TR-I-0283

Composition of Noise and Clean-Speech HMMs for Recognition of Noisy Speech

HMM の合成を用いた雑音下での音声認識

Franck Martin　　杉山 雅英　　嵯峨山 茂樹

内容梗概

雑音 HMM と音素 HMM との合成モデルを用いた雑音下での音声認識方式について述べる。近年、本方式に関連した研究が注目され、各種の試みが行なわれてきている。本報告では、ガウス型出力確率分布で表される雑音及び音素 HMM の合成分布をガウス分布で近似する手法について述べ、その合成 HMM による雑音下での音声認識方式を定式化し、日本語音素認識実験による有効性の評価結果について述べる。

Contents

List of Figures

List of Tables

Abstract

In this report, we develop a method, called HMM composition to cope with the problem of speech recognition in a noisy environment avoiding the tedious training of noisy HMMs. We then consider its application to a speech recognition system based on LPC cepstrum parameters. The method was tested against a variety of noises, stationary and non-stationary with signal to noise ratios ranging from 0dB to 20dB and provides an error reduction over 75% comparing with the clean-speech HMM. It is believed that this technique, by its efficiency, its flexibility and its adaptability to new noises and SNRs could constitute the heart of a real-time speech recognizer robust to noise.

Chapter 1

Introduction

1.1 Foreword

The research project covered the period going from September, 1 1992 to October, 2. Five weeks is a short time so it was proposed that the goal of the training would be to pursue the investigation of the capacities of HMM composition (HMM = Hidden Markov Model), a technique developed at the Research Center for Advanced Science and Technology (RCAST) and at NTT Human Interface Research Laboratories [1].

1.2 What is HMM composition ?

HMM composition belongs to a new category of methods in the area of speech recognition for coping with noise in the background. As a matter of fact, recognition in a noisy environment is a very serious problem because:

- The performance of current speech recognizers is much affected by the presence of noise
- Noise is difficult to model
- Noise characteristics can evaluate rapidly...etc

Roughly, speech-recognition methods against noise can be classified in four groups

- *Cancellation of noise* at the beginning of the speech processing: use of good microphones, application of low-pass filtering... These methods were the first to be applied, they represent a good start but they are insufficient.
- *Estimation from the noisy speech (e.g the speech contaminated by noise) of robust models for speech*: adaptive filtering, spectral subtraction, Maximum A Posteriori estimation with hidden Markov modeling etc. Such methods can lead to high recognition scores. Nevertheless, they are usually based on an iterative process (optimization of a Wiener-filter coefficients for instance) costful from the viewpoint of computation and of course they are noise dependent. Therefore, they are difficult to apply to a realistic task where the SNR (signal to noise ratio) and the characteristics of the noise change.
- *Compensation at the recognition level* by modeling the noisy speech using existing knowledge:

  - HMM decomposition first proposed by Varga and Moore ([2] and [3]).
  - HMM composition described in this report and very close to HMM decomposition.

- *Noise-robust parameter estimation, noise-robust distortion measures* [7]

1.3 Principle

HMM composition assumes that the NOVO HMM (NOVO stands for voice mixed with noise) obtained by combining two or more "source HMMs" will adequately model the complex phenomena (e.g the noisy speech) resulting from the interaction of these sources. The source HMMs may model the clean speech recorded in noise-free conditions or the various noise sources such as stationary or unstationary noises, background voices..etc. The tremendous advantage of HMM composition is that it avoids the HMM training on noisy speech each time speech is corrupted by a "new" noise. By "new", we mean that the characteristics of the background noises have changed so much that a new model is necessary. For example, consider the case of a person with a portable telephone who walks in the street, enters different shops, passes a construction site and so on. In general, we already have a speaker-dependent or independent model for the clean speech. To deal with the problem, we only need either to train a new noise model on

the noise data extracted for instance during the silence periods or recognize the noise characteristics (stationary, SNR...) and extract the most appropriate noise model from our library of source HMMs. Remark that the noisy model is usually made up of a few states (2 or 3) so that the training is fast and does not require a lot of training data. The composition process itself is almost instantaneous with current computers. The gain is considerable even with the increase in decoding time when comparing the HMM composition solution with the training of 2000 to 3000 triphone models on noisy speech.

1.4 Objectives

The evaluation of the possibilities and the optimization of HMM composition will constitute the framework of my master thesis. Before the training at ATR, HMM composition had essentially been tested against a stationary noise modeled by a one-state ergodic HMM. This noise is called "keisanki" because it was recorded in a computer room. The objective during the short training at ATR was to test if we could apply HMM composition to different noise sources, especially non-stationary noises, and could increase the recognition score by using more complex noise-HMMs, 2 and if possible 3-states ergodic HMMs. Extra experiments were carried out to investigate other HMM-composition characteristics:

- Evaluation of the robustness of the NOVO model for the noise "keisanki" when the SNRs of the NOVO model and of the test data differ.

- Use of a Gaussian mixture for the noise model
- Use of the female speaker FSU from the ATR database (5240 Japanese words) as a background noise.

After explaining the theoretical background of HMM composition, we will analyze the results of the experiments carried out at ATR to finally conclude on the achievements of the training.

Chapter 2

Theoretical Framework

The speech recognition system we used is based on LPC (Linear Predictive Coding) cepstrum coefficients. The only task considered was the recognition of 23 Japanese phonemes (5 vowels and 17 consonants). This simple task was suitable for starting our investigation on HMM composition. We used continuous HMMs which output probabilities are modeled by one or a mixture of Gaussian distributions. Finally, we used the 5240 words ATR database. For training, we used two extra sets of data: *one is phoneme-balanced Japanese 216 words* and *another is 101 Japanese syllables.*

The theory framework will be developed in two steps:

- We will define the process of HMM composition.
- We will describe the application of HMM composition to LPC cepstrum parameters.

2.1 Notations

The complex phenomenon we are considering here results from the interaction of a clean-speech source (symbol S) and one noise source (symbol N). The noisy speech will be represented by the symbol R.

The subscripts *cp*, *lg* and *ln* will represent the LPC cepstrum, the logarithm and the linear spectrum domains.

A Gaussian distribution will be represented by $N(\mu, \Sigma)$ where μ represents the mean vector and $\Sigma = \{\sigma_{uv} | 0 \leq u \leq p, 0 \leq v \leq p\}$ represents the covariance matrix.

In this section, we have to deal with many random variables. We need a convention to denominate them. X_b will represent the source X in the b domain where $b \in \{cp, lg, ln\}$. For instance, if we consider the random variable associated to the noisy speech in the linear spectrum, we will write $\mathbf{R}_{ln}$. If this random variable is a multivariate Gaussian, we will write $\mathbf{R}_{ln} = N(\mu^{R_{ln}}, \Sigma^{R_{ln}})$.

2.2 HMM Composition

LPC cepstrum is at this time one of the best set of parameters for representing clean speech (cf remark 2.3.2) and therefore we can hope to obtain a very good source HMM for the clean speech . Nevertheless, they perform very poorly when noise exists in the background. That is why we decided to apply HMM composition to the case the speech recognizer is based on

LPC cepstrum coefficients. But it could be applied to other sets of coefficients like FFT cepstrum in a similar way Young et al.[4] did it with HMM decomposition.

2.2.1 Parameters of an Hidden Markov Model

Our method is called HMM composition because we combine an HMM for the clean speech and an HMM for the noise into one HMM, the NOVO HMM modeling the noisy speech. We need then to determine all the parameters of the NOVO HMM from the source HMMs. We can do it simply for all parameters except for the HMM output probabilities which case is discussed in the next subsection.

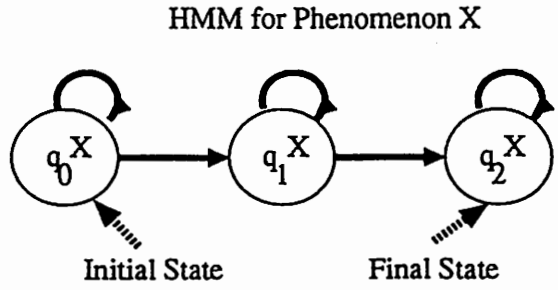

Figure 2.1: Left-to-right Hidden Markov Model

An HMM is defined by the following parameters where X reads S, N or R depending whether we are considering the clean-speech, the noise or the noisy HMM.

- $O = \{O_1, O_2, \ldots, O_T\}$ is the observation. T is the length of the input sequence (speech utterance). The observation O_t at time t is a random

vector. In our case, it will have 17 coordinates which are random variables corresponding to the 17 LPC cepstrum coefficients.

- n^X is the number of states
- $Q^X = (q_1^X, q_2^X, \ldots, q_{n^X}^X)$ is the set of the states of the model. Q^X contains two subsets I^X and F^X, one containing the i^X initial states of the Hidden Markov automaton, the other the f^X final states.
- $A^X = (a_{ij}^X)$ is the transition matrix. a_{ij} is the probability of doing the transition from state q_i^X to state q_j^X.
- $T^X = \{t_{ij}^X\}$ is the set of the HMM transitions. t_{ij}^X represents the arc transition going from state q_i^X to state q_j^X with a non-zero transition probability ($a_{ij}^X \neq 0$). We call such arcs **non-zero arcs.**
- Let ϕ^X be the number of such non-zero arcs.
- $B^X = (b_{ij}^X)$ is the output probability matrix. b_{ij}^X is the probability of outputting an observation O_t when doing the transition t_{ij}^X from state q_i^X to state q_j^X. In our case, B^X will always be modeled by a mixture of m^X Gaussian distributions. In that case, each mixture is weighted by a coefficient λ_{ijk}^X such as :

$$b_{ij}^X(O_t) = \sum_{k=1}^{m^X} \lambda_{ijk}^X b_{ijk}^X(O_t)$$

where of course:

$$\sum_{k=1}^{m^X} \lambda_{ijk}^X = 1$$

so that

$$\int_{-\infty}^{\infty} b_{ij}^{X}(\omega)d\omega = 1$$

for all couples (i, j).

- $\Pi^X = \{\pi_i^X\}$ is the initial probability vector. π_i^X is the probability of being in the initial state q_i^X $(i \in I^X)$.

- We will call $\mathcal{M}_{ij}^X$ the set of mixtures associated to the transition t_{ij}^X. $\mathcal{M}_{ij}^X$ contains m_{ij}^X mixture components.

2.2.2 Parameters of the NOVO HMM

We know deduce the parameters of the NOVO HMM. The Figure 2.2 represents a typical example of the kind of combination that can be done.

- *Number of states:* $n^R = n^S n^N$

- *Number of initial states:* $i^R = i^S i^N$

- *Number of final states:* $f^R = f^S f^N$

- *Number of mixture components of the output probabilities if we are using a mixture of Gaussian distributions:* $m^R = m^S m^N$

- *Number of non-zero arcs:* $\phi^R = \phi^S \phi^N$

To describe the states of the NOVO HMM and their relation to the source HMMs, we introduce some mapping functions such as $\mathcal{N}$ which maps the two source-HMM states onto the NOVO HMM states:

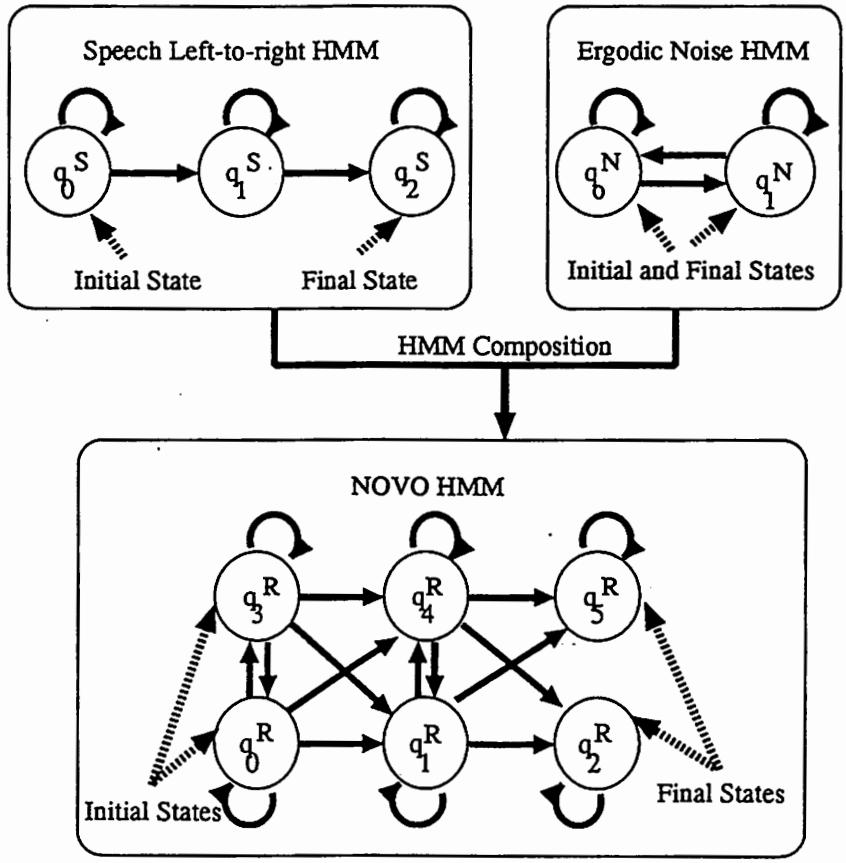

Figure 2.2: Example of HMM combination

$$\begin{aligned} \mathcal{N}: \quad & Q^N \bullet Q^S \longrightarrow Q^R \\ & (q^N_{i_N}, q^S_{i_S}) \longrightarrow q^R_{i_R} \\ & 0 \le i_N \le N^N, 0 \le i_S \le N^S \\ & 0 \le i_R \le N^R, \qquad \mathcal{N}(i_N, i_S) = i_R = i_N n^N + i_S \end{aligned} \tag{2.1}$$

The operator "$\bullet$" represents the space product operator. The subscripts i_N, i_S, i_R serve to count the objects (states, mixture components etc.) of the noise, speech and noisy HMMs. In addition, a state is determined by its number so we can make the association between the state and its number

that is $q^N_{i_N} \longleftrightarrow i_N$. That is why we used the couple (i_N, i_S) instead of $(q^N_{i_N}, q^S_{i_S})$ as argument to $\mathcal{N}$. We can thus simplify our notations.

Consult the Appendix B for details about the following other mapping functions:

- $\mathcal{T}$ for arc transitions
- $\mathcal{I}$ for the initial states
- $\mathcal{F}$ for the final states
- $\mathcal{M}$ for the mixture components

Then,

- The initial probability of an initial state defined by the couple $(i_N, i_S) \in I^N \bullet I^S$ or defined by $\mathcal{I}(i_N, i_S) = i_R$ is

$$\pi^R_{i_R} = \pi^N_{i_N} \pi^S_{i_S}$$

- The transition probability of the arc transition $\mathcal{T}(i_N, i_S, j_N, j_S) = (i_R, j_R)$ is:

$$a^R_{i_R j_R} = a^N_{i_N j_N} a^S_{i_S j_S}$$

where

$$q^N_{i_N} \in Q^N,\ q^S_{i_S} \in Q^S,\ q^R_{i_R} \in Q^R,\ q^N_{j_N} \in Q^N,\ q^S_{j_S} \in Q^S,\ q^R_{j_R} \in Q^R$$

- And for the corresponding non-zero arc, the weight of the mixture component $\mathcal{M}(k_N, k_S) = k_R$ will be:

$$\lambda_{i_R j_R k_R} = \lambda_{i_N j_N k_N} \lambda_{i_S j_S k_S}$$

- The observation probability will take the general form [2]:

$$b^R_{i_R j_R} = \int P((O^N_t, O^S_t) | \, i_R j_R) \tag{2.2}$$

The observation O^R_t is represented by the couple (O^N_t, O^S_t) above. The integration is over all couples and therefore is very difficult to compute in practise. So some approximation is necessary. The form of the approximation depends on the parameterization used, in this case, LPC cepstrum coefficients. This is the purpose of the next subsection.

2.3 Application of HMM composition to LPC cepstrum parameters

2.3.1 Anchor of the method

We need to find a domain where the relation between the sources can be stated explicitly and as simply as possible in order to deal easily with the distributions of the corresponding random variables. In our case, we chose the linear spectrum where the clean-speech and the noise samples are additive, that is:

$$x^R(t_i) = x^S(t_i) + k(SNR)\, x^N(t_i),\ 1 \leq i \leq L \tag{2.3}$$

where SNR is the clean speech to noise ratio, L the length of the sample sequence and t_i the time. $x^X(t_i)$ represents the sample sequence of the variable X. $k(SNR)$ is a weighting factor that determines the SNR of the sample sequence ($x^R(t_i)$). It is defined by equation 3.5.

From equation 2.3, we can represent the noisy speech from the noise and clean-speech sources but to do so, we need to infer the distributions of clean speech S_{ln} and noise N_{ln} in the linear spectrum from the available distributions we have in the cepstrum domain that is, S_{cp} and N_{cp}. For that purpose, we will apply a series of transformations that are based on the fact that data in the cepstrum domain are obtained after applying a Fourier transform on the logarithm of the linear cepstrum. The Figure 2.3 represents the path we follow to obtain the output probabilities matrices of the NOVO HMM from the ones of the source HMMs. The relations between the random variables are indicated. The method for calculating the parameters of the distributions at each step is described in the next sections.

2.3.2 Definition of the cosine transform matrix

We have a finite set (refer to equations 3.8) of $2p + 1$ even LPC cepstrum coefficients $(c_{-p}, \ldots, c_0, \ldots, c_p)$ on which we apply a finite Fourier transform, we obtain a set of $2p + 1$ coefficients $(\kappa_{-p}, \ldots, \kappa_0, \ldots, \kappa_p)$ which are related

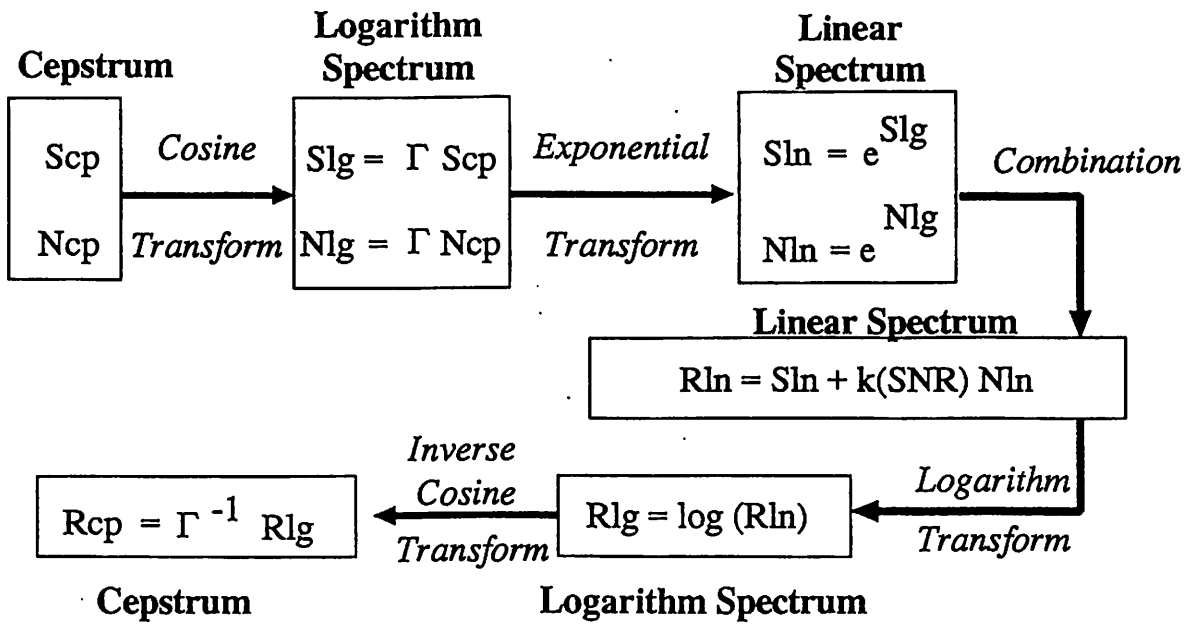

Figure 2.3: NOVO transform

by:

$$-p \leq \forall\, u, v \leq p,$$

$$\kappa_u = \sum_{v=-p}^{p} c_v \exp(\frac{i2\pi uv}{2p+1}) \tag{2.4}$$

$$\kappa_u = c_0 + \sum_{v=1}^{p} \left[c_{-v} \exp(\frac{-i2\pi uv}{2p+1}) + c_v \exp(\frac{i2\pi uv}{2p+1}) \right] \tag{2.5}$$

Since LPC cepstrum coefficients are even (see 3.9 for details), by the next change of variables, we can reduce the number of parameters and define a symmetric form for the cosine transform matrix which will simplify our calculations for the next steps. We set

$$c'_0 = \frac{c_0}{2} \tag{2.6}$$

$$c'_v = c_v,\ 1 \leq v \leq p \tag{2.7}$$

And obtain:

$$\kappa_u = \sum_{v=0}^{p} 2c'_v \cos(\frac{2\pi uv}{2p+1}),\ 1 \leq u, v \leq p \tag{2.8}$$

The previous equation defines a symmetric cosine transform matrix $\Gamma = (\gamma_{uv})_{0 \leq u,v \leq p}$ given by:

$$\gamma_{uv} = 2\cos(\frac{2\pi uv}{2p+1}),\ 0 \leq u, v \leq p \tag{2.9}$$

Remarks:

1. Because of the division by 2 in equation 2.7, we have to do the following modifications on the (Gaussian) distributions of the c' coefficients:

$$c'_u \longleftrightarrow c_u \tag{2.10}$$

$$\mu_0^{'N_{cp}} = \frac{\mu_0^{N_{cp}}}{2} \tag{2.11}$$

$$\mu_0^{'S_{cp}} = \frac{\mu_0^{S_{cp}}}{2} \tag{2.12}$$

$$\sigma_{00}^{'N_{cp}} = \frac{\sigma_{00}^{N_{cp}}}{4} \tag{2.13}$$

$$\sigma_{00}^{'S_{cp}} = \frac{\sigma_{00}^{S_{cp}}}{4} \tag{2.14}$$

$$\sigma_{u0}^{'N_{cp}} = \frac{\sigma_0^{N_{cp}}}{2},\ 1 \leq \forall u \leq p \tag{2.15}$$

$$\sigma_{u0}^{'S_{cp}} = \frac{\sigma_0^{S_{cp}}}{2},\ 1 \leq \forall u \leq p \tag{2.16}$$

And then after finishing the series of transformation, we reverse the modification:

$$c_u \longleftrightarrow c'_u \tag{2.17}$$

$$\mu_0^{R_{cp}} = 2\mu_0^{'R_{cp}} \tag{2.18}$$

$$\sigma_{00}^{R_{cp}} = 4\sigma_{00}^{'R_{cp}} \tag{2.19}$$

$$1 \leq \forall u \leq p,\ \sigma_{u0}^{N} = 2\sigma_0^{'R_{cp}} \tag{2.20}$$

All the transforms described hereafter apply to the coefficients marked with a prime ' but in order to avoid an excessive use of superscripts, we will drop from now the prime knowing what we are talking about.

2. Rigorously, the sum should be over an infinite number of samples. Here, we are assuming that the coefficients which index is over p are null. It is a common way to solve the problem. It results in an error in accuracy that we did not have time to examine.

3. One of authors (Masahide SUGIYAMA) suggested that we could use a rectangular cosine matrix (γ_{uv}) where $0 \leq u \leq p', 0 \leq s \leq p$ with $p' > p$ to obtain a better precision. I studied this point trying to use various forms of cosine matrices. A major difficulty raises when we need to apply the inverse cosine transform. I did not succeed in obtaining a rectangular matrix which would verify $C^T C = I_p$ where I_p is the square unit matrix with dimension p. All my attempts gave instead of I_p a matrix I'_p which off-diagonal terms were very small but non-null and therefore would introduce round-off errors. It also increases the number of calculations. Therefore, the gain in using such a rectangular matrix is not obvious. The extended discussion will be described.

2.3.3 Cosine transform

We now describe how to infer the distributions S_{lg} and N_{lg} for speech and noise in the logarithm spectrum from S_{cp} and N_{cp}. The process is represented in the Figure 2.4.

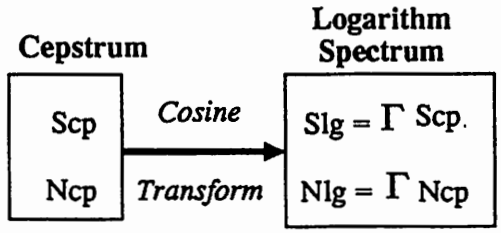

Figure 2.4: Cosine Transform

Hypothesis:

We suppose that S_{cp} and N_{cp} have a multivariate Gaussian distribution.

We will apply the next theorem to solve our problems.

Theorem 1 *Let $X_1, \ldots, X_p$ be p normal random variables such as:*

$$E[X_i] = \mu_i, \ var[X_i] = \sigma_{ii}, \ covar[X_i, X_j] = \sigma_{ij} \ (0 \leq i, j \leq p)$$

The linear combination $Y = \sum_{i=1}^{p} a_i X_i$ whether the X_i are dependent or independent is itself normally distributed and:

$$Y = N(\sum_{i=1}^{p} a_i \mu_i, \sum_{i=1}^{p} \sum_{j=1}^{p} a_i a_j \sigma_{ij}) \tag{2.21}$$

According to equation 2.8, we have:

$$N_{lg} = \Gamma N_{cp} \tag{2.22}$$

$$S_{lg} = \Gamma S_{cp} \tag{2.23}$$

Therefore, according to the above theorem, we have:

$$0 \leq \forall\, u, v \leq p$$

$$\mu_u^{N_{lg}} = \sum_{j=1}^{p} \gamma_{uj} \mu_j^{N_{cp}} \tag{2.24}$$

$$\sigma_{uv}^{N_{lg}} = \sum_{i=1}^{p} \sum_{j=1}^{p} \gamma_{ui} \gamma_{vj} \sigma_{ij}^{N_{cp}} \tag{2.25}$$

$$\mu_u^{S_{lg}} = \sum_{j=1}^{p} \gamma_{uj} \mu_j^{S_{cp}} \tag{2.26}$$

$$\sigma_{uv}^{S_{lg}} = \sum_{i=1}^{p} \sum_{j=1}^{p} \gamma_{ui} \gamma_{vj} \sigma_{ij}^{S_{cp}} \tag{2.27}$$

Remark: You should now be accustomed to our notations. If we write the above equations in a more compact way, we obtain:

- For the noise,

$$\mu^{N_{lg}} = \Gamma \bullet \mu^{N_{cp}} \tag{2.28}$$

$$\Sigma^{N_{lg}} = \Gamma \bullet \Sigma^{N_{cp}} \bullet \Gamma^T \tag{2.29}$$

  where $N_{cp} = N(\mu^{N_{cp}}, \Sigma^{N_{cp}})$ and $N_{lg} = N(\mu^{N_{lg}}, \Sigma^{N_{lg}})$.

- Similarly, for the clean speech,

$$\mu^{S_{lg}} = \Gamma \bullet \mu^{S_{cp}} \tag{2.30}$$

$$\Sigma^{S_{lg}} = \Gamma \bullet \Sigma^{S_{cp}} \bullet \Gamma^T \tag{2.31}$$

  where $S_{cp} = N(\mu^{S_{cp}}, \Sigma^{S_{cp}})$ and $S_{lg} = N(\mu^{S_{lg}})$

2.3.4 Exponential transform

We are considering the following process represented in Fig. 2.5.eq.

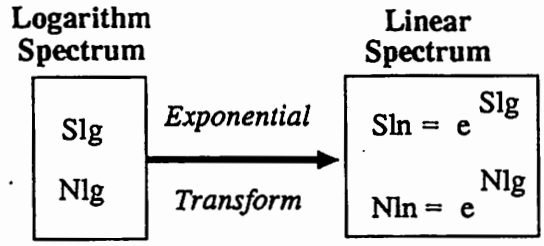

Figure 2.5: Exponential Transform

We have to infer the distributions S_{ln} and N_{ln} from S_{lg} and N_{lg}. This problem is a classic exercise in probability theory. It consists in determining the distribution of $Y = \exp^X$ when X is a normal random vector. Details of the calculation of the mean vector and the covariance matrix of Y are given in Appendix A. They are obtained by directly integrating and using the fact that X is normal. S_{ln} and N_{ln} have a "lognormal" distribution.

For the noise source, we obtain,

$$0 \leq \forall\, u, v \leq p,$$

$$\mu_u^{N_{ln}} = \exp\left[\mu_u^{N_{lg}} + \frac{\sigma_{uu}^{N_{lg}}}{2}\right] \tag{2.32}$$

$$\sigma_{uv}^{N_{ln}} = \mu_u^{N_{ln}} \mu_v^{N_{ln}} \left[\exp(\sigma_{uv}^{N_{lg}}) - 1\right]. \tag{2.33}$$

Similarly for the clean speech source, we obtain,

$$0 \leq \forall\, u, v \leq p,$$

$$\mu_u^{S_{ln}} = \exp\left[\mu_u^{S_{lg}} + \frac{\sigma_{uu}^{S_{lg}}}{2}\right] \tag{2.34}$$

$$\sigma_{uv}^{S_{ln}} = \mu_u^{S_{ln}} \mu_v^{S_{ln}} \left[\exp(\sigma_{uv}^{S_{lg}}) - 1\right]. \tag{2.35}$$

2.3.5 Linear-spectrum addition

We now combine the source elements. This process is represented in Fig. 2.6.

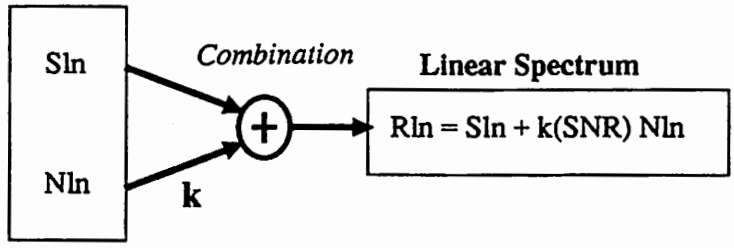

Figure 2.6: Combination of the speech and noise sources

From Eq.(2.3), we deduce the following relation between the random variables:

$$R_{ln} = S_{ln} + k(SNR) \bullet N_{ln} \tag{2.36}$$

We assume that S_{ln} and N_{ln} are independent. Then,

$$\mu^{R_{ln}} = \mu^{S_{ln}} + k(SNR)\, \mu^{N_{ln}} \tag{2.37}$$

$$\Sigma^{R_{ln}} = \Sigma^{S_{ln}} + k^2(SNR)\, \Sigma^{N_{ln}}, \tag{2.38}$$

where $k(SNR)$ is a function of the signal-to-noise ratio [Eq.(3.5)] chosen in such a way that the global SNR of the noisy speech database has the value we want it to have.

2.3.6 Logarithm transform

This process is displayed in Fig. 2.7.

Approximation:

R_{ln} is "lognormal" distributed or, equivalently, R_{lg} is normal.

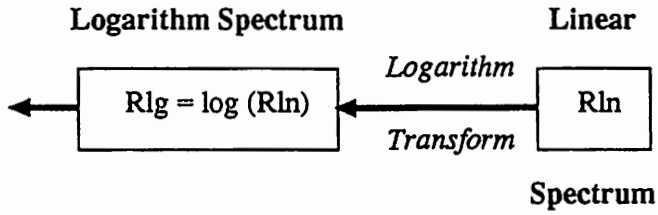

Figure 2.7: Logarithm Transform

This approximation is reasonable when the variances of R_{ln} are small compared with 1. Then, the parameters of R_{lg} are obtained by inversing Eq.(2.33) for the exponential transform and apply it to the noisy speech. We obtain,

$$0 \leq \forall\, u, v \leq p,$$

$$\mu^{R_{lg}} = \log\left[\mu_u^{R_{ln}}\right] - \frac{1}{2}\log\left[\frac{\sigma_{uu}^{R_{ln}}}{\mu_u^{R_{ln}}\mu_u^{R_{ln}}} + 1\right] \tag{2.39}$$

$$\sigma_{uv}^{R_{lg}} = \log\left[\frac{\sigma_{uv}^{R_{ln}}}{\mu_u^{R_{ln}}\mu_v^{R_{ln}}} + 1\right] \tag{2.40}$$

Notice that R_{ln} is a positive random variable because N_{ln} and S_{ln} are positive and $k(SNR)$ is always positive.

2.3.7 Inverse cosine transform

This step is represented by Figure 2.8.

Therefore, we can apply the theorem 1. The linear combination is given by the inverse of the cosine transform matrix. We obtain:

$$\mu^{R_{cp}} = \Gamma^{-1}\mu^{R_{lg}} \tag{2.41}$$

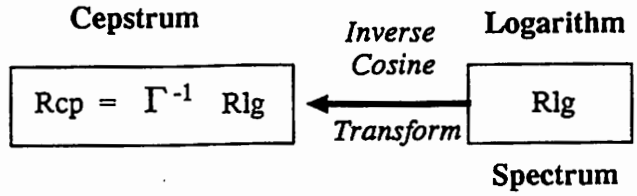

Figure 2.8: Inverse Cosine Transform

$$\Sigma^{R_{cp}} = \Gamma^{-1}\Sigma^{R_{lg}}(\Gamma^{-1})^T \tag{2.42}$$

Remembering that the cosine matrix is symmetric, we then obtain for the covariances:

$$\Sigma^{R_{cp}} = \Gamma^{-1}\Sigma^{R_{lg}}(\Gamma^{-1}) \tag{2.43}$$

2.3.8 About the variance of the power

When we first applied this transformation to the noise "keisanki" (computer room) in the case it is modeled by one Gaussian and the LPC power normalized residual is not used (see next section for details), we obtained a very low recognition score below the one of the clean speech model. In addition, the values of the parameters of the output distributions of the NOVO HMMs where almost identical to those of the corresponding clean-speech HMMs.

We observed we could have a satisfying recognition score by setting the variance of the power to 0 for both source, namely:

$$\sigma_{00}^{S_{cp}} = \sigma_{00}^{N_{cp}} = 0 \tag{2.44}$$

Except if mentioned, the experiments described in this report use this experimental initialization. Since this choice affects the coefficients related to the power, we decided not to use the power LPC cepstrum during the recognition phase, that is to only use 16 coefficients.

Table 2.1: Main Notations

N_{cp}, N_{lg}, N_{ln}	Random variables corresponding to the noise in the cesptrum, the logarithm spectrum and the linear spectrum
S_{cp}, S_{lg}, S_{ln}	Random variables corresponding to the clean speech in the cesptrum, the logarithm spectrum and the linear spectrum
R_{cp}, R_{lg}, R_{ln}	Random variables corresponding to the noisy speech in the cepstrum, the logarithm spectrum and the linear spectrum
$\mu^X = (\mu_u^X)$	Mean vector of the Gaussian variable X
$\Sigma^X = (\sigma_{uv}^X)$	Covariance matrix of the Gaussian variable X
$\Gamma = (\gamma_{uv})$	Cosine transform matrix
$B = (b_{ijk})$	b_{ijk} Output probability of the k^{th} mixture of the transition going from state i to state j.
(c_u)	LPC cepstrum coefficients

Chapter 3

Experimental Procedure

3.1 Procedure

We followed the following procedure to carry out our experiments. Details about each step can be found in the next subsections. The Figure 3.1 gives an overview of the experimental procedure we describe in this chapter.

Note: The process of making a database involves the constitution of two databases, one is used for training and the other for recognition.

1. Fabrication of the clean-speech PCM databases
2. Fabrication of the corresponding clean-speech LPC databases (17 LPC cepstra)
3. Training of the clean-speech HMMs (in our case, there are 23 Japanese phonemes).
4. For each new noise

(a) Extraction of the noise

(b) Fabrication of the noise and of the noisy PCM databases at a given SNR

(c) Fabrication of the corresponding LPC cepstrum databases

(d) Training of the noise HMM (17 LPC cepstra)

(e) Combine this noise HMM with each speech HMM to obtain the NOVO HMMs modeling the noisy speech.

(f) Recognition test on the noisy data using the NOVO HMM (16 LPC cepstra)

(g) To compare the NOVO HMM with other HMMs

i. HMM training on the noisy data to obtain noisy HMMs

ii. Recognition test on the noisy data using the noisy HMMs and the clean speech HMMs.

5. Repeat the process for another SNR, another noise

In our case, our task was phoneme recognition so the HMMs are phoneme HMMs.

3.2 Practical Aspects

3.2.1 Fabrication of the noise and noisy databases

In our case, we already had some noise databases available from the ASJ. The task we considered is somewhat artificial because we simply added the

noise database to the clean speech database, thus avoiding some distortions like the Lombard effect. On the other hand, such an approach enables to quickly make noisy databases at any wanted SNR. This is maybe why many researchers used it so far.

A more realistic task should use a noisy database where the speech is recorded with noise in the background. We would then need a system to evaluate the SNR of the database and to extract some "pure" noise data during silences for instance in order to train the noise HMMs which are small and therefore do not require a lot of training data. I believe the interaction of such a SNR detection system with the HMM composition method is important because the results given in the next chapter tend to show that the NOVO HMM is sensitive to the SNR.

Remark: The power of the clean-speech and noise sample sequences are computed over all the samples of the database that is:

$$N_{pow} = \frac{1}{M_N}\sum_{i=1}^{M_N}(x^N(t_i))^2 \tag{3.1}$$

$$S_{pow} = \frac{1}{M_S}\sum_{i=1}^{M_S}(x^S(t_i))^2 \tag{3.2}$$

where M_N and M_S are the number of samples in the noise and speech databases. Therefore,

$$SNR = 10\log(\frac{S_{pow}}{N_{pow}}) \tag{3.3}$$

Thus, we have the following relation between the noise, speech and noisy sample sequences:

$$x^{NS}(t_i) = x^S(t_i) + k(SNR)x^N(t_i),\ 1 \leq i \leq L \tag{3.4}$$

where:

$$k(SNR) = \sqrt{\frac{S_{pow}}{N_{pow}}} 10^{\frac{-SNR}{20}} \tag{3.5}$$

Finally, the noise database is usually much smaller than the speech database. So we may run short of noise data in the middle of the constitution of a "noisy" word. In that case, we go back to the beginning of the noise database. So the result would be the same as if we had duplicated the noise recording it on a tape many times and we were recording a speaker with the tape recorder playing the noise tape in the background. We are sure that way that noise and clean speech overlap quite randomly as it happens in the real world. This remark corresponds to the word "duplication" on Figure 3.1.

3.2.2 Preprocessing

Preprocessing was done as follows:

- Windowing: The window length was 32 ms with a frame shift of 8 ms.
- Computation of 17 autocorrelation coefficients: The length of the sample sequence was 32 (ms) * 12 (kHz) = 384 (samples).
- Preemphasis: The function $1 - \beta z^{-1}$ where $\beta = 0.97$
- LPC analysis: Computation of the linear prediction coefficients that we shall call $\alpha_u \, (0 \leq u \leq p \, (p = 16))$, 16 per sample sequence. p is called the order of the LPC analysis.

3.2.3 LPC analysis

Computation of the linear predictive coefficients α

They are computed in two steps:

- Computation of the autocorrelation coefficients from the signal samples.
- Computation of the linear predictive coefficients $\alpha_u (0 \leq u \leq p(= 16))$ from the autocorrelation coefficients using the PARCOR recursive algorithm. Simultaneous computation of the linear-prediction residual power ρ which is used to compute the power LPC cepstrum coefficient.

LPC cepstrum analysis

Please refer to the literature for a precise description like Furui [5] . The formulae we used were the following:

$$c_1 = -\alpha_1 \tag{3.6}$$

$$c_u = -\alpha_u - \sum_{m=1}^{u-1}(1 - \frac{m}{u})\alpha_m c_{u-m} \; (1 \leq u \leq p) \tag{3.7}$$

$$c_u = \sum_{m=1}^{p}(1 - \frac{m}{u})\alpha_m c_{u-m} \; (p < u) \tag{3.8}$$

where p is the order of the LPC cepstrum analysis. u is called the truncation order of the LPC analysis. Remember that LPC cepstrum coefficients are even that is:

$$\forall u > 0, \; c_u = c_{-u} \tag{3.9}$$

The computation of the LPC cepstrum corresponding to the power is particular [6] and given by:

$$c_0 = \log(\rho\sigma) \tag{3.10}$$

where ρ represents the normalized residual of the linear prediction analysis and σ the power of the sampling sequence. Here we can add two remarks:

1. The value of ρ is usually large for noise or when noise is added.

2. The residual term introduced in the computation of the LPC cepstrum enables to obtain a speech envelope closer to the real speech envelope for the LPC cepstrum than for the FFT cepstrum. See [5], p68.

3.2.4 HMM training

We used 16 LPC cepstrum coefficients and the power LPC cepstrum. Except if mentioned, the variance of the power was set to 0.

We took about 3 to 4 minutes of noise data to train the noise models using the same algorithm as for the clean speech. Since the noise models are quite small, (the maximum reached sized was for 3 ergodic states and 1 Gaussian and for 1 state and 4 Gaussian components), there was enough data for training.

We used the speaker MHT of the ATR database. 23 Japanese phonemes are trained. The data is hand-labelled. We use the odd numbered words for the training.

3.2.5 HMM composition

Refer to Chapter 2 for details. Except if mentioned, the variance of the power cepstrum is set to 0. The distributions are normal, the covariance matrices, diagonal.

Algorithm

1. Read noise model.
2. Modify it according to equations 2.10 and 2.44.
3. "Bring" it to the linear domain
4. Loop over clean speech models
   (a) Read the current speech model
   (b) Modify it according to equations 2.10 and 2.44
   (c) "Bring" it to the linear domain
   (d) For each state, mixture component of the two source HMMs, combine the corresponding distributions to obtain the NOVO distribution using the mapping functions described in the Appendix B.
   (e) "Bring" the NOVO HMM of the linear spectrum to the cepstrum domain.
   (f) Cancel modifications according to equations 2.17.
   (g) Write the NOVO cepstrum HMM
5. Repeat the process for another clean speech model

3.2.6 Recognition

Using the NOVO HMMs built during the previous step, we carried out the recognition without using the power LPC cepstrum coefficient.

In order to evaluate the performance of the NOVO HMM, we compared its recognition score with those obtained by the clean-speech HMM and the noisy HMM.

For each SNR we wanted to test, after making the noisy speech database at the corresponding SNR, we trained the noisy HMM (16 coefficients, no LPC cepstrum power) and the noise HMM (17 coefficients). We then used the latter to built the NOVO HMM.

The output probability matrices corresponding to the transitions outgoing from a state are tied.

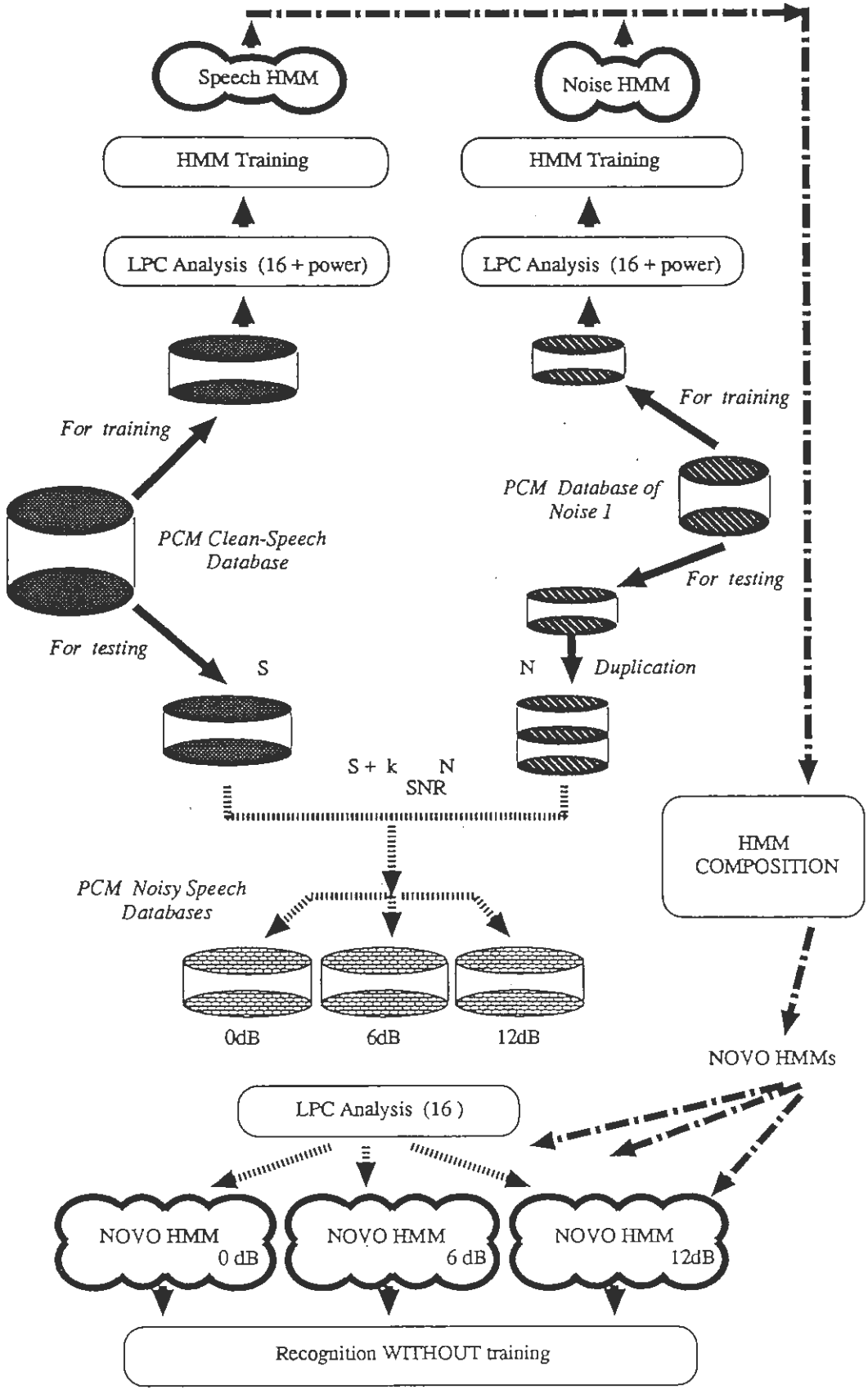

Figure 3.1: Overview of the Experimental Procedure

Chapter 4

Results

4.1 Reminder

We had some results before starting the training at ATR all obtained with the noise "keisanki" modeled by an HMM with one "ergodic" state.

- The HMM combination gave results interesting enough if we were setting the variance of the power cepstrum to zero (consult Chapter 2).

- Using one Gaussian both for clean speech and noise, the recognition score was only one or 2 percents higher for full covariance matrices than for diagonal ones. Refer to [1]. In addition, the computation load becomes more important when using full covariance matrices, Also, to be coherent, if we were using full covariance matrices for the recognition, we should train the source models with full covariance matrices. We also tried this (training full covariance matrices only for the clean speech). The results hardly changed. So, we decided to

use diagonal covariance matrices for the rest of our study. These results are presented in the table below.

Table 4.1: Input possibilities for the source HMM

Noise HMM	Speech HMM	Output HMM	Diagonal Approximation
Diagonal	Diagonal	Full	Valid
Diagonal	Full	Full	Valid
Full	Diagonal	Full	Not tried
Full	Full	Full	Not tried

- The HMM combination gives better results when increasing the number of mixture components.
- The use of the LPC residual (cf equation 3.10) improves the results.
- There are several combinations for making the NOVO HMM. After training the clean-speech HMM with 17 LPC cepstrum coefficients (if power included) or 16 (power not included), we obtain two sets of 16 (power not included) slightly different coefficients because the likelihood is not maximized in the same way whether we have 16 or 17 coefficients. We call set(1) the one obtained with training with 17. We call set(2) the other one. Consequently, we can make two NOVO HMMs:

  1. Using for the speech source HMM all the coefficients of set(1), we will obtain NOVO(1).

2. Using for the speech source HMM, the power coefficient of set(1) and the coefficients of set(2). We will obtain NOVO(2).

The same problem raises when doing the recognition with the clean speech model. We can use the first or the second set. Of course, we will use the first set if we want to study NOVO(1) and the second set for NOVO(2). For the noisy model, the problem also raises but we did not study the impact because we always trained the noisy HMM with 16 coefficients. After investigation, we have the following results for the noise "keisanki" without using the LPC residual:

Table 4.2: Comparison between set(1) and set(2) for "Keisanki" at 10dB

Model type	set(1)	set(1)	Set(2)
	16 + power	only 16	all 16
Trained	75.8	76.9	76.9
Novo (1s1m)	Impossible	61.2	62.4
Clean	52.7	48.2	47.5

The numbers correspond to phoneme recognition rates. There are 23 phoneme HMMs. We can deduce that:

- The use of set(1) is favorable to the clean speech and noisy HMMs but unfavorable to the NOVO HMM.
- The difference is small.

So we know that the choice of the parameter set can raise a small difference in the recognition score. We preferred to use the set(1) rather than the set(2) because that way, we did not need to train the clean speech model on set(2).

The objectives were clearly stated in the introduction, so after the next section where we explain how to read the tables and charts, we will give the results.

4.1.1 How to read the tables and charts

You might want have a rapid glance at the next pages to see how the tables and charts look like before reading through this section. To evaluate the NOVO HMM, we compared the recognition score of the NOVO HMM with the one of the clean HMM (column "Clean") and the noisy HMM (column "Noisy"). The clean HMM corresponds to the clean-speech source HMM. The noisy HMM is the one we obtain after training on each noisy-speech database. Therefore, the noisy HMM depends on the SNR of that database.

A code is written under HMM. It characterizes the type of models that was used. For instance, "4m1f" under "Clean" means that the clean HMM is modeled by a mixture (m=4) of 4 Gaussian distributions using one set of features (f=1), the LPC cepstrum. In the future, we plan to do sensor fusion and use the delta LPC and delta power cepstrum. That is why we introduced the notation "f" for the number of features. The NOVO HMM is the combination of the clean HMM and the noise HMM. Since the clean

HMM is already defined, only the code describing the noise HMM is given. For instance, "2es3m" means that the noise model has two ergodic states (s=2) and the output probability is modeled by a mixture of 3 Gaussian distributions. "es" means ergodic state. "s" means state.

"Error" corresponds to the error reduction between the code of the NOVO HMM located under "Error" and the clean and the noisy HMMs. The error reduction is computed as followed:

$$error\ reduction = \frac{NOVO - CLEAN}{NOISY - CLEAN} \tag{4.1}$$

NOVO, NOISY and CLEAN correspond to the recognition scores obtained by the corresponding HMMs. It measures the importance of the improvement respectively to the Clean HMM.

"Test SNR" corresponds to the SNR of the test database. Because we used the coefficient $k(SNR)$ (equation 3.5) computed for making the training database to make the recognition or "test" database, there is a **mismatch** between the SNR of the NOVO HMM (0, 6 and 20dB) and the one of the database it is tested against. This choice corresponds to the fact that we supposed we could not accurately compute the SNR of the incoming data during a real experiment but we only had a rough estimation.

4.1.2 Noise "Keisanki" (Computer Room)

Characteristics

Noise recorded in a computer room. Its main component comes from the ventilation. This noise is stationary and has a very large band going from 0 to 10 kHz. It thus covers almost all the speech characteristics of a human voice (usually situated between 0 and 4 kHz).

Influence of the number of states

Table 4.3: "Keisanki": Influence of the number of states

	Clean	Novo	Novo	Novo	Novo	Noisy HMM	Error
Test SNR (dB)	4m1f	1s1m	2es1m	2es2m	3es1m	4m1f	/2es2m
- 0.6dB	11.8	59.5	59.5	59.8	59.5	73.1	78
5.4dB	28.6	73.5	73.5	73.6	73.5	83.0	83
19.4dB	76.5	90.3	90.3	90.3	90.3	93.2	83

Comments:

1. We see that the gain is null when the number of states increases. The only gain is obtained when increasing the mixture and this result will be commented in the next subsection.

2. The error reduction is about 80% and increases with the SNR of the test 0% database.

Influence of the number of mixture components

Table 4.4: "Keisanki": Influence of the number of mixture components

	Clean	Novo	Novo	Novo	Noisy HMM
Test SNR (dB)	4m1f	1s1m	1s2m	1s4m	4m1f
- 0.6dB	11.8	59.5	59.8	59.9	73.1
5.4dB	28.6	73.5	73.5	73.6	83
19.4dB	76.5	90.3	90.3	90.4	93.2

Comments:

1. Increasing the number of mixture components (2es2m) results in a slight increase of the recognition score. The improvement was null when increasing the number of states.

2. The configuration { 1 state, 2 Gaussians} is better than { 2 ergodic states, 1 Gaussian } and equal to { 2 ergodic states, 2 Gaussians}.

Robustness of the NOVO HMM against SNR variations

Remark: The curve traced for 3dB and 10dB were obtained using set(2) and the three others with set(1). Though, they are not strictly comparable, we kept them because the curves for 3dB and 10dB using set(1) would have been probably very close to the traced ones. At least, the tendency they reflect is the same. **Comments:**

1. Looking simultaneously at the figure, we can see that the NOVO HMMs are in general optimum at the SNR they were made and their performance

Table 4.5: "Keisanki": Robustness of the NOVO HMM against SNR variations

Test DB SNR	NOVO-HMM SNR				
	0dB	3dB	6dB	10dB	20dB
- 0.6dB	59.5	60.2	54.2	42.2	21.7
5.4dB	55.7	68.2	73.5	71.5	53.5
19.4dB	28.4	41.4	54.3	70.6	90.3

is significantly lower (by a few percents) when the SNR fluctuates by more than 2dB.

2. The 10dB HMM is robust for SNRs above 10dB but it does not improve its recognition score when the SNR increases.

Discussion

We must first think about the physical meaning of a state in a HMM and the meaning of using a mixture of Gaussian distribution for modeling the HMM output probabilities.

The states of a phoneme HMM correspond to the stationary parts of the phoneme. From the experiments, we know that 3 states is a good compromise. The first state models the coarticulation or the transition (silence...) with what is before or after the phoneme. The middle state characterizes more the phoneme itself. So a state generally models a stationary part of the speech phenomenon.

If we increase the number of Gaussian components of the output probability distribution of a given state, we can model the distribution of the speech

phenomenon modeled by that state more accurately.

The above results correspond quite well to what one would expect from the theory. "Keisanki" is a stationary noise so if we add states to the noise model, they will reproduce the characteristics of the first state so the recognition score should not vary. We obtain a slight improvement when using the number of mixture components because "keisanki" is not the "Gaussian" noise often used as model in the literature. On the other hand, it might be not too far to that model because the improvement between 4 components and 2 is low.

The data we collected for studying the robustness are scarce so we can only say that the NOVO HMMs tend to be optimum at the SNR at which they were made and that they are not very robust. So a good evaluation of the SNR of the recognition data will be necessary when using HMM combination for a real task.

4.1.3 Noise "Kousyuu" (Car passing by)

Characteristics

Recorded outside by holding a microphone in the air while a car turns around. It contains a weak stationary background noise (other cars passing by but far away from the microphone) with some peaks when the car passes by and its motor roams.

Influence of the number of states

Comments:

1. Increasing the number of states results in an increase in the recognition score.

Table 4.6: "Kousyuu": Influence of the number of states

Test SNR	Clean	Novo	Novo	Novo	Noisy HMM	Error Red.
(dB)	4s4m	1s1m	2es1m	3es1m	4s4m	3es4m
2.65	66.3	80.8	84.1	82.4	86.6	79.0
6.65	79.6	87.1	88.8	88.2	91.1	75.0
20.65	92.3	92.5	92.6	92.7	93.6	31.0

2. The recognition score for 2 ergodic states is higher than for 3 ergodic states at low SNRs. This is one of the rare irregularities we met in all our experiments.

3. There is an important mismatch between the SNR of the test database and the SNR for which the NOVO HMMs were made.

4. The error reduction decreases while the SNR increases. Nevertheless, at 20dB, the recognition scores are very close to each other.

5. The error reduction is about 80% at low SNRs.

6. The gain at low SNRs is relatively important (in respect to the other noises) between one state and 2 states.

Influence of the number of mixture components

Comments:

1. We do not have the irregularity we had for the states in the progression.

2. The improvement is roughly the same whether we increase the number of Gaussians or states.

Table 4.7: "Kousyuu": Influence of the number of mixture components

Test SNR	Clean	Novo	Novo	Novo	Noisy HMM	Error Red.
(dB)	4s4m	1s1m	1s2m	1s4m	4s4m	1s4m
2.65	66.3	80.8	84.2	84.2	86.6	88.0
6.65	79.6	87.1	88.4	88.5	91.1	77.0
20.65	92.3	92.5	92.6	92.7	93.6	31.0

3. The error reduction is quite high at 0dB but very low at 0dB. Nevertheless, at 0dB, all three models have similar recognition scores.

Discussion

The noise "Kousyuu" is not very rich. When listening, it has essentially two components:

- An important stationary background noise "A" made up of cars passing by with a lot of unstationary noises but which level is not very high and which are therefore probably masked by "A".
- A "talkative" component "B" that is the roaming of the engine when the car passes by or accelerates.

This might explain why the configurations 2es1m performs quite well, one state modelling the "A" part, the other the "B" part.

It is important to note that the test database has a SNR quite far from the one it was expected to have. Considering the previous study regarding the robustness of the "keisanki" NOVO HMM, this might have lowered the performance of the models. Notice that a recognition score above 80% is usually considered as a

minimum for a phoneme model in a real task. We see here that HMM composition succeeded in satisfying this condition at a SNR of 2.65dB.

4.1.4 "Elevator"

Characteristics

Recorded in a hall. The main component is made of footsteps from different persons and therefore having different "rythms". We can hear some voices in the background: a baby shouting, men laughing, calling each other... Unstationary.

Influence of the number of states

Table 4.8: "Elevator": Influence of the number of states

Test SNR	Clean	Novo	Novo	Novo	Noisy HMM	Error Red
(dB)	4s4m	1s1m	2es1m	3es1m	4s4m	3es4m
0.3	76.1	86.9	86.9	87.1	89.4	83.0
6.3	85.3	90.6	90.4	90.5	92.1	76.0
20.3	93.3	92.8	92.9	92.9	93.7	-100.0

Comments:

1. The recognition score is slightly affected when increasing the number of states, positively at 0dB, negatively at 6dB.
2. At 20dB, the 3 HMMs have very close recognition score. It is the only time in all our experiments that the clean speech model performed better than its NOVO counterpart.
3. Once more the error reduction in the low SNRs is about 80%.

Influence of the number of mixture components

Table 4.9: "Elevator": Influence of the number of mixture components

Test SNR	Clean	Novo	Novo	Novo	Noisy HMM	Error Red.
(dB)	4s4m	1s1m	1s2m	1s4m	4s4m	1s4m
0.3	76.1	86.9	86.9	87.0	89.4	82.0
6.3	85.3	90.6	90.3	90.3	92.1	74.0
20.3	93.3	92.8	92.8	92.8	93.7	negative

Comments:

1. Increasing the number of GD (Gaussian distributions) does not give satisfying results. At 0dB, the effect is positive and at 6dB, negative.

Discussion

"Elevator" is a very rich noise. It contains a lot of different impulse noises (footsteps) and voices. So the number of states or GD we used was probably not enough to model all these components. So the HMM could only model the average stationary parts of all these models. And for that, the configuration 1s1m was enough. Results are difficult to interpret. The main conclusion we can make is that HMM composition works with the same error reduction level (about 80 %) at low SNRs as for the previous noises.

4.1.5 Noise "Hitogomi" (Crowd)

Characteristics

Could have been recorded in a restaurant with a ventilator. This noise contains many voices, laughs... Unstationary.

Influence of the number of states

Table 4.10: "Hitogomi":Influence of the number of states

Test SNR	Clean	Novo	Novo	Novo	Noisy	Error Red.
(dB)	4s4m	1s1m	2es1m	3es1m	4s4m	3es4m
0.5	36.9	66.8	67.4	67.9	75.8	80.0
6.5	58.1	78.5	78.9	79.2	85.8	76.0
20.5	88.8	91.4	91.4	91.5	92.9	66.0

Comments:

1. Increasing the number of states results in an increase in the recognition score.
2. The gain between 2 and 3 ergodic states is smaller than the one between 1 and 2 states.
3. The error reduction decreases while the SNR increases. Nevertheless, at 20dB, the recognition scores are very close to each other.
4. The error reduction is about 80%.

Influence of the number of mixture components

Table 4.11: "Hitogomi": Influence of the number of mixture components

	Clean	Novo	Novo	Novo	Noisy	error red.
Test SNR (dB)	4s4m	1s1m	1s2m	1s4m	4s4m	1s4m
0.5	36.9	66.8	67.2	68.7	75.8	82.0
6.5	58.1	78.5	78.8	80.0	85.8	79.0
20.5	88.8	91.4	91.5	91.5	92.9	66.0

Comments:

1. The increase in the recognition score with the number of GD is small but steady.

2. The gain obtained by increasing the number of GD increases with less importance while the SNR increases. This tendency is neat.

Discussion

"Hitogomi" contains a lot of voices. So the performance of all models (CLEAN, NOVO, NOISY) go down when the SNR decreases. The decrease is more important than for the previous unstationary noises.

There is no clear tendency whether increasing the number of states or GD gives the best result. Both factors play favorably. In addition, the progression indicates that the recognition score gets higher the more states or GD we use. So it is legitimate to hope to be able to obtain even higher recognition scores but at the expense of an increase in computations.

4.1.6 Noise "Speaker"

Characteristics

The noise speaker is the female FSU who has a similar database as the speaker MHT we used for the clean-speech source. We made two types of experiment:

Semi-open We had the time to do it at ATR in the end of the training. The experiment is semi-open because we used the same noise data for training and for making the test database. We call that noise "FSU-216" because we used the set of the 216 balanced words of FSU as a noise database.

Open This experiment was carried out at NTT. We extract every 20th word from the ATR database of 5240 words in order to avoid the succession of too similar words. The set of words thus extracted was used to make the test database. We call that noise "FSU-5240". The noise model was trained using "FSU-216".

Then, according to what we said about the variance of the power, we are precisely in the case where the noise has similar power characteristics as the noise source and where at least the masking can work properly using the theory developed in the second Chapter . This was tried after returning to NTT.

Influence of the number of states

Comment: Since leaving the variance of the power unchanged does not bring any improvement, we set the variance of the power to 0 for the open experiment.

Table 4.12: FSU-216: Variance of the power is set to 0

Test SNR	Clean	Novo	Novo	Novo	Noisy HMM	Error Red.
(dB)	4s4m	1s2m	2es2m	3es2m	4s4m	/3es
0	67.4	69.3	71.1	71.8	71.0	122.0
6	73.0	75.5	77.7	75.9	78.5	52.0
12	78.8	80.5	82.0	83.8	84.1	94.0
20	85.4	86.2	87.3	88.1	88.2	96.0

Table 4.13: FSU-216: Variance of the power is left unchanged

Test SNR	Clean	Novo	Novo	Novo	Noisy HMM	Error Red.
(dB)	4s4m	1s2m	2es2m	3es2m	4s4m	/3es
0	67.4	69.0	69.8	71.4	71.0	122.0
6	73.0	75.1	77.1	76.1	78.5	52.0
12	78.8	80.3	81.7	82.9	84.1	94.0
20	85.4	85.8	87.1	87.5	88.2	96.0

Table 4.14: FSU-5240: Variance of the power set to 0

Test SNR	Clean	Novo	Novo	Novo	Noisy HMM	Error Red.
(dB)	4s4m	1s2m	2es2m	3es2m	4s4m	/3es
0	69.4	70.4	71.8	72.6	72.7	97.0
6	74.6	76.0	77.8	76.4	78.2	50.0
12	79.5	80.8	81.8	83.2	83.7	88.0
20	85.6	86.1	87.2	87.6	89.1	57.0

Comments

1. We notice that the approximation "variance of power set to 0" works slightly better than the theoretical case "variance left unchanged".

2. At 0dB, the NOVO HMMs in both experiments give better results than the noisy HMM.

3. At 6dB, the error recognition has an abrupt change that we do not know how to explain.

4. The error reduction otherwise is very high.

5. The results are better (in absolute) for the open experiment that for the semi-open one. This might be accounted to the fact that the choice we made of the words for making "FSU-5240" is not very balanced.

Surprisingly, the clean speech model performs quite well with the noise speaker. This might be due to the fact that the unstationarity is homogeneous, monolithic, "pure" because FSU's voice was recorded in very low noise conditions. So the experiment is even more artificial. If a stationary noise component were added in the background, the task would be more realistic the scores significantly lowered.

The fact that the NOVO HMM can perform better than the noisy HMM at low SNR is very encouraging and confirms the particular ability of HMM composition to deal with unstationary noises.

Chapter 5

Conclusion

The purpose of the training was to study the capacities of HMM composition when applied to the LPC cepstrum feature set.

In Chapter 1, we extensively explained the theoretical background for building the NOVO HMM using LPC cepstrum coefficients, emphasizing the hypotheses we were using when transforming the covariance matrices of the Gaussian distributions.

In Chapter 2, we described the conditions in which we carried out our experiments.

In Chapter 3, we commented the results we obtained and tried to explain them.

In this Chapter, we would like to conclude on the characteristics of HMM

that we can retain from our study:

- HMM composition provides an error reduction above 75% at SNRs of 0 and 6dB.
- HMM composition can be applied to all kinds of noise: stationary ("keisanki") and unstationary noises ("elevator", "kousyuu") including voices ("hito-gomi" and noise "speaker" (FSU)).
- HMM composition provides **real-time** adaptation to new SNRs. Having one HMM of the background noise at a given SNR is enough to make a NOVO HMM at any SNR.
- HMM composition can be applied for **real time** adaptation to new incoming noises provided the clean-speech models are available which is more than often the reality.
- The results given for the noise speaker FSU indicate that the NOVO HMM can perform better in certain conditions than the noisy HMM.
- HMM composition has a very high modularity. One way of benefiting from this modularity would be to make a library of standard noises or in the context of a given application for a factory for instance, of "factory" noises. It is even possible to combine the simple noise models to make complex ones that will be then composed with the one of the clean-speech source.
- Though it was not showed, it is possible to recognize the sources through the decoding HMM algorithm. For instance, it means that for the experiment with the noise speaker FSU at 0dB, we could have recognized not

only MHT as we did but also FSU by just studying the state sequence of the NOVO HMM. This aspect confers to HMM composition a solid base for approaching such problems as the famous "cocktail party" effect.

- Especially, because HMM composition enables to avoid the training on the noisy speech, this method can be applied to large speech recognition systems which are using thousands of HMMs and for which it is not realistic to train the models for each new noise.

In conclusion to all these points, we can say that HMM composition is a very **rich method** that can be applied to a wide range of problems (including the cocktail party effect) with a fair success. Of course, a speech recognizer robust to noise will include a lot of different methods for coping with noise but we believe that HMM composition could be the **central method** of such a system.

Acknowledgements

I would like to thank all the people working in ATR for their encouragement and the help they provided me either regarding the technical aspects of the training, the use of the computer facilities and the administrative procedures.

I would like to express my gratitude to my professor, Yoichi Okabe from the University of Tokyo and Dr. Kiyohiro Shikano from NTT Human Interface Laboratories for giving me a chance to do that training.

Appendix A

Threory of Normal Variables

Let $X = (x_1, \ldots, x_n)^T$ be a multivariate Gaussian (dimension n) then its probability density function (pdf) is given by:

$$f(X) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} e^{-\frac{(X-\mu)^T (\Sigma^{-1})(X-\mu)}{2}} dX \tag{A.1}$$

$$\int_{\Re^n} f(X) dX = 1 \tag{A.2}$$

where $\Re^n$ is the region of all possible values of x. Σ and μ are respectively the covariance matrix and the mean vector associated to X. We will add the superscript X to μ and σ when there is a risk for ambiguity.

$$\mu = (\mu_1, \ldots, \mu_n)^T$$

$$\Sigma = \{\sigma_{uv} \mid 1 \leq u, v \leq n\}$$

Let us now consider the random variable:

$$Y = e^X = (y_1 = e^{x_1}, \ldots, y_n = e^{x_n})^T \tag{A.3}$$

We will now determine the first and second moments of the distribution Y.

Computation of the mean of Y

We directly calculate the mean of the i^{th} variable from the definition:

$$\begin{aligned} \mu_i^Y &= E[e^{x_i}] \\ &= \int_{\Re^n} \frac{1}{\sqrt{(2\pi)^n|\Sigma|}} e^{x_i} e^{-\frac{(X-\mu)^T(\Sigma^{-1})(X-\mu)}{2}} dX \end{aligned} \tag{A.4}$$

The operator E is the expectation operator. Let us change variables: $Z = X - \mu$ or $0 \leq \forall i \leq n$, $z_i = x_i - \mu_i$ We define $e_i = [0, 0, \ldots, 0, 1, 0, \ldots, 0]$ where the 1 is in the i^{th} position.

We have $z_i = e_i^T Z$. Therefore,

$$\mu_i^Y = \int_{\Re^n} \frac{1}{\sqrt{(2\pi)^n|\Sigma|}} e^{z_i+\mu_i} e^{\frac{(-Z)^T(\Sigma^{-1})Z}{2}} dX$$

Remarking that Σ is symmetric and therefore that $\Sigma^T\Sigma^{-1}$ is the identity matrix I_n, let us focus on :

$$K = 2(z_i + \mu_i) - Z^T(\Sigma^{-1})Z$$

Since:

$$\begin{aligned} & (Z - \Sigma e_i)^T\Sigma^{-1}(Z - \Sigma e_i) = \\ = \; & Z^T\Sigma^{-1}Z - (\Sigma e_i)^T\Sigma^{-1}Z + (\Sigma e_i)^T\Sigma^{-1}(\Sigma e_i) - Z^T\Sigma^{-1}(\Sigma e_i) \\ = \; & Z^T\Sigma^{-1}Z - e_i^T Z + e_i^T\Sigma e_i - Z^T e_i \\ = \; & Z^T\Sigma^{-1} - 2z_i + \sigma_{ii} \end{aligned}$$

K can be rewritten as:

$$K = -(Z - \Sigma e_i)^T\Sigma^{-1}(Z - \Sigma e_i) + 2z_i - 2z_i + \sigma_{ii} + 2\mu_i \tag{A.5}$$

$$= -(Z - \Sigma e_i)^T\Sigma^{-1}(Z - \Sigma e_i) + \sigma_{ii} + 2\mu_i \tag{A.6}$$

Therefore,

$$\mu_i^Y = e^{\frac{\sigma_{ii}}{2}+\mu_i} \int_{\Re^n} \frac{1}{\sqrt{(2\pi)^n|\Sigma|}} e^{\frac{(-Z)^T(\Sigma^{-1})Z}{2}} dX$$

The integral is equal to 1 by the definition of the pdf. The mean of Y will therefore be given by:

$$\mu_i^Y = e^{\frac{\sigma_{ii}}{2}+\mu_i},\ 0 \leq i \leq n \tag{A.7}$$

Computation of the covariance matrix

By definition, we have:

$$\sigma_{ij} = E[e^{x_i}e^{x_j}] - E[e^{x_i}]E[e^{x_j}]\ 0 \leq i,j \leq n \tag{A.8}$$

The second term is easily obtained from our previous calculation. To compute the first term, we apply the definition:

$$\begin{aligned} E[e^{x_i}e^{x_j}] &= \int_{\Re^n} \frac{1}{\sqrt{(2\pi)^n|\Sigma|}} e^{x_i+x_j} e^{-\frac{(X-\mu)^T(\Sigma^{-1})(X-\mu)}{2}} dX \\ &= \int_{\Re^n} \frac{1}{\sqrt{(2\pi)^n|\Sigma|}} e^{z_i+\mu_i+z_j+\mu_j} e^{\frac{(-Z)^T(\Sigma^{-1})Z}{2}} dX \end{aligned} \tag{A.9}$$

In a similar way as for the mean, we see that:

$$\begin{aligned} &(Z - \Sigma(e_i + e_j))^T\Sigma^{-1}(Z - \Sigma(e_i + e_j)) \\ &= Z^T\Sigma^{-1}Z - (e_i + e_j)^T Z + (e_i + e_j)^T\Sigma(e_i + e_j) - Z^T(e_i + e_j) \\ &= Z^T\Sigma^{-1}Z - 2z_i + \sigma_{ii} - 2z_j + \sigma_{jj} + 2\sigma_{ij} \end{aligned}$$

Hence,

$$E[e^{x_i}e^{x_j}] = e^{\frac{\sigma_{ii}}{2}+\mu_i} \bullet e^{\frac{\sigma_{jj}}{2}+\mu_j} \bullet e^{\sigma_{ij}^X} \bullet 1$$

The one corresponds to the integration of the pdf in respect to Z (same trick as for the mean). We can write in a more compact way:

$$E[e^{x_i}e^{x_j}] = \mu_i^Y \mu_j^Y \bullet e^{\sigma_{ij}^X}$$

Combining this result with the definition A.8, we obtain the expression of the second moment of Y:

$$\sigma_{ij}^{Y} = \mu_i^{Y} \mu_j^{Y} \bullet (e^{\sigma_{ij}^{X}} - 1),\ 0 \leq i,j \leq n \qquad \text{(A.10)}$$

Appendix B

Mapping Functions

We define here some mapping functions helpful to describe the NOVO HMM resulting from the composition. The operator $\bullet$ is the product space operator. We use the notations defined in Chapter 2. $\mathcal{N}$ maps the HMM states:

$$\begin{aligned} \mathcal{N}: \quad & Q^N \bullet Q^S && \longrightarrow Q^R \\ & (q^N_{i_N}, q^S_{i_S}) && \longrightarrow q^R_{i_R} \\ & 0 \leq i_N \leq N^N, 0 \leq i_S \leq N^S & & \\ & 0 \leq i_R \leq N^R, && \mathcal{N}(i_N, i_S) = i_R = i_N n^N + i_S \end{aligned} \tag{B.1}$$

$\mathcal{I}$ maps the HMM initial states:

$$\begin{aligned} \mathcal{I}: \quad & I^N \bullet I^S && \longrightarrow I^R \\ & (q^N_{i_N}, q^S_{i_S}) && \longrightarrow q^R_{i_R} \\ & 0 \leq i_N \leq I^N, 0 \leq i_S \leq I^S & & \\ & 0 \leq i_R \leq I^R, && \mathcal{I}(i_N, i_S) = i_R = i_N i^N + i_S \end{aligned} \tag{B.2}$$

$\mathcal{F}$ maps the HMM final states:

$$\mathcal{F}: \quad F^N \bullet F^S \longrightarrow I^R$$

$$
\begin{aligned}
& (q_{i_N}^N, q_{i_S}^S) && \longrightarrow q_{i_R}^R \\
& 0 \le i_N \le F^N, 0 \le i_S \le F^S \\
& 0 \le i_R \le F^R, && \mathcal{F}(i_N, i_S) = i_R = i_N f^N + i_S
\end{aligned} \tag{B.3}
$$

$\mathcal{T}$ maps arc transitions:

$$
\begin{aligned}
\mathcal{T}: \quad & Q^N \bullet Q^S \bullet Q^N \bullet Q^S && \longrightarrow Q^R \bullet Q^R \\
& (q_{i_N}^N, q_{i_S}^S, q_{j_N}^N, q_{j_S}^S) && \longrightarrow (q_{i_R}^R, q_{i_R}^R) \\
& 0 \le i_N, j_N \le N^N, 0 \le i_S, j_S \le N^S \\
& 0 \le i_R, j_R \le N^R, \\
& \mathcal{T}(i_N, i_S, j_N, j_S) = (i_R, j_R) = (i_N n^N + i_S, j_N n^N + j_S)
\end{aligned} \tag{B.4}
$$

$\mathcal{M}$ maps the mixture components of one transition:

$$
\begin{aligned}
\mathcal{M}: \quad & M^N \bullet M^S && \longrightarrow M^R \\
& (k_N, k_S) && \longrightarrow k_R \\
& 0 \le k_N \le m^N, 0 \le k_S \le, m^S \\
& 0 \le k_R \le m^R, \\
& \mathcal{M}(k_N, k_S) = k_R = k_N m^N + k_S
\end{aligned} \tag{B.5}
$$

Appendix C

Notations

Table C.1: Translation of the Noise Names

Japanese	English
"Keisanki"	Computer room
"Kousyuu"	Car passing by
"Elevator"	Elevator
"Hitogomi"	Crowd

Table C.2: Main Notations

N_{cp}, N_{lg}, N_{ln}	Random variables corresponding to the noise in the cesptrum, the logarithm spectrum and the linear spectrum
S_{cp}, S_{lg}, S_{ln}	Random variables corresponding to the clean speech in the cesptrum, the logarithm spectrum and the linear spectrum
R_{cp}, R_{lg}, R_{ln}	Random variables corresponding to the noisy speech in the cepstrum, the logarithm spectrum and the linear spectrum
$\mu^X = (\mu_u^X)$	Mean vector of the Gaussian variable X
$\Sigma^X = (\sigma_{uv}^X)$	Covariance matrix of the Gaussian variable X
$\Gamma = (\gamma_{uv})$	Cosine transform matrix
$B = (b_{ijk})$	b_{ijk} Output probability of the k^{th} mixture of the transition going from state i to state j.
(c_u)	LPC cepstrum coefficients

Bibliography

[1] F. Martin, K. Shikano, Y. Minami and Y. Okabe, *Recognition of Noisy Speech by Using the Composition of Hidden Markov Models*, Proc. ASJ Fall Meeting, 1-7-10 (Oct. 1992).

[2] A.P. Varga and R.K Moore, *Hidden Markov Model Decomposition of Speech and Noise*, ICASSP 1990, pp.845-848 (1990).

[3] A.P. Varga and R.K Moore, *Simultaneous Recognition of Concurrent Speech Signals using Hidden Markov Model Decomposition*, Eurospeech Proceedings, (1991).

[4] M. J. Gales and S. Young. *An Improved Approach to the Hidden Markov Model Decomposition of Speech and Noise*,ICASSP, I-233-236 (March 1992).

[5] S. Furui, *Digital Speech Processing, Synthesis and Recognition*, Dekker, Ed. New York and Basel (1989).

[6] K. Shikano, *Study on an Automatic System of Conversational Speech*, Doctor Thesis, NTT technical document (March 1981) (in Japanese).

[7] M.Sugiyama, K.Shikano, *LPC Peak Weighted Spectral Matching Measures*, Electron. Commun. Jap. (Scripta Publishing Co., U.S.A.), Vol.64, pp.50-58 (May 1981).