ELSEVIER

Pattern Recognition Letters 23 (2002) 905–916

Pattern Recognition Letters

www.elsevier.com/locate/patrec

Writer adaptation techniques in HMM based Off-Line Cursive Script Recognition

Alessandro Vinciarelli *, Samy Bengio

[a] IDIAP – Institut Dalle Molle d'Intelligence, Artificielle Perceptive, Rue du Simplon 4, CP 592, 1920 Martigny, Switzerland

Received 22 May 2001; received in revised form 6 September 2001

Abstract

This work presents the application of HMM adaptation techniques to the problem of Off-Line Cursive Script Recognition. Rather than training a new model for each writer, one first creates a unique model with a mixed database and then adapts it for each different writer using his own small dataset.

Experiments on a publicly available benchmark database show that an adapted system has an accuracy higher than 80% even when less than 30 word samples are used during adaptation, while a system trained using the data of the single writer only needs at least 200 words in order to achieve the same performance as the adapted models. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Off-Line Cursive Script Recognition; HMM Bayesian Adaptation; HMM Maximum Likelihood Adaptation; HMM Maximum A Posteriori Adaptation

1. Introduction

In the last years, several efforts were led, in the domain of Off-Line Cursive Script Recognition (CSR), toward the recognition of texts written by a single person (Senior and Robinson, 1998; Marti and Bunke, 1998, 2000, 2001; Lazzerini et al., 1997). In this task, the best performance is achieved using, for training, samples produced by the writer himself. On the other hand, the training is reliable only if there is enough data and this might be difficult because the writer should be required to produce too many samples.

This problem can be solved by applying adaptation techniques. The literature presents several techniques for HMM adaptation (Leggetter and Woodland, 1995b; Digalakis et al., 1995; Gauvain and Lee, 1994). Their aim is to improve the performance of the models over specific subsets of the data they are trained to recognize.

A training set is typically composed of large sets of data produced by different sources. The HMMs trained over such data are not optimal with respect to any source, but they have a good performance on the data produced by all of them. Moreover, they have a good recognition performance also

* Corresponding author. Tel.: +41-27-7217724; fax: +41-27-7217712.

E-mail addresses: alessandro.vinciarelli@idiap.ch, vincia@idiap.ch (A. Vinciarelli), bengio@idiap.ch (S. Bengio).

PII: S0167-8655(02)00021-1

over data produced by sources not represented in the training set.

In some situations, it can be desirable to have models optimal for a certain source, but this is not possible because not enough source dependent data is available. When this is the case, the adaptation techniques can be used to fit the HMM parameters to the distribution of the source dependent data. The resulting models are closer to the optimal solution (represented by models trained using only source dependent data) than the source independent models and are the best solution in absence of sufficient source dependent data.

The adaptation techniques are not influenced by the nature of the sources, it is not necessary to modify them depending on the specific application they are involved in. In the case of the handwriting recognition, the sources can correspond to the different writers.

The models are typically trained over samples produced by many sources, i.e. many writers. The resulting HMMs are not optimal for any single writer, but they are good for all of them. If the training set is sufficiently representative of the different handwriting styles, the models will achieve a good performance also over data written by persons that did not produce samples for the training set.

The adaptation allows to optimize the models for a single writer given a set of its data that can be much smaller than the amount of data actually needed for writer dependent training. For this reason the adaptation process is referred to as, in this case, *Writer Adaptation*. The models obtained through the adaptation are called *Writer Dependent* (WD) in opposition to the original models said *Writer Independent* (WI).

This work presents experiments performed adapting continuous density HMMs (having mixtures of Gaussians as emission probabilities) trained over a WI database to WD data. Models adapted on WD data sets of different sizes are compared with WD HMMs trained over the same sets. The results show that, for our database, $\sim$200 words (a considerable amount for a single writer) are needed to obtain WD models performing better than the adapted ones. Moreover, with less than 30 WD words it is not possible to train WD models because not all the letters are represented in such a small set, while the result with the adaptation method already performs well. This shows that the adaptation can be a good solution to obtain WD models when it is difficult to collect WD data, while reliably trained WI HMMs are available.

This paper is organized as follows: Section 2 presents the CSR approaches, Section 3 describes the HMM adaptation techniques, Section 4 reports experiments and results, and Section 5 presents some conclusions.

2. Off-line Cursive Script Recognition

The research in CSR had its main development in the last ten years (Steinherz et al., 1999; Plamondon and Srihari, 2000). The process leading to the transcription of the handwritten word can be divided into several steps: preprocessing, normalization, segmentation, feature extraction and recognition.

The preprocessing works on the raw data and has as output an image (often binary) showing the word to be recognized without any other disturbing element (background textures, extraneous strokes, etc.). The normalization reduces the variability due to acquisition and handwriting style. This is done by removing *slope* (the angle between the horizontal direction and the direction of the line on which the word is aligned) and *slant* (the angle between the vertical direction and the direction of the strokes that, in an ideal model of handwriting, would be vertical). The segmentation extracts fragments of the word (called primitives or graphemes) considered as the basic units of information. Depending on the approach, these can be letters or parts of them. The feature extraction converts the primitives into vectors that are used in the recognition step. This is performed by using Dynamic Programming techniques or HMMs. The matching of the data with all the words in a list of allowed interpretations (called *lexicon*) is measured. The lexicon word showing the best matching score is retained as correct interpretation.

In the present paper, we will focus on the recognition step done using HMMs and adaptation techniques that can be applied to these models in order to get better performance when small amount of data is available to train such HMMs.

3. Adaptation techniques

The adaptation techniques allow to improve the performance of WI models over WD data when there is not enough data for a reliable training of WD models. The process consists in adapting the parameters $\theta = (\theta_1, \theta_2, \ldots, \theta_M)$ of the WI models using the WD data.

The probability of having a parameter vector θ given the adaptation set of observations $\boldsymbol{O}$ can be written (using Bayes theorem) as follows (Duda et al., 2000):

$$p(\theta|\boldsymbol{O}) = \frac{p(\boldsymbol{O}|\theta)p(\theta)}{p(\boldsymbol{O})}, \tag{1}$$

where $p(\theta|\boldsymbol{O})$ and $p(\theta)$ are, respectively, the *posterior* and *prior* distribution of the parameters, and $p(\boldsymbol{O}|\theta)$ is the likelihood of the HMM with parameter set θ. The aim of the adaptation is finding the vector θ_{ad} maximizing the posterior

$$\theta_{\mathrm{ad}} = \mathrm{argmax}_{\theta} p(\theta|\boldsymbol{O}). \tag{2}$$

This can be done in two different ways depending on the prior distribution of θ. If $p(\theta)$ is *noninformative*, i.e. does not give any information about how the θ components are likely to be, then the adapted parameters θ_{ad} are estimated with Maximum Likelihood (ML). Since a noninformative prior distribution corresponds to a constant uniform distribution $p(\theta) = c$, this amounts to solving the equation

$$\frac{\partial p(\boldsymbol{O}|\theta)}{\partial\theta} = 0. \tag{3}$$

When the prior distribution is *informative*, i.e. the distribution $p(\theta)$ is different than a constant, then the adapted parameters are obtained by solving the equation

$$\frac{\partial (p(\boldsymbol{O}|\theta)p(\theta))}{\partial\theta} = 0. \tag{4}$$

This corresponds to a Maximum A Posteriori (MAP) estimation of θ_{ad}.

When the adaptation is performed with ML, θ_{ad} is estimated so that the probability of the adapted models generating the adaptation data is maximized. When the adaptation is performed with MAP, θ_{ad} is such that the Bayes risk (Duda et al., 2000) over the adaptation set is minimized (hence the name Bayesian Adaptation).

In the experiments, ML will be used to estimate the parameters of a linear regression transforming WI parameters into WD ones (Maximum Likelihood Linear Regression, MLLR). The method will be shown effective for few adaptation data, but quickly converging to a saturation performance that cannot be further improved. On the other hand, the MAP estimation needs, in order to be effective, more adaptation data, but the performance of the adapted models converges to that of the WD ones. A third possibility is given by the combination of the two approaches. A model is first adapted with MLLR, then its parameters are used to obtain a prior distribution information and perform a MAP adaptation.

In this work, the adaptation is applied to continuous density HMMs: the probability of emitting an observation $\boldsymbol{o}$ when being in a given state of the HMM is modeled by a mixture of Gaussians (Rabiner, 1989). The parameters to adapt are thus the means, variances and weights of these mixtures of Gaussians. We can furthermore simplify the adaptation techniques by making the hypothesis that the WD information is carried essentially by the means of the Gaussians. Hence only such parameters will be adapted. The parameter vector θ corresponds to the vector $\mu = (\mu_1, \mu_2, \ldots, \mu_G)$, where G is the total number of Gaussians.

The MLLR technique is conceived to be effective with few adaptation data. Because of this, most Gaussians might be poorly or not at all affected by the adaptation process. This can be overcome by clustering the Gaussians, i.e. by grouping them so that, during the adaptation process, most of them can be updated (Gales, 1996).

The MAP technique adapts each Gaussian separately. This makes necessary more adaptation material, but gives better results as the adaptation set size increases.

Section 3.1 explains how the Gaussians are clustered for the MLLR technique. Sections 3.2

and 3.3 give respectively more details about ML and MAP techniques.

3.1. Gaussian clustering

The first step in MLLR adaptation is the Gaussian clustering performed by grouping the Gaussians into *regression classes*. All the Gaussians of a cluster share the parameters of a linear transform leading from the WI means to the WD ones (see Section 3.2). This allows to adapt also the Gaussians for which there is not enough data in the adaptation set.

The regression classes are determined dynamically according to the amount of data available (in the WI data set) using a binary *regression class tree* (Gales, 1996). This is grown using a centroid splitting algorithm based on a Euclidean distance measure. Given a node to be split, mean and variance from the Gaussians clustered at this node are calculated. Two children are created and their means are initialized to the mean of the parent perturbed in opposite directions by a fraction of the variance. Each Gaussian clustered at the parent node is assigned to one of the children nodes (depending on the distance) and, when all the Gaussians are assigned, the mean and variance for the children nodes are calculated. The process is repeated until the predetermined number of leaf nodes is reached.

Once the tree is grown, the regression classes can be determined following the scheme illustrated in Fig. 1. A solid arrow means that the child node contains a number of observations (each one being attributed to the most likely Gaussian) above an experimentally determined threshold, and can thus form a regression class. In the other cases the data is insufficient and its Gaussians belong to the regression class of the parent node.

Since the tree is grown using the WI data, it is itself independent of the writer and can be used to perform the adaptation to any writer.

3.2. Maximum Likelihood Linear Regression

At the beginning of the adaptation process, the Gaussians are grouped into clusters. When a Gaussian of the cluster is hit by an observation of the adaptation set, all the Gaussians of the same cluster are updated.

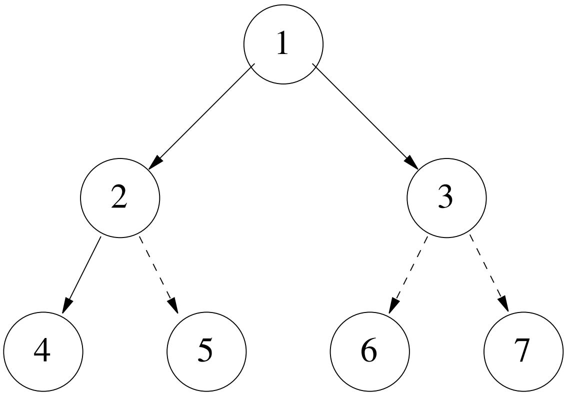


Fig. 1. Regression tree with four base classes. The dashed line means that the child node does not contain sufficient data. In the case of the figure, there are three regression classes. The first one is composed of data clustered at node 4. They are transformed by a matrix estimated with the data themselves. The second one is composed of data clustered at node 5. Their transform matrix is estimated using data clustered at node 2 because the data belonging to node 5 are not enough. The last class is composed of data clustered at nodes 6 and 7. Since the data are insufficient for both nodes, they share a common transform matrix estimated with the data clustered at node 3 and then form a single class.

Consider a Gaussian

$$N(\boldsymbol{o}, \mu_{i_g}, \Sigma_{i_g}) = \frac{1}{(2\pi)^{d/2}|\Sigma_{i_g}|^{1/2}} \times \exp\{1/2(\boldsymbol{o} - \mu_{i_g})'\Sigma_{i_g}^{-1}(\boldsymbol{o} - \mu_{i_g})\}, \tag{5}$$

where Σ is the covariance matrix, μ the mean and d the dimension of the observation space. The index i_g states that this is Gaussian i in cluster g.

Following the hypothesis proposed in (Leggetter and Woodland, 1995a, 1995b; Digalakis et al., 1995), we adapt only the WI means μ using the linear transform

$$\hat{\mu}_{i_g} = W_g \xi_{i_g}, \tag{6}$$

where W_g is a $d \times (d+1)$ matrix, and $\xi_{i_g} = (\omega_g, \mu_{i_g1}, \ldots, \mu_{i_gd})$, where ω_g is an offset. Since ML is used to estimate the parameters of a linear transform, the method is called MLLR.

After the transform, the Gaussians become

$$\hat{N}(\boldsymbol{o}, \mu_{i_g}, \Sigma_{i_g}) = \frac{1}{(2\pi)^{d/2}|\Sigma_{i_g}|^{1/2}} \times \exp\{1/2(\boldsymbol{o} - W_g\xi_{i_g})'\Sigma_{i_g}^{-1}(\boldsymbol{o} - W_g\xi_{i_g})\}. \quad (7)$$

The W_g (the index is g because all of the Gaussians in the cluster share the same matrix transform) elements are selected to maximize the likelihood of the adapted models generating the adaptation data. The ML estimation is performed with the Expectation–Maximization algorithm (Dempster et al., 1977). By formulating the standard auxiliary function, and then maximizing it with respect to the transformed mean the following equation is obtained:

$$\sum_{t=1}^{T}\sum_{r_g=1}^{R_g}\gamma_{r_g}(t)\Sigma_{r_g}^{-1}\boldsymbol{o}_t\xi'_{r_g} = \sum_{t=1}^{T}\sum_{r_g=1}^{R_g}\gamma_{r_g}(t)\Sigma_{r_g}^{-1}\bar{W}_g\xi_{r_g}\xi'_{r_g}, \quad (8)$$

where $\bar{W}_g$ indicates the estimated parameters at a certain iteration, and $\gamma_{r_g}(t)$ the probability of being in state r_g at sample t of the adaptation set. The sum over t involves all the observations $\boldsymbol{o}_t$ belonging to the adaptation set $\boldsymbol{O}$, the sum over r_g involves all the Gaussians belonging to the cluster g. Once the elements of $\bar{W}_g$ are estimated at a certain iteration n, they are used to modify the means μ_{r_g} and, correspondingly, the probabilities γ_{r_g}. These can be used again in Eq. (8) to obtain the estimation at iteration $n+1$ of W_g.

The matrices W_g are randomly initialized, then by repeating the process (until the change in likelihood of the data between two iterations falls below a predefinite threshold), they are iteratively optimized to approximate ML estimations.

Note that the μ adapted by ML will always stay a linear function of the WI μ, which explains why it adapts quickly, but cannot profit from a lot of adaptation material.

3.3. Bayesian adaptation

In the Bayesian framework, the first problem to be solved is finding an appropriate prior distribution $p(\theta)$.

Following (Gauvain and Lee, 1992, 1994; Lee and Gauvain, 1993), we consider a mixture of Gaussians as the marginal pdf of $p(\theta)$

$$p(\boldsymbol{o}|\theta) = \int_{\Theta} p(\boldsymbol{o}, \theta)p(\theta)\,\mathrm{d}\theta. \quad (9)$$

This leads to expressing $p(\theta)$ as a product of a Dirichlet density (Johnson and Kotz, 1972), accounting for the mixture weights ω_i, and a normal-Wishart density (De Groot, 1970), accounting for the other parameters.

Such a prior distribution has the important property of belonging to the conjugate family of the complete data density in the HMM case. This allows to apply the Expectation–Maximization technique to obtain a MAP estimate of the adapted parameters. We make the same assumption as in Section 3.2 and we thus adapt only the means. The use of EM then leads to the following equation for the adaptation of the means:

$$\hat{\mu}_{jm} = \frac{N_{jm}}{N_{jm} + \tau}\bar{\mu}_{jm} + \frac{\tau}{N_{jm} + \tau}\mu_{jm}, \quad (10)$$

where N_{jm} is the occupation likelihood of the adaptation data, i.e. the sum of the probabilities of each observation in the adaptation set being emitted by the Gaussian m in state j (the index of the model is omitted for simplicity), τ is a parameter related to the prior distribution (the value is set empirically), $\bar{\mu}_{jm}$ is the mean of observed adaptation data and μ_{jm} is the WI data mean. When N_{jm} is small, the adapted mean is close to the WI mean, otherwise the adapted values shift toward the mean of the adaptation data. At each iteration of the EM, the $\hat{\mu}_{jm}$ estimated at the previous iteration are used in the Gaussians of the cluster g. This leads to new values of N_{jm} and $\bar{\mu}_{jm}$, then, through Eq. (10), of $\hat{\mu}_{jm}$. This iterative procedure is repeated until the change in the parameters between two following iterations falls below a predefinite threshold.

4. Experiments and results

The experiments are performed using a CSR system based on a sliding window approach

(Vinciarelli and Luettin, 2000). The system converts the handwritten data into a sequence of vectors with a window shifting column by column from left to right (this approach allows to avoid the segmentation). At each window position, a frame is isolated and a feature vector is extracted. The feature extraction consists of partitioning the window into 16 cells regularly arranged in a 4×4 grid and of counting the number of foreground pixels in each cell. This leads to a 16-dimensional feature vector accounting, in component i, for the percentage of foreground pixels (with respect to the total number of foreground pixels in the window) in cell i. The recognition is performed using continuous density Hidden Markov Models to calculate the likelihood of the observation sequence given an HMM corresponding to each word in a lexicon. The most likely word is selected as the interpretation of the handwritten data. Words are modeled as concatenations of single letter HMMs. This makes the system flexible with respect to changes in the lexicon. Given the letter models, any word can be modeled, independently of the presence of its examples in the training set. Training is based on ML estimation (Baum–Welch algorithm), while recognition is based on the estimation of MAP Probability (using the Viterbi algorithm).

The number of states and Gaussians per mixture (S and G, respectively) is the same for every letter model. Their value is set through *cross validation* (Stone, 1974). All the systems corresponding to couples (S, G) falling in a range determined by the amount of training data are trained and tested. The system giving the highest estimated generalization performance is retained as optimal.

Two data sets are used. The first is a collection of samples produced by many writers and is used to obtain a WI system (see Section 4.1). The second is composed of a text written by a single person and is used to train a WD system (see Section 4.2). The WI independent system is adapted to this last database and its performance is compared to that of the WD system for different sizes of the training/adaptation set (see Section 4.3).

4.1. Writer independent model training

The WI system is obtained by training the above described system over a database of words produced by ~200 writers. The data set is composed of 12 178 words extracted from a handwritten page database collected at the University of Bern (Marti and Bunke, 1999). The database was divided into two parts, a training set (8160 words) and a test set (4018 words).

The models are first initialized as follows: they start with 1 Gaussian per state. The training samples are uniformly segmented and mean and variance of the observations attributed to each state of each model are used to initialize mean and variance of its Gaussian. After this step, an embedded training is performed, i.e. the Baum–Welch algorithm is not applied directly to letter models, but to word models obtained by concatenating them. The single letter models are so trained when they are part of a word and not when they are isolated. All the systems corresponding to different values of S in a suitable range (determined experimentally) are trained and tested, and a curve showing the performance in function of S is obtained for $G = 1$.

At this point it is possible to increase G: the most probable Gaussian of each mixture is split by perturbing in opposite directions its mean by a value equal to its variance. This allows to add one Gaussian to each mixture. An embedded training is performed on the so modified models for S ranging in the same interval as the previous step. The parameter G is increased as long as it is possible to improve the performance of the system (see Fig. 2).

Systems with $10 \leqslant S \leqslant 20$ and $1 \leqslant G \leqslant 7$ were trained and tested. The best performance (with lexicon size 100) was achieved using models with 14 states per letter and 7 Gaussians per mixture.

4.2. Writer dependent model training

The WD system is obtained using a database of 4053 words produced by a single writer and

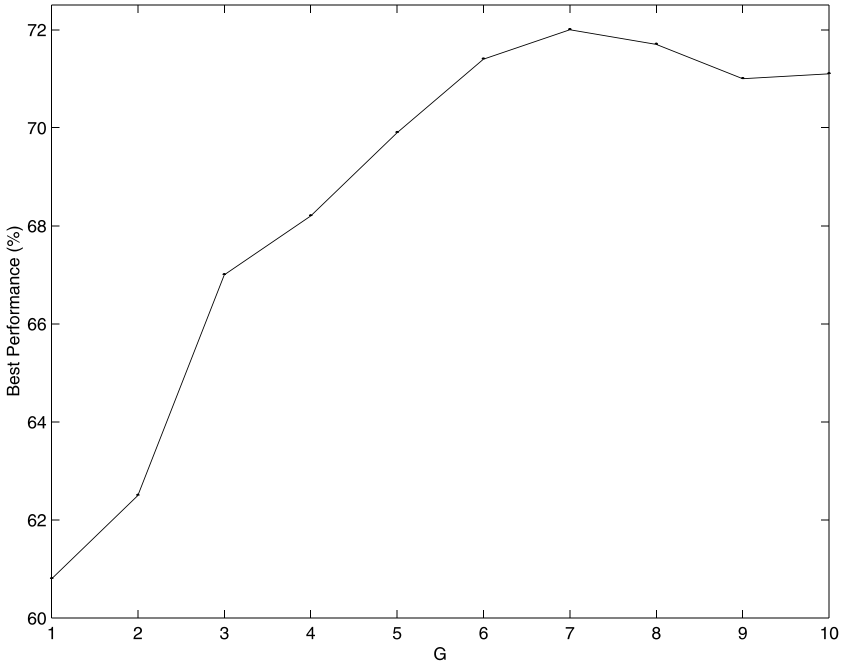


Fig. 2. Best performance of the WI systems as a function of G. For each value of G, the best performance obtained among the systems with $10 \leqslant S \leqslant 20$ is plotted. G is increased until it is not possible to obtain any more improvement of the performance.

described in (Senior and Robinson, 1998).[1] The data set was split into two parts, a training set (2700 words) and a test set (1353 words). The partition is not performed, as usual, with a random process. The words are stored in the same order as they were written and the training set is composed of the first 2700 words. The reason is that the same data set will be used for the adaptation of the WI system and, in a realistic situation, such process is performed over the first n samples to improve the performance over the following ones.

The size of the training set is progressively increased (while keeping unchanged the test set) in order to show the dependence of the system performance on the number of samples used to train (adapt) the WD (WI) models. Until 200 words, the training (adaptation) set size increasing step is 10, above 200 words it becomes 100.

A WD (WI) system is then obtained by training (adapting) the WD (WI) models over the first $10, 20, 30, \ldots, 200$ and over the first $300, 400, \ldots,$ 2700 words of the WD database.

The training technique used for the WD models is the same as the one described in Section 4.1 for the WI models. Models with $6 \leqslant S \leqslant 16$ and $1 \leqslant G \leqslant 5$ were trained and tested. The effect of the G increase is shown in Fig. 3 (for training set size 2700).

4.3. Adaptation results

The results obtained with training/adaptation set size between 10 and 200 (some samples belonging to such set can be seen in Fig. 4) are shown in Fig. 5. The performance is measured in terms of recognition rate with a lexicon of 100 words.[2]

[1] The database is publicly available on the web and can be downloaded at the following ftp address: ftp.eng.cam.ac.uk/pub/data.

[2] The lexicon of the WD database is composed of 1370 words. In order to perform the recognition against 100 words, each sample uses a specific lexicon composed by the correct transcription and 99 words selected randomly from the original 1370 word lexicon.

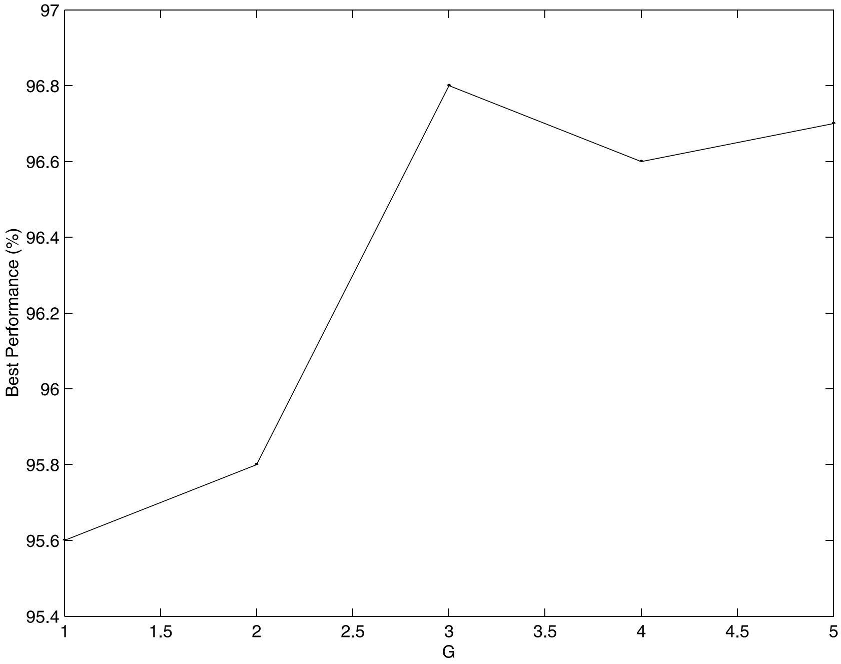


Fig. 3. Performance of the WD systems as a function of *G*. For each value of *G*, the best performance obtained among the systems with $6 \leqslant S \leqslant 16$ is plotted. *G* is increased until it is not possible to obtain any more improvement of the performance.

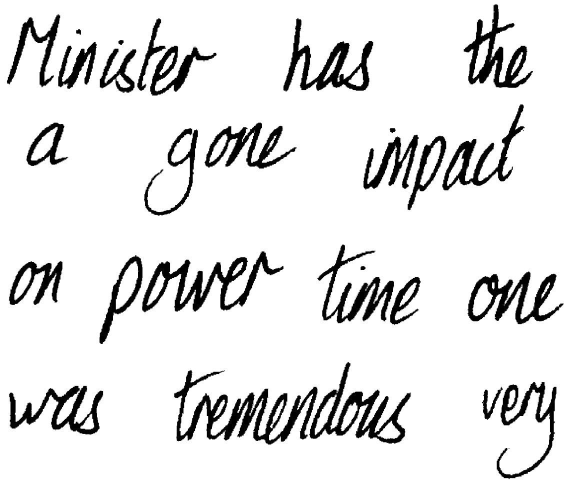
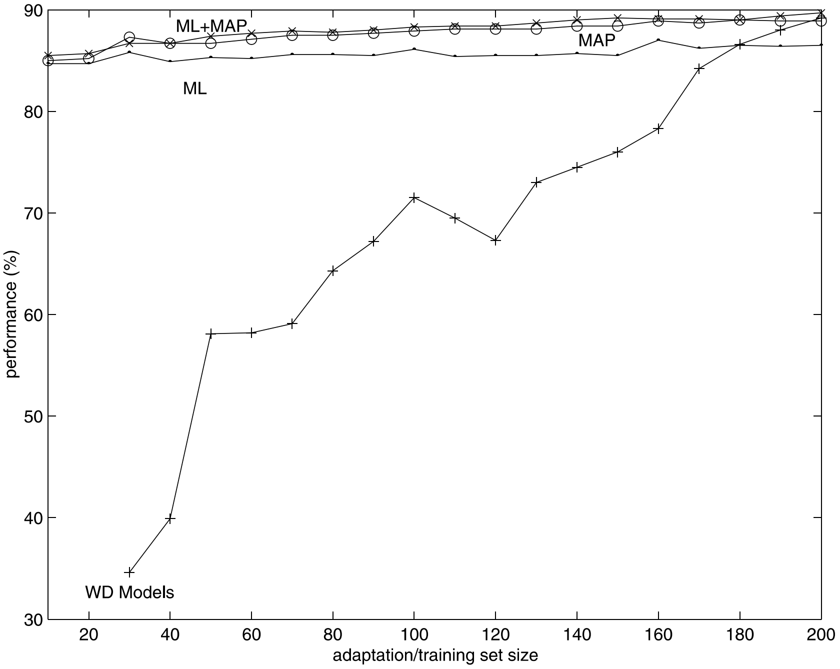


Fig. 4. Some samples used for the adaptation. The words belong to the transcription of a text extracted from the LOB Corpus.

The upper plot shows that, in this range of training/adaptation set size, the adapted models perform significantly better than the WD ones. The lower plot shows the performance of the adapted models more clearly. The models adapted with ML increase their performance slowly with respect to the others and their accuracy falls below that of the WD models before the models adapted with other techniques. The MAP method produces models slightly more accurate than the combination of ML and MAP.

The results obtained with training (adaptation) set size between 300 and 2700 are shown in Fig. 6. When more than 200 words are available in the training set, the WD models become better than the adapted ones. This limit must be considered a lower bound because in our experiments the WD system was trained using the test set for cross validation, then its performance is biased towards it. This leads to an overestimation of the recognition rate of the WD models and it is more correct to say that they need then *at least* 200 words in order to perform better than the adapted models.

The models adapted with ML do not take any advantage of the increase of the adaptation set, while the models adapted with other techniques



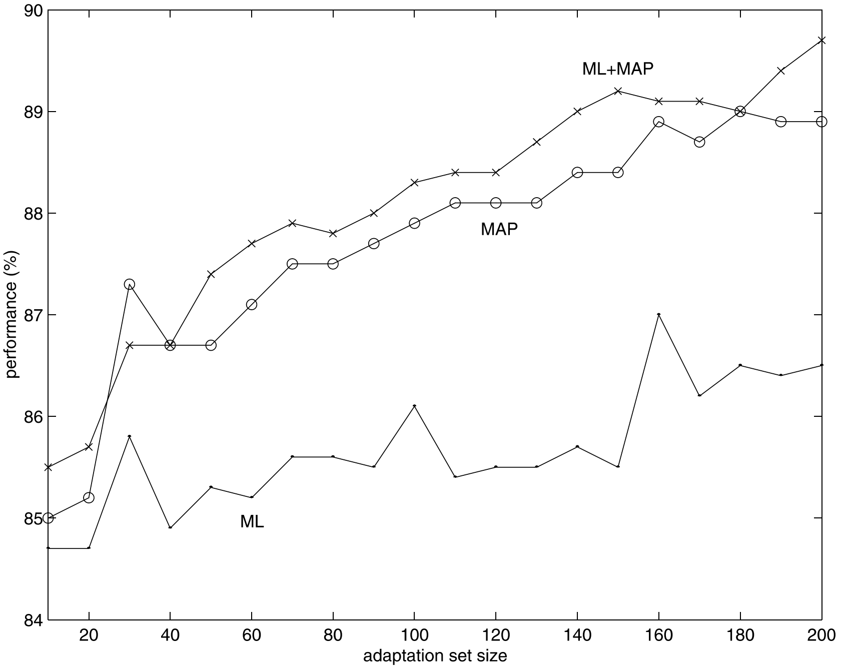


Fig. 5. The upper figure reports the accuracy vs. the training/adaptation set size for the WD models and the WI models adapted with ML, MAP and ML + MAP techniques. In the lower plot, the curves of the adapted models are shown in more detail. The values are reported for training/adaptation set sizes less than or equal to 200. When the training/adaptation set size is less than 30, no WD system can be trained.

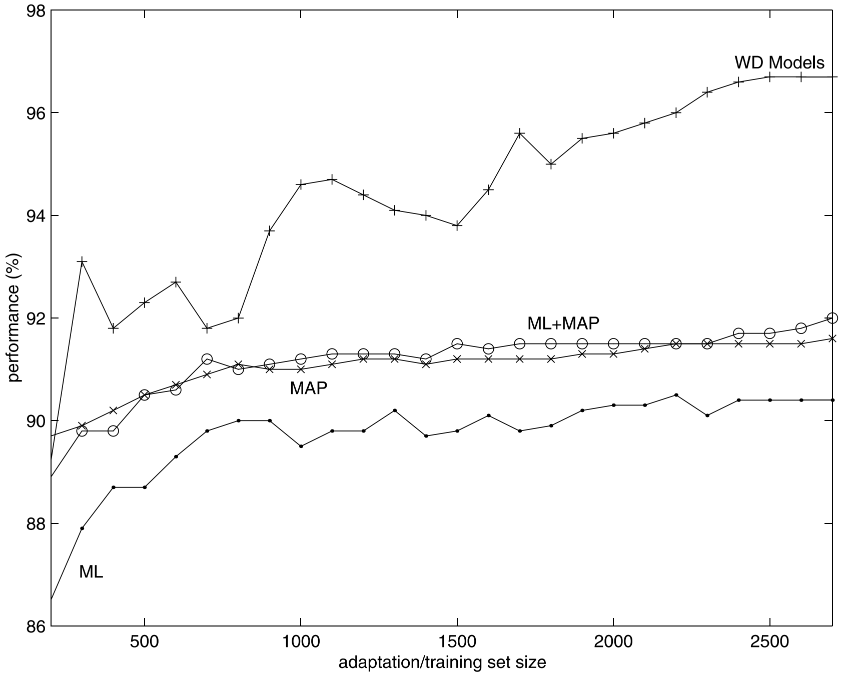


Fig. 6. The figure shows the accuracy of WD and adapted models for sizes of training/adaptation set ranging from 300 to 2700.

show some improvements of the accuracy. The models adapted combining ML and MAP become more accurate than those adapted with MAP.

The main source of errors are the short words (less than three letters). This happens because, being composed of few characters, a bad alignment of a single letter model with the observations of the related character is enough to cause errors. Moreover, the normalization scheme applied to the samples before the recognition is conceived for longer words and, sometimes, produces bad results over the shortest samples. Another source of errors comes from the ambiguity of certain word couples like *been* and *beer* or *year* and *years*. A possible solution to the last problem can be the application of a language model. This allows to select the correct classification using the word context. The accuracies reported are obtained with a 100 word lexicon. The use of bigger lexica would result in a lower performance. In any case, in the limits of the statistical fluctuations, the difference in performance would remain unchanged. The important aspect of the HMM adaptation techniques is not in the performance, but in the fact that, with few WD samples, it is possible to improve considerably the performance of WI models. Once enough data is available for WD training, the adaptation cannot improve the performance of WD models, but the collection of WD data can be sometimes difficult.

As mentioned above, the adaptation set is composed by the first *n* samples produced by the writer. This corresponds to a realistic condition where the WD samples available at a certain moment are used to improve the performance over the data written in the following.

If the experimental conditions allow to select arbitrarily the samples (e.g. by asking the writer to write some predetermined words), the system can be made effective with fewer data by using words containing all the characters. In this way, all the letter models are affected by the adaptation process and the system is adapted more quickly.

The selection of the samples can have influence on the speedness of the adaptation, but cannot in any case improve the performance of the adapted models. The HMMs adapted with MLLR are limited by the fact that the new parameters are a linear function of the old ones. The models adapted through the Bayesian approach cannot in any case perform better than models trained directly (once enough training data are available) over WD data.

A different technique to adapt a CSR system to a single writer is presented by Lazzerini et al. (1997). In this work, the CSR system segments the words into letters and then recognizes them separately with a Neural Network. A contextual analysis of the character recognizer results allows to correct eventual misclassifications.

The adaptation consists of retraining the neural network using only characters of the specific writer extracted from the WD samples using a WI system. From this point of view, the system adapted with this technique corresponds to what we call a WD system, i.e. a system trained directly over the samples of the specific writer.

To train over WD data is the optimal solution, but requires, to be effective, a sufficient amount of data. The aim of the HMM adaptation is not the improvement of the system performance (the best accuracy can be achieved in any case only by training directly over WD data), but the possibility of having an effective system when the available WD data is not sufficient for reliable training.

Moreover, in order to retrain the network it is necessary to have samples of all the characters, while in adapting the letter models, it is possible to adapt selectively only the models related to letters presented in the adaptation set (leaving the others unchanged). This allows the adapted system being effective for very small amounts of adaptation data.

5. Conclusions

We have presented in this paper an application of HMM adaptation techniques to the problem of Off-Line Cursive Script Recognition. A CSR system trained over a database of samples produced by many writers was adapted to the words of a single writer data set.

The experiments showed that, for our database, the models trained over sets containing less than $\sim$200 words have an accuracy inferior to that of models obtained adapting WI models to the same data. This estimated amount must be considered a lower bound since the WD models are trained using the test set for crossvalidation and this results in an overestimation of their performance.

For very small training sets (less than 30 words) it was not even possible to train WD models, while there was no problem in adapting WI models. For a 30 words set, the WD system had an accuracy of 34.6% while a WI system adapted with the combination of ML and MAP techniques had an accuracy of 87.3%.

Both adaptation techniques and training methods for continuous density HMMs have been described in detail. The effect of increasing the number of Gaussians in the mixtures has also been shown.

Some improvements can be obtained by using more advanced adaptation techniques and better WI models. Moreover, by studying an optimal composition of the adaptation set, it will be possible minimize the number of necessary samples for an effective adaptation. The absence of some letters in the first words is determined by the fact that our database is a transcription of a text, but it is possible to create smaller sets of words containing all the letters by asking the writer to produce them.

The adaptation techniques are a natural solution for all the applications where it is necessary to recognize samples produced by a single writer without having enough samples for a reliable training.

Acknowledgements

This work was done under the grant no. 21-55733.98 issued by Swiss National Science Foundation. The authors wish to thank M. Barnard, F. Camastra, J. Meriethoz and A. Morris for commenting on the draft.

References

De Groot, M., 1970. Optimal Statistical Decisions. McGraw-Hill, New York.

Dempster, A., Laird, N., Rubin, D., 1977. Maximum Likelihood estimation from incomplete data. J. Roy. Statist. Soc. B 39 (1), 1–38.

Digalakis, V., Rtischev, D., Neumayer, L., 1995. Speaker adaptation using constrained estimation of Gaussian mixtures. IEEE Trans. Speech Audio Process. 3 (5), 357–366.

Duda, R., Hart, P., Stork, D., 2000. Pattern Classification. Wiley, New York.

Gales, M., 1996. The generation and use of regression class trees for MLLR adaptation. Tech. Rep. TR263, Cambridge University Engineering Department.

Gauvain, J., Lee, C., 1992. MAP estimation of Continuous Density HMM: theory and applications. In: Proc. DARPA Speech and Natural Language Workshop. Morgan Kaufmann, Los Altos, CA.

Gauvain, J., Lee, C., 1994. Maximum a Posteriori estimation for multivariate gaussian mixture observations of Markov chains. IEEE Trans. Speech Audio Process. 2 (2), 291–298.

Johnson, N., Kotz, S., 1972. Distribution in Statistics. Wiley, New York.

Lazzerini, B., Marcelloni, F., Reyneri, L., 1997. Beatrix: A self-learining system for off-line recognition of handwritten texts. Pattern Recognition Lett. 18, 583–594.

Lee, C., Gauvain, J., 1993. Speaker adaptation based on MAP estimation of HMM parameters. In: Proc. IEEE Conf. on Audio Speech Signal Process., Vol. 2.

Leggetter, C., Woodland, P., 1995a. Flexible speaker adaptation for large vocabulary speech recognition. In: Proc. 4th European Conf. on Speech Commun. Technol.

Leggetter, C., Woodland, P., 1995b. Maximum likelihood linear regression for speaker adaptation of continuous density HMMs. Computer Speech Language 9, 171–185.

Marti, U., Bunke, H., 1998. Towards general cursive script recognition. In: Proc. Sixth Internat. Workshop on Frontiers in Handwriting Recognition, Korea.

Marti, U., Bunke, H., 2000. Handwritten sentence recognition. In: Proc. 15th Internat. Conf. on Pattern Recognition, Barcelona, Vol. 3.

Marti, U., Bunke, H., 2001. Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition system. Int. J. Pattern Recognition Artificial Intell. 15, 65–90.

Marti, U.-V., Bunke, H., 1999. A full english sentence database for off-line handwriting recognition. In: Proc. 5th Internat. Conf. on Document Anal. Recognition, Bangalore, Vol. 1.

Plamondon, R., Srihari, S., 2000. On-line and off-line handwriting recognition: A comprehensive survey. IEEE Trans. Pattern Anal. Machine Intell. 22 (1), 63–84.

Rabiner, L., 1989. A tutorial on Hidden Markov Models and selected applications in speech recognition. In: Waibel, A., Lee, K.F. (Eds.), Readings in Speech Recognition. Morgan Kaufmann, Los Altos, CA, pp. 267–296.

Senior, A.W., Robinson, A.J., 1998. An off-line cursive handwriting recognition system. IEEE Trans. Pattern Anal. Machine Intell. 20 (3), 309–321.

Steinherz, T., Rivlin, E., Intrator, N., 1999. Off-line cursive script word recognition – a survey. Internat. J. Document Anal. Recognition 2 (2), 1–33.

Stone, M., 1974. Cross-validatory choice and assessment of statistical prediction. J. Roy. Statist. Soc. 36 (1), 111–147.

Vinciarelli, A., Luettin, J., 2000. Off-line cursive script recognition based on continuous density HMM. In: Proc. 7th Internat. Workshop on Frontiers in Handwriting Recognition, Amsterdam.