Marcin PŁONKOWSKI[1], Pavel URBANOVICH[1,2]

The John Paul II Catholic University of Lublin, Poland (1), Belarusian State Technological University, Belarus (2)

doi:10.15199/48.2018.07.25

Using Full Covariance Matrix for CMU Sphinx-III Speech Recognition System

Abstract*. In this article authors proposed a hybrid system in which the full covariance matrix is used only at the initial stage of learning. At the further stage of learning, the amount of covariance matrix increases significantly, which, combined with rounding errors, causes problems with matrix inversion. Therefore, when the number of matrices with a determinant of 0 exceeds 1%, the system goes into the model of diagonal covariance matrices. Thanks to this, the hybrid system has achieved a better result of about 11%.*

Streszczenie. *W niniejszym artykule autorzy zaproponowali system hybrydowy, w którym pełna macierz kowariancji wykorzystywana jest tylko w początkowym etapie procedury treningowej. W dalszym etapie uczenia, znacząco wzrasta liczba macierzy kowariancji, co w połączeniu z błędami zaokrąglania powoduje problemy z odwróceniem tego typu macierzy. Dlatego też, gdy liczba macierzy o wyznaczniku równym 0 przekracza 1%, system przechodzi do modelu wykorzystującego macierze diagonalne. Dzięki temu system hybrydowy osiągnął wynik lepszy o około 11%. (**Wykorzystanie pełnej macierzy kowariancji w systemie rozpoznawania mowy CMU Sphinx III**).*

Keywords: speech recognition, CMU Sphinx, covariance matrix
Słowa kluczowe: rozpoznawanie mowy, CMU Sphinx, macierz kowariancji

Introduction

CMU Sphinx-III is one of the most popular speech recognition systems [1]. It works very well in continuous speech recognition tasks with a lot of words, regardless of speaker. However, to achieve satisfactory results, system must be trained on the appropriate set of utterances with the reference transcription.

The whole process of speech recognition by decoder starts with acquisition of utterance. Then, the extraction process is performed of the most desirable features (from the point of view of speech recognition system). Decoder analyzes these features using acoustic model, language model and vocabulary. Block diagram is shown in Fig.1.

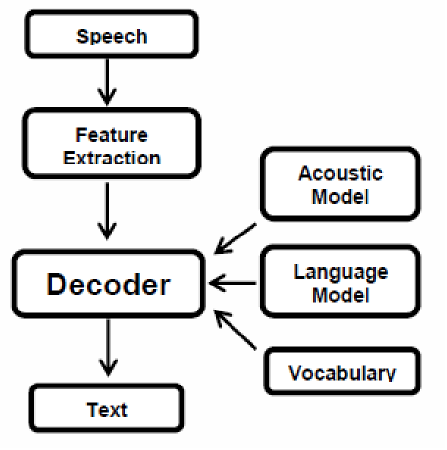

Fig.1. Block diagram of speech recognition system

CMU Sphinx-III is a system that uses statistical methods. Namely, this system is based on a hidden Markov model (HMM). It is now the dominant solution for the most recently designed speech recognition systems. If we have a good learning set (of appropriate size and of appropriate quality) the system gives very good results (word error rate is approximately 15%).

To obtain very good results training set size should take into account the following recommendations:

- 1 hour of recording for command and control for single speaker
- 5 hours of recordings of 200 speakers for command and control for many speakers
- 10 hours of recordings for a single speaker dictation
- 50 hours of recordings of 200 speakers for many speakers dictation

We briefly describe the signal processing front end of the Sphinx III speech recognition system. The front end transforms a speech waveform into a set of features to be used for recognition, specifically, mel-frequency cepstral coefficients (MFCC).

The front end processing performed by the Sphinx-III:

- pre-emphasis
- windowing (Hamming window)
- power spectrum
- mel spectrum
- mel cepstrum.

Sphinx III uses the following features:

- Sample rate: 16000 Hz
- FFT Size: 512
- Frame Size: 410
- Window size: 25ms
- Window shift: 10ms
- Pre-emphasis coefficient: 0.97

Sphinx III uses the following MFCC features:

- 12 MFCC (mel frequency cepstral coefficients)
- 1 energy feature
- 12 delta MFCC features
- 12 double-delta MFCC features
- 1 delta energy feature
- 1 double-delta energy feature
- Total 39-dimensional features

The Sphinx III uses MFCC coefficients for each frame calculated in the same way.

The Sphinx-3 HMM trainer, goes through the following stages [2]:

1. Initialization of Context Independent (CI) models
   a. Creation of model definition file for CI phones
   b. Initialization of models with flat distribution or based on previous segmentation (assignments of phonetic units to speech segments)
2. Training of CI models
   a. Split training data into blocks and compute Baum-Welch variables
   b. Normalize, that is, use the Baum-Welch variables to actually compute the updated transition probabilities, mixture weights, means, variances, etc.

c. Iterate Baum-Welch and normalization until convergence, i.e., until total likelihood changes less than threshold

3. Initialization of Context-Dependent (CD) models
    a. Creation of model definition file for CD phones, by creation of all possible CD phones in the dictionary, and then pruning based on frequency in the training transcripts
    b. Initialization of models based on CI models
4. Training of untied CD models
    a. Split training data into blocks and compute Baum-Welch variables
    b. Normalize
    c. Iterate Baum-Welch and normalization until convergence
5. Building trees
    a. Make linguistic questions
    b. Build classification and regression trees, so as to classify the untied states based on proximity
6. Pruning trees
    a. Prune trees to the desired number of senones, that is, a number of tied states
7. Initialization of tied CD models
    a. Creation of tied CD models definition
    b. Creation of initial set of models from the CI models
8. Training tied CD models
    a. Split training data into blocks and compute Baum-Welch variables
    b. Normalize
    c. Iterate Baum-Welch and normalization until convergence

Covariance matrix

Sphinx III uses HMM with continuous observations modeled as multivariate Gaussian with a probability density function given as (1).

$$f(x)=\frac{1}{\sqrt{(2\pi)^k|\Sigma|}}\exp\left(-\frac{1}{2}(x-\mu)^T(\Sigma)^{-1}(x-\mu)\right), \tag{1}$$

where: *k* - dimension of the feature vector, *x* - future vector, μ - mean of the feature vector, Σ - covariance matrix of the feature vector,

So in order to calculate the probability functions we have to invert the covariance matrix. Due to the fact that we use 39-dimensional features vectors, the covariance matrix will have a dimension of 39x39. The inverse operation of that matrix is computationally expensive. The second problem is the fact that, large amounts of training data are required for reliable full covariance estimation. If we do not have a very large dataset than the matrices are often poorly-conditioned, and do not generalise well [3]. In addition, rounding errors that will occur during the calculation will lead a determinant of a matrix to very small values and consequently to 0. This makes it impossible to invert the matrix.

Despite this shortcomings, full covariance systems have been successfully used for large vocabulary ASR. The most notable example being in the 2004 IBM system [4], where the computational cost was reduced by aggressively pruning Gaussians during the full covariance likelihood computation.

In speech recognition, we frequently assume that the feature vector dimensions are all independent of each other [5]. Then we might reduce the covariance matrix to a diagonal form. The determinant of the diagonal matrix and its inverse are easy to compute. However, due to this simplification, we lose information about the correlation of features.

One of the compromise methods is the use of block block-diagonal covariance matrix [6]. However, we will have to deal with the same problems described in the full covariance matrices only to a lesser extent.

The matrix inverse procedure is implemented in the Baum-Welch algorithm. So we see that stages: 2. ("Training of CI models"), 4. ("Training of untied CD models") and 8. ("Training tied CD models") are important for us. If the number of matrices with a determinant of 0 increases sharply, it will signal to us that it would be a good idea to move from full covariance matrices to diagonal covariance matrices.

In this article we propose hybrid learning method. Namely, the full covariance matrices will be used only in the first stage of learning (stage 2: "Training of CI models"). In this stage we have only 102 matrices (because we have 34 phonemes x 3 states). In the stage 4: "Training of untied CD models" we have 4839 matrices (because we have 1579 triphones x 3 states + 34 monophone x 3 states). So in this step we use the diagonal covariance matrices. These matrices are formed by selecting elements only from the main diagonal.

Tests and results

In this article we analyze the speech recognition accuracy based on the publicly available AN4 database [7]. The database has 948 training and 130 test utterances. All data are sampled at 16 kHz, 16-bit linear sampling. All recordings were made with a close talking microphone.

The directory with training data has 74 sub-directories, one for each speaker. 21 of them are female, 53 are male. The total number of utterances is 948, and the average duration is about 3 seconds, totaling a little less than 50 minutes of speech. The directory with test data has 10 sub-directories, one for each speaker. 3 of them are female, 7 are male. The total number of utterances is 130, totaling around 6 minutes of speech.

There is a protection against inversion of a matrix with a determinant of 0 in the Sphinx III system. Namely, if the determinant is equal to or smaller than 0, the off-diagonal elements are set to 0. The elements from the main diagonal will be set to a value of at least 0.0001.

However, in such a situation we lose a significant amount of information. Therefore, we need to choose the right moment in which it is worth to replace the full covariance matrices into diagonal covariance matrices.

In step 2 ("Training of CI models"), no matrix with a determinant of 0 was recorded. In step 4 ("Training of untied CD models") we recorded as many as 42.44% of the matrices with a determinant equal or less than 0. Such a large number of matrices (with a determinant of 0) distorts the learning procedure. Therefore, the matrices conversion (from full to diagonal) should take place after step 2.

We estimate the accuracy of using number of incorrectly recognized words WER (word error rate), which is defined as:

$$WER=\frac{S+I+D}{N}, \tag{2}$$

where: *S* is the number of substitutions, *I* is the number of insertions, *D* is the number of deletions, *N* is the number of words in the reference.

The word error rate (WER) is the most common way to evaluate speech recognizers. The word error rate is defined as the sum of these errors divided by the number of reference words. It is worth noting that according to the formula (2) WER value may be greater than 100%.

When reporting the performance of a speech recognition system, sometimes word accuracy (WAcc) is used instead:

$$WAcc = 1 - WER \tag{3}$$

The error for the baseline system is equal to 14.8771% (WER). In this situation we use a diagonal covariance matrices. Our tests we also performed with the use of a full and the block-diagonal matrices. These standard training procedures we compared with our hybrid learning method. These results can be seen in Table 1.

Table 1. Word error rate (WER) depending on the type of covariance matrix

Covariance matrix	Diagonal	Block-diagonal	Full	Hybrid (Full to Diagonal)
WER	14.8771%	14.62%	20.43984%	13.58344%

The first surprise may be that the worst result was obtained for the full covariance matrix (WER over 20%). However, it is worth recalling that in the 4th learning stage, as many as 42% of the matrix had a determinant of 0.

In the 4. step, we Iterate Baum-Welch algorithm and normalization until convergence. So it is important to know precisely what percentage of the matrix has a determinant of 0. In Table 2, we see that only in the first step, the number of matrices with a determinant of zero is at a relatively low level. And in step 2 it already reaches nearly 29%.

Table 2. Percentage of matrices with zero determinant

Step	1	2	3
% matrices with zero determinant	0,018%	28,96%	42,51%

Therefore, it is best to move to the diagonal form after the first step of the Baum-Welch algorithm. In practice, this is accomplished by monitoring the number of matrices with a determinant of 0. If this value exceeds 1% then we need to go back to the previous step and change the form of the covariance matrix to diagonal. This algorithm step should be repeated but for the diagonal covariance matrix.

In this situation we will get even better results, namely WER will reach 13.19534% (see Table 3). This means improving the recognition quality (relative to the baseline system) by 11.3%.

Table 3. Word error rate (WER) depending on the type of covariance matrix

Covariance matrix	Diagonal	Block-diagonal	Full	Hybrid2 (Full to Diagonal)
WER	14.8771%	14.62%	20.43984%	13.19534%

The second important parameter is the duration of the training algorithm. We can expect the longest execution time for a model using the full covariance matrix and the shortest for the diagonal covariance matrix model. These results can be seen in Table 4.

Table 4. Duration of the training algorithm

Covariance matrix	Diagonal	Full	Hybrid (Full to Diagonal)	Hybrid2 (Full to Diagonal)
Time (mm:ss)	3:55	26:22	7:42	8:17

As expected, the best time was obtained for the model using the diagonal covariance matrix. In this case, the learning algorithm's duration was almost 4 minutes. Whereas for the model using the full covariance matrix, the algorithm's duration is over 26 minutes. The operating times of our algorithms are about twice as long as for a model with the diagonal matrix but about 3 times shorter than for the full covariance matrix.

Conclusion and future work

In this article authors analyzed use of full covariance matrix in speech recognition systems. The use of this type of matrix involves many problems, which in practice often worsen the results of the system. By using only a diagonal matrix, we lose a great deal of information about the correlation of learning vector coefficients. Hence, the authors proposed a hybrid system in which the full covariance matrix is used only at the initial stage of learning. At the further stage of learning, the amount of covariance matrix increases significantly, which, combined with rounding errors, causes problems with matrix inversion. Therefore, when the number of matrices with a determinant of 0 exceeds 1%, the system goes into the model of diagonal covariance matrices.

Thanks to this, the hybrid system has achieved a better result of about 11%. The disadvantage of this solution is almost twice the length of the algorithm's time.

***Authors**: dr Marcin Płonkowski, prof. dr hab. Pavel Urbanovich, Katolicki Uniwersytet Lubelski Jana Pawła II, Instytut Matematyki, Katedra Systemów Operacyjnych i Sieciowych, ul. Konstantynów 1H, 20-708 Lublin, E-mail: marcin.plonkowski@kul.lublin.pl,*

REFERENCES

[1] The Carnegie Mellon Sphinx Project: CMU Sphinx. http://cmusphinx.sourceforge.net/, Apr 2017

[2] The Carnegie Mellon Sphinx Project: CMU Sphinx Trainer. http://cmusphinx.sourceforge.net/wiki/sphinx4:sphinx4trainer, Apr 2017

[3] Bell P., Full Covariance Modelling for Speech Recognition. PhD thesis, The University of Edinburgh 2010

[4] Chen S., Kingsbury B., Mangu L., Povey D., Saon G., Soltau H., Zweig, G., Advances in speech transcription at IBM under the darpa ears program. IEEE Transactions on Audio, Speech and Language Processing, 14 (2006), nr 5, 1596–1608

[5] Płonkowski M., Urbanowicz P., Tuning a CMU Sphinx-III Speech Recognition System for Polish Language, *Przegląd Elektrotechniczny,* 90 (2014), nr 4, 181-184

[6] Wang R., Zhu X., Chen Y., Liu J., Liu R., Fast likelihood computation method using block-diagonal covariance matrices in Hidden Markov Model. In Proceedings of ICSLP 2002 Taipei, Taiwan, August (2002)

[7] The CMU Audio Databases, AN4 database, http://www.speech.cs.cmu.edu/databases/an4/, Apr 2017