WEAKLY SUPERVISED KEYWORD LEARNING USING SPARSE REPRESENTATIONS OF SPEECH

Joris Driesen, Jort Gemmeke, Hugo Van hamme

Department Electrical Engineering-ESAT, Katholieke Universiteit Leuven
Kasteelpark Arenberg 10, Bus 2441, B-3001 Leuven Belgium
{joris.driesen,jort.gemmeke,hugo.vanhamme}@esat.kuleuven.be

ABSTRACT

When applied to speech, Non-negative Matrix Factorization is capable of learning a small vocabulary of words, foregoing any prior linguistic knowledge. This makes it adequate for small-scale speech applications where flexibility is of the utmost importance, e.g. assistive technology for the speech impaired. However, its performance depends on the way its inputs are represented. We propose the use of exemplar-based sparse representations of speech, and explore the influence of some of these representation's basic parameters, such as the total number of exemplars considered and the sparseness imposed on them. We show that the resulting learning performance compares favorably with those of previously proposed approaches. ***Index Terms***— Vocabulary Acquisition, Non-negative Matrix Factorization, Sparseness, Lasso, Exemplars

1. INTRODUCTION

As technology more and more finds its way into our daily lifes, in the form of handheld devices, GPS, home automation, etc., there is an ever increasing need to interface with it in an easy and natural way. Speech processing can play a central role in meeting these demands, if it can be made accurate and flexible enough to suit any user's needs. And it's the flexibility that is at present in some contexts insufficient. To recognize words, an Automatic Speech Recognition (ASR) system must have a model for each. In the most straightforward case, these models are defined in advance, describing their 'standard' acoustic realizations. Such generic models do not cover for all the possible alterations that may occur in word pronunciations.

There are specialized algorithms that can adapt the models in terms of speaker-related parameters such as age, gender, dialect, speaking rate, etc., e.g. [1], but these algorithms have their limits. Their adaptation can only alter low-level localized acoustic descriptions of the words, but not their phonemic identity: anything beyond variations to the standard pronunciation falls outside the scope of such techniques.

Because of this, speech recognition is for instance largely inaccessible for users with severely reduced speech capabilities. This is especially unfortunate for those whose speech pathology results from an illness that also affects the function of their upper limbs, limiting them in their use of classical human-machine interfaces, such as buttons and slides. Given their dependence on assistive technology, it is exactly this group of users that could benefit the most from voice interfaces.

There is thus need for a recognition system that automatically learns to recognize uttered words, without making any prior assumptions about their phonemic structure and their acoustic properties. Although this is not feasible for very large vocabularies, such a system can be very helpful, since a small vocabulary is already sufficient for many applications. The difficulty in such a system is to learn words when they are embedded in a sentence. Moreover, in the case of commands, not only the acoustics have to be learned, but also their physical significance.

The learning framework used in [2], which is based on Non-negative Matrix Factorization (NMF), is capable of doing all these things. A central question that presents itself, however, is how to represent the inputs of this method. In [2], Van hamme proposes the Histogram of Acoustic Co-occurrences (HAC), a method that essentially clusters and labels temporally localized parts of the signal, and accumulates the occurrences of label combinations at a certain time offset into a histogram. There are disadvantages to this method, however, that make it less tractable. For instance, since it is based on clustering and classifying very short segments of speech, typically no more than 10ms, without considering any temporal context, much of the variability in the speech signal is left for the NMF algorithm to deal with. Also, information is lost in the quantization of the continuous-valued speech segments. Last but not least, the HAC-representation contains a feature for *all possible label combinations*, giving them a tendency to be of a very high dimensionality.

In this paper, we propose to replace the histogram-based HAC representation by a sparse vector of coefficients which describe the input signal as a linear combination of *speech exemplars*, i.e. selected segments of real speech. Conceptually, the use of exemplars offers several advantages: since the exemplars have a longer duration, these representations are more robust against temporally fine-grained variations in speech. Also, since time information is already encoded in these coefficients, there is no need to consider combinations of them, limiting the dimensionality of the resulting vector representations. Finally, it has been shown previously that the

This research was funded by the IWT-SBO project ALADIN (contract 100049)

use of exemplars offers convenient ways of dealing with noise [3].

Applying this idea on a keyword learning task, we explore the influence of some of its parameters, such as the number of exemplars, and the sparsity of the linear approximations. We make a comparison of the resulting learning performance with that of HAC. This paper is organized as follows: in section 2, we explain the keyword learning framework. In section 3, we explain the derivation of HAC and the exemplar-based representations. Experiments in which both speech representations are applied to the learning framework are discussed in section 4. We finish with some concluding remarks and future perspectives in section 5.

2. KEYWORD DETECTION

NMF is an algorithm that, as its name implies, factorizes a non-negative $M \times N$ matrix $\mathbf{V}$, containing speech data, into a non-negative $M \times R$ matrix with word models, $\mathbf{W}$, and a non-negative $R \times N$ matrix $\mathbf{H}$, containing word activations: $\mathbf{V} \approx \mathbf{W} \cdot \mathbf{H}$. The solution can be found by minimizing a distance between $\mathbf{V}$ and its approximation $\mathbf{WH}$. There is a variety of distance measures that can be chosen, but in this paper we use the Kullbeck-Leibler divergence[1] $D_{KL}(\mathbf{V}||\mathbf{WH})$. The minimization of this divergence can be done iteratively, using the multiplicative updates presented in [4]. Because R is usually much lower than either M or N, this leads to a low-rank approximation of the data. NMF imposes a linear model on the data, describing each column of $\mathbf{V}$ as a weighted addition of R basic atoms, given by the columns of $\mathbf{W}$, while storing the weights of these additions in the columns of $\mathbf{H}$. NMF may thus discover latent structure in the data, if it is present.

In this work, where we want to apply NMF on speech, we introduce this structure through the method by which we convert the time signal into a non-negative vector. Concretely, if we have an utterance U_j, which consists of a sequence of n words $\{w_1, w_2, \ldots, w_n\}$, and we name the conversion operator $\psi(\cdot)$, we demand that

$$\mathbf{V}_{(:,\mathbf{j})} = \psi(U_j) \approx \psi(w_1) + \psi(w_2) + \ldots + \psi(w_n) \qquad (1)$$

in which $\mathbf{V}_{(:,\mathbf{j})}$ is the j'th column of the data matrix. Ideally, NMF should discover the vector representations of all the different words in the data $\psi(w_k)$, in which $k = 1..n$, as the basic atoms in $\mathbf{W}$. However, even if the data meets the above condition, this is far from guaranteed. The multiplicative updates by which NMF is solved do not necessarily lead to a global minimum. They merely minimize the divergence, finding a local optimum that depends on the random initialization of $\mathbf{W}$ and $\mathbf{H}$. Additional supervision is therefore required for NMF to robustly learn the word representations from speech. To this end, renaming the data matrix to $\mathbf{V_1}$, we append a matrix $\mathbf{V_0}$ to it, which is defined as

$$V_{0,ij} = \begin{cases} 1 & \text{if } U_j \text{ contains word } w_i \\ 0 & \text{otherwise} \end{cases}$$

[1] KL divergence is technically not a distance measure, since it is not symmetric

Put differently, it contains a row for each recognizable word, which indicates that word's presence in each utterance. An additional matrix $\mathbf{W_0}$ is also appended to $\mathbf{W}$ (henceforth called $\mathbf{W_1}$), and is defined as an identity matrix, to link every keyword to a column in $\mathbf{W}$. If R is larger than the number of recognizable words, $\mathbf{W_0}$ is padded with zero columns. The NMF factorization thus becomes:

$$\mathbf{V} = \begin{bmatrix} \mathbf{V_0} \\ \mathbf{V_1} \end{bmatrix} \approx \begin{bmatrix} \mathbf{W_0} \\ \mathbf{W_1} \end{bmatrix} \mathbf{H} = \mathbf{WH} \qquad (2)$$

The weights in $\mathbf{H}$ which map $\mathbf{W_0}$ onto $\mathbf{V_0}$, are the same as those that map $\mathbf{W_1}$ onto $\mathbf{V_1}$. Because of the way $\mathbf{W_0}$ is defined, this constitutes supervision, since it promotes solutions in which a single word $\psi(w_k)$ is assigned to each of the columns in $\mathbf{W_1}$. To ensure a sufficient influence of this supervision on the factorization of NMF, $\mathbf{V_0}$ is scaled with a constant factor such that the sum of its elements equals that of $\mathbf{V_1}$. One can regard this supervision as the addition of a regularization term to the cost function, penalizing deviations of $\mathbf{H}$ from its known optimal value:

$$(\mathbf{W}^*, \mathbf{H}^*) = \min_{(\mathbf{W},\mathbf{H})} \left(D_{KL}(\mathbf{V_1}||\mathbf{W_1H}) + D_{KL}(\mathbf{V_0}||\mathbf{W_0H})\right)$$

Applying NMF to a set of training data $[\mathbf{V_0^{trn}}^T \mathbf{V_1^{trn}}^T]^T$ with this supervision thus leads to word models in $\mathbf{W}$. Given a set of test data $\mathbf{V_1^{tst}}$, in which the corresponding meta-information $\mathbf{V_0^{tst}}$ is evidently not known, these models can be used to calculate the matrix $\mathbf{H^{tst}}$, for which holds $\mathbf{V_1^{tst}} \approx \mathbf{W_1 H^{tst}}$. The product of $\mathbf{W_0}$ with $\mathbf{H^{tst}}$ is a labelling matrix $\mathbf{A}$, that predicts the unobserved data $\mathbf{V_0^{tst}}$. The framework is depicted in figure 1.

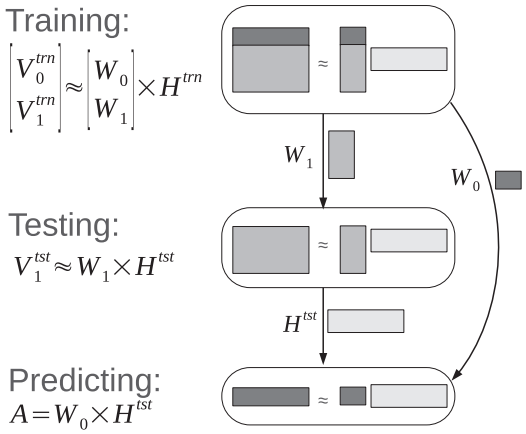

Fig. 1. *A schematic overview of the learning framework.*

3. REPRESENTATIONS OF SPEECH

The non-negative vectors $\mathbf{V}_{(:,\mathbf{j})}$ have to be representative for the original signal, and have to meet the condition put forward in (1). In a first step, the signal is framed, using time windows of 25ms, shifted in increments of 10ms. In each of

these frames, 22 MEL-coefficients, along with their first and second order differences are determined. The resulting 66-dimensional space containing these vectors is then reduced to 36 dimensions with MIDA (Mutual Information Discriminant Analysis), an enhanced variant of Linear Discriminant Analysis based on mutual information. For details, see [5].

3.1. Histogram of Acoustic Co-occurrences

To determine the Histogram of Acoustic Co-occurrences (HAC), all the frames in a large set of data are first clustered into C classes, using a standard algorithm like K-means clustering, e.g. [6]. Using the resulting set of cluster centers as a Vector Quantization (VQ) codebook, the frames in each input utterance are converted into a sequence of VQ-labels. One way to meet condition (1) is to construct a histogram, counting the number of times each label occurs in the utterance. However, all but the most fine-grained temporal structure of the utterance is lost in this operation. To preserve some of this information, it is better to count the number of occurrences of label *combinations* at a certain time offset τ. An example is given in figure 2 where label co-occurrences at an offset of 30ms (3 frames of 10ms) are shown. Histograms resulting from multiple values of τ can be concatenated, capturing even more time information in the final HAC-vector. This comes at a cost, however, since each additional value for τ raises the dimensionality by C^2.

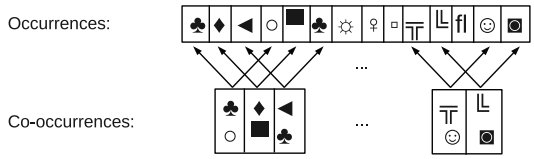

Fig. 2. *An example in which co-occurrences of symbols are derived with a time offset τ of 3 frames*

3.2. Exemplar-based Activations

As an alternative to HAC, we propose a method which uses the same spectro-temporal input as before, but operates on a broader time scale. P windows of T consecutive frames are taken from a large set of training utterances, stacked into vectors of length $T \cdot D$ for which holds $TD \ll P$, and placed as columns in a matrix $\mathbf{X}$. Here, D is the dimensionality of the MIDA features, in our case equal to 36. Each speech segment of T frames, taken from an utterance U_j at time t, is then stacked the same way, resulting in a vector $\mathbf{y_{t,j}}$ which is approximated by a linear weighted addition of the exemplars:

$$\mathbf{y_{t,j}} \approx \alpha^{(1)}_{t,j}\mathbf{X}_{(:,\mathbf{1})} + \alpha^{(2)}_{t,j}\mathbf{X}_{(:,\mathbf{2})} + \ldots + \alpha^{(P)}_{t,j}\mathbf{X}_{(:,\mathbf{P})} = \mathbf{X}\boldsymbol{\alpha_{t,j}} \tag{3}$$

As $||\mathbf{y_{t,j}} - \mathbf{X}\boldsymbol{\alpha_{t,j}}||_2^2$ is minimized, a sparseness constraint is imposed on $\boldsymbol{\alpha}$. To determine equation (3) in this paper, we have made use of LASSO (Least Absolute Shrinkage and Selection Operator), an off-the-shelf algorithm that allows strict control over the resulting sparsities by limiting the number of iterations, [7].

A sliding window of 10 frames is shifted over the signal in increments of 1 frame, and in each position an activation vector $\boldsymbol{\alpha_t}$ is calculated, with a pre-selected number of non-zero elements S. The vector representation of the speech signal is then created by summing these sparse activation vectors over time.

$$\mathbf{V}_{(:,\mathbf{j})} = \sum_t \boldsymbol{\alpha_{t,j}} \tag{4}$$

4. EXPERIMENTS

Both vector representations were created for real speech data. The database was recorded for the ACORNS project (Acquisition of Communication and RecogNition Skills), and is specifically designed to benchmark keyword learning techniques [8]. It consists of 13160 short, syntactically simple utterances, produced by 10 different *unimpaired* speakers. There are 50 different keywords, 1 to 4 of which occur in every utterance. These keywords are always embedded in a carrier sentence with unrelated terms. A training set is defined, containing 9821 randomly selected utterances. The testing set consists of the remaining 3272. The utterances of both train and testing set are vectorized. HAC is applied, as explained in section 3.1, with VQ codebook sizes ranging from 100 to 500. Time offset values τ of 20ms, 50ms and 90ms are combined, as was done in [9]. The sparse activation vectors from section 3.2 are derived using randomly selected sets of 1000, 5000 and 10000 exemplars to explore to what extent higher numbers of exemplars improve recognition accuracy. The sparsity of the activations is varied between 4 and 40, because in related exemplar-based work it has been shown that increasing the sparsity may improve recognition accuracy, even though this can lead to higher reconstruction errors (cf. [10]). The average reconstruction error in a single utterance is shown as an example in figure 3.

The learning framework from section 2 is then applied on all the data matrices. The number of columns in $\mathbf{W}$, R is chosen higher than the number of keywords at 75, assigning 25 columns to model inputs unrelated to keywords, as is recommended in [2]. Evaluation of the keyword prediction on each testing set is done as follows: if test utterance j contains K_j keywords, they are compared with the K_j largest elements in the corresponding column of the prediction matrix, $\mathbf{A}_{(:,\mathbf{j})}$. Each substitution encountered is counted as an error; Insertions and deletions are not possible in this comparison. The total error rate is:

$$\text{ER} = 100 \cdot \frac{\sum_{j=1}^{3272} \#substitutions}{\sum_{j=1}^{3272} K_j}\% \tag{5}$$

Although in a real application the number of keywords K_j may be unknown, this measure does allow a comparison between different kinds of input. To remove the element of randomness in the experimental results, caused by the initialization of $\mathbf{W}$ and $\mathbf{H}$ in NMF, each experiment was repeated 5 times and the results were averaged. The results are shown in table 1.

As can be observed in this table, there is clearly an optimal sparsity for the exemplar-based inputs in this learning

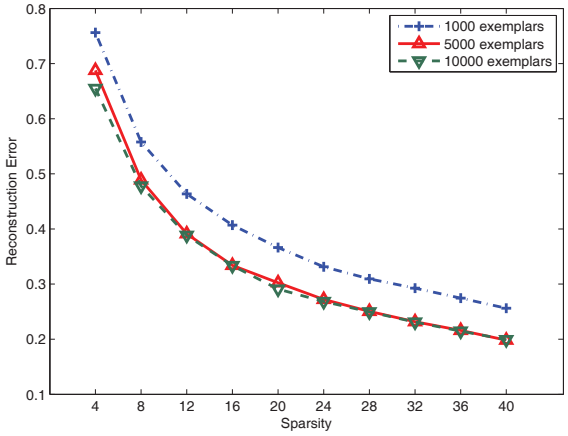

Fig. 3. *Reconstruction errors, averaged over 20 sparse representations, for different numbers of exemplars and different sparsities, using increasing numbers of exemplars.*

Exemplars				HAC	
P $\neq 0$	1000	5000	10000	C	
4	**6.73**	4.13	4.01	100	4.59
8	6.87	3.25	2.87	150	3.61
12	7.52	**3.24**	**2.50**	200	3.02
16	8.34	3.49	2.60	250	2.63
20	9.17	3.75	2.64	300	2.53
24	10.53	3.99	3.03	350	2.59
28	11.30	4.39	3.04	400	**2.49**
32	12.47	4.54	3.51	450	2.56
36	13.57	4.88	3.66	500	2.62
40	14.86	5.07	3.82		

Table 1. *The resulting unordered error rates, expressed in %. The best result for each experiment is printed in bold.*

task. At the same time, the average reconstruction error displayed in figure 3 does decrease as the number of non-zero elements increases. This matches the findings in [10] where it was shown that word identities are to a degree encoded in the $\boldsymbol{\alpha}$-vectors, which is lost if the sparsity becomes too low or too high. The use of more exemplars is also found to have a positive influence on the error rates, with the best results obtained with 10000 exemplars with 12 nonzero elements, resulting in a 2.50% ER. This yields a comparable performance as HAC features using a 400 dimensional codebook, which yields 2.49% ER.

For the HAC features, higher dimensional codebooks do not increase the performance, whereas for the exemplar-based representation it is likely larger numbers of exemplars can further reduce the error rates. It should also be noted there is a large disparity between the dimensionalities of their word models, i.e. 10000 and 480000 ($3 \cdot 400^2$) for exemplar-based and HAC features, respectively. These findings, together with the possibility of noise robustness outlined in [3], make the use of exemplar-based sparse representations a promising candidate for keyword learning in realistic settings.

5. CONCLUSION

In this work we proposed an exemplar-based sparse representation of speech as the basis for weakly supervised keyword learning. It was shown that the best results are obtained by imposing a stronger sparsity than the one leading to the optimal reconstruction error of the sparse representation, and that the more exemplars are used, the better the results. Furthermore, it was found that the proposed approach compares favourably to the histogram-based features used in earlier work.

Future work will include exploring to what extent even larger numbers of exemplars can further improve the performance, investigate the use of different approaches to obtain a sparse representation and investigate the noise robustness of both exemplar-based and histogram-based representations.

6. REFERENCES

[1] Roland Kuhn, Jean-Claude Junqua, Patrick Nguyen, and Nancy Niedzielski, "Rapid speaker adaptation in eigenvoice space," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 6, 2000.

[2] H. Van hamme, "Hac-models: a novel approach to continuous speech recognition," in *Interspeech 2008*, Brisbane, Australia, 2008.

[3] Jort F. Gemmeke, Tuomas Virtanen, and Antti Hurmalainen, "Exemplar-based sparse representations for noise robust automatic speech recognition," *IEEE Transactions on Audio, Speech and Language processing*, vol. 19, no. 7, pp. 2067–2080, 2011.

[4] D.D. Lee and H.S. Seung, "Learning the parts of objects by nonnegative matrix factorization," *Nature*, , no. 401, pp. 788–791, 1999.

[5] Kris Demuynck, *Extracting, Modelling and Combining Information in Speech Recognition*, Ph.D. thesis, K.U.Leuven, ESAT, feb 2001.

[6] S. P. Lloyd, "Least square quantization in pcm," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137.

[7] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani, "Least angle regression," *Annals of Statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[8] "Acquisition of communication and recognition skills," http://www.acorns-project.org/, 2006–2009.

[9] Joris Driesen, Louis ten Bosch, and Hugo Van hamme, "Adaptive non-negative matrix factorization in a computational model of language acquisition," in *Proc. Interspeech 2009*, Brighton, UK, 2009.

[10] Qun Feng Tan, Panayiotis G. Georgiou, and Shrikanth S. Narayanan, "Enhanced sparse imputation techniques for a robust speech recognition front-end," *IEEE Transactions on Audio, Speech and Language Processing*, to appear.