Quasi Text-Independent Speaker-Verification based on Pattern Matching

Michael Gerber, René Beutler, Beat Pfister

Speech Processing Group, Computer Engineering and Networks Laboratory,
ETH Zurich, Switzerland

{gerber,beutler,pfister}@tik.ee.ethz.ch

Abstract

We present a new approach to quasi text-independent speaker verification based on pattern matching. Our method first seeks phonetically matched segments in two speech signals. For all aligned frame pairs of these segments we compute the probability that they were uttered by the same speaker. Based on these frame-level probabilities we take the decision whether the two signals were spoken by the same speaker or not. Our method to find phonetically matched segments does not depend on a speech recognizer. We show that our system performs better than a baseline speaker verification system based on Gaussian mixture models when the signals are long enough. Especially interesting is the fact that a combination of the devised system with the baseline system performs much better than either of the systems alone.

Index Terms: speaker verification, pattern matching, neural networks, dynamic programming

1. Introduction

Statistical modeling, mostly based on Gaussian mixture models (GMMs) is widely used for text-independent speaker verification (SV). The method is computationally not very complex but it does per se not make use of the sequential order of the frames apart from the sequential information implicitly contained in the used features (e.g. delta features). We argue that the sequential order of the frames provides additional information which is helpful to discriminate between speakers. Pattern matching (PM) which is often used for text-dependent SV takes the frame order into account. In this paper we present a new approach which makes pattern matching applicable for a quasi text-independent task.

Various approaches have been presented which aim at including sequential information for text-independent SV. In [1] a method was presented which uses automatic speech recognition (ASR) transcriptions to find common, about phoneme-long segments in two speech signals. The common segments are then compared with a pattern matching approach. The method presented in [2] seeks predefined keywords (frequently used words like *and, I, that, yeah, ...*) in the ASR transcriptions of the two signals. Then GMMs are trained for these keywords. A similar approach which is based on ASR as well is described in [3]. Instead of GMMs, word-level HMMs are trained for the keyword signals. All these approaches are dependent on ASR which makes them language-dependent. For most of these approaches it is observed that they outperform the classical GMM approach only if plenty of data is available to train the speaker models.

Since our way of finding common segments in the two speech signals is not based on ASR it is not restricted to specific predefined words and is therefore in principle language-independent.

2. New Speaker Verification approach

Our method to decide whether two speech signals are spoken by the same speaker or not includes the following 3 steps: First phonetically matched segments (e.g. common words or common syllables) are sought in the two speech signals. This step provides a series of frame pairs where both frames in a pair are phonetically matched (see Section 2.1). In a second step the probability that the two frames in a pair come from the same speaker is computed for every frame pair (see Section 2.1). A multilayer perceptron (MLP) is used for this task. In [4] was shown that an MLP is suitable to calculate this probability and leads to better results than the Euclidean distance which is generally used in PM approaches. Thus we obtain a series of frame-pair-level probabilities. Finally, a global indicator that the two speech signals were spoken by the same speaker can be calculated from these frame-pair-level probabilities. This is described in Section 2.3.

2.1. Seeking equally worded segments

The new method to find common segments in two speech signals uses a so-called phonetic probability matrix as shown in Figure 1. The elements of this matrix are $P_{ij}(x_{1i}, x_{2j})$, i.e., the probability that frame i of signal 1 given as feature vector x_{1i} and frame j of signal 2 given as feature vector x_{2j} are from the same phoneme. Such posterior probabilities can be computed with an appropriately trained MLP as described in Section 4.

In the 3-dimensional representation of this probability matrix, equally worded speech segments show as ridges in about 45° direction. Our procedure to detect such ridges is outlined in Section 3. The output of this procedure are the segments that are common to both speech signals. More precisely, the algorithm delivers a series of phonetically matching frame pairs which are contained in these common segments.

2.2. Calculation of frame-level SV scores

In a second stage for every frame pair a score is computed which stands for the probability that the two phonetically matching frames are from the same speaker. For this purpose we use MLPs which are trained to calculate the posterior probability that the two phonetically matching frames are from the same speaker.

We observed that better results can be achieved if the static features and their corresponding derivatives are not fed into one big MLP but that two MLPs are used (i.e. one for the static features and the other for their derivatives). The reason for this

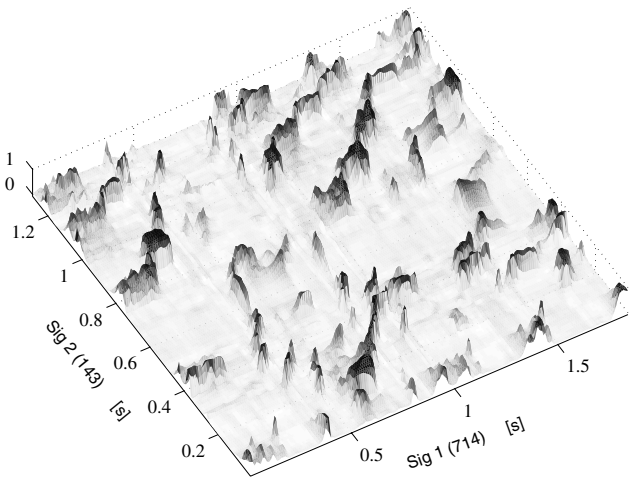

Figure 1: Probability matrix spanned by the German numbers 714 ("siebenhundertvierzehn") and 143 ("hundertdreiundvierzig") uttered by two different speakers. Common segments show as ridges, e.g., the word "hundert" which occurs in both numbers shows as a ridge between 0.6 and 1 s in signal 1 and between 0.1 and 0.4 s in signal 2.

is that the derivatives implicitly already contain some context information. Derivative features of successive frames are therefore more strongly correlated. The static features on the other hand are less intercorrelated since no information beyond the frame boundary is used. In a subsequent averaging step, as we used it to calculate a global score (see Section 2.3), the correlation is disadvantageous. This effect is described in more detail in [4].

2.3. Final decision

Finally we evaluate from the scores of all frame pairs the global score that can be used to decide whether the two speech signals were spoken by the same speaker or not. The global score is the weighted average over the frame-level scores of all found common segments. As a weighting factor the mean intensity of the two frames in a pair is taken. With this intensity-weighting we can reduce the negative effect of silence frames.

3. Detection of common segments

In order to find equally worded segments in two speech signals we have to detect ridges in a phonetic probability matrix P_{ij} such as the one shown in Figure 1. This is done by means of dynamic time warping (DTW). In contrast to the standard application of DTW where processing starts and ends at more or less constrained starting and end points and is unidirectional, we use bidirectional DTW that starts at the peak of a potential ridge and proceeds in both directions until the ridge ends. More precisely the detection is as follows:

1. The processing starts at the point where P_{ij} is maximal.
2. From this initial point DTW is used to follow the potential ridge, i.e. to construct a warping curve in both directions. As usual, slope constraints are used that allow to compensate a maximum local speaking rate difference of the two speech signals of a factor of 2.
3. The ridge ends where P_{ij} is smaller than threshold P_b for two DTW steps.
4. The points of P_{ij} constituting the found ridge are excluded from further processing.
5. If the found ridge is longer than the minimum length L_m, all frame pairs along the warping curve are stored for the SV step described in Section 2.2.
6. If the maximum of all not yet excluded P_{ij} is greater than threshold P_a, processing loops back to step 1

The parameters of the ridge detection, i.e. the thresholds P_a and P_b for starting and ending a ridge, resp., and the minimum length L_m have to be optimized by means of an appropriate data set for maximum SV performance (see Section 5.1).

From the description above it is clear that it can happen that no common segments are detected. This is especially true for short signals. For the fairness of the comparison we omitted this by stepwise lowering thresholds P_a and P_b until at least one segment was detected. For a combined system it would be possible to use only the output of the GMM system in these cases.

An advantage of this approach to detect common segments is its language independence since it is not based on transcriptions generated with ASR.

4. MLP-based probability estimation

In our approach to SV we need two probability measures:

- The first one is required to compute the probability matrix shown in Figure 1. The probability measure used there indicates how likely it is that two frames are from the same phoneme.
- The second probability measure is used to evaluate the frame pair score (cf. Section 2.2). This score indicates some sort of likelihood that the two frames are from the same speaker, given they are from the same phoneme.

Consequently, both probability measures express a kind of posterior probability that two frames are from the same class (class may in our case either stand for phoneme or speaker):

$$P(class(x_1) = class(x_2)|x_1, x_2)$$

where x_1 and x_2 are the feature vectors of the respective frames from the two signals.

Note that in the classical PM approach to speaker-verification distance measures such as the Euclidean distance are used instead of the two posterior probabilities mentioned above.

We have seen that these posterior probabilities can be estimated by an appropriately trained MLP. In this way we have a means to get probability measures which are tailored for specific tasks. The structure of the used MLPs is shown in Figure 2. As can be seen, two feature vectors are fed in parallel to the inputs of the MLP.

We trained such MLPs for our two tasks: a phoneme verification MLP which outputs the posterior probability that two frames (from any speakers) are from the same phoneme and a speaker verification MLP which outputs the posterior probability that two frames are from the same speaker, given they are from the same phoneme.

The networks were trained by means of the backpropagation algorithm to output 1 if the two input vectors are from the same class and -1 otherwise. The output values were linearly mapped to the probability space [0 -1]. All units had a hyperbolic tangent activation function. The number of neurons

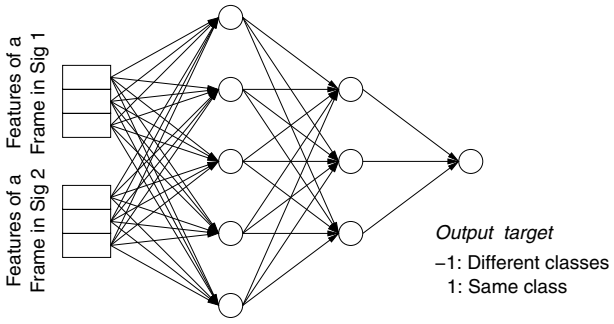

Figure 2: Structure of the used 3-layer perceptrons used to estimate the probability that two feature vectors are from the same class.

can be seen from Table 1. Often a softmax output is used for MLPs which compute probabilities (see e.g. [5]). Empirically we have seen however that for our task a softmax output does not improve the performance of the system. A more detailed description of this application of MLPs can be found in [6].

Task	Phon. verif.	Spkr. verif.
Input dimension	52 ($2 \cdot 26$)	32 ($2 \cdot 16$)
Neurons in 1^{st} hidden layer	80	70
Neurons in 2^{nd} hidden layer	35	18

Table 1: Numbers of neurons in the used MLPs

5. Experiments

For our experiments we used recordings with German three-digit numbers. The speech signals were recorded from male speakers over various land-line and mobile telephones. The signals of every speaker were recorded in 15 sessions which were typically spread over one month. The average duration of a spoken three-digit number is 1.5 s. For convenience we will from now on use the term *word* for a three-digit number.

5.1. Data sets

All data was divided into three disjoint speaker subsets. The *training set* had data from 26 speakers. It was used to train the MLPs and the universal background model (UBM) of the baseline system (cf. Section 5.4). Therefore, the MLPs as well as the UBM are not speaker-specific. The *validation set* contained 10 speakers. It was used to stop the training of the MLP at the right time such that no overfitting effects occurred and to optimize the parameters of the algorithm to detect common segments. The *test set* included 13 speakers.

5.2. Test setup

We tested three systems: our PM method, a GMM-based baseline system and a combination of both in a text-independent SV task. More precisely, the task for each system was to verify if two given speech signals are from the same speaker or not. In every test run the speech signals contained a given number of words in the range of 1 to 7. None of the words (i.e. a full three-digit number) of the first signal occurred in the second one. Therefore only common subwords (e.g. digits occurring in both words) can be detected.

5.3. Feature extraction

We used Mel-frequency cepstral coefficients (MFCCs) from 37.5 ms long frames at a frame rate of 100 Hz. 34 Mel-filters were placed in the frequency range from 100 to 4000 Hz. The cepstral mean of the used signals was subtracted to compensate for linear channel distortions.

For the computation of the probability matrix in Figure 1 we used the MFCCs $0 \ldots 12$ plus their first derivatives and for the speaker verification task we used MFCCs $1 \ldots 16$ plus their first derivatives.

5.4. Baseline GMM system

As a baseline system we used a GMM system as described in [7]. An optimization on the validation set has shown that the optimal number of Gaussians is 1024. The UBM was trained with data from the training set. For each SV trial a speaker model was created by adapting the UBM with the first of the two given signals using a maximum a posteriori (MAP) adaption (see [8]). We have seen, that the system performed best when only the means of the Gaussians were adapted. This is in accordance with the results given in [7]. The log-likelihood (LL) of the second signal was calculated for the adapted speaker model and for the UBM. The difference of the two LL values is the desired output of the GMM system. For the training of the UBM, the MAP adaption and the calculation of the LL we have discarded silence frames.

5.5. Combined system

Since our new SV method and the classical GMM-based SV are completely different we also tested a combination of the two systems, i.e., we used a weighted average of the scores delivered by the two systems. The empirically evaluated optimal weights for the pattern matching system and for the GMM system were found to be $\frac{1}{3}$ and $\frac{2}{3}$, resp. The scores of both systems were normalized to zero mean and unit variance for the optimization of these weights.

5.6. Results

Figure 3 shows the detection error trade-off (DET) curves for the three systems with speech signals that contained 7 words.

An alternative view of the SV results is given in Figure 4, where the equal error rate (EER) for all systems is plotted in function of the number of words in the speech signals. Table 2 additionally gives the mean percentage of the signal that could be used in the PM approach (percentage of the signal covered by the detected common segments).

It can be observed that the presented PM system gets better compared to the GMM system when the number of words in the speech signals increases. This is probably because the portion of the usable signal in the PM method increases more than

Number of words in signal		1	4	7
Total length of speech [s]		1.5	6.0	10.5
Percentage of used signal in PM	same speaker	33	67	79
	diff. speakers	18	41	55
EER [%]	PM	23.4	6.6	3.5
	GMM	17.4	5.9	5.0
	Combined	22.6	4.8	2.3

Table 2: EER of the three systems for several signal lengths

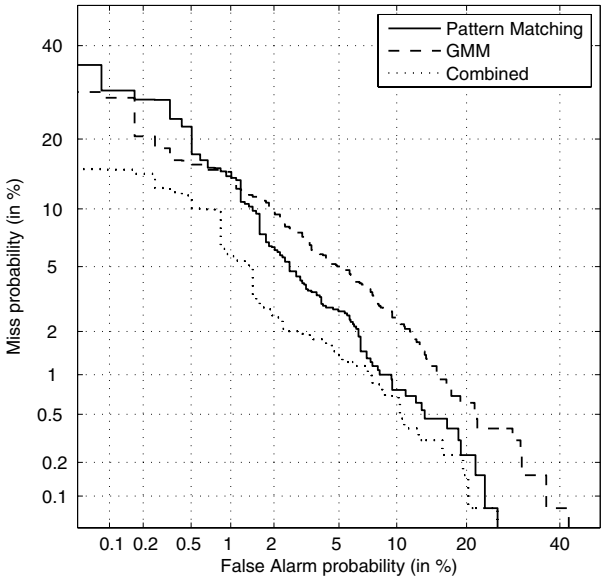

Figure 3: DET curves for the SV test with 7-word signals

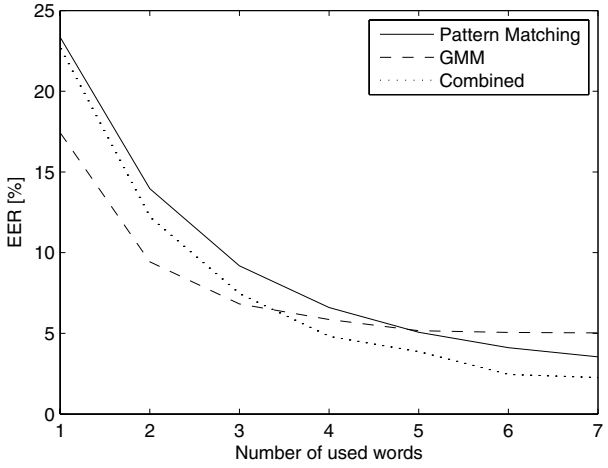

Figure 4: The EER of the GMM, the presented PM system and the combination for speech signals with 1 to 7 words

proportionally with the signal length.

When the number of available words is extremely low, the GMM system outperforms the PM system. When the signals contain more than 3 words the PM system performs about equally well as the GMM system. The PM system even statistically significantly (significance level of 0.3 % according to the McNemar test, see for example [9]) outperforms the GMM system when 7 words are present in the two signals.

As expected, the combined system outperformed the GMM and the PM system when several words were present in the signals. For the cases with at least 4 words in the signals we could detect statistically significant improvements of the SV results of the combined system compared to the GMM system alone according to the McNemar test. The significance levels were 4 % for 4 words, 0.8 % for 5 words and < 0.1 % for 6 words and more.

6. Conclusions

The presented approach to quasi text-independent speaker verification by means of pattern matching is speaker and language-independent, provided the used MLPs have been appropriately trained. In our test, the new SV approach performed comparably to the well-known GMM approach when at least 5 words were present in the two speech signals and even outperformed it when 7 words were present. Moreover we have seen that for signals with at least 4 words a combination of the suggested PM system with a GMM system is better than either of the systems alone. This suggests that the devised PM system exploits other speaker discriminating information from the signals than the GMM system.

Figure 4 indicates that signal length dependent weights could improve the combined system. We will investigate this in our future work.

The experiments reported here were done with speech signals that contained exclusively 3-digit numbers which is obviously advantageous to find common segments. Further experiments with normal speech are therefore required to give a more complete evaluation of our approach.

7. Acknowledgements

This work was partly funded by the Swiss National Center of Competence in Research IM2.

8. References

[1] A. Corrada-Emmanuel and M. Newman et al., "Progress in speaker recognition at Dragon Systems," in *Proc. ICSLP-98,Sydney,Australia*, 1998.

[2] D. E. Sturim and D. A. Reynolds et al., "Speaker verification using text-constrained Gaussian mixture models," in *Proceedings ICASSP '02*, 2002.

[3] K. Boakye and B. Peskin, "Text-constrained speaker recognition on a text-independent task," in *Odyssey 2004 - The Speaker and Language Recognition Workshop, Toledo*, June 2004.

[4] U. Niesen and B. Pfister, "Speaker verification by means of ANNs," in *Proceedings of the ESANN'04, Bruges (Belgium)*, April 2004, pp. 145–150.

[5] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, Wiley Interscience, 2001.

[6] M. Gerber, T. Kaufmann, and B. Pfister, "Perceptron-based class verification," in *Proceedings of Non Linear Speech Processing (NOLISP)*, 2007, (accepted for publication).

[7] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Processing*, vol. 10, pp. 19–41, 2000.

[8] J.-L. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate Gaussian mixture observations of markov chains," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 2, pp. 291–299, April 1994.

[9] L. Gillick and Cox S., "Some statistical issues in the comparison of speech recognition algorithms," in *Proceedings of the ICASSP 1989*, 1989, pp. 532–535.