An *a priori* Indicator of the Discrimination Power of Discrete Hidden Markov Models

F. Grandidier [1,2,3] R. Sabourin [1,2] M. Gilloux [3] C.Y. Suen [2]

[1] CENPARMI, Concordia University
1455 de Maisonneuve Blvd West
Montréal, H3G 1M8, Canada
{grandidier, suen}@cenparmi.concordia.ca

[2] LIVIA, ETS
1100, rue Notre Dame Ouest
Montréal, H3C 1K3, Canada
sabourin@gpa.etsmtl.ca

[3] Pôle ILA, SRTP
10, rue de l'le Mabon, BP 86334
44263 Nantes Cedex 02, France
Michel.Gilloux@laposte.fr

Abstract

During the development of a hidden Markov model-based handwriting recognition system, the testing phase takes a non-negligible amount of computation time. This is especially true for real application where the lexicon size is large. In order to shorten the development process we propose an indicator of the system discrimination power. This indicator is calculated during training and its final value is obtained at the end of the training phase, without more calculation. Its definition consists of a modification of the observation probability of the validation corpus by the trained system. Some experiments were carried out and the results show clearly the correlation between this indicator and recognition rates.

1. Introduction

During the past decades hidden Markov models (HMMs) [8] have become a technique widely used in the field of pattern recognition. Their success in speech recognition [1] led many researchers to apply them in the field of handwriting recognition [2, 9, 3, 6]. The main interest of HMMs comes from their ability to model time variant phenomena and also from the existence of effective procedures to automatically adjust model parameters [8], [1]. In the case of real applications with large vocabulary, such as address reading, the testing of a system based on hidden Markov models requires non-negligible computation time. This phase could slow down the development process especially if the system designer wants to experiment several solutions, including different model architectures or different feature sets.

In this paper we want to introduce a new predictor of the quality or discrimination power of hidden Markov models. This evaluation is performed just after the training phase without requiring system testing. Moreover it requires only few calculations carried out during the training phase. The next section will review some basics about HMMs. In Section 3 the definition of a discrimination power estimator is presented. Then some experiments are described in Section 4. Finally some conclusions are presented.

2. Basics of Hidden Markov Models

A good description of hidden Markov models and their use can be found in [8] and [1]. As mentioned in the introduction HMMs are well suit to model time-variant phenomena. Thus, for the 1-dimension HMMs, the data used are observation sequences $O = O_1 O_2 \cdots O_T$, extracted from the stochastic process to be modeled. Each sequence symbol O_t represents the process state at different consecutive moments. In the discrete case, each of those observations comes from a finite alphabet usually called feature set or codebook. Considering models where observations are produced by transitions, the classic A and B matrices are combined into one: C. Then a fully described HMM is characterized by 4 elements:

- N, the number of states, individual states are denoted $S = \{s_0, s_1, \ldots, s_{N-1}\}$ and the state at time t : q_t;
- M, the number of possible observations, individual observations are denoted $V = \{v_1, v_2, \ldots, v_M\}$;
- $C = \{c_{ij}(k)\}, \quad c_{ij}(k) = Pr(O_t = v_k; q_{t+1} = s_j \mid q_t = s_i)$ the probability of going from state s_i at time t to s_j at time $t+1$ and producing at the same moment the observation v_k ;
- $\Pi = \{\pi_i\}$, the initial state distribution.

The most popular method used to perform HMM parameter estimation is the Baum-Welch algorithm. This is an implementation of the EM (Expectation-Maximization) algorithm that performs a Maximum-Likelihood Estimation

of model parameters and guarantees a convergence to a local maximum of the probability of training samples [8]. This algorithm is iterative and requires the calculation of the forward-backward variables at each iteration.

In order to optimize the learning phase, the use of two data corpuses is preferable. The training set is used to reestimate the model parameters while this new model is evaluated on a validation set after each iteration. Finally the process stops when the improvement of the validation corpus likelihood by the model falls below a given threshold. This strategy allows to optimize model generalization over unknown examples.

The recognition problem is usually solved using the Viterbi algorithm [8], [1]. This procedure looks for the best state sequence for a given observation sequence and a letter sequence. In a real application a test could be run over 1000 different letter sequences, then this procedure is time consuming. The testing of our system [3], using a set of 4674 city names, with a lexicon size of 1000, takes close to 3 hours on a Sun Enterprise 450 (400 MHz).

3. Definition of a new indicator of discrimination power

Our aim is to define an indicator of the discrimination power of the trained HMMs without resorting to the test of the system. Such a strategy will permit to gain computational time and speed up the process of development of a new recognition system especially when the designer wants to evaluate many different models. As we mentioned in the previous section, the use of two corpuses of data during system learning allows the optimization of the system discrimination power. We want to characterize this discrimination power with the help of the likelihood of the validation corpus.

3.1. Definition of the likelihood of the validation corpus

During system learning, the validation corpus is used to test the system improvement after each iteration. In fact the increase of the validation corpus likelihood is measured. This quantity is evaluated with the help of the *forward* procedure. Let $Pr(O \mid \lambda)$ be the probability of the observation sequence $O = O_1 O_2 \cdots O_T$ by the model λ, the *forward* algorithm allows to calculate iteratively this quantity. Let $\alpha_t(i)$ be the probability of emitting the partial observation sequence $O_1 O_2 \cdots O_t$ and to be in state s_i at time t. Considering an initial state (0) and models where observations are produced by transitions, the *forward* procedure differs from the version usually presented in the literature [8]:

- Initialization: $t = 0$

$$\begin{cases} \alpha_0(0) &= 1 \\ \alpha_0(i) &= 0 \qquad \text{for} \quad i \neq 0 \end{cases} \tag{1}$$

- Induction: $j = 0, 1, \ldots, N-1$ and $t = 1, 2, \ldots, T$

$$\alpha_t(j) \;=\; \sum_{i=0}^{N-1} \alpha_{t-1}(i)\, c_{ij}(O_t) \tag{2}$$

- Termination

$$Pr(O \mid \lambda) \;=\; \sum_{i=0}^{N-1} \alpha_T(i) \tag{3}$$

These equations show clearly that the observation probability $Pr(O \mid \lambda)$ is a product of probabilities $c_{ij}(O_t)$. Thus $Pr(O \mid \lambda)$ is strongly dependent on the observation sequence length and also from the mean value of the $c_{ij}(O_t)$ probabilities. During training, this observation probability is calculated for each validation example i, with the help of its labeling w_i. Thus the correct notation of this quantity must be $Pr(O^i \mid w^i, \lambda)$. For a validation corpus of Q examples its log likelihood P_V is defined as:

$$P_V \;=\; \frac{1}{Q} \sum_{i=1}^{Q} \log Pr(O^i \mid w^i, \lambda) \tag{4}$$

In fact this quantity represents the mean log probability of an observation sequence from the validation corpus. The function $\log$ is used in order to get a consistent value. The system parameters conducing to the best value of P_V during training correspond to the trained or final system.

In order to evaluate and compare the quality of different trained systems, we can consider to use their likelihood values P_V. The best system must have the greater likelihood value. However, given a data corpus, this quantity depends on $c_{ij}(O_t)$ and this latter depends strongly on M, the number of possible observations. Given a transition $i \rightarrow j$, its mean probability $\overline{c_{ij}}(O_t)$ with no *a priori* information is equal to $\frac{1}{M}$. Then P_V can be used only to compare systems sharing the same number of possible observation symbols.

In order to go over this problem, this likelihood must be modified to be comparable. For a given validation corpus and different systems, the only constant in the expression $Pr(O^i \mid w^i, \lambda)$ is the labeling w^i. Then it is found to modify the likelihood expression according to this remark. The evaluation of the *a posteriori* probability of the labeling w^i given the observation sequence and the model: $Pr(w^i \mid O^i, \lambda)$ is more consistent to compare different systems. This *a posteriori* probability is related to the observation probability according to Bayes rule:

$$Pr(w^i \mid O^i, \lambda) \;=\; \frac{Pr(O^i \mid w^i, \lambda) \times Pr(w^i)}{Pr(O^i)} \tag{5}$$

Here $Pr(w^i)$ is the *a priori* probability of the labeling also called language model. For any application this quantity is difficult to evaluate. As it is constant for all systems, it does not influence comparison results, thus it can be discarded. The denominator $Pr(O^i)$ is the *a priori* observation sequence probability, *i.e.* before the system parameter estimation. This quantity is directly related to the number of features M. For different systems, the range of this probability differs but corresponds to a same state. This quantity can be considered as a reference or a normalization factor.

3.2. Definition of the new indicator and its properties

We proposed to use as indicator of the discrimination power of a trained system λ, the mean over the validation corpus of the modified *a posteriori* probability or more formally:

$$F = \frac{1}{Q}\sum_{i=1}^{Q}\log\frac{Pr(O^i \mid w^i,\lambda)}{Pr(O^i)} \quad (6)$$

As a validation step is performed during training, the observation probabilities $Pr(O^i \mid w^i,\lambda)$ are available from the *forward* procedure. $Pr(O^i)$ can be estimated in several ways. With really no information, equiprobability between all features can be used. With the help of a data set, its estimation can be obtained by computing feature n-grams. The $Pr(O^i)$ are calculated for each validation example and stored before the first training iteration. Thus, the evaluation of our indicator F needs only few computations at each iteration.

As mentioned previously, the training procedure guaranties an increase of the observation probability, then F will increase during this phase. It will reach its maximum for the best observation probability value. This point corresponds to the best system parameter set $\hat{\lambda}$. Depending on the denominator evaluation, F values could be positive or negative. As long as it is negative, the denominator evaluation is better than the model observation probability.

4. Experimentation

In order to evaluate this indicator, a discrete HMM-based off-line handwriting recognition system, using analytic approach with explicit segmentation [3] was used. A strategy was developed to improve its performance [4]: features with low discriminative power are gathered in feature classes, then each of them is substituted by a new feature set. The increase of the number of features replacing one class allows the construction of different systems. We want to evaluate and compare those systems without performing tests, in order to reduce the computation time. Feature sets are obtained by increasing the number of clusters during the vector quantization procedure. As the LBG algorithm [7] is used, the number of centroids is a power of 2. Seven systems were built using respectively 100 (84+16), 116 (84+32), 148, 212, 340, 596 and 1108 features.

The observation sequence probabilities were evaluated in four different ways considering: equiprobability between features, feature frequencies, feature bigrams and feature trigrams (evaluated on the training data set). In order to avoid data sparseness for n-grams evaluation, we use the Katz back-off model [5] for distribution smoothing. It is based on the Good-Turing smoothing principal. The Good-Turing estimate states that for any n-gram that occur r times, we should consider that it occurs r^* times: $r^* = (r+1)\frac{n_{r+1}}{n_r}$, where n_r is the number of n-grams that occur exactly r times. Then the distributiuon is evaluated according to this new count. Katz smoothing extend this idea by adding the combination of higher-order models with lower-order models. In the case of bigrams, those with non-zero count are discounted according to a discount ratio d_r. Then the subtracted counts are distributed among the zero count bigrams according to the next lower-order distribution (the unigram model here). Moreover, the d_r are calculated taken in account the reliability of the count. Large counts are considered reliable, so they are not discounted: $d_r = 1$ for all $r > k$, with k classicaly in the range of 5 to 8. The discount ratio for the lower counts are derived from the Good-Turing estimate applied to the global bigram distribution. Thus the Katz bigram: $Pr_K(O_i \mid O_{i-1})$ is obtained from the equations below.

$$\begin{cases} C(O_{i-1},O_i)/C(O_i) & \text{if } r > k \\ d_r C(O_{i-1},O_i)/C(O_i) & \text{if } k \geq r > 0 \\ \alpha(O_{i-1})P(O_i) & \text{if } r = 0 \end{cases} \quad (7)$$

where

$$d_r = \frac{\frac{r^*}{r} - \frac{(k+1)n_{k+1}}{n_1}}{1 - \frac{(k+1)n_{k+1}}{n_1}} \quad (8)$$

$$\alpha(O_{i-1}) = \frac{1 - \sum_{O_i:r>0} Pr_K(O_i \mid O_{i-1})}{1 - \sum_{O_i:r>0} Pr(O_i)} \quad (9)$$

Katz smoothing for higher n-grams is defined according to the Katz $(n-1)$-gram model. In our experiments, the value of the parameter k has been chosen empiricly and fixed to 6.

At the end of each iteration of the training phase, the four different estimations of F are computed. Experiments were performed using three data sets: the learning set containing 12023 examples, the validation set (3475 examples) and the test set (4674 examples). In table 1 each column characterizes one of the seven built systems, the lines show respectively the validation corpus likelihood, the different evaluation of F, both at the training end. The last four lines

Number of Features	100	116	148	212	340	596	1108
P_V	-23.76	-25.58	-27.39	-29.25	-31.29	-33.36	-35.52
F_{Equip}	13.61	12.70	12.34	12.65	13.45	14.75	16.33
$F_{Unigram}$	7.30	7.69	8.04	8.36	8.48	8.56	8.55
F_{Bigram}	6.38	6.57	6.76	6.86	6.83	6.81	6.75
$F_{Trigram}$	5.74	5.87	6.10	6.26	6.42	6.71	6.96
RR (Lex. 10)	98.67	98.82	98.74	98.86	98.84	98.78	98.70
RR (Lex. 100)	95.27	95.32	95.96	95.96	95.85	95.72	95.67
RR (Lex. 1000)	87.59	88.49	88.83	89.80	89.18	89.13	88.53
RR (Lex. 5000)	78.80	80.04	81.05	82.05	81.80	81.48	81.01

Table 1. Evaluation of the indicator and recognition rate for several systems

are the recognition rates RR obtained with several lexicon sizes.

First, we can observe the evolution of P_V for the different systems. As we mentioned before, the codebook size influences this measure and the effect of this system input can be directly estimated here: the increase of feature number involves a reduction of the likelihood.

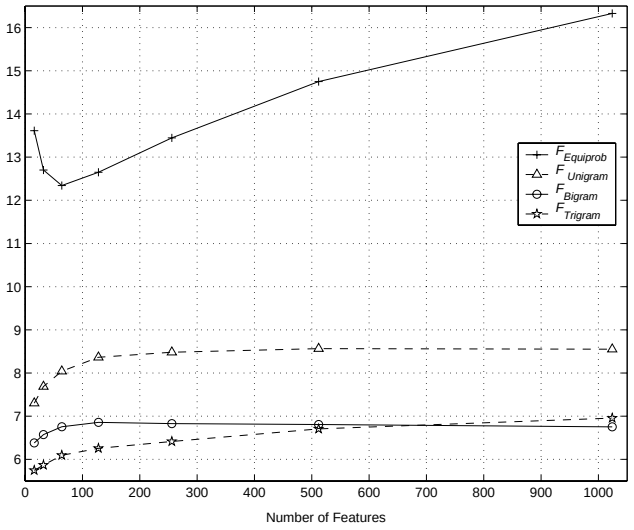

Figure 1. Indicator values estimated in three different ways

In figure 1, the four series of indicator evaluation are presented. The curves are not in the same range, this is due to the $Pr(O^i)$ estimation. When equiprobability is considered, the estimation is really raw and the observation probability is low. As we refine the estimation technique, the observation probability increases and our indicator value decreases. Equiprobabilty estimation is not appropriate for our improvement strategy because feature probabilities from the new set are considered to be equal to those from the primary set. This explains the behavior of the associated curve. The three other curves show also a different behavior: only the bigram evaluation curve gets a maximum. When the number of feature becomes large, with the same data, the estimation of probability suffers from the lack of data. In the case of the frequency estimation, the increase of feature conduct to a lower reliable probability estimation. The use of a smoothing technique during n-gram evaluation allows to overtake this problem. However for the trigram evaluation, when the number of feature becomes large, the ratio of observed trigrams becomes really low (up to 30% for the 3 first systems and lower to 5% for the 2 last systems). The smoothing technique used can not interpolled correctly for trigrams. To obtain a better estimate, we must increase the data set size. Thus, the bigram evaluation gives the best estimate of the observation probability.

In figure 2 the recognition rate curves are presented. The x-axis representes the number of features used by each system. We can clearly observe a decrease of the recognition rate curves when the number of features becomes large. This phenomena is called over-learning: the number of system parameters is too large for the training data. In such a case the system memorizes the learning set and cannot generalize correctly on the unknown data. With the help of the two figures, we can notice the correlation between recognition rates and the F_{Bigram} values. This observation confirms the hypothesis that our indicator is a good estimation of the discrimination power and also that it could be used to detect the over-learning problem.

5. Conclusion

In this paper we introduce a new indicator to evaluate the discrimination power of a recognition system based on HMMs. This indicator is calculated with the help of the *forward* algorithm during the learning phase. This proce-

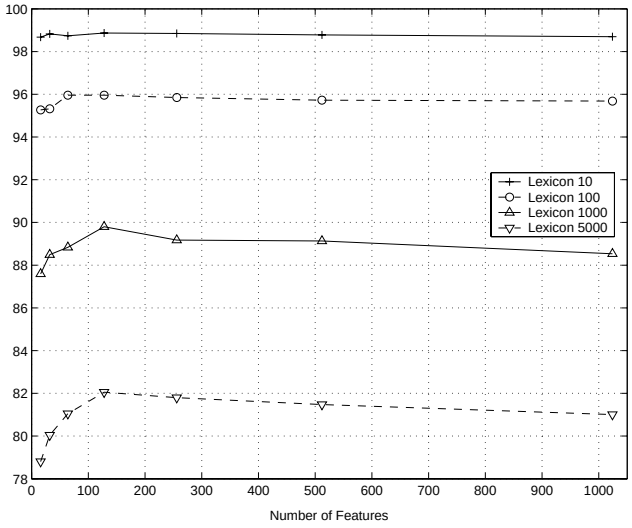

Figure 2. System recognition rates for several lexicon sizes

dure allows the observation probability calculation of each example. This quantity is modified using the *a priori* observation sequence probability in order to obtain a modified *a posteriori* estimation of the labeling. Our new estimator is then defined as the mean value of this quantity over the validation corpus. All calculations of this indicator are done with the help of data not used for the parameter estimation then it is possible to consider it as discrimination power estimator.

Several experiments were carried where the number of features used by the handwriting recognition system is increased. The indicator values are obtained in several ways. The use of smooth bigrams to estimate the denominator give the best estimate. After training we compare the value of our estimator of discrimination power using the recognition rates and concluded that there is a correlation between those two indicators. The estimator defined in (6) gives information about the discrimination power of a discrete HMM-based recognition system just after its learning phase. Its main interest is that during the system development, it could be used to avoid some testing and also detect over-learning problems. Moreover the test set can be preserved for the final test of the system and thus saves the designer from having to tune its system parameters according to this data set.

Our study was conducted in a discrete framework. A possible future work is to extend this discrimination power estimator to the continuous case. We use a smoothing technique during the n-gram estimation. Another could be used to evaluate the trigram in a better way and thus obtain a better probability density estimation from small amount of data.

6. Acknowledgements

This work was supported by the Service de Recherche Technique de La Poste (SRTP) at Nantes, France, the Ecole de Technologie Supérieure and the Centre for Pattern Recognition and Machine Intelligence at Montréal, Canada.

References

[1] L. Bahl, F. Jelinek, and R. Mercer. A maximum likelihood approach to continuous speech recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 5(2):179–190, 1983.

[2] M. Chen, A. Kundu, and J. Zhou. Off-line handwritten word recognition using a hidden markov model type stochastic network. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(5):481–496, 1994.

[3] A. El-Yacoubi, M. Gilloux, R. Sabourin, and C. Suen. An hmm-based approach for off-line unconstrained handwritten word modeling and recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(8):752–760, 1999.

[4] F. Grandidier, R. Sabourin, C. Y. Suen, and M. Gilloux. A new strategy for improving feature sets in a discrete hmm-based handwriting recognition system. In *Proc. of the 7th International Workshop on Frontiers in Handwriting Recognition*, pages 113–122, Amsterdam, Netherlands, September 11-13, 2000.

[5] S. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 35(3):400–401, March 1987.

[6] J. Kim, K. Kim, and C. Suen. An hmm-mlp hybrid model for cursive script recognition. *Pattern Analysis and Applications*, 3(4):314–324, 2000.

[7] Y. Linde, A. Buzo, and R. Gray. An algorithm for vector quantizer design. *IEEE Trans. on Communication*, 28(1):84–95, 1980.

[8] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285, 1989.

[9] A. Senior and A. Robinson. An off-line cursive handwriting recognition system. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(3):309–321, 1998.