Structure Adaptation of HMM applied to OCR

Kamel AIT MOHAND*, Thierry PAQUET*, Nicolas RAGOT**, Laurent HEUTTE*

** Université de Rouen, LITIS EA 4108*
BP12, 76801 Saint Etienne du Rouvray, France
*** Université François Rabelais Tours,*
Laboratoire d'Informatique, EA 2101
64 avenue Jean Portalis, 37200 Tours, France
{thierry.paquet, laurent.heutte}@univ-rouen.fr
kamel.ait-mohand@etu.univ-rouen.fr, nicolas.ragot@univ-tours.fr

Abstract

In this paper we present a new algorithm for the adaptation of Hidden Markov Models (HMM models). The principle of our iterative adaptive algorithm is to alternate an HMM structure adaptation stage with an HMM Gaussian MAP adaptation stage of the parameters. This algorithm is applied to the recognition of printed characters to adapt the character models of a polyfont general purpose character recognizer to new fonts of characters, never seen during training. A comparison of the results with those of MAP classical adaptation scheme show a slight increase in the recognition performance.

1. Introduction

Although proposed a long time ago, Hidden Markov Models (HMM) [1] are still of current interest and widely used for statistical sequence modelling in many application fields. They are particularly popular for applications in Speech and Handwriting recognition. The main interest of such models is their ability to take account of all the information available to build a decision, by the combination of a data model and a model of the expected solutions (the language model in its wider sense).

One of the major difficulties of statistical learning methods (including HMMs) and learning methods in general, is that the training data set may exhibit statistical difference with the data set on which the system is finally used. Pattern classifiers are generally trained so as to correctly identify the widest variety of patterns that can be encountered. This leads to building speaker independent systems, or polyfont systems, or omni writer systems (in case of handwriting recognition). It is generally agreed that such general purpose systems have lower performance than systems that are dedicated to deal with particular shapes, as for example monofont systems or speaker dependent systems. Unfortunately, sufficient labelled data required to train such systems are generally missing. One therefore looks at adaptation techniques that consist in tuning a general purpose system so as to better recognize the new data.

Most of the algorithms proposed in the literature dedicated to HMM adaptation only deal with the adaptation of the data model, by modifying the probability distribution of the data (Gaussian Mixture Models – GMM, in case of continuous density HMM - CDHMM), the HMM structure (the language model) remaining unchanged. In this paper we present a novel algorithm that jointly and iteratively adapts the HMM structure as well as the GMM. We give a full presentation of the algorithm and compare the results to those obtained when using the classical adaptation algorithms for HMM, namely MAP (Maximum A Posteriori) [2].

The paper is organized as follows. In section 2 we give an overview of the learning and adaptation algorithms for HMM. Classical MAP algorithm is detailed in section 3. Section 4 describes the algorithm we propose for jointly adapting HMM structure and parameters. Section 5 presents the OCR system that is

used to test the adaptation algorithm, as well as the data set used for the experiments. Section 6 presents the experimental results.

2. Learning and adaptation of HMM

Once the structure of HMM has been chosen (number of states, size of the Gaussian Mixture, structure of the transition matrix), training the model is performed using the Baum-Welch EM algorithm or the Viterbi EM algorithm. In most cases the HMM structure is tuned by trial and errors over multiple structure configurations. In [3] the authors adjust the number of states of a left / right model to a fraction of the average number of the input frames associated to each character occurrence in the training dataset. Some other authors have introduced algorithms for HMM selection. These algorithms converge to a final optimal structure starting from an initial structure by merging states [4][5], splitting states [6] or doing both operations [7].

Regarding the adaptation algorithms proposed in the literature so far, they have only considered the adaptation of the data model (e.g. GMM). A popular algorithms known as MAP (Maximum a Posteriori) has been proposed in this respect [8], [9]. In the field of handwriting recognition, one interesting study has been devoted to the adaptation of a recognition system to a specific writer [10]. To the best of our knowledge only the work presented in [11] has addressed the structure adaptation of HMM. In this study, the author increments the number of states, at each iteration of the adaptation process, until maximization of the normalized likelihood or recognition rate. At each iteration, the new model is trained with EM. The system is dedicated to the adaptation of a general purpose handwriting recognizer to a particular writer.

3. MAP principle

The MAP adaptation principle is based on the modification of the current parameters of the models (each Gaussian center μ_{im}) using the following updating equation:

$$\hat{\mu}_{im} = \alpha \bar{\mu}_{im} + \beta \mu_{im} \qquad (1)$$

where $\hat{\mu}_{im}$ accounts for the updated Gaussian center, μ_{im} is the actual Gaussian center and $\bar{\mu}_{im}$ is the Gaussian center estimated on the adaptation dataset only.

4. Structure and parameters adaptation of HMM

The proposed algorithm proceeds by iteratively adapting the parameters and then the structure. We restrict our discussion to the case of left / right models as it is the case in speech or text recognition. Parameter adaptation relies on MAP, while structure adaptation of each HMM involves two basic operations i.e. splitting or merging states. We first describe these two basic operations before presenting the algorithm.

4.1. Basic operations on HMM structure

Splitting one state into two states is performed by duplication of the state with the GMM of higher variance. This simply comes from the fact that by allowing one more state, we expect a variance reduction and thus a better model with a higher likelihood on the adaptation data. Self transition probabilities of these states are then updated so as to obtain a mean duration in the two states which is half the duration of the initial split state. If *A* accounts for the self transition probability, then *A/(1-A)* is the mean duration of the state.

Merging two successive states is performed on the two states having the closest emission probability densities (using Kullback-Leibler divergence). As states have the same number of Gaussian components, the data model of the remaining state is obtained by iteratively combining the two closest Gaussian components until the desired number of components is reached. The self transition probability of this new state is computed so that the average length of the new state is the sum of the length of the two initial states.

4.2. Structural MAP algorithm

We now give, in a concise manner, the main steps of the adaptation procedure. Notice that in this algorithm, MAP adaptation is performed on the adaptation database, whereas likelihood of the adapted models is computed on the test dataset.

```
/* Initialisation */
N = # of HMM models
Th = N / 10

for each HMM C, do
   model(C) = MAPadaptation(model(C))

/* Iteration */
T = Th+1;
while T > Th  do
begin
   T= 0
   for each HMM C, do
```

```
   begin
      SplitModel = SplitState(Model(C))
      MergedModel = MergeState(Model(C))
      SplitModel = MAPadaptation(SplitModel)
      MergedModel = MAPadaptation(MergedModel)
      NewModel = ArgMax(Likelihood(Model(C)),
                 Likelihood(SplitModel),
                 Likelihood(MergedModel)
      if NewModel != Model(C) then
         T = T + 1
         Model(C) = NewModel
      end
   end
end
```

5. Application to OCR

We have evaluated the proposed HMM adaptation algorithm to the recognition of printed characters (OCR) with fonts unknown from the system (no training was done using these particular fonts). We briefly describe the general OCR system and the various databases on which the experiments have been conducted.

5.1. Polyfont OCR system

Each text block is first segmented into text lines. Then a sliding window is applied over each text line from left to right and extracts a set of features at each position in the line. The features are similar to those used for handwriting recognition [12], [13]. They account for black pixel density in the window, number of black to white vertical transitions, and for some specific basic features that occur in the window. 30 real valued features are extracted at each position and provide a feature sequence associated to each line of text. Training the HMM character models is performed using embedded Baum-Welch. The number of states for each character is optimized using a validation set according to the method proposed in [4]. The OCR system is case sensitive and includes one inter word space model and punctuation models, for an overall of 69 HMM character models.

Recognition is performed on each line using the Viterbi algorithm. The optimal decoded character sequence is compared to the ground truth label sequence associated to each line image, thus allowing for performance evaluation.

5.2. Adaptation of the polyfont OCR system

After the learning of the polyfont system has completed, we adapt it to an unknown font (Which perhaps does not belong to the training set). The adaptation consists in applying a supervised adaptation algorithm using a data set of 10 images (with their exact transcript) of the unknown font. The adapted system is no longer a polyfont system but is closer to a mofont system specific to the font which the polyfont system has been adapted to (the influence of the size of the adaptation data set and the interest of adapting a polyfont system rather than training from scratch have been studied in [14]).

The performance of the adaptation on this particular font is then tested on images of the same font (but different from the images used during adaptation). We repeat these operations on each of the 30 test fonts.

5.3. Training and test data sets

The system is currently tested using synthesized data that allow for an extended and easy performance evaluation in various configurations. Using a standard text processor it is very easy to generate text line images covering a large set of fonts and for which the ground truth labels are directly available. In order to be as much realistic as possible, the text images are degraded using the degradation model of Baird [15]. In addition, such a database can be reproduced very easily. Images at a resolution of 300 dpi and using 100 fonts of size 12 are generated. 70 fonts will serve for learning, while the remaining 30 will be used for testing. Fonts have been chosen so as to cover the whole family set established by the Vox-Atypi classification [16]. Figure 1 gives some examples from the generated database:

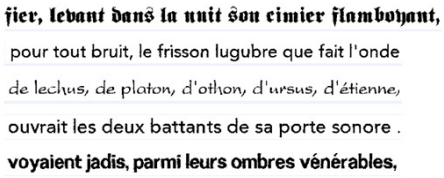

Fig.1: Some examples of the font set.

6. Experimental results

For the experiments we train the general purpose system using a set of 2100 text images covering the 70 fonts selected. The test set comprises 60 text images covering the remaining 30 fonts, whereas the adaptation data sets are made of 10 lines of text of each of the 30 fonts. The following tables report the results obtained respectively using the classical MAP adaptation and the proposed algorithm.

Font	Polyfont	MAP	Struct MAP	Mono Font
Average (30 fonts)	86.59	93.67	96.02	99.55
Berkeley old style	96.62	97.66	97.2	98.98
Banco	34.08	57.14	73.46	98.77
Mistral	46.63	80.61	92.16	94.89
Fette Kanzlei	68.36	90	95.74	99.43

Table 1. Mean performance and detailed for 4 particular fonts (recognition rates are in %).

In mots cases the structural MAP adaptation algorithm outperforms the standard MAP adaptation. With an average improvement of 3% over the 30 fonts evaluated. This improvement is particularly important on the fonts less well recognized by the polyfont system such as "Banco" and "Mistral", for which there is a significant improvement of the adaptation capacity using structural MAP compared to the standard MAP, because structural MAP adds to the efficiency of MAP in adapting the emission probability densities of each state a sort of *model selection* driven by the likelihood which simultaneously optimizes the structure of the HMM models.

But as one can see in table 1 the structural MAP performance are still far from those of monofont systems. This gap leaves place for further research development, if one is considering the goal as to transform a polyfont system into a monofont one using adaptation techniques.

7. Conclusion

We have presented in this paper a new algorithm for the adaptation of HMM models, that jointly and iteratively adapts the HMM structure and the GMMs. We have shown that it can be applied successfully to the recognition of printed characters in order to adapt the character models of a polyfont general purpose character recognizer to new fonts of characters. Future work will concern testing the adaptation algorithm over a larger data set, so as to better understand the exact cause of performance degradation. As for now, we have only considered the MAP adaptation principle as the core of our parameter adaptation scheme, but the literature also report on other adaptation or learning schemes that must be investigated also. Among them we will particularly explore the Maximum Likelihood Linear Regression (MLLR) for adaptation, as well as semi-supervised learning using EM algorithm.

References

[1] Rabiner L., A tutorial on hidden markov models and selected applications in speech recognition, Proc. IEEE 77(2), p. 257-285, 1989.

[2] Gauvain J-L., Lee C.H., Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains, IEEE Transactions on Speech and Audio Processing 2, p. 291–298, 1994.

[3] Zimmermann M., Bunke H., Hidden Markov Model Length Optimization for Handwriting Recognition Systems, IWFHR , pp. 369-374, 2002.

[4] Stolcke, A., Omohundro, S. M., Hidden Markov Model Induction by Bayesian Model Merging, NIPS, pp. 11-18, 1992.

[5] Binsztok H., Artières, T., Learning Model Structure from Data: An Application to On-Line Handwriting, ELCVIA, 5(2):30-46, 2005.

[6] Siddiqi S., Gordon G., Moore A., Fast state discovery for HMM model selection and learning. International Conference on Artificial Intelligence and Statistics, pp. 492–499, 2007.

[7] Brants, T., Estimating Markov Model Structures, ICSLP-96, pp. 893–896, 1996.

[8] Wang S. et Zhao Y., Online bayesian tree-structured transformation of HMMs with optimal model selection for speaker adaptation, IEEE Trans. Speech Audio Processing, vol. 9, pp. 663–677, 2001.

[9] Povey D., Gales M.J.F., Kim D.Y., Woodland P.C., MMI-MAP and MPE-MAP for acoustic model adaptation, EUROSPEECH, pp. 1981–1984, 2003.

[10] Vinciarelli, A. and Bengio, S., Writer adaptation techniques in HMM based off-line cursive script recognition, Pattern Recogn. Lett. 23(8), 905-916, 2002.

[11] Schambach, M., Model Length Adaptation of an HMM based Cursive Word Recognition System, ICDAR, p. 109-114, 2003.

[12] El-Hajj R., Likforman-Sulem L., Mokbel C., Arabic Handwriting Recognition Using Baseline Dependant Features and Hidden Markov Modeling, Proc. ICDAR, pp. 893-897, 2005.

[13] Kessentini Y., Paquet T., Benhamadou A., Off-line handwritten word recognition using multi-stream hidden markov models. Pattern Recognition Letters, 31:60–70, 2010.

[14] Ait-Mohand K., Heutte L., Paquet T., Ragot N.. «Font adaptation of an HMM-based OCR system». Document Recognition and Retrieval XVII, Part of the IS&T/SPIE International Symposium on Electronic Imaging 2010, San Jose, California, USA, January 2010.

[15] Baird H. S., Document Image Defect Models, IAPR Workshop on SSPR, 1990.

[16] Laliberté J., Formes typographiques: historique, anatomie, classification, Presses de l'Université Laval, 2004.