

Discovering Phone Patterns in Spoken Utterances by Non-negative Matrix Factorisation

Veronique Stouten*, *Member, IEEE*, Kris Demuynck, Hugo Van hamme, *Member, IEEE*

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

Email : {veronique.stouten,kris.demuynck,hugo.vanhamme}@esat.kuleuven.be

Tel : {+32-16-321827} {+32-16-321860} {+32-16-321842} Fax : {+32-16-321723}

Abstract

We present a technique to automatically discover the (word-sized) phone patterns that are present in speech utterances. These patterns are learnt from a set of phone lattices generated from the utterances. Just like children acquiring language, our system does not have prior information on what the meaningful patterns are. By applying the non-negative matrix factorisation algorithm to a fixed-length high-dimensional vector representation of the speech utterances, a decomposition in terms of additive units is obtained. We illustrate that these units correspond to words in case of a small vocabulary task. Our result also raises questions about whether explicit segmentation and clustering are needed in an unsupervised learning context.

Index Terms

matrix factorisation, word segmentation, phone lattices, language acquisition.

EDICS Category: SPE-RECO

I. INTRODUCTION

It is remarkable that infants are able to automatically learn the acoustic, lexical and grammatical patterns of a language. Indeed, human learners appear to synthesise units out of large and apparently

*Corresponding author.

unsegmented streams, using the statistical structure embedded within this stream [1], [2]. Moreover, they can do this significantly better than current automatic speech recognition (ASR) systems. Nevertheless, ASR systems make use of expert knowledge that arises from audiology, phonology and linguistics. In the resulting beads-on-a-string approach [3], the speech is represented as a sequence of elementary sounds that are combined to sentences in a hierarchical way (sub-phones are combined to phones, which are linked together to words, and finally to sentences). The automatic *discovery* of the structure of speech - a task in which billions of infants have succeeded - is out of the question.

Conventional pattern recognition systems rely on supervised learning, which requires that the patterns to be recognised as well as the primitive elements from which complex patterns can be formed, are defined a priori. This is a necessary prerequisite for training statistical models of the primitive elements and the patterns. In speech recognition, the patterns to be recognised are almost invariably words, while the primitives are related to the phonemes of the language. By deciding beforehand what the words are, and how the words are composed of sub-word units, essential constraints are imposed onto the learning system. This is undoubtedly advantageous from the point of view of a meta-level description of how speech is organised. However, children gradually learn new words and the corresponding sequences of speech sounds during the language acquisition process. In humans these are emergent properties, formed on the basis of speech input in a meaningful semantic/pragmatic context.

We intend to build a system that retrieves the phone patterns within the speech input without prior knowledge of a pre-defined set of patterns linked to a fixed and pre-defined set of concepts. The core of our system consists of an unsupervised algorithm, namely non-negative matrix factorisation (NMF) [4]. Thanks to the non-negativity constraints, NMF decomposes a matrix in additive (not subtractive) components. Moreover, the result can be given a probabilistic interpretation, which is not the case for the result of latent semantic analysis (LSA) or principal component analysis (PCA). In [5], the equivalence (down to almost identical update rules) is shown between probabilistic LSA (pLSA) and NMF with a Kullback-Leibler divergence criterion. Compared to the multigram model [6], our approach is more elegant in that it is less complex. Namely, NMF does not need a threshold on the maximal length of the units, nor does it require a heuristic that determines which units to keep. Instead of using a clustering technique as in [7], a mathematical factorisation is applied.

In the next section, we briefly review the basics of NMF. To construct the input matrix, a fixed-length high-dimensional vector representation must be obtained from the speech utterances. In this step, we do not start from scratch, but apply phonetic knowledge sources as explained in section III. In section IV, we describe how the phone patterns are automatically discovered from this matrix. Hence, the lexical

information is extracted in an unsupervised way. We also show that these lexical units can be found back in previously unseen test data. We illustrate our findings with experiments on a small vocabulary database. Conclusions can be found in section V.

II. THE NMF ALGORITHM

Let us consider an $(n \times t)$ input matrix \mathbf{V} that contains only elements that are positive or zero (e.g. probabilities, counts, ...). An approximate factorisation of the input matrix into an $(n \times r)$ matrix \mathbf{W} and an $(r \times t)$ matrix \mathbf{H} is obtained by optimising an objective function under the constraint that all matrix elements should be non-negative.

$$\mathbf{V} \approx \mathbf{W}\mathbf{H} \quad (1)$$

Usually the value of r is chosen such that $r(n + t) < nt$, which results in a reconstructed matrix \mathbf{V} with reduced dimensionality. Several forms of the objective function (such as divergence, mean-squared error, Frobenius norm, ...) have been proposed in literature. In this paper, the divergence criterion seems appropriate since \mathbf{W} and \mathbf{H} have a probabilistic interpretation :

$$D(\mathbf{V}||\mathbf{W}\mathbf{H}) = \sum_{i,j} \left([\mathbf{V}]_{ij} \log \frac{[\mathbf{V}]_{ij}}{[\mathbf{W}\mathbf{H}]_{ij}} - [\mathbf{V}]_{ij} + [\mathbf{W}\mathbf{H}]_{ij} \right) \quad (2)$$

The notation $[\mathbf{A}]_{ij}$ denotes the element in the i th row and j th column of matrix \mathbf{A} . It can be shown [4] that the algorithm converges to a local optimum of this objective function by iterating the following update rules for \mathbf{W} and \mathbf{H} :

$$[\mathbf{H}]_{k\ell} \leftarrow [\mathbf{H}]_{k\ell} \frac{\sum_i [\mathbf{W}]_{ik} [\mathbf{V}]_{i\ell} / [\mathbf{W}\mathbf{H}]_{i\ell}}{\sum_j [\mathbf{W}]_{jk}} \quad (3)$$

$$[\mathbf{W}]_{ik} \leftarrow [\mathbf{W}]_{ik} \frac{\sum_\ell [\mathbf{H}]_{k\ell} [\mathbf{V}]_{i\ell} / [\mathbf{W}\mathbf{H}]_{i\ell}}{\sum_m [\mathbf{H}]_{km}} \quad (4)$$

Because these update rules are multiplicative and do not change the sign of \mathbf{W} or \mathbf{H} , it is sufficient to initialise the matrix elements of \mathbf{W} and \mathbf{H} to strictly positive values in order to satisfy the non-negativity constraints. Eq. (1) indicates that each data vector $[\mathbf{V}]_{:j}$ is written as a linear combination of the columns of \mathbf{W} weighted by the coefficients $[\mathbf{H}]_{:j}$. Since relatively few basis vectors $[\mathbf{W}]_{:k}$ are used ($r \ll t$), a high quality approximation can only be achieved if the basis vectors discover structure that is latent in the data. An interesting property of NMF is that it usually produces sparse representations, such that the input can be interpreted in terms of a few components.

III. SPEECH UTTERANCE REPRESENTATION

One of the key issues in applying latent variable methods such as NMF in an ASR context, is to find a fixed-length vector representation of an utterance such that the model underlying the latent variable method applies. In this paper, we explore the use of (weighted) phone lattice transition probabilities to represent a speech utterance (or a part thereof).

For each speech utterance (that typically contains several words), a dense phone network is constructed using the first layer of the FLaVoR architecture [8], which performs an acoustic-phonetic decoding. In this decoding step, a search algorithm determines the network of most probable phone strings F , given the acoustic features Z of the incoming signal. The employed knowledge sources are an acoustic model $p(Z|F)$ and a phone transition model $p(F)$. The phone network contains the set of matching (context-dependent) phones with their start and end times. In this acyclic directed graph, the arcs correspond to phones and the nodes impose time and context-dependency constraints. The acoustic score that is associated with each arc is transformed to a posterior probability. The ratio of the acoustic model likelihood and the phone transition model likelihood is adjusted on an independent development set as to optimise the mutual information between the arc probabilities and the true phone identity.

The NMF input matrix V is then obtained as follows. The information that is contained in each phone network of an utterance is summarised into one column of V . Thereto, the probability of every two consecutive phones ϕ and ψ is accumulated over the network :

$$c(\phi, \psi) = \sum_{\{\alpha: h(\alpha)=\phi\}} \sum_{\{\beta: h(\beta)=\psi\}} p(\alpha) p(\beta) \Delta_{\alpha\beta} \quad (5)$$

in which $h(\alpha)$ and $h(\beta)$ return the phone identity, and $p(\alpha)$ and $p(\beta)$ the posterior probability of the arcs α and β , respectively. If the start node of β is equal to the end node of α , then $\Delta_{\alpha\beta}$ is the inverse of the probability of the common node, else it is 0. This node probability is given by the sum of the posterior probabilities of the incoming (or outgoing) arcs of the corresponding node.

The values $c(\phi, \psi)$ form the entries of the $(n \times t)$ matrix \mathbf{V} , in which n is equal to the square of the number of phone identities and t is given by the number of utterances in the database. Since the elements of \mathbf{V} represent weighted co-occurrence counts, they will never become negative.

IV. SYSTEM SETUP

In this section, we illustrate the performance of the system on a speaker independent database. The speech data are taken from TI-Digits [9] which contains recordings of US-American adults, downsampled to 16 kHz. The training set consists of 6159 connected digit sequences of length 1 through 7.

Because of the small vocabulary of the task, the phone lattices are generated without too much prior information. A set of 43 different phone identities is used, including the phone ‘SIL’ (= silence). The acoustic-phonetic decoding makes use of a state-of-the-art acoustic model $p(Z|F)$ that consists of a HMM with cross-word context-dependent tied states (GMMs). This model is trained on the Wall Street Journal (WSJ0 plus WSJ1) database such that it is general enough. For the phone transition model $p(F)$, a bigram is estimated on the same data (but a unigram proved to be just as good).

A. Discovering phone patterns

First, the input matrix \mathbf{V} is constructed for the TI-Digits training set as we explained in section III. Then, the NMF algorithm is applied. The columns of \mathbf{W} represent recurring units in the data, while the columns of \mathbf{H} indicate which units are active in each utterance. When matrix \mathbf{V} is obtained from real phone networks, the algorithm has to cope with several difficulties, which can be divided into two categories:

1. uncertainties in the phone lattice, e.g. paths that deviate from the canonical transcription, errors induced by the acoustic-phonetic decoder, pronunciation variants,...
2. cross-model effects, e.g. non-negligible probabilities for transitions between the phones of two consecutive words (without silence in between), co-articulation effects, etc.

Due to these distortions, the model that is imposed by NMF is not fully satisfied. Namely, the input is not an exact linear combination of basis vectors. To investigate their effect on the performance of our algorithm, we consider 3 cases. The first one is the idealised case in which the lattice consists only of the path corresponding to the canonical phonetic transcription of the utterance and all words are embedded in silence. In the second case, cross-model effects are included in a worst case scenario by removing all silence between words and by applying the most typical co-articulation rules (degemination, voice assimilation). In the third case, the real lattice is used such that both distortion effects are to some extent present.

In figure 1, the value of the NMF objective function is shown versus the number of basis vectors $[\mathbf{W}]_{:k}$ that are extracted for each of the above mentioned cases. The matrix factors \mathbf{W} and \mathbf{H} were initialised with positive random values. The NMF update rules were applied for 2000 iterations. Local optima were avoided through a ‘simulated annealing’ technique: an additive distortion (≥ 0) of which the magnitude decreases with the iteration number according to a sigmoid function, was applied to \mathbf{W} and \mathbf{H} . From figure 1, it can be observed that the slope of the curves changes at $r = 11$, which corresponds to the number of words present in the database. Hence, as long as the extra basis vector can be used to model a

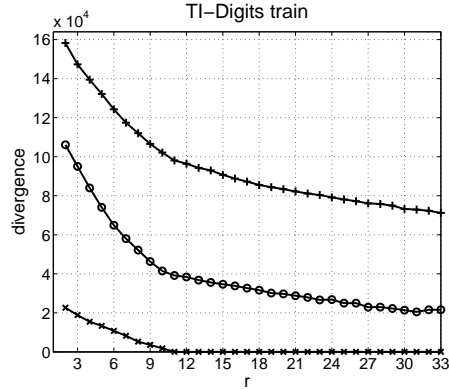


Fig. 1. Value of the NMF objective function versus the number of basis vectors that are extracted. Idealised case (x), idealised lattice but cross-model effects (o), real case (+). TI-Digits training set.

new word from the database ($r < 11$), the divergence criterion decreases fast in all three cases. When the number of basis vectors gets larger than the number of words ($r \geq 11$), the cost function still decreases in case 2 and case 3 but at a slower rate, while it asymptotically converges to zero w.r.t. the number of iterations in the idealised case. Finally, it can be observed that the uncertainties in the lattices cause a larger increase of the cost function than the cross-model effects when $r = 11$. This conclusion is validated by the fact that in case 2 more silence between words was removed than in case 3 (‘worst case’ vs. ‘real case’ cross-model effects). Also, the ‘knee’ tends to become weaker in the case of real lattices.

We will now investigate how the basis vectors are related to phone patterns. In table I, we show an extract from matrix \mathbf{W} when the input matrix \mathbf{V} is obtained from real phone networks and $r = 11$. The matrix factors \mathbf{W} and \mathbf{H} are normalised such that the columns of \mathbf{W} have unit L1-norm. The values in each column are sorted in descending order and only the upper part is shown. For each value, the corresponding phone transition (related to the index in \mathbf{W}) is also given. The obtained basis vectors $[\mathbf{W}]_{:,k}$ indicate which phone pairs frequently occur together in so-called *phone patterns*. No explicit timing (or even exact ordering) information is present in $[\mathbf{W}]_{:,k}$. Yet, the corresponding sequence of phones can be deduced from the phone pair values. It is striking that the dominant phone pairs in each $[\mathbf{W}]_{:,k}$ form a *connected* sequence, although this ‘temporal smoothness’ was not imposed. However, determining its start or end phone is out of the scope of this letter. This result questions the need for explicit segmentation and clustering as is done in [6]. Due to the normalisation, the weights in the rows $[\mathbf{H}]_{k,:}$ give an indication of the length of the sequence.

From table I, it can be seen that the phone transitions that actually occur in the digit are dominant. Often, the ‘deviant’ transitions that have a reasonably large weight can be linked to the first or the last phone of one of the (other) digits. E.g. in the first column ‘N W’ could have been obtained from the digit sequences 1-1, 7-1 or 9-1. On the other hand, ‘AA N’ in the same column arose from a pronunciation variant of the word ‘one’. The same applies for ‘Z IY’ in the basis vector of ‘zero’. We can conclude that NMF is able to extract the phone patterns that are present in the TI-Digits database, although the basis vectors contain some distortion (e.g. non-zero weights for inter-word transitions) because the NMF model assumptions are not fully satisfied in a real phone network.

It is interesting to analyse the behaviour of the algorithm w.r.t. r . When e.g. $r = 20$, all 11 digits are still discovered, while the remaining 9 basis vectors are used to model the following effects: frequently occurring recognition errors in the phone lattice (such as ‘SIL B AY’) are modelled [4 vectors], already modelled words are added with different dominant inter-word transitions [3 vectors], and alternative pronunciations of a word are disunited (e.g. a separate $[\mathbf{W}]_{:k}$ for ‘Z IH R OW’ and ‘Z IY R OW’) [2 vectors].

B. Labelling the phone patterns

In this section, a labelling algorithm is described to *automatically* link each basis vector to a sequence of phones. Let us consider the set of labels that represents all possible speech units. In our case, this set contains the lexicon of the Resource Management (RM) database, augmented with all sub-sequences of phones. E.g. the word ‘any’ gives rise to the following labels: {‘SIL EH N IY SIL’, ‘SIL EH N IY’, ‘SIL EH N’, ‘SIL EH’, ‘EH N IY SIL’, ‘EH N IY’, ‘EH N’, ‘N IY SIL’, ‘N IY’, ‘IY SIL’}. For each label b the $(n \times 1)$ vector $\mathbf{c}_{:b}$ is constructed that contains the number of occurrences of every phone transition in this label. Both $\mathbf{c}_{:b}$ and $[\mathbf{W}]_{:k}$ are given unit L1-norm. The labelling algorithm then assigns to each basis vector $[\mathbf{W}]_{:k}$ a label b for which the Kullback-Leibler divergence between $\mathbf{c}_{:b}$ and $[\mathbf{W}]_{:k}$,

$$D(\mathbf{c}_{:b} || [\mathbf{W}]_{:k}) = \sum_{\{i: \mathbf{c}_{ib} \neq 0\}} \left(\mathbf{c}_{ib} \log \left(\frac{\mathbf{c}_{ib}}{[\mathbf{W}]_{ik} + \epsilon} \right) \right) \quad (6)$$

is minimal. Here, $\epsilon (= 10^{-50})$ is a small number. Since RM contains not only the 11 digits but also more than 900 other words, the quality of the basis vector must be high in order to obtain a correct label. Although the dominance of the correct phone transitions is not always very pronounced in $[\mathbf{W}]_{:k}$, this method resulted in the automatic assignment of the correct digit label to each basis vector of table I. This illustrates that \mathbf{W} contains word-sized phone patterns. The basis vectors $[\mathbf{W}]_{:k}$ also provide a flexible way

TABLE I
EXTRACT FROM BASIS VECTORS $[\mathbf{W}]_{:,k}$ FOR REAL PHONE LATTICE.

basis vectors							
one		two		three		four	
AH N	0.2094	T UW	0.3210	R IY	0.2033	AO R	0.2617
W AH	0.1978	SIL T	0.1224	TH R	0.1735	F AO	0.2326
SIL W	0.0836	UW SIL	0.1051	IY SIL	0.0700	SIL F	0.0868
N SIL	0.0708	N T	0.0456	SIL TH	0.0572	R SIL	0.0861
AA N	0.0513	UW S	0.0439	IY S	0.0386	N F	0.0484
N W	0.0504	UW F	0.0435	N TH	0.0370	R S	0.0386
W AA	0.0403	OW T	0.0390	IY F	0.0367	R F	0.0331
five		six		seven		eight	
F AY	0.2200	K S	0.1745	S EH	0.1669	EY T	0.2717
AY V	0.1709	S IH	0.1303	AH N	0.1539	SIL EY	0.1333
SIL F	0.0633	IH K	0.1138	EH V	0.1415	T SIL	0.1227
V SIL	0.0567	S SIL	0.0611	V AH	0.1363	N EY	0.0560
N F	0.0433	SIL S	0.0601	SIL S	0.0546	EY D	0.0478
AY D	0.0369	S T	0.0378	N SIL	0.0473	T EY	0.0375
D SIL	0.0263	IH K	0.0322	N S	0.0388	T S	0.0288
nine		oh		zero			
AY N	0.2655	SIL OW	0.0697	R OW	0.1472		
N AY	0.2567	OW SIL	0.0442	IH R	0.1200		
SIL N	0.0909	AH L	0.0335	Z IH	0.1018		
N SIL	0.0819	L SIL	0.0323	OW SIL	0.0492		
N D	0.0348	AE N	0.0297	N Z	0.0366		
AH N	0.0286	OW L	0.0237	IY R	0.0365		
N T	0.0239	AE T	0.0228	Z IY	0.0296		

of describing pronunciation variants of the words, as illustrated in table I. In other words, each $[\mathbf{W}]_{:,k}$ can be seen as a phone-level statistical pronunciation model.

C. Activation of speech units in unseen data

We will now show how the obtained phone patterns can be found back in previously unseen data. To this end, NMF is applied to the TI-Digits test utterances to calculate the optimal coefficients (matrix \mathbf{H}), while keeping the set of phone patterns that has been discovered automatically in the training data (matrix \mathbf{W}) fixed. In this case, the information that is contained in each phone network (utterance) is

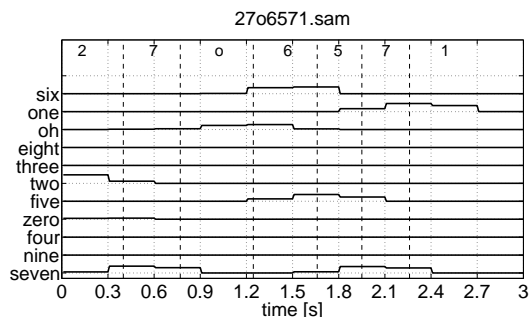


Fig. 2. Graphical representation of the rows of matrix \mathbf{H} . For each basis vector (ordinate), the value in $[\mathbf{H}]_k$ is related to the evidence that the word is present at a certain time instant.

summarised into multiple columns of \mathbf{V} by using a sliding window. The entries in \mathbf{V} (eq. (5)) are then a summation over the time instants within this sliding window only. We choose a window length of 600 ms with a shift of 300 ms. Note that some ‘artifacts’ are introduced when a word or syllable is only partly contained in the sliding window. However, the algorithm did not have to cope with this kind of artifacts when learning \mathbf{W} since one utterance was summarised into one column of \mathbf{V} in the training phase. This results in basis vectors of higher quality.

Figure 2 shows the rows of matrix \mathbf{H} for the utterance ‘27o6571’. The ordinate represents the (fixed) basis vectors $[\mathbf{W}]_{:k}$, while the abscissa represents the time index. As a reference, the true word boundaries are shown as vertical dotted lines. As can be seen, all digits are ‘recognised’. Often, evidence is found for the presence of more than one word (or a part thereof). Some words (e.g. ‘five’, ‘seven’) get a high weight for quite a long time w.r.t. the segmentation file. In this respect it should be noted that the use of a sliding window causes a smoothing over time. Hence, the information becomes available (up to half a window length) before the event actually takes place.

V. CONCLUSION

We have presented a technique to automatically discover the phone patterns that are present in speech utterances. First, a dense phone network is constructed from each utterance, given the set of basis units (phones). Then, the non-negative matrix factorisation (NMF) algorithm is applied to find the recurring patterns in the data in an unsupervised way. An advantage is that no tuning of parameters is required. We have illustrated that the obtained basis vectors correspond to words in case of a small vocabulary task. Pronunciation variants of the words are also automatically discovered. This novel approach to unsupervised word discovery and pronunciation modelling does not require explicit segmentation or

clustering operations. These promising results open up perspectives to deviate from the conventional beads-on-a-string approach to model speech.

ACKNOWLEDGEMENT

This research was funded by ‘Research Fund (Onderzoeksfonds) KULeuven’ (project no.OT/03/32/TBA), by the IWT - SBO project ‘SPACE’ (project no.040102), and by the European Commission under contract FP6-034362. The authors wish to thank Prof. Chin-Hui Lee for useful discussions.

REFERENCES

- [1] J. Saffran, R. Aslin, and E. Newport, “Statistical learning by 8-month-old infants,” *Science*, vol. 274, no. 5294, pp. 1926–1928, 1996.
- [2] E. Newport and R. Aslin, “Innately constrained learning: Blending old and new approaches to language acquisition,” in *Proc. 24th Annual Boston Univ. Conf. on Language Development*, S. Howell, S. Fish, and T. Keith-Lucas, Eds. Somerville, MA: Cascadilla Press, 2000, pp. 1–21.
- [3] M. Ostendorf, “Moving beyond the ‘beads-on-a-string’ model of speech,” in *Proc. ASRU 1999*, Keystone, Colorado, USA, Dec. 1999.
- [4] D. Lee and H. Seung, “Algorithms for non-negative matrix factorization,” *Advances in Neural Information Proc. Syst.*, vol. 13, pp. 556–562, 2001.
- [5] E. Gaussier and C. Goutte, “Relation between PLSA and NMF and implications,” in *Proc. ACM SIGIR conference on research and development in information retrieval*, Salvador, Brazil, 2005, pp. 601–602.
- [6] S. Deligne and F. Bimbot, “Inference of variable-length linguistic and acoustic units by multigrams,” *Speech Communication*, vol. 23, no. 3, pp. 223–241, 1997.
- [7] A. Park and J. Glass, “Towards unsupervised pattern discovery in speech,” in *Proc. ASRU*, San Juan, Puerto Rico, 2005, pp. 53–58.
- [8] K. Demuynck, T. Laureys, D. Van Compernelle, and H. Van hamme, “Flavor: a flexible architecture for LVCSR,” in *Proc. EUROSPEECH*, Geneva, Switzerland, Sept. 2003, pp. 1973–1976.
- [9] R. Leonard, “A database for speaker-independent digit recognition,” in *Proc. ICASSP*, San Diego, CA, U.S.A., 1984, pp. 42.11/1–4.