cdot \frac{\exp\left(\bar{\lambda}_{s_l q_{i_{l+1}-1} F}\right) v_{f(s_l,F)}}{\psi v_{f(s_l,q_{i_{l+1}-1})}} \Big], \tag{A.1}$$

where by grouping elements we get

$$\frac{1}{\psi^{T+L}} \frac{h_\lambda(\mathbf{i}, \mathbf{q}; S)}{\exp(\sum_{l,t} \zeta_{s_l q_t})} \prod_{l=1}^{L} \Big[\frac{v_{f(s_l,F)}}{v_{f(s_l,I)}} \frac{v_{f(s_l,q_{i_l})}}{v_{f(s_l,q_{i_{l+1}-1})}} \prod_{t=i_l}^{i_{l+1}-2} \frac{v_{f(s_l,q_{t+1})}}{v_{f(s_l,q_t)}} \Big]. \tag{A.2}$$

Note that, in each segment l the telescope product over $\frac{v_{j'}}{v_j}$ is equal to $\frac{v_{f(s_l,F)}}{v_{f(s_l,I)}} \cdot 1$, and then equation (A.2) is reduced to

$$p_\theta(\mathbf{i}, \mathbf{q} \mid S) = \frac{1}{\psi^{T+L}} \Big[\prod_{l=1}^{L} \frac{v_{f(s_l,F)}}{v_{f(s_l,I)}} \Big] \frac{h_\lambda(\mathbf{i}, \mathbf{q}; S)}{\exp(\sum_{l,t} \zeta_{s_l q_t})}. \tag{A.3}$$

References

Bianne-Bernard, A.L., Menasri, F., Al-Hajj Mohamad, R., Mokbel, C., Kermorvant, C., Likforman-Sulem, L., 2011. Dynamic and contextual information in hmm modeling for handwritten word recognition. Pattern Analysis and Machine Intelligence, IEEE Transactions on 33, 2066 –2080.

Dempster, A.P., Laird, N.M., Rubin, D.B., 1977. Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society. Series B (Methodological) 39, 1–38.

Duda, R.O., Hart, P.E., 1973. Pattern Classification and Scene Analysis. J. Wiley and Sons.

Giménez, A., Alkhoury, I., Juan, A., 2010. Windowed Bernoulli Mixture HMMs for Arabic Handwritten Word Recognition, in: ICFHR' 10, Kolkata (India). pp. 533–538.

Giménez, A., Andrés-Ferrer, J., Juan, A., Serrano, N., 2011. Discriminative Bernoulli Mixture Models for Handwritten Digit Recognition, in: ICDAR' 11, Beijing (China). pp. 558–562.

Giménez, A., Juan, A., 2009. Embedded Bernoulli Mixture HMMs for Handwritten Word Recognition, in: ICDAR' 09, Barcelona (Spain). pp. 896–900.

Graves, A., Schmidhuber, J., . Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks.

Grosicki, E., Carree, M., Brodin, J.M., Geoffrois, E., 2009. Results of the RIMES Evaluation Campaign for Handwritten Mail Processing, in: ICDAR' 09, Barcelona(Spain). pp. 941–945.

Grosicki, E., El Abed, H., 2009. ICDAR 2009 Handwriting Recognition Competition, in: ICDAR '09, Barcelona (Spain). pp. 1398–1402.

Heigold, G., Deselaers, T., Schlüter, R., Ney, H., 2008a. A GIS-like training algorithm for log-linear models with hidden variables, in: ICASSP' 08, Las Vegas (USA). pp. 4045–4048.

Heigold, G., Lehnen, P., Schluter, R., Ney, H., 2008b. On the equivalence of Gaussian and log-linear HMMs, in: INTERSPEECH' 08, Brisbane (Australia). pp. 273–276.

Märgner, V., El Abed, H., 2010. ICFHR 2010 - Arabic Handwriting Recognition Competition, in: ICFHR' 10, Kolkata (India). pp. 709–714.

Pastor i Gadea, M., 2007. Aportaciones al reconocimiento automático de texto manuscrito. Ph.D. thesis. Dep. de Sistemes Informàtics i Computació. València, Spain.

Povey, D., 2003. Discriminative Training for Large Vocabulary Speech Recognition. Ph.D. thesis. Cambridge University Engineering Dept.

Rabiner, L., Juang, B.H., 1993. Fundamentals of speech recognition. Prentice-Hall.

Rao, C.R., Rao, M.B., 1998. Matrix algebra and its applications to statistics and econometrics. World Scientific.

Riedmiller, M., Braun, H., 1993. A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm, in: IEEE International Conference on Neural Networks, pp. 586–591.

Schluter, R., Macherey, W., 1998. Comparison of discriminative training criteria, in: ICASSP' 98, pp. 493–496 vol.1.