Available online at www.sciencedirect.com
ScienceDirect
Speech Communication 51 (2009) 1124–1138
SPEECH COMMUNICATION
www.elsevier.com/locate/specom

Unsupervised learning of time–frequency patches as a noise-robust representation of speech

Maarten Van Segbroeck *, Hugo Van hamme

Katholieke Universiteit Leuven, Dept. ESAT, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium

Received 12 February 2009; received in revised form 6 May 2009; accepted 9 May 2009

Abstract

We present a self-learning algorithm using a bottom-up based approach to automatically discover, acquire and recognize the words of a language. First, an unsupervised technique using non-negative matrix factorization (NMF) discovers phone-sized time–frequency patches into which speech can be decomposed. The input matrix for the NMF is constructed for static and dynamic speech features using a spectral representation of both short and long acoustic events. By describing speech in terms of the discovered time–frequency patches, patch activations are obtained which express to what extent each patch is present across time. We then show that speaker-independent patterns appear to recur in these patch activations and how they can be discovered by applying a second NMF-based algorithm on the co-occurrence counts of activation events. By providing information about the word identity to the learning algorithm, the retrieved patterns can be associated with meaningful objects of the language. In case of a small vocabulary task, the system is able to learn patterns corresponding to words and subsequently detects the presence of these words in speech utterances. Without the prior requirement of expert knowledge about the speech as is the case in conventional automatic speech recognition, we illustrate that the learning algorithm achieves a promising accuracy and noise robustness.
© 2009 Elsevier B.V. All rights reserved.

Keywords: Acoustic signal analysis; Language acquisition; Matrix factorization; Automatic speech recognition; Noise robustness

1. Introduction

It is remarkable how babies exposed to a language acquire it naturally without deliberate efforts of teaching or learning. Before they can even speak, infants gather immense amounts of information while listening to human voices in their surroundings. During this stage, their brains are being tuned to a specific language. With only a surprisingly small amount of supervision, they succeed in learning new words from a spoken language. Most words are not being explained to them, but are learned from their significance in the world they live in. Although these streams of language data are large and appear to be unsegmented, infants show the ability to distinguish the different units of the language and to acquire how these units are linked in meaningful patterns, such as words.

Evidence exists (Chomsky, 2000) that the basic ability to acquire language is innate to the child, e.g. the basics of human *speech* is built into babies' brains. However, no specific structural property of *language* has yet been proven to be innate and any infant seems equally capable of acquiring any language. Future research still has to reveal what in human language is inborn into the infant's brain and how they succeed in learning the language through experience and exposure to a specific speech community. Moreover, during their lifespan, humans are exposed to variations of what is being uttered. These variations can be acoustical (such as different speaking styles, accents or speech distortions caused by e.g. background noises) as well as on the level of interpretation in the context used. Nevertheless, humans have the ability to continuously learn and adapt to these new situations.

* Corresponding author. Tel.: +32 16 32 11 41; fax: +32 16 32 17 23.
E-mail addresses: maarten.vansegbroeck@esat.kuleuven.be (M. Van Segbroeck), hugo.vanhamme@esat.kuleuven.be (H. Van hamme).

0167-6393/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.specom.2009.05.003

Although current systems for automatic speech recognition (ASR) show to be successful in some aspects, their performance can only be guaranteed if these systems are task-specifically programmed and adjusted to the (predicted) acoustic challenges in which they will operate. This way, ASR-systems are unable to adapt to situations different from the one seen during training and are mostly unreliable in real life situations.

Our work is motivated by the idea that engineering approaches have fallen short in the design of ASR-systems and that inspiration has to come from human language learning and speech perception, an idea that was also postulated in other research work (Baker et al., 2006; Scharenborg et al., 2005). This paper does not claim to explain human language learning, neither does it have the intention to learn grammar, world knowledge or pragmatics. However, we will show that a small vocabulary can be learned from scratch using a bottom-up approach from a spectral analysis of speech signals. To this end, we intend to build a system that automatically discovers the structure in the data, learns the patterns, links them with the words of a vocabulary and finally recognizes them in unseen (noisy) speech data. Our work is related to previously reported approaches of unsupervised language learning (Scharenborg et al., 2007; Qiao et al., 2008; Brugnara et al., 1993; Aversano et al., 2001; Siivola et al., 2003). However, in these approaches the units are phones, phonemes or sub-word items, while in this work we search for recurring acoustic patterns in the time–frequency plane. Moreover, instead of acquiring the words of the language by concatenating these units, we assume that words can be represented by a sparse combination of these patterns and can be learned by discovering similarities in the activations of these patterns.

A first step in language acquisition is to build representations of speech that are to a great extent speaker-independent and robust to noise. The first part of the paper explains how recurring acoustic patterns are discovered in speech data without supervision, a problem that was also addressed in a variety of other research work, see e.g. (Park and Glass, 2005; Stouten et al., 2008; Smaragdis, 2007; Meyer and Kollmeier, 2008). The learning algorithm involved makes use of non-negative matrix factorization (NMF) introduced by Lee and Seung (2001). Thanks to the non-negativity constraints, NMF decomposes a matrix in additive (not subtractive) components, resulting in a parts-based representation of the data. NMF can therefore be seen as a learning algorithm that, when applied to an appropriate feature space, finds the parts or objects that the training data are built of.

We will apply NMF to magnitude spectrograms in order to discover typical patterns in the time–frequency plane (the parts) that can be combined additively to form spectrograms of speech. We will consider spectral analyses over longer time windows than the centisecond scale usually considered in automatic speech recognition. Instead, the spectral patterns that are found have a duration in the order of hundreds of milliseconds. Other researchers have also observed that speech features spanning a longer time interval such as TRAPs (Hermansky and Sharma, 1998) and its variants show improved robustness to noise (Hermansky and Sharma, 1997). Other examples of long-span features are MRASTA filtering (Hermansky and Fousek, 2005) or modulation spectra (Kingsbury et al., 1998; Tyagi et al., 2003). Some work also explicitly looks at time–frequency representations (Meyer and Kollmeier, 2008; Kleinschmidt, 2003; Ezzat et al., 2007). An important difference with the current work is that our time–frequency representation results from a parts-based representation that is learned without supervision. Other authors have used NMF for this purpose. Our approach is most closely related to that of Virtanen (2007), who additionally imposes temporal continuity. Convolutional NMF by Smaragdis (2007) or the variant by O'Grady and Pearlmutter (2008) can also be used to find speech patterns in the time–frequency plane. The discovered speech units seem to be best described as phones, while our units are best described as "acoustic events", such as bursts or formant trajectories. The current work differs in other respects. Firstly, the patterns are discovered from a combination of two complementary feature representations that either reveal timing or frequency structure and which are derived from a time–frequency reassignment spectrogram. Subsequently, these spectral features are segmented into two-dimensional overlapping time-slices which are stacked into column vectors. Recurring time–frequency dependent patterns and bases are then found by applying NMF on these vectors. By enforcing sparsity constraints in NMF, both timing and frequency information are modeled by the obtained bases. Secondly, we use conventional NMF instead of convolutive NMF (cNMF). Although the convolutive variant is appealing from a theoretical point of view, we have found from analyses of parts based on cNMF that it is less resistant to noise. Also, the computational requirements are significantly higher for cNMF. Thirdly, we also add a pattern recognition step to show speech recognition based on the discovered time–frequency patterns and demonstrate the robustness to noise thus obtained.

The bases are acoustic patterns and they will also be referred to as *time–frequency patches* of speech. From a neuroscience point of view, we could relate the process of discovering and acquiring these patches with the learning and/or evolutionary process by which humans have developed an auditory system that is exceedingly sensitive to speech sounds, though we do not claim that what we present here is a validated model of the neurophysiological mechanism. By describing speech in terms of these patches, we show how meaningful objects of the language such as words are linked to patch activation patterns. These patterns appear to be unrelated to speaker-specific properties and remain clearly visible when noise is added to the speech signals. Similarly to our auditory system, the proposed model seems to be skilled in easily distinguishing speech

from other environmental sounds, without the use of noise models or noise reduction techniques.

In the second part of this paper, our aim is to let a computer system acquire the vocabulary of a language by detecting, segmenting and learning the recurring activation patterns from the continuous stream of speech. To this end, the above mentioned speech model will be exploited in a language learning system. Similar to human speech recognition, the proposed system is able to acquire a language on clean training speech using weak supervision without knowing the words until after the acquisition process is completed. Key to the success of the system is the ability to discover recurring patterns in the activations of the *time–frequency patches* in speech across time. Therefore, the speech data is transformed into a high-dimensional vector representation, called "histograms of acoustic co-occurrence" (HAC) which are computed by accumulating the co-occurrence counts of acoustic events (Van hamme, 2008b). For this work, these events are quantized patch activation vectors. Subsequently, a learning algorithm with weak supervision and which is again based on non-negative matrix factorization (NMF) is proposed to discover recurring patterns through the use of HAC-features and link them with the lexical items of a language. Hence, in contrast to hidden Markov model (HMM) based speech recognition systems, no expert knowledge from audiology or phonology is incorporated in our system, neither do we need any a-priori information about what the words are and how they are composed. After the learning process, the system shows a remarkably good performance in detecting the words of the language in both clean and noisy speech data. Hence, the learning system could form the basis of an alternative framework for robust speech recognition.

Fig. 1 shows the structure of the proposed system. In the learning stage, the NMF in the first layer is performed on the time–frequency representation to acquire time–frequency patches in clean training speech. From these learned patches, the patch activations along the time axis are then computed. In the second layer of the system, the patch activation vectors are quantized and transformed to the HAC representation. On these HAC-features, another NMF is performed to learn the HAC of the vocabulary words. During testing, the first NMF computes the patch activations from the learned time–frequency patches and the second NMF computes the word activations from the learned HAC-models to detect the words in the utterance. To assess the efficacy of the language learning, a third layer was added to the system to detect the words from the word activations on unseen speech data.

The outline of the paper is as follows: Section 2 explains how the time–frequency patches of the first layer are learned from speech and how their activations along the time axis are computed. In Section 3, the concept of HAC-models is restated from which the HAC of the words are learned in the second layer. Section 4 describes how this learning system can be applied to detect words in speech utterances. A small vocabulary word discovery experiment was conducted on the Aurora2 digit database and experimental results are given in Section 5. Finally, conclusions can be found in Section 6.

2. Layer 1: Time–frequency patch discovery

The production of human speech can be regarded as a process of combining a small number of spectral patterns into many more different sequences. In this section, our goal is to find a set of patterns by analyzing continuous speech recordings. Moreover, we would like that (i) these patterns accurately represent a wide variety of acoustic speech events with well-localized energy regions to model e.g. formant tracks or energy bursts due to plosives, and in a later stage (ii) that they can be robustly detected in unseen noisy speech. To this end, we transform the speech data into a time–frequency reassignment spectrogram (Auger and Flandrin, 1995) which is subsequently smoothed in time and frequency domain. The reassignment method (briefly restated in Section 2.1) produces sharpened time and frequency estimates for each spectral component from partial derivatives of the short-time phase spectrum.

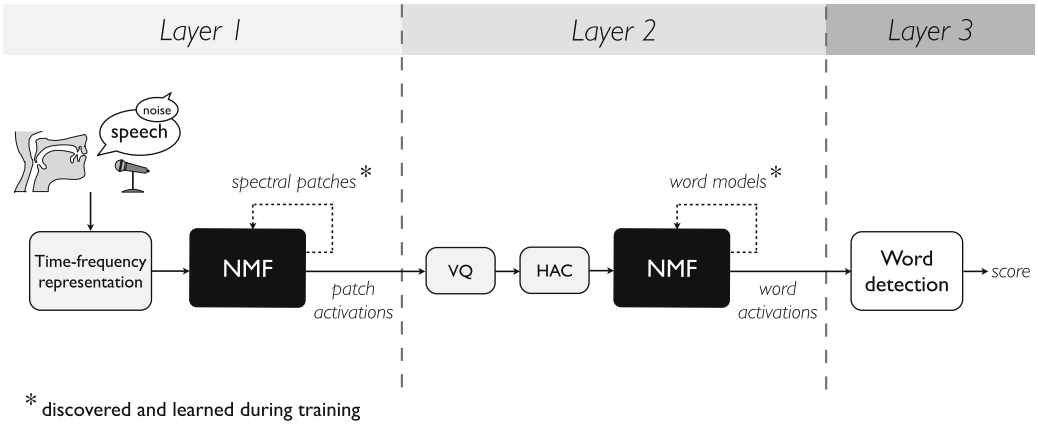

Fig. 1. Structural representation of the proposed learning system.

Instead of locating the spectral density value at the geometrical center of the analysis window, as in traditional short-time spectral analysis (e.g. STFT), the components are reassigned to the center of gravity of the energy distribution. In a next step, we perform a non-negative matrix factorization (NMF) on a data matrix containing consecutive spectral slices of this speech representation. By imposing non-negativity constraints, NMF generates parts that are additive, unlike factorization techniques such as PCA or SVD. Furthermore, NMF was chosen among other unsupervised learning method such as hierarchical clustering, self-organizing maps or neural networks, since (i) it is a recent and promising technique that has shown it merits in other research questions; (ii) it provides a more stable, intuitive and meaningful decomposition of non-negative data. By combining two complementary time–frequency reassignment representations that either reveal timing or frequency structure, the discovered speech patterns are acoustic patches of correlated energy that are well-localized in both time and frequency. The involved smoothing process allows to make these patches speaker-independent. Besides the fact that the reassignment method produces highly detailed patterns, another motivation to use RTFR is the impact it has on the higher level of word learning (Section 3). This will be illustrated in the experiments of Section 5 by comparing the approach of RTFR with the approach where the patches are derived from the short time Fourier transform (STFT) of the speech.

2.1. Time–frequency reassignment

Time–frequency reassignment (Auger and Flandrin, 1995; Plante et al., 1998; Hainsworth and Macleod, 2003) allows perfect localization of (well-separated) impulses, cosines and chirps, which constitute a reasonable model for speech. The corresponding reassigned time–frequency representation (RTFR) has an increased sharpness of localization of the signal components without sacrificing the frequency resolution.

In this paper, the reassignment principle is applied to the short time Fourier transform (STFT) although it can be applied to different time–frequency representations each characterized by a different analysis kernel. The STFT is often used as the basis for a time–frequency representation of speech signals and is written as

$$\mathrm{STFT}\{x(t)\} = \int_{-\infty}^{+\infty} x(u)h^*(t-u)\mathrm{e}^{-j\omega u}\mathrm{d}u \tag{1}$$

where $x(t)$ is the analyzed signal and $h(t)$ is the analysis kernel function. The spectrogram is then defined as the magnitude of the STFT and can also be expressed as a two-dimensional smoothing of the Wigner–Ville distribution (Auger and Flandrin, 1995)

$$|\mathrm{STFT}\{x(t)\}|^2 = \frac{1}{2\pi}\int\int_{-\infty}^{+\infty} W_x(u,v)W_h(t-u,\omega-v)\mathrm{d}u\,\mathrm{d}v \tag{2}$$

with

$$W_x(t,f) = \int_{-\infty}^{+\infty} x\left(t+\frac{1}{\tau}\right)x^*\left(t-\frac{1}{\tau}\right)\mathrm{e}^{-j\omega\tau}\mathrm{d}\tau. \tag{3}$$

From expression (2) it can be seen that the spectral density value of each time–frequency component is the weighted sum of all the Wigner–Ville distribution values at the points $(t-u,\omega-v)$ and thus located at the geometrical center (t,ω) of the spectral analysis kernel function. The principle of the reassignment method is then to reallocated the energy from the geometric center of the kernel function to the center of gravity of the energy distribution. Therefore, the RTFR takes into account the phase of the STFT, which is omitted in the classical spectrogram, but contains important temporal information and this results in an improved localization of the energy in the time–frequency plane.

The reassignment points can be computed from the partial derivatives of the phase of the STFT using the principle of stationary phase (Kodera et al., 1978). According to this principle, the maximal contribution to the values of (2) occurs at the points where the phase is changing most slowly with respect to time and frequency. If $\phi(t,\omega)$ denotes the short-time phase spectrum, then these points are computed as (Plante et al., 1998)

$$(\hat{t},\hat{\omega}) = \left(t - \frac{1}{2\pi}\frac{\partial}{\partial\omega}\phi(t,\omega), \omega + \frac{1}{2\pi}\frac{\partial}{\partial t}\phi(t,\omega)\right) \tag{4}$$

which represents the group delay and instantaneous frequency of the windowed signal. It has been shown in (Auger and Flandrin, 1995), that a more efficient implementation is possible using two additional STFTs rather than using the derivatives of the phase. Let $H(t,\omega)$, $D(t,\omega)$ and $T(t,\omega)$ denote the STFT of the signal obtained with the window of choice $h(t)$, the derivative of $h(t)$ and the time weighted $th(t)$ respectively and let $\Re(X)$ and $\Im(X)$ be the real and imaginary part of X, then the energy at (t,ω) is reassigned to the center of gravity (Auger and Flandrin, 1995):

$$(\hat{t},\hat{\omega}) = \left(t - \Re\left[\frac{T(t,\omega)}{H(t,\omega)}\right], \omega + \Im\left[\frac{D(t,\omega)}{H(t,\omega)}\right]\right) \tag{5}$$

where the time and frequency offsets are now computed from the ratios of the three STFTs.

To improve the visibility of acoustic events with short duration, we further enhance the localization of the energy along the time axis. Therefore, we first search for zero-crossing points in the time offsets of (5) and only those of them that are connected in the vertical direction (i.e. along the frequency axis) are retained. Finally, the corresponding energy of the RTFR is assigned to the retained points. When applied to speech with a sufficiently short analysis window, the enhanced RTFR clearly shows the vertical (i.e. well-localized in time) lines that are related to the burst of plosives and affricatives and energy releases by the vocal folds. By repeating the same procedure using the frequency offsets of (5), the tracks of time-varying spectral features

such as pitch and formants can be clearly localized in frequency. We have found that formant structure is more apparent if we use shorter windows in the RTFR.

The different steps of the enhanced reassignment procedure are shown Fig. 2 for the word "two". Firstly, a time–frequency representation is computed using a 128-point STFT. Subsequently, a RTFR is produced by reallocating the spectral energy to the gravity centers according to (5). The above mentioned enhancement steps are then applied to the RTFR to reveal either the timing or the frequency structure. Experiments have shown that an optimal choice for the window length is respectively 11 ms and 7 ms for male speakers and 6 ms and 4 ms for female speakers. The analysis window is shifted by 1 ms. To prevent ambiguity in later formulations, we will use the word *subframe* to denote a frame of the enhanced RTFR.

2.2. Constructing the input matrix

In this section, we explain how the input matrix is created to which we will apply NMF for finding acoustic time–frequency patterns in speech signals. These patterns are discovered on clean training data. After pre-emphasizing the speech signals, we compute the enhanced RTFRs by the approach described in Section 2.1. Both representations are used to exploit the spectral information that is more apparent in either the vertical or the horizontal direction.

Two additional steps are also performed; a smoothing in time and frequency followed by a cube root compression. If these steps are not be applied, speaker dependent "bases" will be learned to model the different pitch characteristics of training speakers. For reasons that become clear in the next section, however, we want to prevent overfitting to the training set, e.g. we want the resulting time–frequency patches not to be speaker dependent. Time smoothing is performed by reframing the enhanced RTFR by a sliding triangular window with a length of 30 subframes and a frameshift of 10 subframes. After conversion of the frequency axis from the Hertz scale to the Mel scale followed by a frequency smoothing using $N = 128$ triangular overlapping windows with a window size of 3 frequency bins using a weight of 1 for the center bin and 0.5 for the adjacent bins, we obtain the final N-dimensional feature vectors. Subsequently, spectral changes are emphasized by adding first and second order derivatives resulting in a static (S), a velocity (V) and an acceleration (A) stream. For these feature representations we use the word *frames* and these are shown in Fig. 3a for the same uttered "two" as was used in Fig. 2. Note that the vertical lines corresponding to pitch bursts are dissolved by the smoothing process, but the overall energy bursts and releases are retained. In Fig. 3b we also show the feature vectors derived by applying the STFT using the same time window parameters as in the enhanced and smoothed RTFR. Note that, despite the smoothing process, formant contours still remain clear in Fig. 3a and are not confused with pitch harmonics as is the case in conventional Fourier transform. In Section 5, we will compare the final accuracy results obtained by the STFT features with those of the enhanced and smoothed RTFR features.

The feature representations corresponding to the timing and frequency structure contain complementary information and therefore both will be used in the discovery of the speech patches. Let us now define the spectral vector at a certain frame t for a feature stream ρ ($\rho = S$, V or

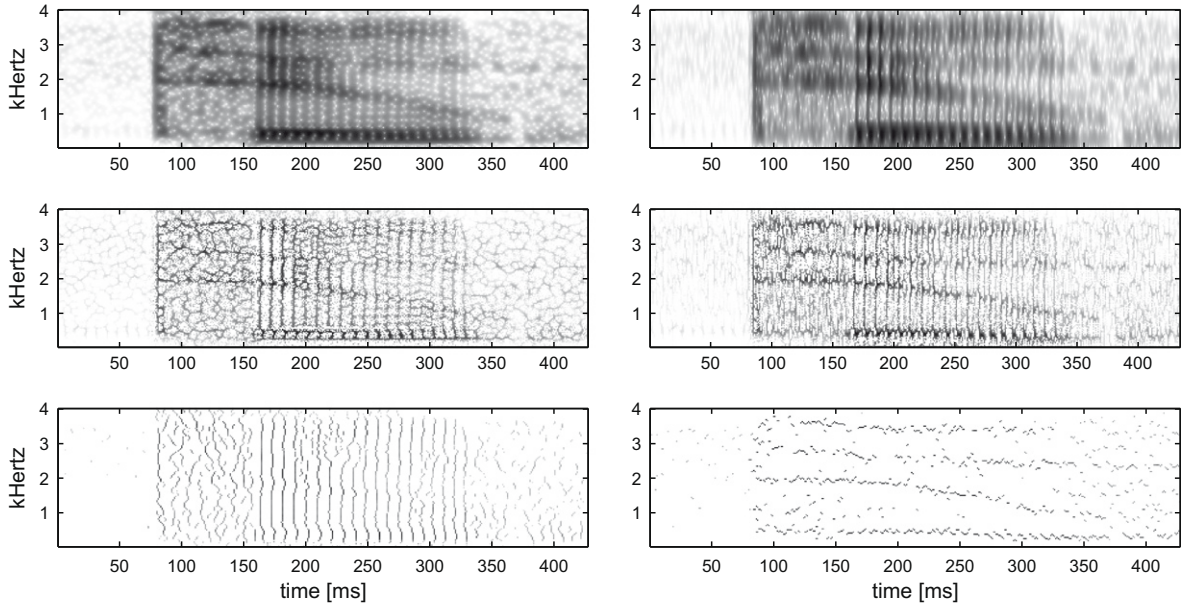

Fig. 2. STFT representation (top), RTFR (middle) and enhanced RTFR (bottom) for the word "two" with focus on timing (left) and frequency (right) structure. In the bottom left panel, vertical lines correspond to plosive burst and vocal fold bursts, while the enhanced RTFR at the bottom right reveals the horizontal structure in the word, e.g. pitch and formant contours.

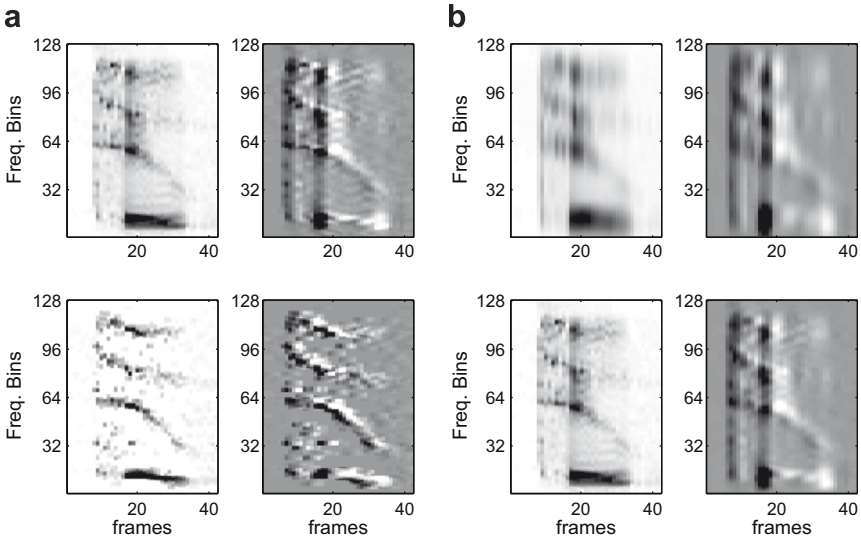

Fig. 3. (a) Static and velocity feature vectors derived from the enhanced RTFRs that reveal timing (top) and frequency (bottom) structure for the word "two". (b) The corresponding STFT feature vectors using the same time window parameters.

A) as $\boldsymbol{v}_{\rho,t}$ and $\boldsymbol{h}_{\rho,t}$, corresponding to the feature representation that reveals respectively the timing (vertical) and frequency (horizontal) structure. Since NMF requires the data to be comprised of non-negative values only, we split these vectors into a positive ($\boldsymbol{v}^+$ and $\boldsymbol{h}^+$) and a negative ($\boldsymbol{v}^-$ and $\boldsymbol{h}^-$) stream by zeroing out those values that are respectively <0 and >0 and taking absolute value of the negative stream. Finally, we stack all these vectors in one real and non-negative column vector of dimension $4N$:

$$\boldsymbol{c}_{\rho,t} = \begin{bmatrix} \boldsymbol{v}^+_{\rho,t} \\ |\,\boldsymbol{v}^-_{\rho,t}\,| \\ \boldsymbol{h}^+_{\rho,t} \\ |\,\boldsymbol{h}^-_{\rho,t}\,| \end{bmatrix} \tag{6}$$

For static features, $\boldsymbol{v}^-_{\rho,t}$ and $\boldsymbol{h}^-_{\rho,t}$ are all-zero vectors and their rows can be removed from (6). Note that the above mentioned procedure to handle input data with mixed sign in NMF, can also be seen as an alternative for the semi-NMF as was proposed by Ding et al. (2006).

At each frame step t, we take k consecutive frames of $\boldsymbol{c}_{\rho,t}$ representing the spectro-temporal structure of a short-time speech segment (i.e. of length $10k$ ms). These k frames are then reshaped into a column vector $\boldsymbol{C}_t^{k,\rho}$ of dimension $4kN$ as schematically illustrated in Fig. 4. From these column vectors, we construct a data matrix:

$$C^k = \begin{bmatrix} C_1^{k,S} & \cdots & C_t^{k,S} & \cdots & C_T^{k,S} \\ C_1^{k,V} & \cdots & C_t^{k,V} & \cdots & C_T^{k,V} \\ C_1^{k,A} & \cdots & C_t^{k,A} & \cdots & C_T^{k,A} \end{bmatrix} \tag{7}$$

with T the total number of frames used from the clean training set.

2.3. Matrix factorization for unsupervised learning

By applying non-negative matrix factorization to the matrix C (dropping index k for notational convenience), it is approximated by the product of factors B and A which are of size $4kN \times P$ and $P \times T$:

$$C \approx BA \tag{8}$$

subject to the constraint that all matrices are non-negative and where the common dimension P of B and A is much smaller than T and $4kN$. Hence, Eq. (8) contains only additive linear combinations such that the factorization leads to a parts-based representation, where P parts are found in the columns of B and their activation across time are given by the corresponding rows of A.

In order to capture all feature streams in each basic vector, namely the timing and frequency spectral structure and their corresponding positive and negative part, additional sparsity constraints must be enforced on A. Otherwise, NMF tends to model all these parts in multiple columns of B. Therefore, we use sparse NMF (Hoyer, 2004) where the factorization is approximated by minimizing the objective function:

$$G(C\|BA,\lambda) = D(C\|BA) + \lambda \sum_{i,j} A_{ij} \tag{9}$$

The first term in (9) is a generalized version of the Kullback–Leibler divergence (Lee and Seung, 2001), defined as:

$$D(C\|BA) = \sum_{ij} \left(C_{ij} \log \frac{C_{ij}}{(BA)_{ij}} - C_{ij} + (BA)_{ij} \right) \tag{10}$$

The second term in Eq. (9) enforces sparsity on A by minimizing the L_1-norm of its columns. The trade off between reconstruction accuracy and sparseness is controlled by the parameter λ.

An algorithm for finding B and A given C based on multiplicative updates and with the additional sparseness constraint can be found in (O'Grady and Pearlmutter, 2008). To address scaling, the constraint that each column of B sums to unity is imposed. Experiments have shown that

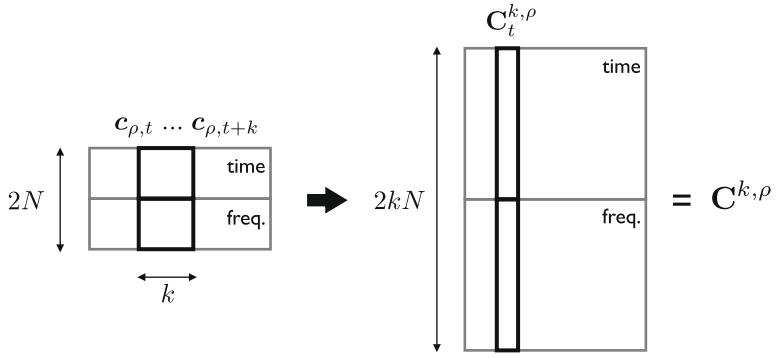

Fig. 4. Schematic representation of the construction of data matrix $C^{k,\rho}$ from feature vectors $c_{\rho,t}$ which contain the feature representations that contain timing ($v_{\rho,t}$) and frequency ($h_{\rho,t}$) structure.

with the settings used in this paper, a good choice for λ is 1000.

2.4. Interpreting the time–frequency patches and their activation in time

The columns of matrix B correspond to spectral patches which describe the recurrent time-varying spectra of speech. A selection of these patches are shown in Figs. 5 and 6. Just for visualization, the rows of each feature stream were extracted from B and were reshaped back into $N \times k$ matrices, then the positive and negative parts were recombined and the feature representations corresponding to the timing and frequency structure were plotted onto each other by means of the max-operator. These figures illustrate the patches found for the static and velocity features. The parameter k of Section 2.2 is set to 10 (Fig. 5) and 20 (Fig. 6) corresponding to a patch length of resp. 100 and 200 ms. Most patches describe formant movements over the duration of about a phone. A smaller set of time–frequency patches resemble wideband sounds and short-time energy bursts. Others are modeling the beginning or ending of phones and phone-pair transitions. Since we have discovered the acoustic patterns from recordings composed of a sufficiently large set of different speakers, the patches are assumed to be speaker-independent (will be confirmed in Section 5).

To discover the patches that are present in test utterances of the Aurora2 database, the same procedure as in training is used except that we compute A in Eq. (8) by holding B fixed to the one obtained from training. As an example, Fig. 7 shows the time–frequency representations (a) and their corresponding patch activation matrices (b) of three examples of the word "four" each uttered by different male speakers in clean conditions (left column) and noisy conditions (right column). A set of $D = 100$ time–frequency patches were discovered from the training set as described above with a patch length of 100 ms. As can be seen from the clean speech examples in Fig. 7b (left pane), only few patches are highly active (black) at a certain time and sparse patterns can be discovered in the activation data. Despite variations in speaking style and speaker characteristics, the figure also suggest that each word corresponds to similar, speaker-independent activation patterns and that different words can be discriminated by comparing these patterns.

For the noisy speech examples, the babble noise type of the aurora2 database were added at different levels of

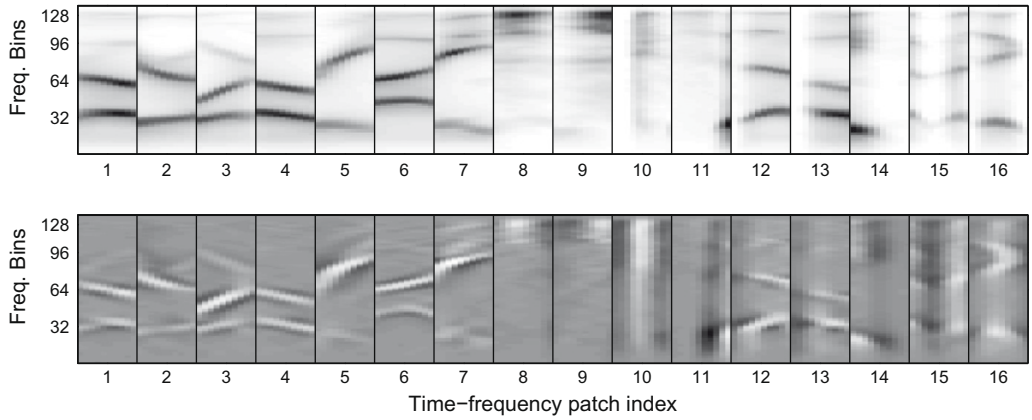

Fig. 5. A collection of the discovered time–frequency patches for static (top row) and velocity features (bottom row) with a duration of 100 ms. Some patches show formant patterns, wideband spectra and bursts; others model inter-phone or silence-phone transitions.

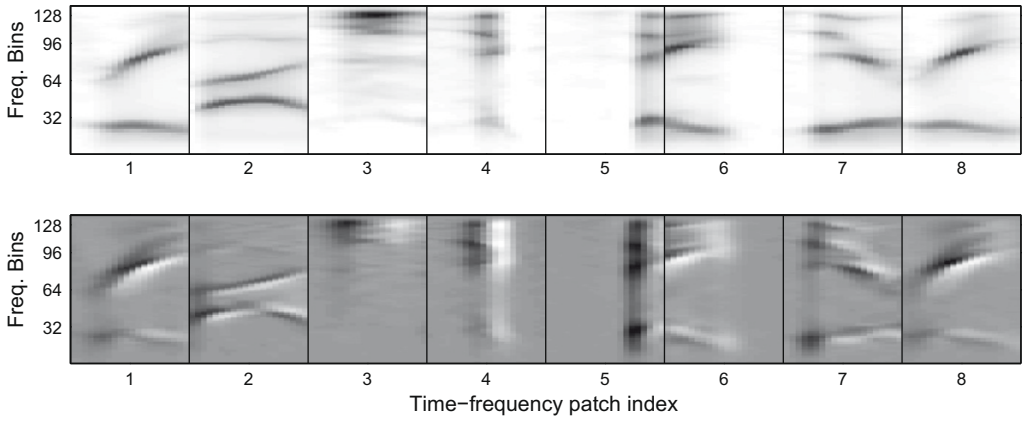

Fig. 6. A collection of the discovered time–frequency patches with a duration of 200 ms.

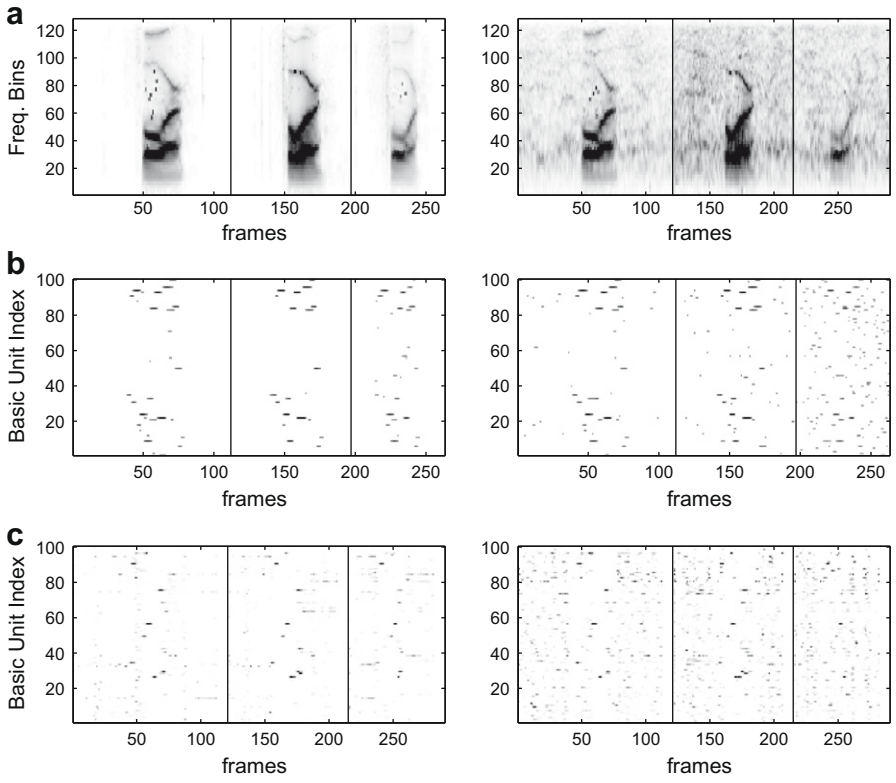

Fig. 7. (a) Time–frequency representations of the word "four" uttered by three different speakers in clean conditions (left) and noisy conditions (right) and their corresponding patch activation matrix A for patches derived from the procedure with (b) NMF (Section 2.3) and (c) cNMF (Section 2.5).

signal-to-noise ratio (SNR) to the clean word utterances of Fig. 7a, namely at 15, 10 and 5 dB SNR. In these noisy conditions, the activation patterns of the words remain similar and are hardly distorted by the different noise types. The noise robustness properties of the proposed speech model will be investigated in more detail later on (Section 5).

Finally, Fig. 8 displays ten time–frequency patches for the word "four". The selected patches have a patch length of 100 ms and are those with the highest activation values in the utterance. The patches are ordered chronologically, e.g. patch 1 is activated from frames 31 to 33, patch 10 from frames 61 to 65.

2.5. Comparison with convolutive NMF

Alternatively, convolutive NMF (cNMF) could be used to obtain a parts-based representation of the data (O'Grady and Pearlmutter, 2008). Therefore, cNMF can be applied onto the sequence of feature vectors $c_{\rho,t}$ of Eq. (6). However, experiments show two major drawbacks in disfavor of cNMF. Firstly, the computational requirements for

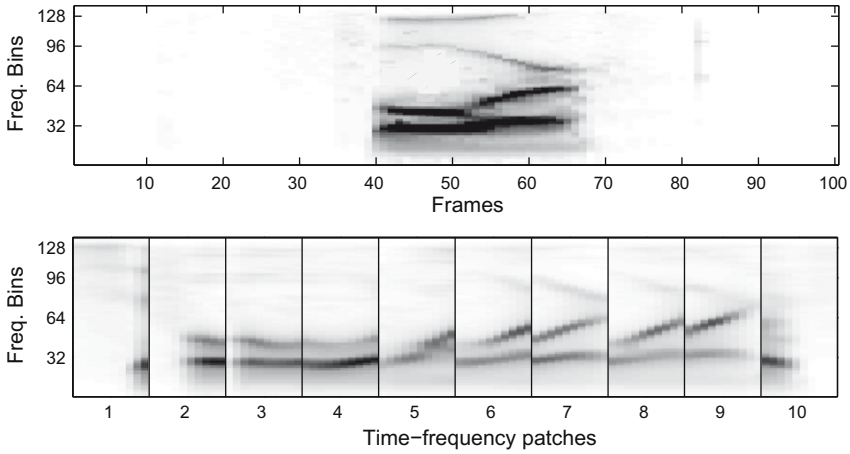

Fig. 8. Time–frequency representation of the word "four" (top) and the time–frequency patches (bottom) with highest activation values in the utterance.

cNMF are higher than those of the NMF procedure as described in Section 2.3. The training of the time–frequency patches with cNMF involves more computational time than NMF for the same number of iteration steps. During testing, the processing time spent per iteration is similar for both factorization techniques, but cNMF requires more iterations to convergence. Secondly, although cNMF also produces speaker-independent activation patterns, it turned out that these patterns are less robust to additional noise sources. From Fig. 7c it can be seen that the activations patterns of cNMF are more distorted in noisy speech than the patterns of NMF. These observations will be confirmed in Section 5 by discussing and comparing the final results.

3. Layer 2: Acquiring activation patterns of time–frequency patches

Our bottom-up approach for language learning is driven by the assumption that a particular language is characterized by similarities in the activation patterns of time–frequency patches, as was illustrated by Fig. 7. If we assume that the patches correspond with (groups of) auditory neurons, each sensitive to a specific time–frequency pattern, then a "snapshot" of their firing rate at a certain time is represented by a column of A. Hence, if recurring acoustic patterns in speech correspond to recurring neural firing patterns, one can hypothesize that the meaningful objects in a language (e.g. words) are characterized by similar activation patterns of time–frequency patches.

In this section, a learning algorithm is proposed that is able to acquire the objects of a language. The objects that are found are words, but could also be phone-like units in a different setting. The algorithm will discover the latent structures in patch activations by using "histograms of acoustic co-occurrence" (HAC) which are described by Van hamme (2008a) and will be restated next. HAC-features can represent a given segment of speech in a unique high-dimensional vector without requiring a segmentation of the segment, time warping or a constraint of its duration. Moreover, each word in the utterance contributes additively to the HAC of the utterance which motivates to apply the HAC representation in association with NMF. Word identities are being provided to the algorithm to bring the discovered activation patterns in relation with the words in the vocabulary.

3.1. Histograms of acoustic co-occurrences

HAC-models can describe speech by the co-occurrence statistics of acoustic events. In this work, these models are used to recover recurring patterns in patch activation events which are here the occurrence of quantized vectors in the activation matrix A.

The activation data of the time–frequency patches will be characterized by its similarity to examples. Therefore, the columns of the activation matrix A are clustered into Σ centroids using the K-means algorithm. Given the Euclidean distance metric used in clustering, each centroid can be represented by a Gaussian with spherical covariance. As a consequence, the posterior probabilities $P_{i,n}$ of all centroids n characterize any frame i of A in terms of its similarity to each of the centroids. For each frame i, the posterior probabilities satisfy:

$$\sum_{n=1}^{\Sigma} P_{i,n} = 1 \tag{11}$$

A special case is obtained in a "winner takes all" setting, where all posteriors are zero except for the centroid closest to the observation, which is assigned the value 1. This setting is related to a vector quantization (VQ) approach where the centroids are the codebook entries labeled from 1 to Σ. After decoding, each frame of the activation matrix A is then replaced by the best matching centroid of the codebook, which allows to reduce the activation matrix to a single row vector of VQ-labels.

The HAC-representation is then the number of times all VQ-label pairs $(m, n) \in \Sigma \times \Sigma$ are observed τ frames apart.

In other words, a histogram of lag-τ co-occurrences is constructed where each co-occurrence signifies that the input of activation frames is encoded into a VQ-label m at time i, while encoded into VQ-label n at time $i+\tau$. For a given utterance u, the lag-τ co-occurrence is weighted with the (approximated) probability of the event

$$\left[V_u^{\tau}\right]_{mn} = \sum_{i=1}^{I_u-\tau} P_{i,m}P_{i+\tau,n} \quad \text{with } m,n = 1,\ldots,\Sigma \tag{12}$$

where I_u is the number of frames in the utterance. Also note that $[V_u^{\tau}]_{mn} \neq [V_u^{\tau}]_{nm}$, such that these co-occurrences are directed.

By stacking all (m,n)-combinations, each utterance can be represented by a single column vector V_u^{τ} where the elements express the sum of all Σ^2 possible lag-τ co-occurrences. We will refer to this vector as a *HAC* (histogram of acoustic co-occurrence).

This procedure can be performed for different τ-values and for a given set of time–frequency patches with a patch length of k frames. For a set of U utterances, the data matrix for a choice of k and τ is then formed by

$$V^{k,\tau} = \left[\, V_1^{k,\tau} \quad \cdots \quad V_u^{k,\tau} \quad \cdots \quad V_U^{k,\tau} \,\right] \tag{13}$$

Note that thanks to the vector quantization approach, matrix $V^{k,\tau}$ has a high sparsity. Furthermore, all its entries are non-negative such that NMF-methods can be applied.

3.2. Semi-supervised learning with NMF

Suppose that the utterances are composed of R recurring acoustic events such as words, each constructed from the set of time–frequency patches. Since (13) is a sum over time of activations, the words will contribute additively to the corresponding column of $V^{k,\tau}$. As each word is characterized by a HAC, the HAC of each utterance will be a (integer) linear combination of these histograms.

If the HAC of the words are placed in separate columns of a matrix W, and if the corresponding rows of H would contain the presence of each word in each utterance, one would have (leaving out indices k and τ):

$$V \approx WH \tag{14}$$

Given their interpretation, all entries of W and H are constrained to be positive or zero. Because of these constraints and given the fact that Eq. (14) will not hold exactly since the observed symbols are subject to variability and uncertainty, W and H are estimated by NMF. Factorization of V is performed using the approach of Section 2.3 without enforcing sparsity constraints, e.g. we set λ to 0.

Once W is estimated on a training set, new utterances can be analyzed with factorization (14) by estimating H. The degree to which each discovered activation pattern is present in each new utterance is then found by examining the columns of H.

In this work, the words are unknown and NMF is used to separate them out from the utterances. However, since utterances can be seen as a sequence of words, but also as, for instance, a sequence of phones, constraints have to be imposed on (14) by exploiting grounding information. If it is known which words occur in each utterance, this information can be exploited to associate a word identity to each column of W. Therefore, the $L \times U$ grounding matrix G is formed, which holds in its lth row and uth column the number of times the lth word occurs in the uth utterance. Here, L is the number of word identities and U is the number of training utterances. Subsequently, we compute:

$$\begin{bmatrix} G \\ V \end{bmatrix} = \begin{bmatrix} W_g \\ W_v \end{bmatrix} H \tag{15}$$

which expresses that word identities need to be explained jointly with the acoustic data by common model activations H. Given the properties of multiplicative updates (Lee and Seung, 2001), grounding forces the NMF decomposition to associate word models in W_v also to the utterances containing those words. Without augmenting V with the grounding matrix, NMF tends to spend columns of W_v preferably on the more frequent acoustic patterns since this has the most impact on minimizing the modeling error. Experiments have shown that the common dimension R is better overestimated, hence $R \geqslant L$. This allows to model acoustic events that have no relevance to grounding such as silence or filler words.

3.3. Improving learning by modeling multiple streams

As explained in Section 3.1, the data matrix $V^{k,\tau}$ contains lag-τ co-occurrences in the activation data of time–frequency patches with a duration of k frames. For each individual configuration (k,τ), patterns can be learned using the approach of Section 3.2 by connecting acoustic information and by assigning a meaning to these patterns. We may assume that the performance of the learning algorithm will increase if multiple configurations are combined in the input matrix. This idea has already been exploited by jointly capturing static, velocity and acceleration feature information in each time–frequency patch. On the activation level, the data matrix of (14) can be further extended by incorporating different sets of patches and including co-occurrence data at different time lags. By allowing the use of patches with different durations, we could also compensate for the time differences of phones. For instance, one can expect that plosives cause more neural activation at neurons modeling time–frequency patches with a duration around 50 ms, while neurons corresponding to patches of 100 ms better represent diphthongs and vowels. Units with even longer duration (e.g. 200 ms) can be used to model intra- and inter-phoneme transitions. Therefore P sets of time–frequency patches are included in the model, each exploiting Q values of τ, augmented with grounding information that relates to the spoken words:

$$\begin{bmatrix} G \\ V^{k_1,\theta} \\ \vdots \\ V^{k_P,\theta} \end{bmatrix} = \begin{bmatrix} W_g \\ W_v \end{bmatrix} H \quad \text{with} \quad V^{k_i,\theta} = \begin{bmatrix} V^{k_i,\tau_1} \\ \vdots \\ V^{k_i,\tau_Q} \end{bmatrix} \tag{16}$$

For these joint streams, the generative parts-based model still holds: the joint stream co-occurrences of utterances can be written as an additive combination of parts. As we will show in the experiments of Section 5, it is indeed advantageous to exploit multiple combination of k and τ in the HAC-model.

4. Layer 3: Detecting words in activation patterns

After the semi-supervised training procedure, W_g and W_v is known. Recognition on unseen test utterances (from which grounding information G is unknown), is achieved by first computing the histograms of co-occurrence V and then estimating the matrix factor $\widehat{H}$ in $V \approx W_v \widehat{H}$ by holding W_v fixed. This matrix $\widehat{H}$ reveals to which extent the internal representations of the trained words are present in the new test utterance. By estimating the grounding information as:

$$\widehat{G} = W_g \widehat{H} \tag{17}$$

we obtain estimates for the presence of the words in the test utterances. Hence, for a word that is present, the corresponding element of $\widehat{G}$ tends to 1 and to 0 if it is absent. This way, a word detection system can be build from the content of matrix $\widehat{G}$ by comparing $\widehat{G}$ with a threshold ξ. Two types of errors are involved in the system: missed detections ($\widehat{G}_{ij} < \xi$ while utterance j contains word i) and false alarms ($\widehat{G}_{ij} \geqslant \xi$ while utterance j does not contain word i). The trade-off of both error types can be visualized by means of a Detection Error Trade-off (DET) curve (Martin et al., 1997). In Fig. 9, the DET-curve is shown for the word detection task where the model is trained on lag-10 co-occurrence counts ($\tau = 10$) computed on a set of time–frequency patches with a length of 100 ms ($k = 10$). The performance of the system in clean speech conditions is compared with a noisy test case where the speech is distorted by babble noise at 10 dB SNR. The estimated grounding matrix $\widehat{G}$ of five different utterances for both test cases are shown in Fig. 10 where high values (black) indicate that the corresponding words have a high probability to be present in the utterance.

In the experimental evaluation of Section 5, we will not apply this "per word detection" paradigm, which is relevant for tasks such as keyword spotting. Instead we will measure correct word recognition per utterance. Assuming that the number D_u of different digits occurring in the uth test utterance is given, the D_u candidates with highest activation according to Eq. (17) are selected. Notice that the recognition result is unordered, a problem that is addressed in (Van hamme, 2008a) by a sliding window decoder that estimates at which time each word occurs in the utterance.

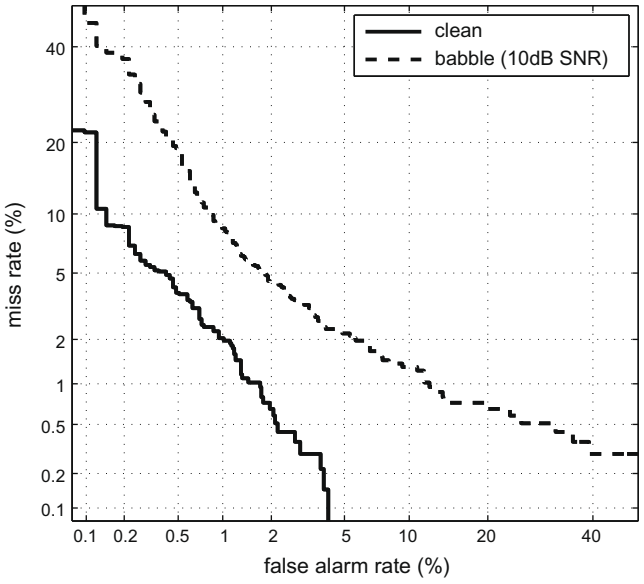

Fig. 9. DET-curves of the word detection system for clean and noisy speech (babble noise at 10 db SNR).

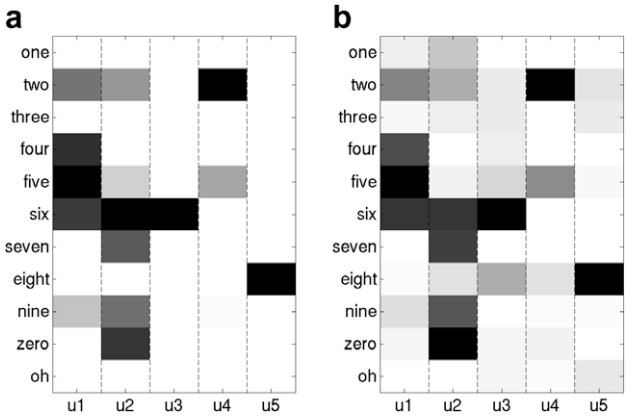

Fig. 10. Estimated grounding matrix $\widehat{G}$ for (a) clean and (b) noisy speech (babble noise at 10 db SNR) of the utterances "four six two five" (u1), "zero seven nine six two" (u2), "six" (u3), "two five two" (u4) and "eight" (u5).

Word error rate is thus defined as the sum of the number of incorrect digits that end up in the top D_u, divided by the sum of D_u over the complete test set.

5. Experiments

The speech data are taken from Aurora2, a small vocabulary, speaker-independent database for connected digit recognition as defined by Hirsch and Pearce (2000). All utterances are derived from the TI-Digits database that contains recordings of male and female US-American adults, downsampled to 8 kHz sampling frequency. The database contains isolated digits and sequences of up to seven digits. The Aurora2 clean-condition training set consists of 8440 utterances that are used for the discovering the time–frequency patches and their activation patterns from

which the HAC-models are derived. The test experiments are conducted on test set A consisting of 4004 utterances from the TI-Digits test data, split into 4 subsets of 1001 utterances each and to which 4 different noises are added at different SNR-levels: noise recorded in a subway (N1), babble noise (N2), car noise (N3) and noise recorded in an exhibition hall (N4).

5.1. Reference experiment

Baseline recognition results are produced by a conventional HMM-based recognition system using the complex back-end configuration as defined by the ETSI Aurora group (ETSI Standard Document, 2000b). Whole word digit models were trained on the clean speech training database using the HTK software package version 2.2 from Entropic (Young et al., 1999). The digit models have 16 emitting states with 20 Gaussians per state. The optional inter-word silence is modeled by 1 or 3 states with 36 Gaussians per state, while leading and trailing silence have 3 states. The total number of Gaussians is 3628. Feature were extracted by the Aurora WI007 front-end (ETSI Standard Document, 2000a), a cepstral analysis scheme where 12 Mel-scaled cepstral coefficients and c_0 (no log-energy) are determined for a speech frame of 25 ms length using a frame shift of 10 ms. These features are combined with their dynamic coefficients to yield in 39-dimensional feature vectors for recognition, as explained in (Macho et al., 2002).

Here, we assume that for each test utterance u the number N_u of digit sequences is known. This information is then used in the language model by forcing the decoder to recognize exactly N_u digits. From the recognition result, only the different digits are retained to obtain an unordered string result of at most N_u digits. Similarly as in Section 4, a detection error is accounted for each digit from the set of D_u different and correct digits in the utterance that is not present in the unordered recognition result. The error rate of this HMM-based word detection system is shown in Table 1. Results were averaged over the four noise types of Aurora2.

5.2. Training procedure

In the experiments, multiple sets of time–frequency patches, modeling acoustic patterns of different durations, were trained on the clean training set of Aurora2. The patches are learned from a data matrix constructed from static, velocity and acceleration features as explained in Section 2, while using the following values for k: 5, 10, 15, 20. For each set, the number of patches P to be discovered is 100. Experiments, not reported in this paper, have shown that this number suffices to model the different spectral patterns of the small vocabulary task of Aurora2. The obtained patches are modeled by the columns of the four matrices B^k which are stored for the recognition task on test data.

For each set of time–frequency patches, the patch activation vectors in A^k are quantized using a codebook of 250 entries, resulting in 4 sequences of VQ-labels. Subsequently, the VQ-label co-occurrence histograms are computed for all utterances using different lag-τ values: 5, 10, 15, 20. The VQ histogram counts are divided by a fixed constant (100) such that the acoustic and grounding information have roughly the same weight in the data matrix $V^{k,\tau}$. Experiments have shown that the value of this constant is not critical: it can be changed over several orders of magnitude without significant impact. To acquire all eleven words of Aurora2, namely the digits "one" to "nine", "zero" and "oh", the training procedure as described in Section 3 was performed with $R = 12$ using the utterances of the clean training set. After factorization (15), W_g and W_v are stored for recognition.

Table 1
Unordered word error rate of an HMM-based word detection system on the Aurora2 database averaged over the four noise types.

Clean	15 dB	10 dB	5 dB
0.16	2.98	11.92	34.88

5.3. Evaluating the results

To discover the digits that are present in the test utterances, the same procedure as in training is used except that we compute the patch activation matrix A in (8) by holding B fixed to the one obtained from training. Similarly, the word activation matrix H is found by holding W fixed in (14) to the one estimated from the training set.

Table 2 shows the unordered word error rate on the Aurora2 test set averaged over the four noise types, using different stream configurations (k, τ). For clean speech, the self-learning algorithm performs worse than the HMM-based system that makes use of expert speech knowledge that arises from audiology and linguistics. However, our system performs comparably to the HMM-based recognizer at 15 dB SNR and has a remarkably higher accuracy for noisy speech at lower SNRs without using any noise compensation techniques. From the Table 2, we can also observe that the robustness can be increased by exploiting more knowledge sources. The reason for this noise robustness is three-fold: (i) thanks to the parts-based representation of speech, the system easily detects, even in noisy conditions, which time–frequency patches are active; (ii) these time–frequency patches provide static and dynamic spectral information over large time windows; (iii) multi-window time–frequency representations can be exploited by the joint modeling of different streams.

For comparison, we added the results of the STFT features with the same time windows parameters for the timing and frequency structure (see Table 3) and those where cNMF are performed onto the feature vectors of Eq. (6) (see Table 4). As can be seen from both tables, the word error rates are worse than those shown in Table 2.

Table 2
Unordered word error rate results on the Aurora2 database averaged over the four noise types for the proposed recognition system using a combination of different sets of time–frequency patches and different lag-τ co-occurrence counts. The time–frequency patches are derived from the enhanced and smoothed RTFR features using the NMF procedure described in Section 2.3. The $\times$-symbol indicates which configurations (k, τ) are integrated in the input matrix of Eq. (16).

k				τ				Clean	15 dB	10 dB	5 dB
5	10	15	20	5	10	15	20				
	×			×				2.10	4.29	7.07	12.54
	×				×			1.98	3.74	6.37	11.65
	×					×		2.22	4.02	6.78	11.58
	×						×	2.67	4.90	7.33	12.64
	×				×		×	2.17	3.80	5.79	10.34
	×			×	×	×		1.89	3.58	5.86	10.83
	×				×	×	×	1.96	3.74	5.82	10.86
	×			×	×	×	×	1.93	3.56	5.64	10.67
	×		×	×	×	×	×	1.94	3.41	5.39	9.35
×	×	×		×	×	×	×	**1.83**	3.26	5.24	9.35
	×	×	×	×	×	×	×	2.01	3.36	5.12	8.93
×	×	×	×	×	×	×	×	1.87	**3.08**	**4.94**	**8.67**

Table 3
Unordered word error rate results for Aurora2 averaged over the four noise types using time–frequency patches derived from STFT features by the NMF procedure.

k				τ				Clean	15 dB	10 dB	5 dB
5	10	15	20	5	10	15	20				
	×		×	×	×	×	×	2.06	4.27	6.48	11.62
×	×	×		×	×	×	×	**1.81**	4.23	6.78	12.08
	×	×	×	×	×	×	×	2.06	4.20	**6.40**	**11.06**
×	×	×	×	×	×	×	×	1.91	**4.12**	6.43	11.56

Table 4
Unordered word error rate results for Aurora2 averaged over the four noise types using time–frequency patches derived from the enhanced and smoothed RTFR features by the cNMF procedure discussed in Section 2.5.

k				τ				Clean	15 dB	10 dB	5 dB
5	10	15	20	5	10	15	20				
	×		×	×	×	×	×	3.45	6.57	10.23	18.16
×	×	×		×	×	×	×	**2.76**	5.83	9.48	16.36
	×	×	×	×	×	×	×	3.18	5.90	9.44	16.49
×	×	×	×	×	×	×	×	2.85	**5.50**	**9.08**	**15.78**

This indicates respectively that the enhanced and smoothed RTFR features are more robust than STFT features and that the activation patterns acquired by the proposed approach using NMF to discover time–frequency patches deliver a more robust input stream to the word detection system than those obtained by the cNMF approach.

6. Conclusions

In this paper, we proposed a bottom-up approach for learning the words of a language. An unsupervised technique was presented to discover a set of spectral patches that can describe speech. We exploited the use of two complementary feature representations derived from a reassigned time–frequency spectrogram to obtain a representation that can cope with acoustic events with short and long durations. The non-negative matrix factorization (NMF) algorithm using sparsity constraints was applied to discover latent recurring patterns in static and dynamic features. The obtained basis vectors correspond to phone-sized spectral patterns which we referred to as time–frequency patches. Experiments on the Aurora2 database revealed that these patches are activated in speaker-independent patterns which are related to the words of a language.

Next, a learning algorithm was built that automatically discovers and acquires the recurring patterns in the activation data by applying NMF on the co-occurrence counts of activation events. The obtained patterns were associated with the words of a language and finally the system was able to detect the words in unseen (noisy) speech data.

Experimental evidence was given for the noise robustness of the proposed word detection system, based on the Aurora2 digit recognition task. Although a conventional HMM-based approach using cepstral features obtained better results on clean speech data, the proposed learning algorithm showed a superior performance for speech that is distorted by the noise down to 5 dB SNR. The NMF learning algorithm was shown to be sufficiently versatile to apply it at both levels of speech representations for discovering structure in the data. NMF has less parameters to be tuned in comparison to HMM-based systems. The most important parameters are the number of time–frequency patches P, the sparsity parameter λ in the NMF of the first layer and the number of VQ-labels Σ in the second layer. Moreover, experiments not reported here have shown that for the small vocabulary task as was considered in this paper, the performance of the system is not very sensitive to these parameters.

Inspired by research on the auditory cortex of mammals, researchers have suggested that ASR-systems should trigger on the presence of spectro-temporal patches. Such biologically inspired systems might exhibit properties of human audition such as robustness to noise. In this paper, we have shown that such an auditory representation with good robustness can be obtained through unsupervised learning (the first layer). We have also shown how the activation patterns can be exploited to build a speech recognizer. Further work involves extending the word detection system to a speech recognition system that also provides information related to the order in which the words occur in the test utterances. Therefore, the HAC-models in the second layer can be extended by moving a sliding window over the utterance to detect the time of occurrence of the different words in the utterance and hence the word order. In our current implementation, the noise has been left uncompensated and we would like to investigate to which extent the performance of the recognition system can be further improved by exploiting noise reduction techniques. Though, the layered architecture offers scalability towards vocabularies in the sense that the patch set is reusable across words, more research is required to reveal how well the proposed system is suited for large vocabulary continuous speech recognition. At his point, the presented three-layered architecture is not capable to deal with these vocabularies because of the large data requirements per word. However, the scalability of the system can be increased by adding more layers in cascade to model the words as a combination of sub-word patches instead of learning all the words from scratch.

To conclude, we believe that the presented system is an ideal platform for future research as in its baseline implementation it already yields competitive results and could open new avenues of research on automatic speech recognition.

Acknowledgements

This research was funded by the Institute for the Promotion of Innovation through Science and Technology in Flanders, Belgium (I.W.T.-Vlaanderen) and by the European Commission under contract FP6-034362 (ACORNS).

References

Auger, F., Flandrin, P, 1995. Improving the readability of time–frequency and time-scale representations by the reassignment method. IEEE Trans. Signal Process. 43 (5), 1068–1089.

Aversano, G., Esposito, A., Marinaro, M., 2001. A new text-independent method for phoneme segmentation. In: Proc. IEEE Internat. Workshop Circuits and Systems 2, 516–519.

Baker, J.M., Deng, L., Khudanpur, S., Lee, C.-H., Glass, J., Morgan, N., 2006–2007. Minds historical development and future directions in speech recognition and understanding. Tech. Rep., Report of the Speech Understanding Working Group. <http://www.itl.nist.gov/iaui/894.02/MINDS/FINAL/speech.web.pdf>.

Brugnara, F., Falavigna, D., Omologo, M., 1993. Automatic segmentation and labeling of speech based on hidden markov models. Speech Commun. 12 (4), 357–370.

Chomsky, N., 2000. New Horizons in the Study of Language and Mind. Cambridge University Press, Cambridge, UK.

Ding, C., Li, T., Jordan, M., 2006. Convex and semi-nonnegative matrix factorizations for clustering and low-dimension representation. Tech. Rep. LBNL-60428, Lawrence Berkeley National Laboratory, US.

ETSI Standard Document, 2000a. Distributed Speech Recognition; Front End Feature Extraction Algorithm; Compression Algorithm. ETSI ES 201 108 v1.1.2, April.

ETSI Standard Document, 2000b. Speech Processing, Transmission and Quality Aspects (STQ); Distributed Speech Recognition; Advanced Front-end Feature Extraction Algorithm; Compression Algorithm. ETSI ES 202 050 v1.1.1 (2002-10), April.

Ezzat, T., Bouvrie, J., Poggio, T., 2007. Spectro-temporal Analysis of Speech Using 2-D Gabor Filters. In: Proc. International Conference on Spoken Language Processing, Antwerp, Belgium, August, pp. 506–509.

Hainsworth, S., Macleod, M., 2003. Time–frequency reassignment: a review and analysis. Tech. Rep. CUED/FINFENG/TR.459, Cambridge University Engineering Department.

Hermansky, H., Fousek, P., 2005. Multi-resolution RASTA Filtering for TANDEM-based ASR. In: Proc. International Conference on Spoken Language Processing, Lisbon, Portugal, October, pp. 361–364.

Hermansky, H., Sharma, S., 1997. Temporal Patterns (TRAPs) in ASR of Noisy Speech. In: Proc. International Conference on Acoustics, Speech and Signal Processing, vol. 1. Phoenix, Arizona, March, pp. 289–292.

Hermansky, H., Sharma, S., 1998. TRAPS Classifiers of Temporal Patterns. In: Proc. International Conference on Spoken Language Processing, Sydney, Australia, November, pp. 1003–1006.

Hirsch, H.G., Pearce, D., 2000. The AURORA experimental framework for the performance evaluation of speech recognition systems under noisy conditions. In: Proc. ISCA ITRW ASR2000 Workshop, Paris, France, September, pp. 18–20.

Hoyer, P., 2004. Non-negative matrix factorization with sparseness constraints. J. Machine Learn. Res. 5, 1457–1469.

Kingsbury, B., Morgan, N., Greenberg, S., 1998. Robust speech recognition using the modulation spectrogram. Speech Commun. 25, 117–132.

Kleinschmidt, M., 2003. Localized spectro-temporal features for automatic speech recognition. In: Proc. Eurospeech, Geneva, Switzerland, September, pp. 2573–2576.

Kodera, K., Gendrin, R., Villedary, C., 1978. Analysis of time-varying signals with small bt values. IEEE Trans. Audio Speech Language Process. 26 (1), 64–76.

Lee, D., Seung, H., 2001. Algorithms for non-negative matrix factorization. Adv. Neural Inform. Process. Systems 13, 556–562.

Macho, D., Mauuary, L., Noé, B., Cheng, Y., Ealey, D., Jouvet, D., Kelleher, H., Pearce, D., Saadoun, F., 2002. Evaluation of a Noise-

robust DSR Front-end on Aurora Databases. In: Proc. International Conference on Spoken Language Processing, Denver, Colorado, USA, September, pp. 17–20.

Martin, A., Doddington, G., Kamm, T., Ordowski, M., Przybocki, M., 1997. The DET curve in assessment of detection task performance. In: Proc. Eurospeech, Rhodes, Greece, September, pp. 1895–1898.

Meyer, B.T., Kollmeier, B., 2008. Optimization and Evaluation of Gabor Feature Sets for ASR. In: Proc. International Conference on Spoken Language Processing, Brisbane, Australia, September, pp. 906–909.

O'Grady, P.D., Pearlmutter, B.A., 2008. Discovering speech phones using convolutive non-negative matrix factorisation with a sparseness constraint. Neurocomputing 72, 88–101.

Park, A., Glass, J., 2005. Towards unsupervised pattern discovery in speech. In: Proc. ASRU, San Juan, Puerto Rico, December, pp. 53–58.

Plante, F., Meyer, G., Ainsworth, W., 1998. Improvement of speech spectrogram accuracy by the method of reassignment. IEEE Trans. Speech Audio Process. 6 (3), 282–286.

Qiao, Y., Shimomura, N. Minematsu, N., 2008. Unsupervised Phoneme Segmentation Using Transformed Cepstrum Features. In: Proc. Spring Meeting of Acoust. Soc. Japan, 287-2901-11-20.

Scharenborg, O., Norris, D., ten Bosch, L., McQueen, J.M., 2005. How should a speech recognizer work? Cognitive Sci. 29 (6), 867–918.

Scharenborg, O., Ernestus, M., Wan, V., 2007. Segmentation of Speech: Child's Play? In: Proc. International Conference on Spoken Language Processing, Antwerp, Belgium, August, pp. 1953–1956.

Siivola, V., Hirsimäki, T., Creutz, M., Kurimo, M., 2003. Unlimited vocabulary speech recognition based on morphs discovered in an unsupervised manner. In: Proc. Eurospeech, pp. 2293–2296.

Smaragdis, P., 2007. Convolutive speech bases and their application to speech separation. IEEE Trans. Speech Audio Process. 15 (1), 1–12.

Stouten, V., Demuynck, K., Van hamme, H., 2008. Discovering phone patterns in spoken utterances by nonnegative matrix factorisation. IEEE Signal Process. Lett. 15, 131–134.

Tyagi, V., McCowan, I., Misra, H., Bourlard, H., 2003. Mel-cepstrum modulation spectrum (MCMS) features for robust ASR. In: Proc. ASRU 2003 Workshop, St. Thomas, Virgin Islands, December, pp. 399–404.

Van hamme, H., 2008a. HAC-models: A Novel Approach to Continuous Speech Recognition. In: Proc. International Conference on Spoken Language Processing, Brisbane, Australia, pp. 2554–2557.

Van hamme, H., 2008b. Integration of asynchronous knowledge sources in a novel speech recognition framework. In: ISCA ITRW Workshop on Speech Analysis and Processing for Knowledge Discovery, Aalborg, Danmark, June.

Virtanen, T., 2007. Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria. IEEE Trans. Audio Speech Language Process. 15 (3), 1066–1074.

Young, S., Kershaw, D., Odell, J., Ollason, D., Valtchev, V., Woodland, P., 1999. The HTK Book – Version2.2. Entropic.