Efficient method of speech spectrum description using multigrams *

Jan Černocký [1,2], Geneviève Baudoin [1] and Gérard Chollet [3]

(1) ESIEE, Dpt. Signal, Cité Descartes B.P. 99, 93162, Noisy-le-Grand CEDEX, France

tel: +33-1-45.92.66.91/66.46, fax: +33-1-45.92.66.99, emails: cernockj,baudoing@esiee.fr

(2) FEI VUT Brno, Inst. of Radioelectronics, Antonínská 1, 662 09 Brno, Czech Republic

tel: +42-5-41.32.12.75, fax: +42-5-41.21.11.35, email: cernocky@ant.fee.vutbr.cz

(3) ENST, Dpt. Signal, 46 rue Barrault, 75643 Paris CEDEX 13, France

tel: +33-1-45.81.78.84, fax: +33-1-45.88.79.35, email: chollet@sig.enst.fr

Abstract

The multigram method allows us to split a string of symbols into a stream of variable length sequences. We study the application of this method to vector-quantized speech spectra. The results are given in terms of entropy per symbol and multigram dictionary size. To overcome the problem of a great variability of sequences, we suggest a modification of this method with the use of a distance measure. We present the algorithm for modified multigram dictionary training as well as the experimental results. We discuss the possible applications to speech coding and speech recognition.

1 Introduction

In both speech coding and recognition, there is a need of representing the speech signal by as little information as possible. In coding we are limited by the transmission channel capacity. In recognition, it is the generalization of speech events which we expect from the speech signal description. The standard systems work on a segmental basis - the signal is divided into fixed-length frames, which are processed independently, without taking into account the inter-frame dependencies. However, these dependencies exist and when observing a speech spectrogram, we can see on one hand stationnary parts (for ex. in long vowels), on the other hand many rapid spectral transitions. Three techniques: *Matrix Quantization* [1, 2], *Multiframe Coding* [3] and *Phonetic Segmentation* [4] described in

*This work was supported by French Government scholarship No. 94/4516 and by "Elite" programme of Texas Instruments.

the literature, work with speech segments grouping several frames. In these methods, the main problem is the choice of lengths of segments - the fixed length is not the best solution if we take into account the known timing irregularity of different speech events. Several segmentation methods were recently proposed, using either statistical methods or a part of phonetics knowledge. We based our work on dynamic segmentation method for sequences of symbols, proposed by Bimbot et al. in [5, 6] and called *multigram segmentation*. We show that the direct application of this method to speech description on signal-level is not possible and we develop an extension, which we call *modified multigrams* or *multigrams with distance*. This approach is comparable with Variable-to-Variable length Vector Quantization (VVVQ), introduced in [2].

2 Multigram segmentation

The article of Bimbot et al. [5] and following articles [6, 7] were the basis for the first part of our work with multigrams. We will limit us to a brief description and refer the reader to the above mentioned articles. On the input of a segmentation we have a string of symbols $W = w_1 w_2 \dots w_N$. For a segmentation B into sequences $S_1, S_2, \dots S_q$ of length 1 to n we introduce the likelihood

$$L(B, W) = \prod_{k=1}^{q} p(S_k) \tag{1}$$

where q is the number of sequences in this segmentation, S_k is k-th sequence and $p(S_k)$ its probability. The optimal segmentation maximizes this likelihood, therefore

$$L(W) = \max_{\{B\}} \prod_{k} p(S_k) \tag{2}$$

where $\{B\}$ is the set of all possible segmentations. The number of all these segmentations is large and we use a Viterbi-based algorithm to find the optimal one. We define the likelihood of a partial string $w_1 \dots w_{k+1}$ as

$$L(w_1 \dots w_{k+1}) = \max_{1 \le i \le n} p([w_{k-i+2} \dots w_{k+1}]) \times L(w_1 \dots w_{k-i+1}) \tag{3}$$

and by evaluating it for $1 \le k \le N-1$, we obtain the optimal segmentation in the sense of Eq. 2. To be able to segment, it is necessary to dispose of probabilities of sequences, but these are not known a-priori. That is why we build a *dictionary* of such sequences using a training string and an iterative procedure. We initialise the probability of each sequence T, that we can find in the string to $p_{init}(T) = \frac{c_{init}(T)}{C_{init}}$, where $c_{init}(T)$ is the number of occurences of T in W and $C_{init} = N.n$ (number of all possible sequences of lengths 1 to n). Then we reestimate the probabilities in a loop:

- *Segmentation*, where we segment the training string using Eq. 2.

- *Reestimation of probabilities* on the basis of segmented string: $p(S) = \frac{c(S)}{C}$, where c(S) denotes here the number of occurences of S in the *segmented* string and C the total number of sequences after the segmentation. The probability may also be reestimated with a pruning factor a, which helps to eliminate the rare sequences from the dictionary:

$$p'(S) = \frac{c(S)}{C}\left(1 - a\sqrt{\frac{C - c(S)}{C.c(S)}}\right) \tag{4}$$

Article [6] gives another possibility of probability reestimation using ML-EM method.

3 Application to speech spectrum, evaluation

We applied the multigram segmentation and dictionary training to vector quantized speech spectra. We disposed of a telephone database of one speaker, with 8000 Hz sampling frequency and 16 bit quantization. After suppression of silences and high-pass filtering by $1 - 0.95z^{-1}$, we created 20 ms frames with 10 ms overlapping and we calculated 10 LPC coefficients $a_1 \ldots a_P$ (for $P = 10$). These were converted to LPC-cepstrum coefficients $c_1 \ldots c_P$ which formed the vectors used for vector quantization and modified multigrams (see Section 6). We performed a VQ-codebook training (by a simple LBG algorithm) for codebook lengths $L = 2, 4, 8, 16, 32, 64, 128, 256, 512$. We quantized the spectral vectors by these codebooks and we obtained 9 training and 9 test strings of symbols. The length of the former ones was 213270, the length of the later ones 122903.

The results of VQ were evaluated in terms of *average spectral distortion* SD defined as the mean over all frames of the logarithmic spectral distance

$$D = \sqrt{\int_{-1/2}^{1/2} \left[10 \log S(f) - 10 \log \hat{S}(f)\right]^2 df} \quad \text{in dB} \tag{5}$$

where $S(f) = 1/\|A(f)\|^2$ is the power LPC-spectrum and $\hat{S}(f)$ the power spectrum with quantized coefficients. Using the Parceval's relation it can be calculated using LPC-cepstral coefficients $D = \mu\sqrt{2\sum_{i=1}^{\infty}(c_i - \hat{c}_i)^2}$, where c_i and $\hat{c}_i$ are the unquantized and quantized cepstral coefficients respectively and $\mu = \ln(10)/10$ is a constant enabling the conversion to decibels. The infinity in the sum can be replaced by a reasonably chosen number: for ex. for 128, the result is very precise. To limit the computational load, we used only P (the order of prediction filter) as the upper limit of the sum in all evaluations. This simplifiacation introduces an error of up to 12% for SD, but for comparisons, the precision is sufficient.

For the evaluation of "classical" multigrams, it is not necessary to re-evaluate the spectral distortion – it is given by the VQ used. To measure the efficiency of the representation, we compare the entropy of VQ codebook with the entropy per symbol of the multigram dictionary. We define the former as:

$$H(V) = -\sum_{i=1}^{L} p(\boldsymbol{y}_i) \log_2 p(\boldsymbol{y}_i) \tag{6}$$

where $\boldsymbol{y}_i$ are the code-vectors of VQ-codebook and the later as

$$H'(M) = -\frac{\sum_{i=1}^{Z} p(M_i) \log_2 p(M_i)}{\sum_{i=1}^{Z} l(M_i) p(M_i)} \tag{7}$$

where Z is the total number of multigrams in the dictionary, $p(M_i)$ stands for the probability of the multigram M_i and $l(M_i)$ for its length. We note, that the term in the denominator is the average length of multigrams.

As these quantities are based only on the training string, it is necessary to validate them on a test one. We are evaluating the average *rate* for both the VQ and the multigrams. We compare this rate to $H(V)$ and $H'(M)$. The average rate for the VQ is defined by

$$R(V) = -\frac{\sum_{i=1}^{L} c_{test}(\boldsymbol{y}_i) \log_2 p(\boldsymbol{y}_i)}{N_{test}} \tag{8}$$

where $c_{test}(\boldsymbol{y}_i)$ is the number of vectors of the test string represented by code vector $\boldsymbol{y}_i$, and N_{test} is the length of the test string. For the multigrams, we obtain a similar formula:

$$H'(M) = -\frac{\sum_{i=1}^{Z} c_{test}(M_i) \log_2 p(M_i)}{N_{test}} \tag{9}$$

where in this case, $c_{test}(M_i)$ is the number of sequences represented by multigram M_i.

Another important criterion is the size of resulting multigram dictionary, which informs us how easily the appropriate entropy-code can be constructed.

4 Experimental results, discussion

The results of the vector quantization with the codebooks of $2, 4, \ldots 512$ vectors are given in Table 1. We see that the entropy of VQ codebook is validated by the rate obtained on the test string – we do not observe a significant difference between $H(V)$ and $R(V)$.

L	$H(V)$ [bit]	$R(V)$ [bit]	SD [dB]
2	0.999	1.000	4.332
4	1.950	1.930	3.770
8	2.914	2.888	3.199
16	3.881	3.844	2.893
32	4.863	4.830	2.640
64	5.869	5.848	2.413
128	6.858	6.840	2.231
256	7.865	7.853	2.055
512	8.865	8.852	1.907

Table 1: Results of Vector Quantization: codebook size, entropy of the codebook, rate obtained on the test string and spectral distortion.

We performed several test of the multigram segmentation with the maximal lengths of multigrams $n = 2, 4, 6, 8, 10$ and with the penalization factors $a = 0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0$. For each combination, we performed a multigram dictionary training with 10 iterations of the process "segmentation" $\Rightarrow$ "reestimation of probabilities". We evaluated the results in terms of entropy per symbol of the resulting multigram dictionary, we validated these numbers on the test string (calcul of $R(M)$) and we recorded the multigram dictionary size Z.

The optimal results for two VQ codebooks $L = 16$ and $L = 128$ are given in Table 2. Note, that there are two kinds of optimal results:

- (1) where one takes into account only the decrease of multigram dictionary entropy. In this case, we could conclude that for $L = 16$, the bit economy is $H(V) - H'(M) = 3.881 - 1.407 = 2.474$ bits or 63% of the original bit rate. This optimistic result looses its brightness if we look at the dictionary size and at the rate on the test string: 4.051 is more than 4 bits necessary to represent 16 symbols!
- (2) where the optimum is found as the minimal rate on the test string. For $L = 16$ we obtain worse, but more realistic results: the difference $H(V) - H'(M) = 3.881 - 2.264 = 1.617$ bit so that the economy due to the using of multigrams is 41%.

When increasing the number of codebook vectors, we obtain worse results. The difference $H(V) - H'(M)$ (with the (2) criterion) is only $6.858 - 5.414 = 1.444$ bit, or 21% for the VQ codebook $L = 128$.

L	16(1)	16(2)	128(1)	128(2)
n_{opt}	10	6	10	2
a_{opt}	0.5	1.0	0.0	0.0
Z	16225	2593	21368	3451
$H'(M)$	1.407	2.264	1.453	5.414
$R(M)$	4.051	2.328	12.737	5.465

Table 2: Optimal results for the multigram method: optimal length of multigrams, optimal penalization factor, size of resulting multigram dictionary, entropy per symbol of the multigram dictionary and rate obtained on the test string. (1) - optimum for the training string, (2) - optimum for the test string.

Although this method offers good results for written text analysis (see [5, 6, 7]), we conclude, that a direct application to vector quantized speech spectra is impossible. A significant decrease of entropy is obtained only for small codebooks, which are, on the other hand, not enough precise for the spectrum representation. For greater codebooks, the main disadvantage is an enormous multigram dictionary, whose size is comparable to the length of the training string. This is a proof of a strong overlearning, verified on the test strings. We see the reason in a great variability of characteristic symbol sequences, which increases with L: for greater codebooks we can say that almost each longer sequence is unique.

with the corresponding indices of code-vectors (symbols for "classical" multigrams), but we will return to unquantized vectors.

We have defined the distance of two sequences of vectors in a very simple manner as the average Euclidean distance of corresponding vectors. The distance of two sequences $\boldsymbol{X}$ and $\boldsymbol{Y}$ having the same length l is

$$D(\boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{l} \sum_{i=1}^{l} d(\boldsymbol{x}_i, \boldsymbol{y}_i) \tag{10}$$

where $d(\boldsymbol{x}, \boldsymbol{y})$ is the Euclidean distance of two vectors. It is of course possible to use more sophisticated distance definitions (Itakura-Saito for ex.), or to use various weightings in the sum (for ex. depending on energy of corresponding frames). Another possibility is a time alignment of multigrams, which may take advantage of timing variations of characteristic speech parts (various lengths of the same vowel etc.).

In the "modified multigrams" or "multigrams with distance" approach we use a dictionary that consists no more of symbol sequences but of vector sequences (in analogy to VQ we will call them code-multigrams). In the same manner, the input string does not consist of symbols but of unquantized vectors. Similarly as in the "classical" multigram case, we are searching a segmentation into variable length (1 to n) sequences and a procedure for creation of dictionary of typical sequences.

The optimal segmentation of a string $\boldsymbol{X} = \boldsymbol{x}_1 \ldots \boldsymbol{x}_N$ into sequences $\boldsymbol{U}_1 \ldots \boldsymbol{U}_q$ of length 1 to n is given by maximization of the quantity (formerly called *modified likelihood*, but after the protest of Fréderic Bimbot renamed to "*a quantity*")

$$L'(B, \boldsymbol{X}) = \prod_{k=1}^{q} p'(\boldsymbol{M}_k) \tag{11}$$

over the set of all possible segmentations $\{B\}$

$$L'(\boldsymbol{X}) = \max_{\{B\}} \prod_{k} p'(\boldsymbol{M}_k) \tag{12}$$

The main difference from the previous likelihood definition (Eqs. 1 and 2) is the use of the *penalized probability* p'. It is no more a probability of a sequence but a modified probability