Integration of Asynchronous Knowledge Sources in a Novel Speech Recognition Framework.

Hugo Van hamme

Katholieke Universiteit Leuven, dept. ESAT, Belgium
hugo.vanhamme@esat.kuleuven.be

Abstract

Hidden Markov Models have been essential in obtaining today's successes in speech recognition. However, some limitations of HMMs become clear: for example it is difficult to successfully exploit features that are measured at different time scales than the centisecond scale at which the spectral features are measured. Little success has been achieved in integrating utterance level information such as prosody, segmental information and finer detail such as voice onset times. In this paper, we apply latent semantic analysis (LSA) techniques known from the text processing field to histograms of acoustic event co-occurrence (HAC) to propose a novel speech recognition framework. We show that the HAC-method can deal with correlated information and exploit knowledge sources that are asynchronous.

Index Terms: speech recognition, information discovery, information integration, latent semantic analysis, co-occurrence statistics.

1. Introduction

Today's state-of-the-art speech recognizers are mostly based on Hidden Markov Models (HMMs). This model handles the variation in the speech signal in a statistical framework. The sequential nature of speech is represented by states with restricted transitions. In recent years however, there has been a growing awareness that some important information in the speech signal is difficult to model with HMMs. Especially when the speech features are measured at different time scales, the experience is that they are hard to integrate in the model. Areas where HMMs fall short are e.g. duration and prosody modelling, pronunciation variation and context dependency. Dynamic Bayesian networks are an extension of HMMs that offer a framework that has extended capabilities for integrating multiple knowledge sources [1].

This paper advocates a radically different approach inspired by latent semantic analysis (LSA) techniques. A high-dimensional shift-invariant speech representation called *histogram of acoustic co-occurrences* (HAC) is proposed and algorithms for training and recognition are proposed. It is shown that sources of information at different time scales are easily integrated into a joint decision. Though this paper focuses on (weakly) supervised learning, its unsupervised learning capabilities have also been shown in [2]. The core of the method is based on non-negative matrix factorisation ([3] NMF) in a high-dimensional space. NMF has been applied in speech and audio processing for various applications such as music transcription [4], [5], source separation [6] and feature extraction [7]. However, the vector space to which NMF is applied in these applications differs from the one used in this paper.

This paper is organised as follows: in section 2 the histogram of acoustic co-occurrence (HAC) model is introduced gradually. The experiment of section 3 shows the information stream integration capabilities which are discussed in section 4.

2. Histograms of acoustic co-occurrences

2.1. Input data

We consider symbolic input data that can be represented in directed graph Ω as illustrated in Figure 1. Each arc in the graph is labelled with a symbol and an associated positive activation level. The symbols are drawn from a set S with finite cardinality |S|. The activation levels could be posterior probabilities (as will be the case below), the activations of neural networks, signal energies or activations of the type computed in section 2.4. We list a few examples:

- S are phones and the Ω is a phone lattice with posterior probabilities, while the nodes are labelled with the start and end times of the phones. A method for generating phone lattices is described in [9].
- S are vector quantisation (VQ) labels of spectra (full-band or in subbands) and Ω is a chain, i.e. each node is connected to a single successor. The nodes are spaced regularly in time at multiples of the spectral analysis frame shift time. All activations are 1.
- With soft vector quantisation, there are multiple parallel arcs between the nodes, see Figure 2. Soft VQ can be applied to spectra, but phone posterior probabilities evaluated at regular instants also adhere to this form.

The node times will not be used explicitly in this paper.

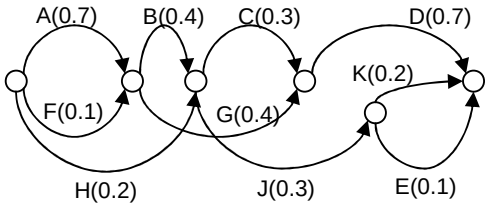

Figure 1*: example of an input graph. The arcs are labeled with a symbol and an activation (a posterior probability in this example).*

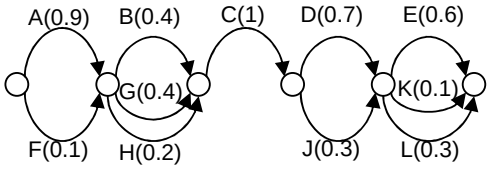

Figure 2: *vector quantization (VQ) and soft VQ can be represented as a chain graph.*

2.2. Shift-invariance and linearity

A graph is converted to a histogram vector of length $N = |S|^2$ by accumulating the co-occurrence activations of all ordered symbol pairs (A,B) at a lag τ over the graph. The lag between the arc pair (α,β) is defined as the minimal number of nodes that need to be visited to travel from α to β. Several options can be considered for defining the co-occurrence activation in a graph. In this paper, we will work in a statistical framework in which the activations are posterior probabilities. Consider all paths that pass through α and β with τ nodes in between. The co-occurrence activation of (α,β) is the sum of probabilities of all such paths. The symbol co-occurrence activation of (A,B) is then the sum of arc pair activations over all arc pairs that carry the symbol A and B respectively. A practical expression for $\tau = 1$ can be found in [2].

The operator *HAC* (histogram of acoustic co-occurrences) transforms Ω to an *N*-dimensional real-valued vector $\mathbf{w}$ by considering all *N* possible symbol pairs. Let $\Omega_1, \ldots, \Omega_R$ be graphs and $\mathbf{w}_r = HAC(\Omega_r)$ for $r = 1 \ldots R$. First notice that all $\mathbf{w}_r$ are points in the same *N*-dimensional space, irrespective of the size of Ω_r. Consider Ω as the concatenation of $\Omega_1, \ldots, \Omega_R$ (assume graphs a single entry and exit node, such that they can be concatenated). $HAC(\Omega)$ differs only from the sum (over *r*) of $HAC(\Omega_r)$ by the contributions of any cross-graph arc pairs that start more than τ nodes before the start node of Ω_r. In that sense, the *HAC*-operator is approximately shift-invariant: it is only affected by very near predecessor arcs in Ω. Hence,

$$\mathbf{w} = HAC(\Omega) \approx \sum_{r=1}^{R} HAC(\Omega_r) = \sum_{r=1}^{R} \mathbf{w}_r \tag{1}$$

Let Ω now be the graph of an utterance and Ω_r be the graphs corresponding to individual words. We have now shown that $HAC(\Omega) = \mathbf{w} \approx \mathbf{W}\,\mathbf{h}$ where $\mathbf{W} = [\mathbf{w}_1 \ldots \mathbf{w}_R]$ and $\mathbf{h}$ is an *R*-dimensional indicator column vector containing a 1 in the *i*-th position if word *i* occurs in the utterance. When used to model speech, we will only dispose of a noisy observation of $HAC(\Omega)$, called $\mathbf{v}$. However, if *T* utterances are available, we can stack them as $\mathbf{V} = [\mathbf{v}_1 \ldots \mathbf{v}_T]$ and likewise $\mathbf{H} = [\mathbf{h}_1 \ldots \mathbf{h}_T]$, so

$$\mathbf{V} \approx \mathbf{W}\,\mathbf{H} \tag{2}$$

2.3. Non-negative matrix factorisation

In [2] and [8] it was shown that we can train (without supervision) the words a collection of *T* utterances is composed of by computing the factorization (2). Each discovered word is then characterized by a histogram vector or a column in $\mathbf{W}$. This mathematical technique is known as non-negative matrix factorization [3]. Given a matrix $\mathbf{V}$ of size $N \times T$ of non-negative real numbers, approximate non-negative matrix factorization rewrites this matrix as the product of factors $\mathbf{W}$ and $\mathbf{H}$ of size $N \times R$ and $R \times T$ respectively and containing non-negative real numbers. The metric to measure the (dis)similarity of the left hand side and the right hand side of (2) is their divergence:

$$D(\mathbf{V} \,\|\, \mathbf{W}\mathbf{H}) = \sum_{i,j} \left(\mathbf{V}_{ij} \log \frac{\mathbf{V}_{ij}}{[\mathbf{W}\mathbf{H}]_{ij}} - \mathbf{V}_{ij} + [\mathbf{W}\mathbf{H}]_{ij} \right) \tag{3}$$

Divergence is preferred over e.g. the Frobenius norm since unobserved data ($\mathbf{V}_{ij}$=0) will contribute weakly to dissimilarity. In this paper, the iterative multiplicative updates for minimizing (3) as outlined in [3] and [10] are used. The related convergence issues will be discussed in section 3.2.

Notice that in (2), $\mathbf{W}$ and $\mathbf{H}$ are not unique. For instance, scaling and permutation of the columns of $\mathbf{W}$ or the rows of $\mathbf{H}$ lead to equivalent solutions. Therefore, the columns of $\mathbf{W}$ will be normalized to sum to 1. Consequently, $\mathbf{H}$ will need to be scaled accordingly and we will refer to the values of $\mathbf{H}$ as *model activations*. The permutation problem will not hamper the analysis below.

2.4. Supervised training and recognition

The NMF-method for unsupervised discovery of recurring acoustic patterns (words in section 2.2), can be extended to include supervision. If the *t*-th utterance is known to contain the *m*-th word G_{mt} times, form the $M \times T$ matrix $\mathbf{G}$ with G_{mt} in its *m*-th row and *t*-th column and compute the NMF:

$$\begin{bmatrix} \mathbf{G} \\ \mathbf{V} \end{bmatrix} \approx \begin{bmatrix} \mathbf{W}_g \\ \mathbf{W}_v \end{bmatrix} \mathbf{H} \tag{4}$$

which expresses that word identity needs to be explained jointly with the acoustic data by common model activations. In case $R = M$ we expect $\mathbf{W}_g$ to be a diagonal matrix (within a permutation, see the previous section). We will set $R > M$ in this work, such that not all models need to explain a word.

After supervised training, i.e. computing factorisation (4), recognition on unseen data is achieved by first computing $\hat{\mathbf{H}}$ in $\mathbf{V} \approx \mathbf{W}_v\,\hat{\mathbf{H}}$ using only the acoustic data and with fixed $\mathbf{W}_v$. The presence of words or their *activation* in the test utterances is subsequently estimated as:

$$\hat{\mathbf{G}} = \mathbf{W}_g \hat{\mathbf{H}} \tag{5}$$

Notice an important difference with an HMM-based speech recogniser: each column of the matrix $\hat{\mathbf{G}}$ will reveal to which extent each trained word is present in the corresponding utterance. However, it will say nothing about the order in which the words occur in the utterance. True decoding still requires further research (see also section 4).

2.5. Integrating asynchronous information streams

If a single utterance can generate multiple graphs, the model of section 2.2 will hold for each of them with a common indicator vector $\mathbf{h}$. Hence, if there are *Q* streams of acoustic information that can each be represented in a graph per utterance, the $HAC(.)$-operator can be applied to each stream *q* to yield the histogram matrices $\mathbf{V}_q$. Training and recognition are still described by equations (4) and (5) with $\mathbf{V}$ replaced by

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_1 \\ \vdots \\ \mathbf{V}_Q \end{bmatrix} \tag{6}$$

Since the graph node times are not used, there are virtually no limitations to the nature and time scale of the acoustic events in the different streams. They can for instance be frame synchronous, segmental, defined at a fine time scale or even at the utterance level like the matrix $\mathbf{G}$.

3. Experiments

To date, the method has been applied to small vocabulary word discovery and recognition tasks only. In this paper, we report on results obtained on the TIDIGITS corpus.

Admittedly, research on more complex tasks will be required to explore the limitations of the method. The information sources that will be integrated are measured at the segmental and at the frame or feature vector level. They are respectively represented by an automatically generated phone lattice and by vector quantised cepstral features and their first and second order time derivatives.

The training speech data are taken from TI-Digits which contains recordings of 55 male and 57 female US-American adults, downsampled to 16 kHz, totaling 6159 connected digit sequences of length 1 through 7. The test set contains 6214 digit strings of length 1 through 7 uttered by a disjoint set of 56 male and 57 female speakers.

3.1. Stream integration

The phone lattices contain 43 different phone symbols, plus the silence symbol. The acoustic-phonetic decoding makes use of a state-of-the-art acoustic model that consists of an HMM with cross-word context-dependent tied Gaussian mixtures, trained on the Wall Street Journal (WSJ0 plus WSJ1) database. For the phone transition model, a bigram is estimated on the same data. With this construction, we ensure that the phone lattices are not biased towards the task and that similar performance can be obtained on other vocabularies. For all utterances of the training database, a phone lattice is generated and the acoustic co-occurrence histogram at lag $\tau = 1$ is computed, resulting in a histogram matrix $\mathbf{V}_1$ of size 1936×6159 ($1936 = 44^2$).

For the frame-synchronous information, 12 MFCC's plus log-energy are computed at a 100 Hz frame rate. A codebook of respectively 150, 150 and 100 for static, velocity and acceleration parameters is trained on the training set of the TIDIGITS corpus using the K-means algorithm. All training utterances are then processed resulting in a hard VQ-label for static, velocity and acceleration features per 10 ms analysis frame. Each of these three streams can be represented as a "chain" graph (special case of Figure 2 with only one arc between two successive nodes). For a given τ, the resulting histogram matrices have size 22500×6159 for the static and the velocity stream and 10000×6159 for the acceleration stream, which are all stacked to yield a (very sparse) 55000×6159 matrix. In the current experiment, we consider co-occurrence lags $\tau = 2$ ($\mathbf{V}_2$), $\tau = 5$ ($\mathbf{V}_3$) and $\tau = 10$ ($\mathbf{V}_4$). Finally, the VQ histogram counts are divided by a fixed constant (100) such that all $\mathbf{V}_q$ have roughly the same weight in the cost function equation (3).

In the sequel, we will investigate how the segmental information $\mathbf{V}_1$ and the three sources of frame-synchronous information $\mathbf{V}_2$ through $\mathbf{V}_4$ complement each other to yield better recognition results. For each combination of information sources, an NMF is computed with $R = 12$ (model order selection was discussed in [2]).

Since the current method is not capable of ordering the recognized words in time, we apply a non-standard evaluation method. For the t-th test utterance which is known to contain K_t different words, we take the K_t largest values of the corresponding t-th column of the estimated word activation $\hat{\mathbf{G}}$ in equation (5). For instance, for a test utterance "nine one nine", we will respond with the 2 best scoring words. Hence, in terms of difficulty, the resulting task is comparable (but not equivalent) to length-constrained digit string recognition. The number of correct words in the list of K_t digits is counted over the test corpus and divided by $\sum_{t=1}^{T} K_t$.

to form the word error rate (WER) shown in the rightmost column of Table 1. In the columns on the left, the included knowledge sources are marked with a "x". The first few rows show the effect of including frame-synchronous acoustic co-occurrence information. Short-term acoustic co-occurrence alone ($\tau = 2$) seems too local to model word-level segments (WER = 6.09 %). Including longer-span co-occurrence histograms has a positive impact on the WER, although the knowledge sources at different τ are obviously correlated. Using segmental information only, the WER is 4.84 %. Combining the sources which contain related information but at different time scales, reduces the error rate to 2.50 %.

Table 1: *digit recognition word error rates (WER) when information sources are combined. Included sources are marked with X.*

segmental	frame synchronous			
$\mathbf{V}_1$	$\mathbf{V}_2$	$\mathbf{V}_3$	$\mathbf{V}_4$	
phone	VQ	VQ	VQ	
$\tau = 1$	$\tau = 2$	$\tau = 5$	$\tau = 10$	WER (%)
	x			6.09
		x		4.22
			x	4.17
	x	x		5.17
	x		x	4.99
		x	x	4.23
	x	x	x	4.88
x				4.84
x	x			3.58
x		x		3.26
x			x	3.29
x	x	x		2.80
x	x		x	2.73
x		x	x	2.57
x	x	x	x	2.50

3.2. Initialisation and convergence analysis

NMF algorithms suffer from problematic convergence behaviour [10]. In the present experiments, we exploited the supervision knowledge in the initialisation of the matrix factors. Assuming that the M (=11 here) leftmost columns of $\mathbf{W}$ each model a single word, the arguments of section 2.2 also allow to set the entries in the top M rows of $\mathbf{H}$ corresponding to the unused words in a particular training utterance to zero. The other entries of $\mathbf{H}$ and all entries of $\mathbf{W}$ are initialized to random strictly positive numbers and 200 multiplicative update iterations are performed. In the experiments of section 3.1, this process was repeated 10 times and the factorisation with the smallest divergence is retained.

If the factorisation outcomes are not identical on each trial, this results in a source of stochastic variation in the observed word error rate, which is now analysed. In fact, HMMs suffer from a similar phenomenon. For the case where the data consist of $\mathbf{V}_1$ and $\mathbf{V}_3$, we performed 180 trials and computed the WER for each outcome. A scatter plot of the observed WER on the test set versus the divergence value at the last iteration on the training set is given in Figure 3. None of the trials leads to a dramatically poor error rate. Most solutions the algorithm converges to are fairly equivalent, though a considerable variation in WER is observed. The lowest divergence on the training set is not a guarantee for the lowest WER on the test set. Like with HMMs, cross validation is required. Notice that the estimation of $\hat{\mathbf{H}}$ with fixed $\mathbf{W}_v$ is a convex problem, so that a similar variations are not observed due to the matrix factorisation required for recognition.

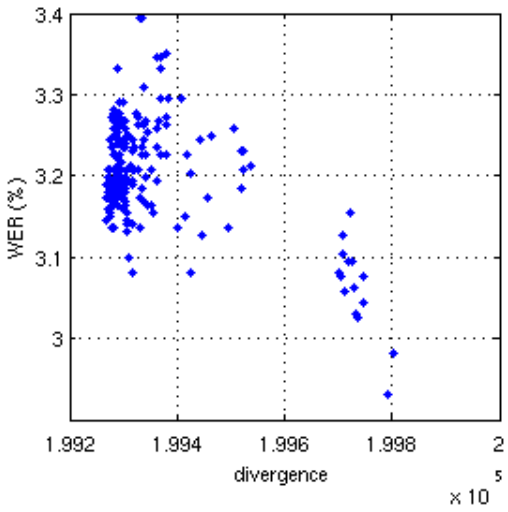

Figure 3: *word error rate as a function of the divergence between data and model during training.*

4. Discussion

The presented training and recognition HAC framework possesses some properties that are markedly different from the wide-spread HMM framework. First, it can work in supervised (this paper) and unsupervised mode [2], [8]. Second, it can easily integrate correlated information as well as asynchronous information streams. Using correlated information streams in HMMs typically requires feature dimension reduction techniques such as linear discriminant analysis. Third, time and sequence are only weakly represented by acoustic event co-occurrence. Much like a word-level bigram is only a weak linguistic representation of grammar, the acoustic event co-occurrence does represent order, but many different sequences can lead to the same or similar histograms. Especially, repetitions of a pattern or cyclic permutations lead to very similar histograms and are hard to distinguish. Fourth, the high-dimensional shift-invariant speech representation does not require segmentation of the audio in words during training or recognition. This obviously has a disadvantage, in that it is not able to locate and order the recognized words within the analysis window (an utterance in this paper). Computing the histograms over a sliding window at least confines the recognised words to this window as was already shown in [2], but this technique needs future research. The absence of a segmentation step also has the advantage that related non-contiguous expressions such as separable verbs in Germanic languages can easily be related in one pattern. It also means that no time warping and dynamic programming are required, unlike in other pattern discovery algorithms [11].

5. Conclusions

In this paper, we have presented an NMF-based method for supervised training of word models for speech recognition. The HAC method is capable of successfully integrating correlated information as well as information at different time scales, which was illustrated with the integration of utterance level information (**G**), segmental information ($\mathbf{V}_1$) and frame-synchronous spectral information ($\mathbf{V}_2$ through $\mathbf{V}_4$). Evidently, other choices of acoustic events could be made.

Though the initial results of this work are very encouraging, future research will need to address ways to generate segmentations in terms of the learned models as well as to explore the accuracy limits of HAC models as the vocabulary increases.

6. Acknowledgements

This research was funded by the European Commission under contract FP6-034362 (ACORNS).

7. References

[1] J. N. Gowdy, A. Subramanya, C. Bartels and J. Bilmes, "DBN-Based Multi-Stream Models for Audio-Visual Speech Recognition", *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 1, pages 993--996, May 2004. Montreal, Canada

[2] V. Stouten, K. Demuynck, K. and H. Van hamme, "Discovering Phone Patterns in Spoken Utterances by Non-negative Matrix Factorisation." *IEEE Signal Processing Letters*, pages 131—134, Vol. 15, 2008.

[3] D. Lee, and H. Seung, "Algorithms for non-negative matrix factorization," *Advances in Neural Information Processing Systems*, vol. 13, pp. 556–562, 2001.

[4] P. Smaragdis, C. Judith and J. C. Brown, "Non-Negative Matrix Factorization for Polyphonic Music Transcription", *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, October 19-22, 2003, New Paltz, NY, pages 177-180

[5] M. N. Schmidt and M. Mørup, "Sparse Non-negative Matrix Factor 2D-Deconvolution for Automatic Transcription of Polyphonic Music", Technical University of Denmark, 2006

[6] B. Raj, R. Singh, and P. Smaragdis, "Recognizing speech from simultaneous speakers", in *Proc. Interspeech 2005*, Lisbon, Portugal, Sept. 4-8, 2005, pages 3317-3320

[7] S. Behnke, "Discovering hierarchical speech features using convolutional non-negative matrix factorization", in *Proc. Of International Joint Conference on Neural Networks (IJCNN'03)*, Portland, OR, USA, vol. 4, pp. 2758-2763

[8] V. Stouten, K. Demuynck, and H. Van hamme, "Automatically Learning the Units of Speech by Non-negative Matrix Factorisation." In *Proc. Interspeech*, pages 1937-1940, Antwerp, Belgium, August 2007.

[9] F. Wessel, R. Schluter, K. Macherey and H. Ney, "Confidence measures for large vocabulary continuous speech recognition", *IEEE Transactions on Speech and Audio Processing*, Volume 9, Issue 3, Mar 2001 pages 288 – 298

[10] B. W. Berry, M. Browne, A. N. Langville, V. P. Pauca and R. J. Plemmons, "Algorithms and applications for approximate nonnegative matrix factorizations", *Computational Statistics & Data Analysis* 52(1), (2007), pp. 155-173

[11] A. S. Park and J. R. Glass, "Unsupervised Pattern Discovery in Speech", *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 16, No. 1, January 2008, pages 186 – 197.