A NOVEL INITIALIZATION METHOD FOR UNSUPERVISED LEARNING OF ACOUSTIC PATTERNS IN SPEECH DEPARTMENT OF COMMUNICATIONS ENGINEERING TECHNICAL REPORT FGNT-2013-01

Oliver Walter, Joerg Schmalenstroeer and Reinhold Haeb-Umbach

Department of Communications Engineering, University of Paderborn, Germany
{walter,schmalen,haeb}@nt.uni-paderborn.de

ABSTRACT

In this paper we present a novel initialization method for unsupervised learning of acoustic patterns in recordings of continuous speech. The pattern discovery task is solved by dynamic time warping whose performance we improve by a smart starting point selection. This enables a more accurate discovery of patterns compared to conventional approaches. After graph-based clustering the patterns are employed for training hidden Markov models for an unsupervised speech acquisition. By iterating between model training and decoding in an EM-like framework the word accuracy is continuously improved. On the TIDIGITS corpus we achieve a word error rate of about 13% by the proposed unsupervised pattern discovery approach, which neither assumes knowledge of the acoustic units nor of the labels of the training data.

***Index Terms*—** unsupervised learning, dynamic time warping

1. INTRODUCTION

Unsupervised acoustic pattern discovery has the task to find recurring patterns in recordings of speech or general audio, which, after clustering, can be assigned labels which may then be used to train a classifier. One application is the unsupervised training of an automatic speech recognizer (ASR), which does not require costly labeled training data. Another application is the classification of audio recordings. The discovered patterns will be used to classify the recordings based on the distribution of these patterns and their sequence in the recordings.

Non-negative matrix factorization (NMF) has been proposed for unsupervised pattern discovery in audio and speech, by factorizing a spectrogram of the audio data into basis vectors and activations. Those basis vectors could be related to phonemes [1]. A major disadvantage of NMF is that temporal correlations and differences in speaking rate are not modeled by standard NMF. Although some extensions of NMF have been proposed to overcome these issues, dynamic time warping (DTW) seems to more elegantly account for time variability. DTW searches a time alignment path between two sequences of feature vectors and thus provides a measure of similarity between the two. Variants of DTW have been extensively investigated for finding recurrent patterns in audio or speech [2, 3, 4].

An issue with DTW, however, is its large computational complexity. To find two sub-sequences of acoustic feature vectors which are similar to each other, segmental DTW (SDTW) has been proposed, where an alignment path is searched for which may be shorter than either of the two feature vector sequences to be compared. Since the sub-sequences may start at arbitrary positions the number of starting points and thus alignment paths to be computed grows quadratically with the database size.

In order to reduce the number of alignment paths to be searched, it was suggested to subdivide the distance matrix into diagonal bands and to search alignment paths for each band separately, followed by a path trimming algorithm to find the subsequences with largest acoustic similarity [3]. A distance matrix, see Fig. 1 further below for an example, contains all pairwise distances between the feature vectors of two utterances. An alternative to SDTW, which avoids the rigid band structure, is unbounded DTW (UDTW) as proposed in [5]. Here, the search space is reduced by requiring the starting points for the alignment paths to lie on specific horizontal or diagonal lines of the distance matrix. Additionally, the allowed transitions within the alignment paths were restricted and the path was searched in forward and backward direction. In [6] we proposed a combination of DTW and clustering using a modified K-Means++ approach, which avoided the computation of all inter-utterance distance matrices.

However, the measures to reduce the complexity usually go hand in hand with reducing the probability of finding all recurrent patterns. For example, with the band structure and the path trimming introduced in [3], only one coincidence can be detected, even if the sequences compared contain two similar feature subsequences lying in the same band. Also the restriction on the location of the starting points in [5] clearly reduces the chances to find all matching subsequences.

In this contribution we propose a novel starting point selection which considerably reduces false alarm and missed hit rates in finding similar acoustic patterns at hardly an increase in computational effort. The acoustic patterns are subsequently clustered and used to train a hidden Markov model (HMM). Here we propose an iterative training, where the clusters deliver only an initial labeling of the training data, which is subsequently improved in the course of the HMM training.

The paper is organized as follows: In the next section we describe the steps required to train a speech recognizer in a completely unsupervised fashion. Emphasis is placed on the novel initialization procedure for recurrent pattern discovery in Section 2.1.1 and the proposed iterations between training and decoding to improve the classification rate in Section 2.3. Experimental results are given in Section 3 and we finish with conclusions in Section 4 and a description of the relation to prior work in Section 5.

The work was in part supported by Deutsche Forschungsgemeinschaft under contract no. Ha 3455/9-1 within the Priority Program SPP1527 ”Autonomous Learning”

2. TASK OVERVIEW

The unsupervised training of a speech or audio classifier can be subdivided into the following steps

i) Discovery of recurrent sequences of audio patterns

ii) Clustering of audio patterns

iii) Training of the classifier

In this work we will focus our discussion on i) and iii).

2.1. Pattern Discovery

Let $\boldsymbol{D}_{a,b}$ be the distance matrix computed from two sequences $\boldsymbol{X}_a = \{\boldsymbol{x}_a(1), \ldots, \boldsymbol{x}_a(L_a)\}$ and $\boldsymbol{X}_b = \{\boldsymbol{x}_b(1), \ldots, \boldsymbol{x}_b(L_b)\}$ of feature vectors of length L_a and L_b frames, respectively. The matrix has entries $[\boldsymbol{D}_{a,b}]_{i,j} = d(\boldsymbol{x}_a(i), \boldsymbol{x}_b(j))$, where $d(\cdot,\cdot)$ is an appropriately chosen distance measure. The goal of DTW is to find an optimal sequence of mappings, i.e., an alignment path $\Phi_k = (i_k, j_k)$, between $\boldsymbol{X}_a$ and $\boldsymbol{X}_b$, where k is the index of a mapping between a pair of feature vectors at the time instances i and j. The sequence of mappings is found by minimizing the cumulative distance

$$\overline{D}\left(\boldsymbol{X}_a, \boldsymbol{X}_b\right) = \sum_{k=1}^{K} d\left(\boldsymbol{x}_a(i_k), \boldsymbol{x}_b(j_k)\right) \qquad (1)$$

with respect to certain constraints on the allowed transitions from (i_k, j_k) to (i_{k+1}, j_{k+q}). Here, K is the length of the alignment path.

Since DTW delivers only a single alignment path over the complete length of the two sequences we use segmental DTW (SDTW) from [3] which aims to find multiple alignment paths. This is achieved by subdividing the search space, i.e., the distance matrix, into diagonal bands, with a predefined width R, and searching for individual alignment paths in each band separately, see Fig. 1.

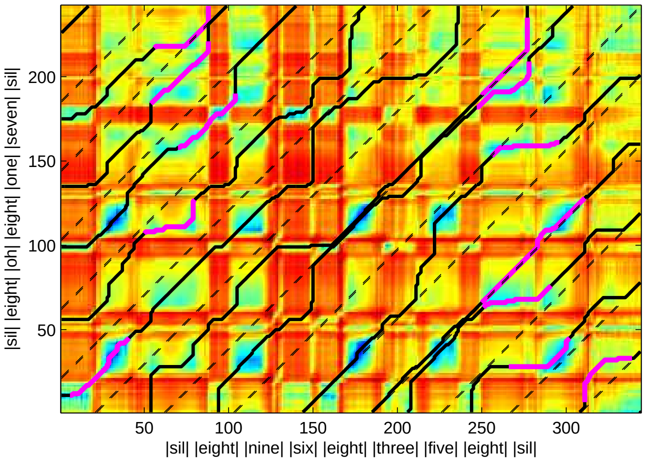

Fig. 1. Distance matrix between the feature vector series of two utterances including the diagonal band limits (dashed lines), the discovered alignment paths (black) and LCMA paths (magenta)

In a second step the found sequences of mappings for each band are refined by searching for the ”length constrained minimum average” (LCMA) path, i.e., the subsequence on the alignment path for which the accumulated average distortion along the path

$$\overline{D}_{\text{LCMA}}\left(\boldsymbol{X}_a, \boldsymbol{X}_b\right) = \frac{1}{k_e - k_s + 1} \sum_{k=k_s}^{k_e} d\left(\boldsymbol{x}_a(i_k), \boldsymbol{x}_b(j_k)\right) \qquad (2)$$

is minimal, under the constraint that the LCMA length $k_e - k_s$ is larger than some given minimum length L_{min}.

The band structure, however, prevents that all relevant pattern matches are found. Firstly, some bands could be positioned such that the edges of them will cut through valid sequences. Secondly, within one band only one sequence of mappings can be found, even if multiple similar patterns exist. Thirdly, a sequence of mappings will be searched (and found) in every sub-band, even if no valid pair might be present, resulting in false detections.

UDTW has similar problems. The starting points are selected along diagonal or vertical lines. This arbitrary selection of starting points is questionable since choosing too many starting points might produce more false detections or poor mappings and with too few starting points not all pattern matches will be found.

2.1.1. *Improved Initialization*

While a completely unconstrained search is prohibitive due to complexity reasons, the following procedure relaxes the constraints imposed by SDTW and UDTW. We propose to choose the centers of the search regions in SDTW and the starting points in UDTW to be the local minima of a smoothed log-distance matrix. If SDTW is used, this choice minimizes the probability that the alignment path will hit the search region boundaries. Further, the search region is limited to a certain length L_{max} which allows to find more than just one recurring pattern in a band. If UDTW is used, the alignment is extended in forward and backward direction alternatingly, starting from the local minimum. To avoid finding adjacent minima which correspond to the same alignment path, exclusion areas around selected minima are introduced within which all local minima are discarded.

The number of minima to be searched for is chosen proportional to the sum of the lengths L_a and L_b of the two sequences $\boldsymbol{X}_a$ and $\boldsymbol{X}_b$ and is given by $N_{\text{min}} = \left\lceil \frac{L_a + L_b}{R_{\text{min}}} \right\rceil$, which results in a similar number as the number of bands used in [3]. Note that the number of minima to be searched for can be chosen independently of the width of the search region, which is in contrast to [3] where the number of regions and thus LCMA paths increases with decreasing band width.

Fig. 2 shows the search regions (dashed boxes), the alignment paths and the LCMA paths obtained with the proposed approach for the same two utterances as in Fig. 1. If the old band limits were drawn in Fig. 2 one would observe that zero, one or multiple LCMA paths can be found per band. Note that the distance matrix has been smoothed in a preprocessing step to avoid the detection of spurious minima, as will be described in the next section. However, the alignment path calculation and the LCMA path search are carried out on the non-smoothed distance matrix.

2.1.2. *Distance Matrix Smoothing*

By smoothing the entries of the distance matrix isolated local minima with otherwise large distance values in its vicinity will be removed. Matches of subsequences resulting in low distance values with a certain temporal extent will, however, be preserved. They correspond supposedly to the recurrent patterns to be discovered.

Smoothing is achieved by sweeping a 2-dimensional hexagonally shaped smoothing kernel $\boldsymbol{K}$ over the logarithmic distance ma-

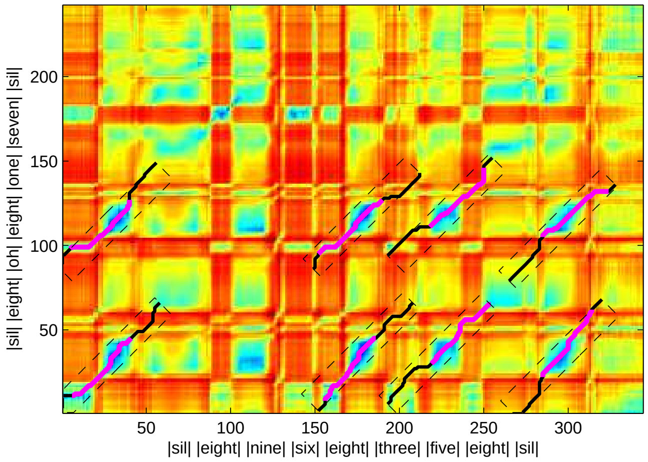

Fig. 2. Distance matrix including search region boundaries (dashed lines), alignment paths (black) and LCMA paths (magenta) according to the proposed method

trix:

$$\left[\widetilde{\boldsymbol{D}}_{a,b}\right]_{i,j} = \sum_{n=-\kappa}^{\kappa} \sum_{m=-\kappa}^{\kappa} \log\left\{[\boldsymbol{D}_{a,b}]_{i+n,j+m}\right\} \cdot [\boldsymbol{K}]_{n,m} \qquad (3)$$

where $\kappa = 2$, and where the kernel is given by

$$\boldsymbol{K} = \begin{pmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 2 & 2 & 1 \\ 1 & 2 & 3 & 2 & 1 \\ 1 & 2 & 2 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix} \qquad (4)$$

After some informal experiments we decided to use the cosine distance metric

$$d(\boldsymbol{x}_a(i), \boldsymbol{x}_b(j)) = \frac{1}{2}\left(1 - \frac{\boldsymbol{x}_a^T(i)\,\boldsymbol{x}_b(j)}{|\boldsymbol{x}_a(i)||\boldsymbol{x}_b(j)|}\right) \qquad (5)$$

and to smooth the logarithmic values of the distance matrix $\boldsymbol{D}$, since this combination yielded the best results.

A straightforward implementation of the smoothing by the hexagonal kernel would have required 19 multiplications and 19 additions in the worst case. A more efficient realization, however, can be found by following [7]. The calculation of the average value of a sliding window along a one dimensional vector requires only one addition of the new value which is added to the window and the subtraction of the old value which is removed from the window. The averaging along the rows results in a horizontal line kernel, the subsequent averaging along each column results in a rectangular kernel and the final averaging along the diagonals results finally in the hexagonal kernel given by eq. (4). This implementation constructs the kernel $\boldsymbol{K}$ step-by-step and requires only 6 additions per element of the log distance matrix $\widetilde{\boldsymbol{D}}_{a,b}$.

2.2. Pattern clustering

The pattern discovery step delivers pairwise similarities of patterns found during the comparison of two feature vector sequences. The next step is to cluster these pattern matching pairs. In order to reduce the amount of false matching pairs, only patterns up to a maximum distance of $D_{\max}$ are used in the two step clustering.

Here, we adopted the graph clustering algorithm of [8] which was also used in [3]. For details, the reader is referred to these publications.

2.3. Iterative Training of Classifier

For the training of the classifier we use the clusters derived during the clustering step for labeling the training data with dummy labels (lacking any meaning in terms of word identity). We assume to have some a priori knowledge about the amount of clusters and thus focused on the 11 biggest clusters, as this is the expected number of digits. If the clustering process ends up with more clusters, the remaining ones are dropped. Admittedly this is a small limitation of the aspect “unsupervised”, however, it allows the performance comparison of our approach with a speech recognizer trained in a supervised manner.

The cluster labels now form the transcriptions to train HMMs for a speech recognizer. Note that only the sequence of patterns is used for an embedded training of the HMMs and that no isolated training of the HMMs was performed, since we do not derive start or stop indices from the found segments. The word boundaries are rather found as a side product of the embedded training.

It is important to note that the cluster labels serve only as initial transcriptions. Exploiting the power of statistical models, the trained acoustic model was used to decode the training data and thus to determine an updated transcription, which in turn is input to the next iteration of the acoustic model training. Overall, the iterative training is governed by the following two equations:

$$\Lambda^{r+1} = \text{argmax}_{\Lambda} \prod P\left(\boldsymbol{X}|T^{r};\Lambda\right) \qquad (6)$$

$$T^{r+1} = \text{argmax}_{T}\, P\left(T|\boldsymbol{X};\Lambda^{r+1}\right) \qquad (7)$$

where r is the iteration counter. At first the HMMs Λ are trained using the transcriptions T from the clustering process and the audio features $\boldsymbol{X}$. Subsequently, the trained HMMs are employed to decode the training data and derive new transcriptions. The new transcriptions are then used in the next training iteration. The iterations could be extended to also train a language model. For the task considered here (connected digit recognition), a language model, however, was not necessary.

This iterative scheme resembles the acoustic unit training presented in [9], where, however, the goal was to discover acoustic events rather than recognize speech.

3. EXPERIMENTAL RESULTS

We performed our experiments on the TIDIGIT Database using the whole subset of training speakers consisting of 112 speakers and 77 digit sequences per speaker, where each speaker was processed separately. We used the ETSI standard front-end to extract the first 13 Mel-frequency cepstral coefficients (MFCC) from the audio data and additionally the first and second order derivatives, resulting in a 39 dimensional feature vector per 10 ms frame.

3.1. Pattern Discovery

As a performance measure for the pattern discovery step we used the false alarm rate FR and the missed hit rate MR defined as follows:

$$\mathrm{FR} = \frac{N_f - N_r}{N_f} \tag{8}$$

$$\mathrm{MR} = \frac{N - N_f}{N}, \tag{9}$$

where N_f is the total number of found alignment paths/pairs, N_r the number of correctly found pairs and N the number of correct pairs in the database. The resulting values of the performance measures are displayed in a receiver operating characteristic (ROC). Different tradeoffs between FR and MR are found by varying the parameter D_{max} in the pattern clustering stage, see Sec. 2.2.

The proposed initialization was applied to both the SDTW-based pattern discovery approach of [3] and to UDTW described in [5]. Note that in UDTW sequences of mappings are searched for by extending the path at the starting point as long as the mean distortion of the path is below a value D_{max} and no other path at the point to which the path would be extended to has a lower mean distortion. This path extension will be done in forward and backward direction around the initialization point.

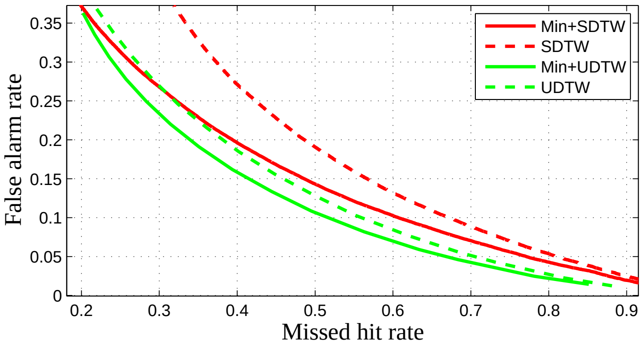

Fig. 3. Receiver operating characteristic of proposed initialization for SDTW (“Min+SDTW”) and for UDTW (“Min+UDTW”), compared to original segmental DTW (SDTW) and unbounded DTW (UDTW)

The experimental results from the TIDIGIT database are depicted in Fig. 3. Obviously, the proposed initialization approach improves the effectiveness of both, SDTW (“Min+SDTW”) and UDTW (“Min+UDTW”), in terms of the defined performance measures. Especially for a fixed false alarm rate the missing rate is lower if the new starting point selection is used.

3.2. Training and Decoding

As explained before the clustering results are used to assign a sequence of labels to each audio file for the HMM training of an automatic speech recognizer. The HMMs consisted of 18 states per word and one Gaussian per state.

In Tab. 1 the experimental results in terms of word accuracy (ACC), word error rate (WER), deletions (DEL), substitution (SUB) and insertions (INS) are given after each training/decoding iteration for up to five iterations. Additionally, the amount of processed files N_{Files} is stated. From iteration 0 to 1 the amount of files increases, since the HMM based recognition allows the addition of files which were dropped during the initial clustering step. As input to the clustering step we used the pattern matching pairs of the “Min+UDTW” approach.

Table 1. Speech recognition results on TIDIGIT database using iterative unsupervised learning of patterns in speech

r	N_{Files}	ACC	SUB	DEL	INS	WER
0	8507	80.90	9.98	9.13	5.51	24.62
1	8623	85.97	9.06	4.97	0.98	15.01
2	8623	87.09	8.94	3.97	0.83	13.74
3	8623	87.60	8.89	3.51	0.82	13.22
4	8623	87.82	8.88	3.29	0.82	12.99
5	8623	87.94	8.92	3.15	0.83	12.89

From the experimental results it can be seen that the amount of deletions, insertions and substitutions significantly decreases in the course of the iterations, and thus the overall word error rate is reduced. Finally, we end up with an WER of below 13%.

4. CONCLUSIONS

We have presented a novel initialization method for the unsupervised discovery of recurrent acoustic patterns in speech. It was shown to exhibit lower false alarm and missed hit rates than two competing state-of-the-art approaches and was shown to be applicable to both the SDTW approach of [3] and the UDTW approach of [5]. The pattern labels obtained from graph clustering were taken as initial labels for a HMM training. The training was carried out completely unsupervised by iterating between updating the HMM model parameters from a given transcription, and recomputing the transcriptions by recognizing the acoustic data using the HMM models trained so far. The word error rate was shown to be reduced in each iteration.

Since both the acoustic patterns, i.e., the dictionary, and the labels, i.e., the transcription are assumed unknown, the approach is applicable to learn a completely unknown language from acoustic observations. Here, language does not only refer to a human language but also to arbitrary acoustic patterns. The approach may thus be used for acoustic scene classification.

5. RELATION TO PRIOR WORK

The use of SDTW and UDTW for pattern discovery was taken from [3] and [5], respectively. Here, we present an improved initialization, resulting in a better pattern discovery performance. The iterative training approach is similar to [9] but has been adopted to a speech recognition task here. This work is a significant extension of our prior work in [6] by moving from isolated digits and acoustic events to digit sequences.

6. REFERENCES

[1] V. Stouten, K. Demuynck, and H. Van hamme, “Discovering phone patterns in spoken utterances by non-negative matrix factorization,” *Signal Processing Letters, IEEE*, vol. 15, pp. 131 –134, 2008.

[2] Mi Zhou and Man Hon Wong, “A segment-wise time warping method for time scaling searching,” *Inf. Sci.*, vol. 173, pp. 227–254, June 2005.

[3] A.S. Park and J.R. Glass, “Unsupervised pattern discovery in speech,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 16, no. 1, pp. 186 –197, jan. 2008.

[4] Yaodong Zhang and James R. Glass, “Unsupervised spoken keyword spotting via segmental dtw on gaussian posteriorgrams,” in *IEEE SLT Workshop, Miami, USA*, 2012.

[5] Xavier Anguera, Robert Macrae, and Nuria Oliver, “Partial sequence matching using an unbounded dynamic time warping algorithm,” in *Int. Conf. on Acoustics Speech and Signal Processing (ICASSP)*, 2010.

[6] Joerg Schmalenstroeer, Markus Bartek, and Reinhold Haeb-Umbach, “Unsupervised learning of acoustic events using dynamic time warping and hierarchical k-means++ clustering,” in *Interspeech 2011*, 2011.

[7] Wojciech Jarosz, “Fast image convolutions,” in *ACM SIGGRAPH workshop, Illinois, USA*, 2001.

[8] M. E. J. Newman and M. Girvan, “Finding and evaluating community structure in networks,” *Physical Review E*, vol. 69, no. 2, pp. 026113+, Feb. 2004.

[9] Sourish Chaudhuri, Mark Harvilla, and Bhiksha Raj, “Unsupervised learning of acoustic unit descriptors for audio content representation and classification,” in *INTERSPEECH*. 2011, pp. 2265–2268, ISCA.