PHONETIC RECOGNITION USING HIDDEN MARKOV MODELS AND MAXIMUM MUTUAL INFORMATION TRAINING

Bernard MERIALDO

IBM-France Scientific Center
36 avenue Raymond Poincare, 75116 Paris FRANCE
Tel: (1) 45 05 14 00

ABSTRACT

In this paper, we study the application of Maximum Mutual Information training to Hidden Markov Models for phonetic recognition. MMI training has been proposed as an alternative to standard Maximum Likelihood (ML) training. In practice, MMI training performs better (produces models that are more accurate) than ML training.
In the first three paragraphs, we review the fundamental notions of training HMM, ML and MMI training. In the fourth and fifth paragraphs, we show how MMI training can be applied easily to the case of phonetic models and phonetic recognition. We propose some computational heuristics to implement these computations practically. In the last paragraph, we detail some experiments (training and recognition) that show that the phonetic error rate decreases significantly when MMI training is used, as compared with ML training.

INTRODUCTION

Hidden Markov Models (HMM) are now a standard in Speech Recognition. Originally, the interest for HMM was motivated by the existence of an efficient training procedure, the Baum-Welch or Forward-Backward algorithm. Given the structure of a HMM (the states and transitions) and some training data, this procedure allows to find the best values of the parameters of the HMM (the probability distributions), according to a Maximum Likelihood (ML) criterion. The convergence of this iterative procedure to a local maximum of the objective function is guaranteed by an inequality discovered by Baum [1].
Besides its computational interest, ML training has motivated some theoretical research to study its optimality in terms of recognition accuracy. A. Nadas [6] has proved that under certain conditions, ML training will find the optimal model. These conditions are:

1. the optimal model belongs to the family of models considered,
2. the true language model is known,
3. the training sample is large, and
4. the performance of the recognizer does not get worse when the trained model comes closer to the optimal model.

Unfortunately, these conditions are very unlikely to be satisfied in the case of Speech Recognition. For example, speech is certainly not produced by a Markov model, and the language models that we can build are only approximations of the reality. In this situation, as pointed by P. Brown [5], some arguments suggest that ML training may not lead to the best possible model, the one which maximizes recognition. Other techniques have been proposed to train HMM, with the intention of maximizing recognition:

- L. Bahl et al. [3], have proposed the Maximum Mutual Information (MMI) training, which tries to maximize the Mutual Information between the text and the acoustic observation. Their experiments suggest that MMI training leads to better performance than ML training.
- L. Bahl et al. [4], have recently proposed a method called 'corrective training', which, despite the lack of theoretical foundations and proof of convergence, seems to give results better than MMI in practice.

In this paper, we study the application of MMI training to phonetic models and phonetic recognition. We first recall the basic formulas for ML and MMI training. We show that, for phonetic recognition, the terms involved in the reestimation formulas for MMI can be efficiently computed using the Looped Phonetic Model (LPM). We also consider some of the computational problems that may occur when performing MMI training, and we propose a heuristic method to overcome them. Finally we describe a set of experiments that show that MMI training improves significantly the phonetic recognition rate over the standard ML training.

MAXIMUM LIKELIHOOD TRAINING

In this paragraph, we recall some basic results on the training of HMM. According to the Information Theory formulation of the Speech Recognition problem [2], the optimal recognizer is the one that, given some acoustic observation A, produces the text $\hat{W}$ such that:

$$p(\hat{W} \mid A) = \max_{W} \frac{p(A \mid W) \cdot p(W)}{p(A)}$$

Designing a speech recognizer consist in defining the acoustic model (which allows to compute $p(A \mid W)$) and the linguistic model (which computes $p(W)$).
The linguistic model is generally obtained independently, by statistics over large samples of texts related to the task that is considered, and therefore does not involve speech data. We do not consider the problem of training language models here.
In this paper, we suppose that the acoustic model is based on Hidden Markov Model (HMM). That is, we have a process to construct a HMM from the text W (for example by concatenating phonetic Markov Machines), and we compute $p(A \mid W)$ as the probability of emission of the observation $A = y_1^T = y_1 y_2 \ldots y_T$ by this Markov model. If we call a_{ij} the probability of transition from state i to state j in this model, b_{ijy} the probability of emitting the symbol y when going from i to j, and c_i the probability that the model is initially in state i, then we have:

$$p(A \mid W) = \sum_{i_1} \cdots \sum_{i_T} c_{i_1} \prod_{j=1}^{T} a_{i_j i_{j+1}} \cdot b_{i_j i_{j+1} y_j}$$

We note θ the vector of parameters $\theta = (a_{ij}, b_{ijy}, c_i)$, and we note $p_\theta(A \mid W)$ to recall the dependency on θ.
The problem of training HMM is, given some recording A_T corresponding to some text W_T, to compute the "best" values of the parameters a_{ij}, b_{ijy} and c_i.
The Maximum Likelihood (ML) training tries to find the values that maximize $p_\theta(A_T \mid W_T)$. The Forward-Backward (FB) algorithm is an iterative procedure that, given initial estimates for $\theta = (a_{ij}, b_{ijy}, c_i)$, produces a new parameter vector $\bar{\theta} = (\bar{a}_{ij}, \bar{b}_{ijy}, \bar{c}_i)$ such that:

CH2561-9/88/0000-0111 $1.00 © 1988 IEEE

$$p_{\bar{\theta}}(A_T|W_T) \geq p_\theta(A_T|W_T)$$

The iteration converges then to a local maximum of $p_\theta(A_T|W_T)$.
We detail the computation of the FB reestimate for a_{ij}. We introduce the standard variables α and β defined by:

$$\begin{cases} \alpha_i(t) = \text{probability that the model is in state i and has produced } y_1^t \\ \beta_i(t) = \text{probability that the model will produce } y_{t+1}^T \text{ when starting from state i} \end{cases}$$

(The values of α's and β's can be computed easily using recurrence relations).
We are able to compute the average count of transition i-j during the production of $A = y_1^T$:

$$c_{ij} = \frac{\sum_t \alpha_i(t-1) \cdot a_{ij} \cdot b_{ijy_t} \cdot \beta_j(t)}{p_\theta(A_T|W_T)}$$

The FB reestimate $\bar{a}_{ij}$ is proportional to c_{ij}:

$$\bar{a}_{ij} = \frac{c_{ij}}{\sum_j c_{ij}}$$

Note that:

$$\sum_{i,j} c_{ij} = T$$

and that:

$$\frac{\partial p_\theta(A_T|W_T)}{\partial a_{ij}} = \frac{1}{a_{ij}} \cdot p_\theta(A_T|W_T) \cdot c_{ij}$$

Similar formulas hold for b_{ijy} and c_i.

MAXIMUM MUTUAL INFORMATION TRAINING

In MMI training as proposed by L. Bahl [4] and P. Brown [5], the objective is to maximize the mutual information between the text and the acoustic observation:

$$\begin{aligned} I_\theta(W,A) &= \log \frac{p(A,W)}{p(A) \cdot p(W)} \\ &= \log p_\theta(A|W) - \log p(A) \\ &= \log p_\theta(A|W) - \log \sum_W p_\theta(A|W) \cdot p(W) \end{aligned}$$

The rationale for MMI training is to minimize the amount of information needed to specify W when A is known (the conditional entropy $H(W|A)$).
Another presentation is that we would like the decoder to recognize perfectly (or with minimum error) the training data, that is we would like:

$$p_\theta(W_T|A_T) \geq p_\theta(W|A_T) \qquad \forall W$$

A plausible way to go towards this objective is to make $p_\theta(W_T|A_T)$ as large as possible, that is, to maximize:

$$\begin{aligned} p(W_T|A_T) &= \frac{p_\theta(A_T|W_T) \cdot p(W_T)}{p(A_T)} \\ &= \frac{p_\theta(A_T|W_T) \cdot p(W_T)}{\sum_W p_\theta(A_T|W) \cdot p(W)} \end{aligned}$$

Since $p(W_T)$ does not depend on θ, this is the same formulation than MMI training.

For MMI training, we do not know of a procedure similar to the Forward-Backward algorithm, so that we solve this optimization problem using standard gradient search techniques with projection on the constraints. We consider as objective function:

$$F(\theta) = \log p_\theta(A_T|W_T) \cdot p(W_T) - \log \sum_W p_\theta(A_T|W) \cdot p(W)$$

the constraints are:

$$\sum_j a_{ij} = 1 \quad \forall i; \qquad \sum_y b_{ijy} = 1 \quad \forall i,j; \qquad \sum_i c_i = 1$$

We detail the computation for the coordinate a_{ij}[1] (computations for b_{ijy} and c_i are similar). The coordinate of the gradient corresponding to a_{ij} is:

$$\begin{aligned} \frac{\partial F(\theta)}{\partial a_{ij}} &= \frac{\frac{\partial p_\theta(A_T|W_T)}{\partial a_{ij}}}{p_\theta(A_T|W_T)} - \frac{\sum_W \frac{\partial p_\theta(A_T|W)}{\partial a_{ij}} \cdot p(W)}{\sum_W p_\theta(A_T|W) \cdot p(W)} \\ &= \frac{1}{a_{ij}} \cdot (c_{ij} - c'_{ij}) \end{aligned}$$

where c_{ij} is exactly the average count of transition i-j in the FB reestimation of a_{ij} for the text W_T, and c'_{ij} looks like a weighted average of average counts for FB reestimates for all possible texts W.
This second term c'_{ij} is rather complex to compute, since the summation is taken over all possible sentences W. In [4], L. Bahl et al. used MMI training for word models, and they approximated the second term by reducing the summation to the only words that were acoustically confusable (according to their recognizer) with the training words.

MMI TRAINING FOR PHONETIC MODELS

In our case, we are interested by phonetic recognition. Therefore, a sentence W is a sequence of phonemes ϕ_1^n. Our standard language model is based on diphone probability:

$$p(W) = p(\phi_1^n) = \prod_{i=1}^n p(\phi_i|\phi_{i-1})$$

(assuming a special phoneme ϕ_0 at the beginning of the sentence).

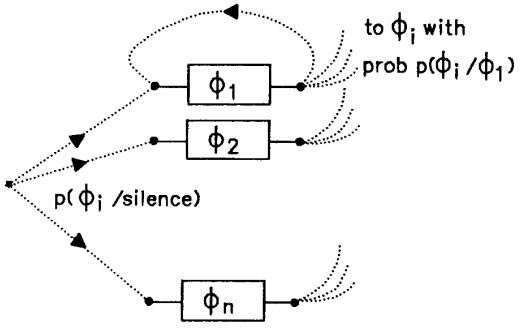


Figure 1: Looped Phonetic Model

1 The precise mathematical formulas for MMI are complicated [4,5] by the tying of the distributions (for example, using several copies of a phonetic machine in the Markov models for different W). For simplicity, we assume that the subscript ij will refer to the set of all transitions tied to transition i-j. The reader is invited to write the exact formulas by himself.

Each phoneme is represented by a Markov machine. We have a set of 40 machines, including silence [8].

The recognition of the most probable sequence of phonemes is performed by a Viterbi search in the Looped Phonetic Model (LPM). To build the LPM, we place all the phonetic machines in parallel, and add null transitions from the final state of every phonetic machine to the starting state of every phonetic machine. The probability of the transition that goes from phoneme ϕ to phoneme ϕ' is taken equal to $p(\phi' \mid \phi)$.

We can remark that:

$$\sum_{W} p_\theta(A_T \mid W) \cdot p(W) = p_\theta(A_T \mid LPM)$$

so that the LPM allows to compute in one shot the second term of the expression of the MMI, although the summation is taken over all possible sequences of phonemes.

This also implies that the term c'_{ij} involved in the computation of the gradient is simply the average count occurring in the FB reestimation for the LPM. This gives us an efficient way of computing directly the value of these sums. Therefore, in the case of phonetic recognition, the computation of the gradient consists only of two Forward-Backward reestimations, one on the training sentence W_T, the other on the LPM.

COMPUTATIONAL VARIANTS

The gradient coordinates can be expressed as:

$$\frac{(c_{ij} - c'_{ij})}{a_{ij}}$$

where c_{ij} and c'_{ij} are the corresponding FB average counts. We can remark that the module of the gradient is not bounded: both c_{ij} and c'_{ij} have values between 0 and T, but we divide by a_{ij} which can have any positive value.

In practice, we encountered some numerical difficulties when computing the gradient, because certain low value of a_{ij} caused the corresponding gradient coordinate to be very high, so that, after projection on the constraints, only these low values were affected by the gradient search algorithm. But these values are unreliable (we don't have sufficient training data to estimate very low probabilities), and they correspond to transitions that are used very little during the production of A_T. Intuitively, we would prefer the gradient search to modify the largest values of a_{ij}, those which are used often during the production of A_T. To avoid these numerical problems, we tried different heuristic variants. One was to replace the gradient by the vector with coordinates:

$$\frac{(c_{ij} - c'_{ij})}{f(a_{ij})}$$

where f could be any positive increasing function, used to weight the influence of low values of a_{ij}. This vector is bounded, so that we can expect the interference caused by low a_{ij} values will disappear. On the other hand, since the scalar product of this vector with the gradient is positive, we are sure that the objective function will also increase if we move in this new direction (simply the increase may be less important than in the direction of the gradient). The choice $f(x) = x$ corresponds to the gradient itself, and we tested the choice $f(x) = 1$.

However, the direction vector:

$$v_{ij} = \left(\frac{c_{ij}}{\sum_j c_{ij}} - \frac{c'_{ij}}{\sum_j c'_{ij}}\right)$$

experimentally provided a faster convergence. Although it is not guaranteed that this direction will always increase the objective function, we can expect that the counts $\sum c_{ij}$ and $\sum c'_{ij}$ will be similar (at least if the initial model is sufficiently good), so that v_{ij} has the same sign that the corresponding gradient coordinate. In practice, it is always the case that the objective function increased in this direction. The following experiments use this last possibility for the direction vector.

EXPERIMENTS

We have tested this method on a set of 4 male speakers, SVS, MOY, CRI and FAR. Each speaker pronounced a training text T_{200} of 200 phonetically balanced sentences, a training text T_{250} of 250 sentences designed for contextual phonemes [11], a training text T_{210} of 210 sentences extracted from letters, and a test text of 79 sentences. All these texts were pronounced in Isolated Syllable mode, that is we require the speaker to make a short pause after every syllable. There was no minimum required length for a pause between two syllables. In fact, some speakers tend to make significant pauses (10-20 centiseconds), while others almost made no pause, talking in "Connected Syllable" mode.

For speaker SVS, speech was recorded at 10kHz, and processed using the system described in [8]. For speakers MOY, CRI and FAR, speech was recorded using the Yorktown acoustic front-end [7]. It consists of a PZM desk microphone, an A/D converter that samples the speech signal at 20 kHz, and a PIE signal processor that performs a FFT analysis followed by an ear-model and vector quantization to produce the acoustic observation.

For each speaker, a standard ML training was performed using FB algorithm (on T_{200} and T_{210} for SVS, on T_{200} and T_{250} for the others). The resulting model was used as the starting point for the MMI training. MMI training used T_{200} and T_{210} for SVS, and T_{200} for the others. The gradient search was conducted as follows:

1. compute the new direction vector, this requires two FB reestimations for the acoustics, one corresponding to the correct sequence of phonemes, one corresponding to the LPM.
2. move the model from a given step along this direction vector and compute the Mutual Information of the new model,
3. if the Mutual Information has increased, iterate with this new model, otherwise reduce the step and retry,

This procedure stops when the improvement on the Mutual Information falls under a specified threshold, or when a maximum number of iterations has been reached, or when too much computer time has been spend. At each step of this iteration, we perform a phonetic recognition on the test data to study its evolution (but we do not use this result in the training itself). The following figures show the evolution of the Mutual Information (increasing curve) and the phonetic error rate (decreasing curve) along the iterations of the MMI training. (some MMI training were stopped rapidly because of computational constraints, others were prolongated to study their asymptotic behavior).

We indicate in a table the numerical values of:

- the phonetic recognition rate,

$$\frac{\text{nb. phonemes recognized}}{\text{nb. phonemes}}$$

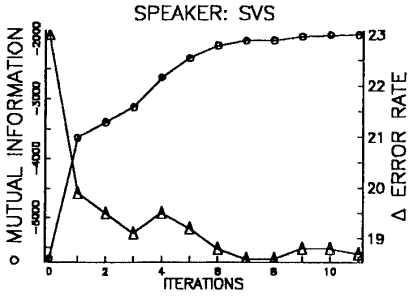
- the insertion rate,

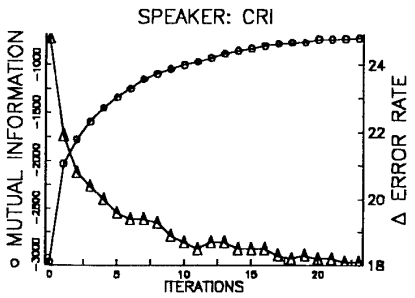
$$\frac{\text{insertions}}{\text{nb. phonemes}}$$

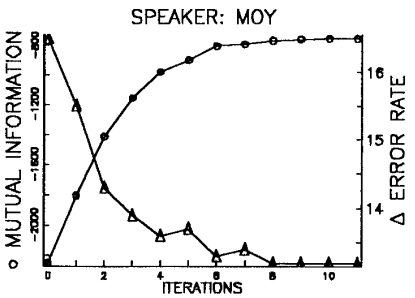
- the global error rate, computed as in [10]:

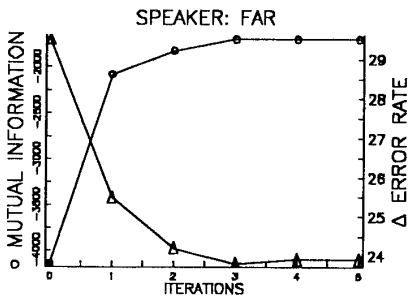
$$\frac{\text{substitutions + deletions + insertions}}{\text{nb. phonemes + insertions}}$$

after ML training and MMI training.









speaker	rec.	ins.	errors
CRI ML	84.1	11.8	24.8
MMI	85.2	4.1	18.1
SVS ML	82.2	6.7	23.0
MMI	84.2	3.6	18.7
FAR ML	79.4	12.6	29.5
MMI	80.4	5.7	23.9
MOY ML	88.2	5.7	16.5
MMI	89.7	3.3	13.2
avg. ML	83.48	9.2	23.45
MMI	84.8	4.18	18.48

In all cases, MMI training was able to reduce the error rate by several percents for every speaker. By looking at the results, most of the gain comes from the reduction of the number of insertions, 5% on the average. The recognition itself raises by 1.3%. On the average, the global error rate is reduced by 5%.

CONCLUSION

We have studied a training procedure based on Maximum Mutual Information (MMI) which attempts to give models that are more accurate that the standard Maximum Likelihood training. We have shown how to apply this procedure in the case of phonetic models and phonetic recognition, and we have given certain heuristics to avoid some numerical problems that may appear in the gradient search.
We have experimented this procedure on a set of 4 different speakers. For every speaker, MMI training builds a model that has a better performance than ML training. Most of the gain is obtained by a decrease of the number of insertions. On the average, the gain on the global error rate is 5%.

Bibliography

[1] L. Baum, *An inequality and association Maximization technique in Statistical Estimation for Probabilistic Function of Markov Processes*, Inequality, Vol III, 1972, pp 1-8.

[2] L. Bahl, F. Jelinek, R. Mercer, *A Maximum likelihood Approach to Continuous Speech Recognition*, IEEE Trans on PAMI, PAMI-5 No 2, March 83.

[3] L. Bahl, P. Brown, P. de Souza, R. Mercer, *Maximum Mutual Information of Hidden Markov Model Parameters*, ICASSP 86, Tokyo, vol 1 pp 49.

[4] L. Bahl, P. Brown, P. de Souza, R. Mercer, *Estimating HMM parameters so as to maximize Speech Recognition accuracy*, Research Report RC-13121, 9/10/87, IBM TJ Watson Research Center, PO Box 218, Yorktown Heights, NY10598.

[5] P. Brown, *The Acoustic-Modeling Problem in Automatic Speech Recognition*, PhD Dissertation, Carnegie Mellon University, May 87.

[6] A. Nadas, *A decision theoretic formulation of a training problem in speech recognition and a comparison of training by unconditional versus conditional maximum likelihood*, IEEE trans. on ASSP, ASSP-31 No 4, August 1983, pp 814-817.

[7] A. Averbuch et al., *An IBM PC Based large-vocabulary isolated-utterance speech recognizer*, ICASSP 86, Tokyo, vol 1 pp 53.

[8] H. Cerf-Danon, A-M Derouault, M. El-Beze, B. Merialdo, S. Soudoplatoff, *Speech Recognition experiment with 10,000 word vocabulary*, NATO Advanced Institute on Pattern Recognition, June 18-20, 1986, Brussels..

[9] B. Merialdo, *Speech Recognition using Very Large Size dictionary*, ICASSP 87, Dallas.

[10] R. Schwartz, Y. Chow, F. Kubala, *Rapid speaker adaptation using a probabilistic spectral mapping*, ICASSP 87, Dallas, vol 2, pp 633-636.

[11] A-M. Derouault, *Context-dependent phonetic Markov models for Large Vocabulary speech recognition*, Proc. of NATO Advanced study Institute on Speech Understanding, 6-18 July 1987, Ed. Springer-Verlag.