Discriminative Bernoulli Mixture Models for Handwritten Digit Recognition

Adrià Giménez, J. Andrés-Ferrer, Alfons Juan and Nicolás Serrano
DSIC, Universitat Politècnica de València, Camí de Vera s/n, 46022 València, Spain
{*agimenez,jandres,ajuan,nserrano*}*@dsic.upv.es*

***Abstract*—Bernoulli-based models such as Bernoulli mixtures or Bernoulli HMMs (BHMMs), have been successfully applied to several handwritten text recognition (HTR) tasks which range from character recognition to continuous and isolated handwritten words. All these models belong to the generative model family and, hence, are usually trained by (joint) maximum likelihood estimation (MLE). Despite the good properties of the MLE criterion, there are better training criteria such as maximum mutual information (MMI). The MMI is a widespread criterion that is mainly employed to train discriminative models such as log-linear (or maximum entropy) models. Inspired by the Bernoulli mixture classifier, in this work a log-linear model for binary data is proposed, the so-called mixture of multi-class logistic regression. The proposed model is proved to be equivalent to the Bernoulli mixture classifier. In this way, we give a discriminative training framework for Bernoulli mixture models. The proposed discriminative training framework is applied to a well-known Indian digit recognition task.**

***Keywords*-Bernoulli mixture; discriminative training; MMI; mixture of multi-class logistic regression; log-linear models**

I. Introduction

In the past few years Bernoulli-based models have been proved to be competitive for handwritten text recognition (HTR). HTR usually involve text images which are only composed by black and white colors. Since Bernoulli-based models are very well suited for such cases, binaryzed input images are directly fed into a Bernoulli-based model without the need of a sophisticated feature extraction process. Bernoulli-based models such as Bernoulli mixtures (BM) or HMMs with Bernoulli mixture emission probabilities at the states (BHMMs), have been successfully applied to both isolated character and continuous HTR [1], [2]. In particular, BHMMs have been used for Arabic handwritten Tunisian town names recognition, achieving the first place prize in the Arabic HTR competition organized during the ICFHR 2010 conference [3], [4].

All previously mentioned Bernoulli-based models are known to be generative models. Generative models are classifiers based on the optimal Bayes classifier [5], in which the posterior probability $p(c|\boldsymbol{x})$ is approximated by a joint probability model $p_{\boldsymbol{\theta}}(c, \boldsymbol{x})$ which is parametrized by $\boldsymbol{\theta}$. Generative models have two great advantages among many others. On the one hand, the parameters of the generative models are easily understandable for researchers. For instance, in the Bernoulli-based models, the model parameters can be displayed as grey level images, so that we can know which pixels are more probable than others within a class. On the other hand, generative models are mostly trained with maximum likelihood estimation (MLE) criterion. One of the advantages of this criterion is that there are well-known algorithms for training generative models with hidden variables such as the EM [6].

Despite the good properties of the MLE criterion, it has a very important drawback when used in classification problems. The MLE is aimed at explaining the probability distribution underlying the training sample, that is, maximizing the likelihood of the joint probability function $p_{\boldsymbol{\theta}}(c, \boldsymbol{x})$. However, we are interested in simply classifying samples, and there is no guarantee that the MLE parameters are the most suitable for classifying.

The discriminative models and criteria are aimed at classifying the data without explaining it, and, hence, they directly approximate the posterior class probability by a model $p_{\boldsymbol{\lambda}}(c|\boldsymbol{x})$ which is parametrized by $\boldsymbol{\lambda}$. 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 $p_{\boldsymbol{\lambda}}(c \mid \boldsymbol{x})$. 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 important for discriminative models with hidden variables.

The generalized iterative scaling (GIS) algorithm [7] finds the optimal discriminative parameters accordingly to the MMI criterion for a special family of discriminative models, the so-called *log-linear or maximum entropy models (LLM)*. However, GIS is not suited for LLM with hidden variables. Recently, in [8] a similar algorithm, namely GGIS, has been proposed for training LLM with hidden variables. Due to its very interesting properties, we will pay special attention to the case of mixture of log-linear models [8], which approximate the posterior probability with a set of parameters $\boldsymbol{\lambda}$ as follows

$$p_{\boldsymbol{\lambda}}(c \mid \boldsymbol{x}) = \frac{1}{\mathcal{Z}(\boldsymbol{x})} \sum_{k} \exp(\boldsymbol{\lambda}^{T} \boldsymbol{f}(\boldsymbol{x}, c, k)) \,, \tag{1}$$

where $\boldsymbol{f}$ is a given vector feature functions, and k is a hidden variable. Although the GGIS finds the optimal parameters accordingly to the MMI criterion, a huge amount of iterations are required [8], and the RPROP algorithm [9] is usually employed instead since it obtains similar results while providing faster convergence.

The contributions of this work are enumerated as follows:

1) We propose a particular case of mixture log-linear models for binary data inspired by BMs, a mixture of multi-class logistic regression (MMLR).
2) We prove the equivalence between BMs and MMLRs for binary data.
3) We provide a MMI training scheme for BMs by means of their equivalence with MMLRs.
4) We provide the capability to understand discriminative parameters of the MMLR from a generative perspective by means of their equivalence with BMs.

The remainder of the paper is organized as follows. In Section II a review of BM is given. In Section III, the MMLR classifier is proposed. The Section IV proves equivalence between both classifiers. The experimental performance of the discriminative BMs is gathered in Section V. Finally conclusions and future work are discussed.

II. Bernoulli Mixture Classifier

Given a binary input vector $\boldsymbol{x} \in \{0,1\}^D$ and a class c from the set of classes $\{1,\dots,C\}$; a Bernoulli mixture (BM) classifier is defined as follows

$$c^* = \arg\max_c \; p_{\boldsymbol{\theta}}(\boldsymbol{x}, c) \,, \tag{2}$$

where $p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid c)$ is a mixture model

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}, c) = \sum_{k=1}^{K} p_{\boldsymbol{\theta}}(\boldsymbol{x}, c, k) \,. \tag{3}$$

Each joint probability is decomposed in two terms

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}, c, k) = \pi_{ck} \, p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid c, k) \,, \tag{4}$$

where π_{ck} is the prior coefficient of the k-th component and class c; and where $p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid c, k)$ follows a multivariate Bernoulli probability distribution

$$p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid c, k) = \prod_d p_{ckd}^{x_d} \cdot (1 - p_{ckd})^{(1-x_d)} \,. \tag{5}$$

with p_{ckd} being the probability of the d-th bit to be one in the k-th mixture component of the class c.

All parameters of the BM classifier $\theta = \{\boldsymbol{\pi}; \boldsymbol{p}\}$ are required to be probabilities, and in particular the mixture coefficients are constrained to sum 1, that is

$$\sum_{c,k} \pi_{ck} = 1 \,. \tag{6}$$

Note that the *Bernoulli prototype*, $\boldsymbol{p_{ck}}$, of a particular class c and mixture component k can be visually represented as a vector of gray values, where 1 stands for black and 0 for white.

Given a training set of samples $\{\boldsymbol{x}_n, c_n\}_{n=1}^N$, the parameters θ of a BM are usually estimated by maximizing the likelihood of the joint probability over the training set using the EM algorithm [1].

III. Mixture of Multi-class Logistic Regression

In this section, we propose a mixture of multi-class logistic regression (MMLR) model inspired by the BM classifier. Given a binary input vector $\boldsymbol{x} \in \{0,1\}^D$ and a class $c \in \{1,\dots,C\}$, the proposed MMLR classifier is defined as

$$c^* = \arg\max_c \; p_{\boldsymbol{\lambda}}(c \mid \boldsymbol{x}) \,, \tag{7}$$

where the posterior probability is modeled as

$$p_{\boldsymbol{\lambda}}(c \mid \boldsymbol{x}) = \sum_{k=1}^{K} p_{\boldsymbol{\lambda}}(c, k \mid \boldsymbol{x}) \,, \tag{8}$$

with k denoting the selected mixture component for the current class, analogously to (3). The component and class posterior probability in (8) is modeled as a log-linear combination of binary features $f_i(\boldsymbol{x}, c, k)$,

$$p_{\boldsymbol{\lambda}}(c, k \mid \boldsymbol{x}) = \frac{\exp(\sum_i \lambda_i f_i(\boldsymbol{x}, c, k))}{\mathcal{Z}(\boldsymbol{x})} \,, \tag{9}$$

where $\mathcal{Z}(\boldsymbol{x})$ is the normalization constant defined as

$$\mathcal{Z}(\boldsymbol{x}) = \sum_c \sum_k \exp(\sum_i \lambda_i f_i(\boldsymbol{x}, c, k)) \,. \tag{10}$$

Note that (9) is a multi-class logistic regression model.

Finally, given an index of features $i = (\tilde{c}, \tilde{k}, d)$ where $\tilde{c}$ ranges in the domain $\{1,\dots,C\}$, $\tilde{k}$ in $\{1,\dots,K\}$ and d in $\{0,1,\dots,D\}$; the feature $f_i(\boldsymbol{x}, c, k) = f_{\tilde{c},\tilde{k},d}(\boldsymbol{x}, c, k)$ is defined as follows

$$f_{\tilde{c},\tilde{k},d}(\boldsymbol{x}, c, k) = \begin{cases} \delta(c,\tilde{c})\delta(k,\tilde{k}) & d = 0 \\ \delta(c,\tilde{c})\delta(k,\tilde{k})x_d & 1 \leq d \leq D \end{cases} . \tag{11}$$

The MMI criterion has the disadvantage that easily overfits to the training data. A typically solution to amend this problem is to add a regularization term to the criterion

$$F_{\text{C}}(\boldsymbol{\lambda}) = F_{\text{MMI}}(\boldsymbol{\lambda}) - C \sum_i (\lambda_i^{(0)} - \lambda_i)^2 \,, \tag{12}$$

where $\boldsymbol{\lambda}^{(0)}$ is either a reliable estimation of the parameters or simply $\mathbf{0}$.

IV. Equivalence Between Classifiers

In this section we prove that the BM classifier defined in Section II is *equivalent* to the MMLR model defined in Section III. A generative classifier is said to be *equivalent* to a discriminative classifier if for a given set of generative parameters $\boldsymbol{\theta}$, discriminative parameters, $\boldsymbol{\lambda}$, can be found such that

$$\arg\max_{c}\ p_{\boldsymbol{\theta}}(\boldsymbol{x}, c) = \arg\max_{c}\ p_{\boldsymbol{\lambda}}(c \mid \boldsymbol{x})\,; \tag{13}$$

and vice-versa.

A. From Generative to Discriminative Parameters

Unlike the converse direction, it is quite simple to prove that given a BM classifier it can be re-parameterized into the model proposed in Section III. Left probability in (13) can be rewritten as

$$\sum_{k=1}^{K} \exp(\log \pi_{ck} + \sum_{d=1}^{D} x_d \log p_{ckd} + (1-x_d)\log(1-p_{ckd}))\,. \tag{14}$$

If we group the terms that depend on x_d in the previous equation, then we obtain

$$\textstyle\sum_k \exp(\log \pi_{ck} + \gamma_{ck} + \sum_d x_d \log \frac{p_{ckd}}{1-p_{ckd}})\,, \tag{15}$$

with

$$\gamma_{ck} = \textstyle\sum_d \log(1 - p_{ckd})\,. \tag{16}$$

Finally, if (15) is re-parameterized as

$$\lambda_{ck0} = \log \pi_{ck} + \gamma_{ck}\,, \tag{17}$$

$$\lambda_{ckd} = \log \frac{p_{ckd}}{(1 - p_{ckd})}\,, \tag{18}$$

then a discriminative MMLR classifier which is equivalent to the generative BM classifier is obtained.

B. From Discriminative to Generative Parameters

In this subsection we prove that the MMLR classifier defined in in section III, is *equivalent* to the BM classifier defined in section II. For doing that, we prove that the log-lineal model in (9) is equivalent to the joint probability defined in (4) but for a constant that does not depend on neither the class nor the component. We start by rewriting (4) in the following form

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}, c, k) = \exp\Big(\log \pi_{ck} + \gamma_{ck} + \sum_{d=1}^{D} x_d \log \frac{p_{ckd}}{1 - p_{ckd}}\Big)\,. \tag{19}$$

The normalization constraint in (6) should be taken into account during the proof, and in order to do so, another parameter, λ_{000} is introduced in the numerator of (9). This new parameter is not related with any feature since it accounts for an additional constraint. Moreover, introducing it does not modify the posterior probability computed by the log-linear model in (9) with the features in (11), provided that introducing the new parameter, λ_{000}, is equivalent to multiply the numerator and denominator by a constant: $\exp(\lambda_{000})$. Therefore, (9) is rewritten as

$$p_{\boldsymbol{\lambda}}(k, c \mid \boldsymbol{x}) = \tfrac{1}{\mathcal{Z}(\boldsymbol{x})} \exp(\lambda_{000} + \lambda_{ck0} + \textstyle\sum_d x_d \lambda_{ckd})\,, \tag{20}$$

for an arbitrary and unknown λ_{000}.

There are two types of parameters in (20) and (19). The first group corresponds to the parameters that multiply the input features, x_d, while the second group multiplies the feature 1. In other words, equivalence between (19) and (20) is proven if the following equivalences are verified

$$\textstyle\sum_d x_d \lambda_{ckd} = \sum_d x_d \big[\log p_{ckd} - \log(1 - p_{ckd})\big]\,, \tag{21}$$

$$\lambda_{000} + \lambda_{ck0} = \log \pi_{ck} + \gamma_{ck}\,, \tag{22}$$

where π_{ck} must verify (6).

Equation (21) is verified if

$$\lambda_{ckd} = \log p_{ckd} - \log(1 - p_{ckd})\,, \tag{23}$$

from where we can work out the value of p_{ckd}

$$p_{ckd} = \frac{\exp(\lambda_{ckd})}{1 + \exp(\lambda_{ckd})}\,. \tag{24}$$

Although the equivalence in (22) is more difficult to verify, the value of π_{ck} can be worked out as

$$\pi_{ck} = \exp(\lambda_{000}) \cdot \exp\left(\lambda_{ck0} - \gamma_{ck}\right)\,, \tag{25}$$

where γ_{ck} is defined in (16) with the values of p_{ckd} defined by (24).

Recall that the prior parameters π_{ck} must sum up to 1 and, hence, by plugging (25) into (6), the following constraint must be verified

$$\textstyle\sum_{c'k'} \exp(\lambda_{000}) \exp\left(\lambda_{c'k'0} - \gamma_{c'k'}\right) = 1\,, \tag{26}$$

from where the value of $\exp(\lambda_{000})$ is worked out

$$\exp(\lambda_{000}) = \left[\textstyle\sum_{c'k'} \exp\left(\lambda_{c'k'0} - \gamma_{c'k'}\right)\right]^{(-1)}\,. \tag{27}$$

Finally, the solution is given by plugging (27) into (25) as follows

$$\pi_{ck} = \frac{\exp\left(\lambda_{ck0} - \gamma_{ck}\right)}{\sum_{c'k'} \exp\left(\lambda_{c'k'0} - \gamma_{c'k'}\right)}\,. \tag{28}$$

Note that if we had not introduced the additional discriminative parameter, λ_{000}, then we could not have found a transformation from the discriminative parameters into the generative ones.

Finally, if we define the generative parameters as indicated in (24) and (28), then the generative model in (19) is equivalent to the discriminative model in (20), but for the constant factor $\mathcal{Z}(\boldsymbol{x})$. Given the previous relationship, it is straightforward to prove the equivalence between the parameters for the full classifiers as follows:

$$\begin{aligned}
\hat{c} &= \arg\max_{c}\ p_{\boldsymbol{\lambda}}(c \mid \boldsymbol{x}) = \arg\max_{c}\ \textstyle\sum_k p_{\boldsymbol{\lambda}}(c, k \mid \boldsymbol{x}) \\
&= \arg\max_{c}\ \textstyle\sum_k \mathcal{Z}(\boldsymbol{x}) p_{\boldsymbol{\lambda}}(c, k \mid \boldsymbol{x}) \\
&= \arg\max_{c}\ \textstyle\sum_k p_{\boldsymbol{\theta}}(c, k, \boldsymbol{x}) = \arg\max_c\ p_{\boldsymbol{\theta}}(c, \boldsymbol{x})
\end{aligned} \tag{29}$$

In summary, the MMLR adds no improvement or flexibility over the BM classifier apart from the possibility of discriminatively training it. The transformations provided in this section not only allow us to initialize the MMLR classifier with the MLE parameters, but also to train a MMLR classifier and then obtain its equivalent generative parameters so that they can be easily analyzed.

V. EXPERIMENTS

Experiments were carried out in order to assess the proposed BM MMI training algorithm with respect to the standard EM training algorithm. We first tested the initialization method, afterwards the model complexity, and finally the impact of the regularization term. Reported results will show that MMI training clearly outperforms conventional MLE training.

For the experimentation, we focused on the non-touching part of the Indian digits database from the well-known Arabic cheque database provided by CENPARMI [10]. The Indian digits database comprises $10\,425$ binary images, which were obtained from $3\,000$ real cheques, distributed in 10 classes which are related to the ten Indian digits. In order to obtain images normalized in size and position, two simple preprocessing steps were applied. First, each digit image was pasted onto a square background whose center was aligned with the digit mass center. This square background was a white image large enough (64×64) to accommodate most samples. Second, each digit image was subsampled into 30×30 pixels, from which its corresponding binary vector was built (with a dimension of $D = 900$ binary bits).

All experiments were carried out using the standard experimental procedure for *classification error rate (CER)* estimation in the CENPARMI Indian digits task, which is a simple partition with $7\,390$ samples for training and $3\,035$ for testing (excluding the extra classes *delimiter* and *comma*). As discussed in the introduction, the RPROP algorithm were used for training the discriminative model.

In the first experiment we have compared different initializations of the BM classifier prior to its transformation into a MMLR model. We have tested two different initializations: an initialization using the EM algorithm and a hypercube initialization. The comparison was carried out using $K = 5$ mixture components, although similar results were obtained for other values of K. In the hypercube initialization all parameters were uniformly initialized and then randomly perturbed. In the EM initialization, the initial BM was trained using the EM algorithm, which in turn, was initialized using a conventional Bernoulli classifier trained with the MLE criterion. Afterwards, several iterations of the RPROP algorithm were performed to discriminatively train the MMLR model. Results are reported in Fig. 1. The results are statistically significant since each point in the plot is the average of 50 repetitions. It is observed that the discriminatively trained BM improves the generative results, which correspond to the left-most point in the EM curve. However, the hypercube initialization outperforms the generative initialization.

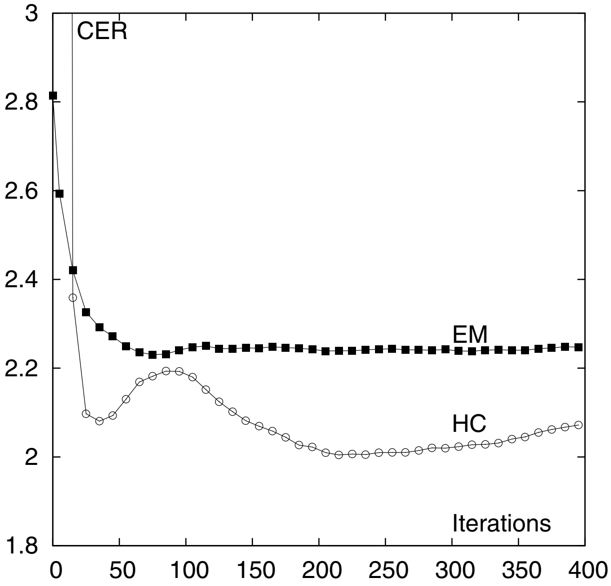

Figure 1. Comparison between EM and and HC initializations using the RPROP algorithm.

In order to assess the repercussion of the number of mixtures components, K, we carried experiments varying it in the range $\{1, 2, 4, 6, 8, 10\}$ and using the hypercube initialization. The results are shown in Fig. 2, each point is the average of 50 repetitions. A big improvement is obtained by only 2 components. The more components are added, the larger the improvement becomes until it is saturated at 6 components. Results for $K > 6$ are not plotted since they are identical to $K = 6$. Note that the behavior differs from the generative BM classifier. According to [1], generative BM achieves the best results around $K = 15$. Moreover, generative BM with $K = 15$ have an error of 2.7% while in Fig. 2 using $K = 6$ we obtain an error about 2.0%. The discriminative BM obtain an improvement of 25% over the generative BM by using half of the parameters. To our knowledge, the best result in this database is approximately 1.9% [2], which is similar to our result but using much more parameters.

A final experiment were performed to assess the behavior of the regularization term. Several values of C were scanned ranging from 0 (no regularization) to 0.5. Results are shown in Fig. 3. It is observed that without regularization the error is unstable and it increases (over-fits) along with the iterations. In contrast, regularization makes the error more stable while providing the same performance. In particular for $C = 0.001$ the CER is stabilized around 2%.

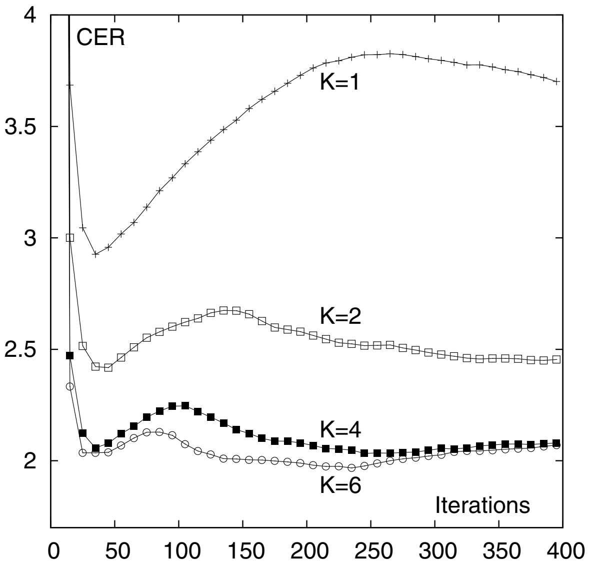

Figure 2. CER (%) of discriminative BM for several number of mixture components (K) using the hypercube initialization.

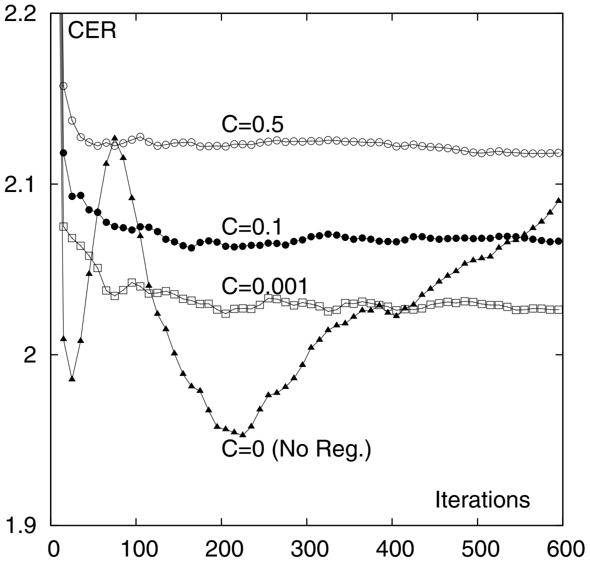

Figure 3. Impact of the regularization term over the CER (%).

VI. Conclusions

A mixture of multi-class logistic regression (MMLR) model has been proposed for binary data. This model was inspired by Bernoulli mixture (BM) model. Afterwards, the equivalence between both classifiers has been proved. Consequently we obtained two results. On the one hand, discriminative parameters can be interpreted by transforming them into generative parameters. On the other hand, we have provided a MMI training scheme for BM classifiers.

This new training scheme has been tested and compared, with the generative MLE training scheme, using different initialization methods and regularization terms, on the well-known CENPARMI Indian digits database. The proposed MMI training scheme outperforms the generative MLE criterion using half of the parameters.

As future work, we intend to extend all the work developed in this paper to Bernoulli HMMs, which have obtained very good results in Arabic HTR when trained with the generative MLE criteria.

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).

References

[1] A. Juan and E. Vidal, "Bernoulli mixture models for binary images," in *ICPR*, 2004.

[2] V. Romero, A. Giménez, and A. Juan, "Explicit Modelling of Invariances in Bernoulli Mixtures for Binary Images," in *IBPRIA*, 2007, pp. 539–546.

[3] A. Giménez, I. Khoury, and A. Juan, "Windowed Bernoulli Mixture HMMs for Arabic Handwritten Word Recognition," in *ICFHR*, 2010, pp. 533–538.

[4] V. Märgner and H. E. Abed, "ICFHR 2010 - Arabic Handwriting Recognition Competition," in *ICFHR*, 2010, pp. 709–714.

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

[6] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[7] J. N. Darroch and D. Ratcliff, "Generalized iterative scaling for log-linear models," *The Annals of Mathematical Statistics*, vol. 43, no. 5, pp. 1470–1480, 1972.

[8] G. Heigold, T. Deselaers, R. Schlüter, and H. Ney, "A GIS-like training algorithm for log-linear models with hidden variables," in *ICASSP*, 2008, pp. 4045–4048.

[9] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: The rprop algorithm," in *IEEE Int. Conf. on Neural Networks*, 1993, pp. 586–591.

[10] Y. Al-Ohali, M. Cheriet, and C. Suen, "Databases for recognition of handwritten Arabic cheques," *Pattern Recognition*, vol. 36, pp. 111–121, 2004.