Internal Use Only（非公開）

TR－SLT－0039

Clean and Noisy Speech Detection using Hybrid HMM/BN Framework

Brian Mak　　Konstantin Markov　　Satoshi Nakamura

2003.04.30

This technical report describes a new method for detection of noisy speech signals and clean/noisy speech decision making based on the hybrid HMM/BN acoustic modeling framework.

（株）国際電気通信基礎技術研究所
音声言語コミュニケーション研究所
〒619－0288「けいはんな学研都市」光台二丁目2番地2 TEL：0774－95－1301

Advanced Telecommunication Research Institute International
Spoken Language Translation Research Laboratories
2-2-2 Hikaridai "Keihanna Science City" 619-0288, Japan
Telephone:+81-774-95-1301
Fax　　:+81-774-95-1308

c2003（株）国際電気通信基礎技術研究所
c2003 Advanced Telecommunication Research Institute International

Contents

1 Introduction

Many approaches have been proposed to deal with noise in robust speech recognition. These approaches can be categorized roughly into feature-based methods and model-based methods. The former includes techniques to eliminate noise in noisy speech such as spectral subtraction [2], and cepstral mean subtraction [6]; or to extract more robust feature as in PLP/RASTA [4], robust auditory features [5], and discriminative features [7]; estimation of clean speech from noisy speech by joint estimation of the additive and channel noise and the signal [10], or by SPLICE [3]. Model-based methods include noise adaptation by e.g. MLLR [1], parallel mode combination [8], and stochastic matching [9], etc. However, a pre-requisite for all these methods usually is a good detection of noisy speech. Usually the problem is called voice activity detection (VAD) and is solved by signal processing techniques. In this paper, we would like to investigate the use of statistical method to compute directly the posterior probability of noise in each speech frame. This is performed in a hybrid hidden Markov model/Bayesian network framework. The advantage of such framework is that it may continue to benefit from current efficient HMM algorithms and yet acoustic factors that affect speech can easily be incorporated into the statistical models.

2 Hybrid HMM/BN Framework

In our hybrid HMM/BN framework, each acoustic model is a hidden Markov model (HMM) but the state probability density function (pdf) is modelled by a Bayesian network. The advantage is that on the one hand, all current HMM algorithms on speech recognition may still be used except for the computation of state likelihoods; on the other hand, the use of Bayesian networks allow us to incorporate other acoustic-dependent information (or factors) such as gender, prosody, and speaking rate, etc. into the estimation of the state pdf easily in a disciplined manner.

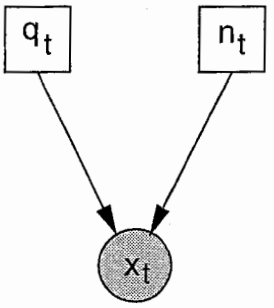

Figure 1: A Bayesian network for the relationship between observation x, state q, and noise factor n.

In this paper, we are considering the effect of noise factors in the state pdf. The interdependency relationship between an observation $\mathbf{x}_t$ at time t, an HMM state q, and a noise factor n is depicted in Fig. 1. As usual, square boxes represent discrete variables and circles represent continuous variables; shaded entities are observable whereas unshaded entities are hidden. The noise factor, n is a discrete quantity describing the noise conditions such as the SNR or noise type. Thus, the probability of an observation $\mathbf{x}_t$ at state q is now

$$\begin{aligned} p(\mathbf{x}_t|q_t) &= \sum_{n_t \in \mathcal{N}} p(\mathbf{x}_t, n_t|q_t) \\ &= \sum_{n_t \in \mathcal{N}} p(n_t|q_t) p(\mathbf{x}_t|q_t, n_t) \\ &= \sum_{n_t \in \mathcal{N}} p(n_t) p(\mathbf{x}_t|q_t, n_t) \quad [n_t \text{ is indept. of } q_t] \end{aligned} \tag{1}$$

where $\mathcal{N}$ is the set of noise factors. If we assume a uniform distribution for $p(n_t)$, then Eqn.(1) may be rewritten as

$$p(\mathbf{x}_t|q_t) \approx \frac{1}{|\mathcal{N}|} \sum_{n_t \in \mathcal{N}} p(\mathbf{x}_t|q_t, n_t) \ . \tag{2}$$

If the pdf $p(\mathbf{x}_t|q_t, n_t)$ is modelled by a mixture of Gaussians, then the state pdf is a mixture of mixtures of Gaussians.

3 Noise Detection

The presence of a noise factor n_t can be detected by computing the posterior probability of n_t for an observation $\mathbf{x}_t$. Depending on how much context — history and lookahead — we have, we may compute the posterior probability as follows.

3.1 Case I: With Context

If the whole utterance $\mathbf{X}$ of duration T is given, then the posterior probability of a noise factor can be computed as

$$\begin{aligned} p(n_t = k|\mathbf{X}) &= \sum_i p(n_t = k, q_t = i|\mathbf{X}) \\ &= \sum_i p(q_t = i|\mathbf{X}) p(n_t = k|q_t = i, \mathbf{X}) \\ &= \sum_i \gamma_t(i) \frac{p(n_t = k) p(\mathbf{x}_t|q_t = i, n_t = k)}{\sum_m p(n_t = m) p(\mathbf{x}_t|q_t = i, n_t = m)} \end{aligned} \tag{3}$$

where $i \in \mathcal{Q}$ which is the set of HMM states; and

$$\gamma_t(i) = p(q_t = i|\mathbf{X}) \tag{4}$$

is the probability of the utterance $\mathbf{X}$ being at state i at time $t \leq T$. Notice that T may be greater than t as we may want to lookahead several frames to determine the noise factor at time t.

3.2 Case II: No Context, but With Time Information

Assuming that you are only given an observation $\mathbf{x}_t$ and the time t it appears. Then we will have,

$$\begin{aligned} p(n_t = k|\mathbf{x}_t) &= \sum_{q_t \in \mathcal{Q}} P(n_t = k, q_t = i|\mathbf{x}_t) \\ &= \sum_{q_t \in \mathcal{Q}} \frac{P(n_t = k, q_t = i, \mathbf{x}_t)}{p(\mathbf{x}_t)} \end{aligned} \tag{5}$$

where $i \in \mathcal{Q}$ which is the set of HMM states.

However, according to the dependencies in the Bayesian network in Fig. 1, the joint probability

$$P(n_t = k, q_t = i, \mathbf{x}_t) = p(q_t = i) p(n_t = k|q_t = i) p(\mathbf{x}_t|q_t = i, n_t = k) \tag{6}$$

can be reduced to

$$P(n_t = k, q_t = i, \mathbf{x}_t) = p(q_t = i) p(n_t = k) p(\mathbf{x}_t|q_t = i, n_t = k) \ . \tag{7}$$

Therefore, Eqn.(5) can be evaluated as

$$p(n_t = k|\mathbf{x}_t) = \frac{p(n_t = k)}{p(\mathbf{x}_t)} \sum_{q_t \in \mathcal{Q}} p(q_t = i) p(\mathbf{x}_t|q_t = i, n_t = k) \ . \tag{8}$$

The term $p(q_t = i)$ may be estimated recursively as follows:

$$p(q_t = i) = \sum_{q_{t-1} \in \mathcal{Q}} p(q_{t-1} = j) a_{ji} \tag{9}$$

While Case I is more sound in theory, Case II is simpler and faster.

4 Implementation Issues

Using the HMM/BN framework requires the implementation of mixture of Gaussian mixture models. This is achieved by modifying the stream implementation of HTK Version 3.2 codes. In HTK, stream are "multiplicative"; that is, the observation likelihood at a state is the product of the likelihood due to each stream. The mixture of mixture models requires "additive streams" instead. That is, the observation likelihood at a state is the weighted sum of the stream likelihoods.

Technically, this implies 3 basic changes to HTK library codes:

1. the addition of a new enum type for the "additive stream":

   ```
   typedef enum { PRODUCT, SUM } BM_Stream_Kind;
   ```

   Thus, "multiplicative streams" are used if a variable of `BM_Stream_Kind` is set to `PRODUCT`, and "additive streams" if it is set to `SUM`.

2. change the way state observation likelihood is computed if additive stream is used.

3. a subtle change has something to do with the way HTK splits an observation vector into streams. Depending on the number of streams, HTK has a standard way to slicing up the observation vector into sub-vectors that is hard-coded. However, for our additive streams, each stream will use the *same* observation vector. Thus, an observation vector has to be "duplicated" instead of "sliced" for each stream. This is done efficiently by only copying the pointer to the vector for each stream.

The two most important utilities are as follows:

- Hadd: to combine additive streams
- Hnoise-prob: to compute the posterior probability of each noise factor for each speech frame, and decide the most likely one among them.

Notice that

- all program files added by Brian Mak have a prefix "`bm_`".
- all C structures added by Brian Mak have a prefix "`BM_`".
- a library file "`bm_subr.c`" is added. It contains some subroutines for float matrices.
- an include file "`bm.h`" is added. It contains 2 new enum types:
  - `BM_Stream_Kind`: for new streams which use addition instead of multiplication for the stream probabilities
  - `BM_Aij_Mode`: governs how to compute the new transition probabilities of a composite state.
- some global variables are added:
  - `HMem.c:BM_Stream_Kind bm_stream_kind = PRODUCT;`
  - `bm_subr.c:extern int trace;`
- a new option "-Z" is added to specify the kind of stream, `BM_Stream_Kind`.

5 Experimental Evaluation

The Aurora2 database is chosen to investigate our work because its multi-condition training set contains speech with the same contents but with various noise factors. For testing, only Test Set A is used because the test set matches with the noise conditions in the training set. Both noise types and SNRs are considered. Furthermore, this preliminary work only studies the noise detection method of Case II.

5.1 Procedure

The basic procedure is as follows:

1. An HMM is trained for each noise factor. A noise factor is a combination of noise type and its SNR.
2. Only one HMM is used to represent all digits.
3. We follow the HMM topology used in the Aurora2 baseline so that the speech HMM has 16 states, but we vary the number of Gaussian mixtures per state from one to three. In addition, there are the noise model and short pause model as in the standard Aurora baseline.
4. The noise-dependent Gaussian mixtures from the corresponding states of all HMMs are combined into a mixture of Gaussian mixtures.
5. Unless otherwise stated, all noise factors are equally likely in the composite HMM.
6. For each state, the transition probabilities of the composite HMM are computed from the corresponding transitions probabilities in the noise-dependent models.

5.2 Experiment 1: 2-class Problem with 1-Mixture per Gaussian

In the first series of experiments, only clean speech and noisy speech with one noise factor are considered. That is, the task is to determine if a speech frame in an utterance is clean or exhibits a noise factor. To do that, a 2-stream composite HMM is constructed from a clean speech model and a noisy speech model. The results for each of the four noise types: subway, babble, car, and exhibition are shown in Table 1 – 4.

Table 1: Result of noise detection using a 2-stream model: clean vs. set A subway noise. (16-state 1-mixture model.)

Model	Detection Accuracy	
	noisy	clean
clean+SNR5	97.3	93.7
clean+SNR10	95.4	90.6
clean+SNR15	93.7	85.8
clean+SNR20	92.2	79.8

Table 2: Result of noise detection using a 2-stream model: clean vs. set A babble noise. (16-state 1-mixture model.)

Model	Detection Accuracy	
	noisy	clean
clean+SNR5	95.6	91.6
clean+SNR10	94.0	87.8
clean+SNR15	92.0	83.4
clean+SNR20	87.2	80.6

We have the following observations:

Table 3: Result of noise detection using a 2-stream model: clean vs. set A; car noise 16-state 1-mixture model.)

Model	Detection Accuracy	
	noisy	clean
clean+SNR5	97.3	95.6
clean+SNR10	96.5	90.8
clean+SNR15	94.8	85.4
clean+SNR20	91.8	79.9

Table 4: Result of noise detection using a 2-stream model: clean vs. set A exhibition noise. (16-state 1-mixture model.)

Model	Detection Accuracy	
	noisy	clean
clean+SNR5	97.6	95.8
clean+SNR10	96.8	93.1
clean+SNR15	94.3	89.1
clean+SNR20	85.9	84.4

- there is a bias towards noisy frames in the sense that the detection is more accurate in noisy speech than in clean speech.
- as expected, it is easier to tell the difference between noisier speech and clean speech.

5.3 Experiment 2: 2-class Problem with 3-Mixture per Gaussian

Experiment 1 is repeated with only subway noise to investigate if more mixtures may be helpful. The result with 3 mixtures per state using subway noise is shown in Table 5. The results are very similar to those obtained with 1 mixture per state in Table 1. Thus, for this simple 2-class problem, a single mixture per state is adequate.

Table 5: Result of noise detection using a 2-stream model: clean vs. set A subway noise. (16-state 3-mixture model.)

Model	Detection Accuracy	
	noisy	clean
clean+SNR5	97.1	93.9
clean+SNR10	95.4	90.6
clean+SNR15	93.8	85.8
clean+SNR20	92.1	79.7

5.4 Experiment 3: 5-class Problem with 1-Mixture per Noise Factor

Experiment 1 is repeated with only subway noise of four different SNRs. A 5-stream composite HMM is constructed from a clean speech model and four noisy speech models, each representing a different SNR noise factor. The results for subway noise with even weights and uneven weights are shown in Table 6 while Table 7 gives the confusion matrix. When even weights are used, each of the 5 noise factors has an equal weight of 0.2. When uneven weights are used, each of the 4 noise factors has an equal weight of 0.125 and they altogether is as likely as clean speech (which has a weight of 0.5).

We have the following observations:

Table 6: Result of noise detection using a 5-stream model: 5 SNRs of set A subway noise compete with each other. (16-state, 1-mixture per noise factor.)

Model (Stream Wts)	Detection Accuracy	
	5*0.2	0.5, 4*0.125
clean	78.8	92.6
SNR20	55.6	41.6
SNR15	42.6	37.2
SNR10	48.4	46.5
SNR5	71.1	69.8

Table 7: Confusion matrix of noise detection using a 5-stream model: 5 SNRs of set A subway noise compete with each other. (16-state, 1-mixture per noise factor.)

	clean	SNR20	SNR15	SNR10	SNR5	Total #frames
clean	77.99	11.24	6.30	2.75	1.72	173792
SNR20	8.47	54.06	25.59	8.88	3.00	173792
SNR15	3.70	24.35	42.17	23.33	6.45	173792
SNR10	1.60	5.61	21.41	47.17	24.20	173792
SNR5	1.02	1.13	7.17	21.50	69.17	173792

- This 5-class problem has a lower detection accuracy than the corresponding 2-class problem. This is expected: accuracy decreases with the increase in the number of classes.
- The middle SNR noise factors are harder to detect because of its confusion with SNR below and above it as can be seen in the confusion matrix. On the other hand, clean speech and speech with 5dB SNR are easier to detect because of fewer competitors.
- On the other hand, if we are only interested in the detection of clean and noisy speech, we may convert the 5-class result using uneven stream weights described in Table 6 to a 2-class classification result of Table 8. We find that the detection accuracies are again biased to noisy speech and are mostly above 90%.

Table 8: Result of noise detection using a 5-stream model: clean model competes with 4 SNRs of set A subway noise. (16-state, 1-mixture per noise factor.)

Data	Clean	Noisy
clean	86.83	13.17
SNR20	20.19	79.81
SNR15	9.43	90.57
SNR10	4.24	95.76
SNR5	2.36	97.64

6 Conclusions and Future Works

The above pilot study is quite preliminary but encouraging. The following improvements are suggested:

- More accurate detection may be achieved by taking into account the speech context as mentioned in Section 3.1.
- Currently, the silence or short-pause portions in the testing utterances are included for detection. They should be detected separately by the trained silence and short-pause models.
- Right now, the HMM of each noise factor is trained separately on its own data, and then combined to form a composite HMM/BN model. The composite HMM should be re-estimated using all data so that the state definition will be consistent across noise factors.

[1] C. Leggetter amd P. Woodland. Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density Hidden Markov Models. *Journal of Computer Speech and Language*, 9(2):171–185, 1995.

[2] S. F. Boll. Suppression of Acoustic Noise in Speech using Spectral Subtraction. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 24:113–120, 1979.

[3] L. Deng, A. Acero, M. Plumpe, and X. Huang. Large Vocabulary Speech Recognition under Adverse Acoustic Environments. In *Proc. ICSLP*, 2000.

[4] H. Hermansky and N.Morgan. RASTA Processing of Speech. *IEEE Transactions on Speech and Audio Processing*, (2):578–589, 1994.

[5] Qi Li, F. Soong, and O. Siohan. An Aditory System-based Feature for Robust Speech Recognition. In *Proc. European Conference on Speech Communication and Technology*, 2001.

[6] F. Liu, R. Stern, A. Acero, and P. Moreno. Environment Normalization for Robust Speech Recognition using Direct Cepstral Comparison. In *Proc. ICASSP*, volume II, pages 61–64, 1994.

[7] B. Mak, Y. Tam, and Q. Li. Discriminative Auditory Features for Robust Speech Recognition. In *Proc. ICASSP*, volume I, pages 381–384, 1994.

[8] M.Gales and S. Young. Robust Continuous Speech Recognition using Parallel Model Combination. *IEEE Transactions on Speech and Audio Processing*, pages 352–359, sep 1996.

[9] A. Sankar and C. Lee. A Maximum-Likelihood Approach to Stachostic Matching for Robust Speech Recognition. *IEEE Transactions on Speech and Audio Processing*, 4(3):190–202, 1996.

[10] Y. Zhao. Spectrum Estimation of Short-time Stationary Signals in Additive Noise and Channel Distortion. *IEEE Transactions on Signal Processing*, 49:1409–1420, jul 2001.