NOISE ROBUST SPEECH RECOGNITION WITH STATE DURATION CONSTRAINTS

Kari Laurila

Speech and Audio Systems Laboratory,
Nokia Research Center, Tampere, Finland
kari.laurila@research.nokia.com

ABSTRACT

In this paper, we present a method to incorporate and re-estimate state duration constraints within the Maximum Likelihood training of hidden Markov models. In the recognition phase we find the optimal state sequence fulfilling the state duration constraints obtained in the training phase. Our target is to get speaker-dependent training and recognition perform well with a very small amount of training data in the case of mismatch between the training and testing environments. We take advantage of the fact that speakers tend to preserve their speaking style in similar situations (e.g. when speaking to a machine) and our main means to reach the target is to force similar state segmentations in the training and recognition phases. We show that with the proposed method we can substantially improve the robustness of a speech recognizer and decrease the error rates by over 93% when compared with a standard approach.

1. INTRODUCTION

The speech recognition field can be divided into speaker-dependent and speaker-independent categories. Speaker-independent recognition can be seen as a much more attractive approach from the users' point of view, but speaker-dependent recognition can still not be totally discarded. In many applications, like in voice-dialling, speaker-dependent models need to be utilized. Typically, the training of speaker-independent models has much less limitations than the training of speaker-dependent models. One of the most severe limitations in speaker-dependent training is the small amount of training material. Due to this, many well performing approaches, like discriminative training, are not very usable. To be able to make a good speaker-dependent recognizer one must apply such means that are suitable and well argumented considering the limitations in the training phase.

In speaker-dependent recognition the models are often not very reliable and the training is easier in a reasonably silent environment, like in an office. Thus, if no restrictions for state-segmentations imposed by the Viterbi decoder are set, in a practical usage environment, e.g. in a moving car in highway, state-segmentations can be far away from the ones obtained in the training environment. Since wrong segmentations cause increased error rates it is in our interest to prevent such as effectively as we can. Our target is to force similar state segmentations in the training phase and in the recognition phase.

It is known that standard HMMs are not able to model the temporal structures of speech effectively. This is a major deficiency, since we would like to set clear restrictions for the state-segmentations. It has been suggested by many researchers that an explicit modelling of state durations can increase the recognition rates [1-9]. State durations are usually modelled with certain distributions [1-6] and the probabilities produced by these distributions are added to the overall log-likelihood calculation. In the literature, bounded state durations used in the recognition phase have also been suggested [7-9].

In [8-9], bounded state durations were estimated after the training scheme by finding the global minimum and maximum durations for each state. The scheme resulted in quite loose state duration constraints which were then used in the final recognition phase. We have noticed that this is not effective enough. We are, therefore, using bounded state durations already in the training phase.

2. THE PROPOSED ALGORITHM FOR STATE DURATION CONSTRAINTS

In the conventional Maximum Likelihood (ML) training, the model parameters are estimated using all state sequences, most of which are unlikely to happen in the real cases but still contribute to the estimation. In another extreme case, Viterbi training, only the best state sequence is used to estimate the model parameters, which cannot give robust estimates. In our new approach, only the state sequences that are likely to happen, and which fulfill given state duration constraints are used in the estimation. In addition, the same state duration constraints are used in the recognition phase.

In the training phase we use state duration constrained ML (SDML) training. Initially, we start with loose state duration constraints, and gradually tighten the constraints. This way we can end up with the optimal state duration constraints producing the desired word duration constraints.

State duration constraints can be added to the ML-training by modifying the existing forward and backward procedures. The modified forward procedure, assuming left-to-right models without skips, is given below:

$$\alpha_t(i) = P(o_1 o_2 ... o_t | q_t = i, \lambda, \varphi_i \le d_i \le \gamma_i), \text{ and}$$

$$\chi_t(i)_d = P(o_1 o_2 ... o_t |$$

$$q_t = i, \lambda, \varphi_{i-1} \le d_{i-1} \le \gamma_{i-1}, d_i = d),$$

where q_t is the state at time t, φ_i and γ_i are the minimum and maximum durations of the state i and d_i is the duration of the state i. The modified forward variable $\alpha_t(i)$ is the probability that $o_1 o_2 ... o_t$ are observed and that the state at time t is i on the condition that durations in states $\{1,2,...,i-1,i\}$ are within the minimum and maximum limits. The additional forward variable $\chi_t(i)_d$ is the probability that $o_1 o_2 ... o_t$ are observed and that the state at time t is i on the condition that durations in states $\{1,2,...,i-2,i-1\}$ are within the minimum and maximum limits and that the duration in state i is d.

1. Initialization

$$\chi_t(i)_d = 0, \text{ except } \chi_t(1)_1 = a_{0,1} b_1(o_1), \text{ and}$$

$$\alpha_t(i) = 0, 1 \le i \le N, \text{ except } \alpha_t(1) = \chi_t(1)_1, \text{ if } \varphi_1 = 1$$

2. Induction

$$\chi_{t+1}(j)_1 = \alpha_t(j-1) a_{j-1,j} b_j(o_{t+1}), 1 \le j \le N, \text{ and}$$

$$\chi_{t+1}(j)_m = \chi_t(j)_{m-1} a_{jj} b_j(o_{t+1}), 2 \le m \le \gamma_j, 1 \le j \le N$$

$$\alpha_{t+1}(j) = \sum_{n=\varphi_j}^{\gamma_j} \chi_{t+1}(j)_n, 1 \le j \le N$$

3. Termination

$$P(O | \lambda, \varphi, \gamma) = \alpha_T(N), \text{ where } T \text{ is the input frame count.}$$

In a similar way, a modified backward procedure can be defined.

Estimation of the state duration constraints is done using loose beginning constraints (notice, that min=1 and max=∞ correspond to the conventional ML-training). Duration constraints are then made more strict during the SDML-training. I.e. the minimum state durations are estimated in the following way (MIN is the desired minimum word duration in frames):

$$\text{IF } (\sum_{s=1}^{s \le N} \varphi_s < MIN) \text{ THEN } \{$$

$$\quad \text{FOR } i = 1 \text{ TO } N \{$$

$$\quad\quad D_i = \varphi_i + 1, \text{ and } D_j = \varphi_j, \text{ where } j \ne i,$$

$$\quad\quad 1 \le j \le N, \text{ and}$$

$$\quad\quad P_i = P(O | \lambda, D, \gamma) = \sum_{k=1}^{T} \alpha_{T_k}(N)$$

$$\quad \}$$

$$\quad m = argmax(P_i), \text{ where } 1 \le i \le N,$$

$$\quad \varphi_m = \varphi_m + 1$$

$$\},$$

where T is the number of training tokens, T_k is the number of frames in the k'th training token, and $\alpha_{T_k}(N)$ is the modified forward variable considering the state duration constraints D and γ evaluated at the T_k'th (that is, the last) input frame in the state N.

Estimation of the maximum state durations can be done in a similar manner as the estimation of the minimum state durations (except that the duration constraints are gradually decreased instead of increased).

In the recognition phase, we use a modified Viterbi algorithm which is performed on a three-dimensional (time, state, duration) space [9]. This way we can find the optimal state observation sequence fulfilling state duration constraints.

An example state duration constrained HMM structure is given in Figure 1. The filled mother states represent the actual HMM states and the unfilled duration states share the same parameters (Gaussian densities) with their mother states on the same vertical lines. State transitions define the minimum and maximum state durations. In the example, the first state has the minimum duration of 3 and the maximum duration of 5.
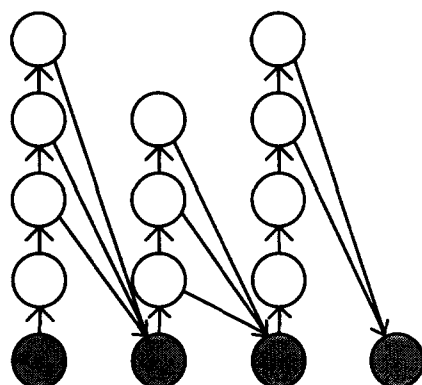

Figure 1. An example state duration constrained HMM structure.

3. RECOGNITION EXPERIMENTS

The algorithm was tested in a speaker-dependent isolated-word recognition task. Standard Mel-frequency cepstral coefficients (13: c0-c12) plus delta (13) and delta-delta (13) values with 30ms frame and 10ms frame shift were used in the tests. Since there was also a mismatch between the training and testing environments, a simple cepstral mean normalization (CMN) technique was applied to the cepstral coefficients c1-c12. The training of the models was done in a clean environment using one or two training samples per model. 30 Finnish first names, including highly confusable ones, were included in the vocabulary. The vocabulary consisted of the following names:

"hannu heikki jani janne jari juhani jukka kaarina kari marko matti mikko minna päivi pasi pekka petri petteri riitta sakari saku sari seppo sirpa tero tiina timo tommi tuula vesa".

Single utterance training

The names were spoken by 4 native male speakers. 1-mixture variable-state HMMs for the names and a 1-mixture 1-state HMM for the backround noise were used in the experiment. Since the training was done with only one or two samples, a global variance vector was estimated from all 30/60 training utterances, separately for each model set. This vector was assigned to each state of each model within the set. The state count of a name model was defined in the following way:

Frame_count / 5,

where *Frame_count* was the number of frames in the endpoint detected training utterance. A simple energy and zero-crossings based end-point detection was utilized.

The conventional ML-training (plus Viterbi decoding in the recognition phase) and the proposed SDML-training (plus the three-dimensional Viterbi decoding in the recognition phase) were compared. In the case of the SDML-training, the word duration constraints for each model were derived from an end-pointed training utterance with the following rules:

Minimum word duration = 0.8 * *Frame_count* and
Maximum word duration = 1.2 * *Frame_count*.

The following beginning (loose) state duration constraints were used:

Minimum constraint: 2, and maximum constraint: 8.

The database for training and testing consisted of 3 sessions (sessions A, B and C) for each person, recorded on different days. In each session each name was uttered four times.

Training of the models was carried out with the session A data from which four different model sets were trained. Each model set was tested with the session B and C data. Noise recorded in a moving car in highway was added to the session B and C utterances with different signal-to-noise ratios (SNR) in order to test the performance in noisy conditions. Table 1 gives the recognition rates in the case of the ML-training. It can be seen that a dramatic drop in the performance occurs around 0 dB SNR.

	Clean env.	SNR: +5dB	SNR: 0dB	SNR: -5dB	SNR: -10dB
Person1	96.67	92.92	70.31	26.15	7.08
Person2	93.65	90.73	67.29	23.13	4.79
Person3	97.40	95.00	81.77	36.04	9.17
Person4	97.29	92.81	84.48	33.75	11.88
Average	**96.25**	**92.87**	**75.96**	**29.77**	**8.23**

Table 1: The results for the ML-training.

Table 2 gives the results in the case of the SDML-training. It can be seen that a significant drop in the performance occurs only in -10 dB SNR. A slight increase of recognition rates in -5 dB SNR is hard to explain knowing that the ML-training performance drops drastically at the same -5 dB SNR. Notice that all the average results for the SDML-training are better than the corresponding results for the ML-training. The lower the SNR, the bigger the difference is.

	Clean env.	SNR: +5dB	SNR: 0dB	SNR: -5dB	SNR: -10dB
Person1	97.81	94.69	93.54	93.23	82.92
Person2	93.02	90.73	90.83	90.94	80.00
Person3	98.23	95.73	95.94	96.46	91.67
Person4	97.29	94.90	93.54	93.23	90.83
Average	**96.59**	**94.01**	**93.46**	**93.47**	**86.36**

Table 2: The results for the SDML-training.

Two utterance training

Two utterance training was carried out almost identically with the single utterance training. The state count of a name model was defined in the following way:

Average_frame_count / 5,

where *Average_frame_count* was the average number of frames in the two end-point detected training utterances. Training of the models was carried out with the session A, B and C data from which total of six different model sets were trained. Each model set was tested with the data from other sessions. For example, if the first two utterances from the session B were used to train the models, then the sessions A and C were used for testing.

Some model sets were discarded since the end-point detected training samples had too different durations.

Notice that the two utterance SDML-training fails if the training utterances have relative durations outside a feasible range. In these cases the corresponding ML-training model sets were also discarded in order to obtain a true comparison. In a practical situation, however, one would not have to discard the trained model but to ask for more repetitions so that the model can be trained.

Table 3 gives the results in the case of two utterance ML-training. The last row values of the table are the error rate reductions when compared with the single utterance ML-training. It can be seen that the error rates decrease significantly in clean and +5 dB SNR conditions but remain almost the same in noisier environments. Notice that the direct averages of the personal results are slightly different that the overall averages due to some discarded training sets.

	Clean env.	SNR: +5dB	SNR: 0dB	SNR: -5dB	SNR: -10dB
Person1	99.17	95.42	71.77	24.69	8.13
Person2	97.50	94.10	70.00	21.46	5.63
Person3	98.23	95.21	80.31	35.63	10.31
Person4	99.50	95.17	79.75	32.58	12.42
Average	**98.81**	**95.10**	**76.47**	**29.81**	**9.81**
Error rate reduction	67%	31%	2%	0%	2%

Table 3: The results for the two utterance ML-case.

The corresponding two utterance SDML-training results can be found in Table 4. Examining the last row of the table, which gives the error rate reductions from the single utterance SDML-case, one can notice that the error rates decrease more heavily than in the ML-training case and more importantly, the reduction of error rates is significant also in noisy environments.

	Clean env.	SNR: +5dB	SNR: 0dB	SNR: -5dB	SNR: -10dB
Person1	99.38	96.15	96.04	94.79	85.83
Person2	97.71	94.17	95.00	93.96	85.21
Person3	98.96	96.25	95.52	96.46	91.98
Person4	99.17	96.50	95.08	94.92	91.67
Average	**98.98**	**96.50**	**95.44**	**95.17**	**89.33**
Error rate reduction	70%	34%	30%	26%	22%

Table 4: The results for the two utterance SDML-case.

Table 5 compares the ML-training and the SDML-training head to head in the two utterance training case. Table values are the error rate reductions that are obtained with the SDML-training over the ML-training. One can notice that the SDML-training is significantly better in all environments, especially in noisy ones, which shows the noise robustness of the SDML-training scheme. Error rate reduction is the highest in -5 dB SNR, in which over 93%

reduction is obtained. In -5 dB SNR, the average recognition rate with the ML-training is less than 30% whereas the rate with the SDML-training is over 95%.

	Clean env.	SNR: +5dB	SNR: 0dB	SNR: -5dB	SNR: -10dB
Error rate reduction	14%	19%	81%	93%	88%

Table 5: Error rate reduction with the SDML-training over the ML-training in the two utterance training case.

4. CONCLUSION

State duration modelling has generally been accepted as an important concept within the HMM framework. In this paper, a method to incorporate state duration constraints into the conventional ML-training was presented. Also, a novel method to estimate the state duration constraints given the overall (word level) duration constraints was presented. State duration constraints applied both in the training and the recognition phases of a speaker-dependent isolated-word recognition system were shown to significantly increase the system robustness compared to the conventional ML-training and Viterbi decoding.

REFERENCES

[1] L.R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", Proc. IEEE, Vol.77, No.2, pp. 257-286, 1989

[2] M.J. Russell, R.K. Moore, "Explicit Modelling of State Occupancy in Hidden Markov Models for Automatic Speech Recognition", Proc. ICASSP 1985, pp. 5-8

[3] S.E. Levinson, "Continuously Variable Duration Hidden Markov Models for Automatic Speech Recognition", Computer Speech and Language, Vol. 1, pp. 29-45, 1986

[4] D. Burshtein, "Robust Parametric Modeling of Durations in Hidden Markov Models", Proc. ICASSP 1995, pp. 548-551

[5] A. Anastasakos, R. Schwartz, H. Shu, "Duration Modeling in Large Vocabulary Speech Recognition", Proc. ICASSP 1995, pp. 628-631

[6] X. Wang, L.F.M. ten Bosch, L.C.W. Pols, "Intergration of Context-Dependent Durational Knowledge into HMM-Based Speech Recognition", Proc. ICSLP 1996, pp. 1073-1076

[7] S.V. Vaseghi, "State Duration Modelling in Hidden Markov Models", Signal Processing, Vol. 41, pp. 31-41, 1995

[8] W.-G. Kim, J.-Y. Yoon, D.H. Youn, "HMM with Global Path Constraint in Viterbi Decoding for Isolated Word Recognition", Proc. ICASSP 1994, pp. 605-608

[9] H. Gu, C. Tseng, L. Lee, "Isolated-Utterance Speech Recognition Using Hidden Markov Models with Bounded State Durations", IEEE Transactions on Signal Processing, Vol. 39, No. 8, pp. 1743-1752, 1991