Mapping Sparse Representation to State Likelihoods in Noise-Robust Automatic Speech Recognition

Katariina Mahkonen.[1], *Antti Hurmalainen.*[1], *Tuomas Virtanen.*[1], *Jort Gemmeke.*[2]

[1]Department of Signal Processing, Tampere University of Technology, Finland
[2]Centre for Language and Speech Technology, Radboud University Nijmegen, The Netherlands

katariina.mahkonen@tut.fi, antti.hurmalainen@tut.fi,
tuomas.virtanen@tut.fi, jgemmeke@amadana.nl

Abstract

This paper proposes learning-based methods for mapping a sparse representation of noisy speech to state likelihoods in an automatic speech recognition system. We represent speech as a sparse linear combination of exemplars extracted from training data. The weights of exemplars are mapped to speech state likelihoods using Ordinary Least Squares (OLS) and Partial Least Squares (PLS) regression. Recognition experiments are conducted using the CHiME noisy speech database. According to the results, both algorithms can be successfully used for training the mapping. We achieve improvements over the previous binary labeling system, and recognition scores close to 70% at -6 dB SNR.

Index Terms: automatic speech recognition, sparse representations, exemplar-based, regression

1. Introduction

There is a general agreement in the speech community about the need for novel approaches for improving Automatic Speech Recognition (ASR) in adverse conditions [1]. Recently, there has been a renewed interest in non-parametric ASR methods as an alternative for Gaussian Mixture Models (GMMs) [2, 3, 4]. In particular, methods that rely on representing speech as a linear combination of a small set of dictionary atoms have been shown to offer higher classification accuracy [2] and better noise robustness [3].

These methods work by first finding the sparsest possible linear combination of predefined atoms that describe the observed speech spectra. With each atom associated with a speech class [2, 4] or Hidden Markov Model (HMM) states [3], decoding is done by using the weights of atoms directly as evidence of the observed speech likelihoods. In Sivaram et al. [4], the phone classification was done by training a neural network to map the weights of atoms directly to phoneme classes.

In this paper, we employ the exemplar-based speech representation described in [3], where the dictionary atoms are spectrograms extracted from training data. The method has the advantage that noisy speech can be represented as a linear combination of speech and noise exemplars, allowing the method to obtain accurate sparse representations of the speech atoms even in the presence of corrupting background noise.

The recognition in [3] was done using state labels associated to the exemplars. This *canonical transcription* of the exemplars was obtained through forced alignment with a conventional GMM-based recognizer. A downside of using the canonical state association is that the sparse representation is formed of exemplars that represent the underlying speech states of the observed speech, i.e. the method is restricted to using exemplars that are realizations of speech. Furthermore, the use of canonical state association of the exemplars is not optimal for the recognition purpose. It may also require cumbersome handling of silence states, because those are not well represented as a linear combination of exemplars [3].

To circumvent these issues, we propose in this paper to *learn* the mapping between exemplar activations and state likelihoods using regression methods. Using Ordinary Least Squares (OLS) and Partial Least Squares (PLS) regression models, we show that learning the mapping between exemplar activations and likelihoods can handle phonetic ambiguity and labeling errors of dictionary exemplars, especially when enough training material is available.

The rest of the paper is organized as follows. In Section 2 we introduce our noisy speech representation model, describe how we retrieve the linear combination of exemplars used to represent speech and explain how the exemplar activations are used for speech recognition. In Section 3 we describe the two regression models used for mapping exemplar activations to speech state likelihoods. The experimental setup using the CHiME database is presented in Section 4. Results and discussion follow in Section 5, and conclusions in Section 6.

2. Exemplar-based representation of speech

The framework we use for robust speech recognition is shown in Figure 1. There are three steps to be done offline, namely dictionary building, training of conversion matrices and HMM training. Other phases of recognition are to be done with fixed system parameters.

We represent noisy speech using magnitudes within Mel-frequency bands. Magnitudes from T consecutive frames are concatenated to form a feature vector $\mathbf{y}$. This feature vector is then modeled as a sparse linear combination of weighted exemplar vectors $\mathbf{a}_m$, $m = 1...M$ from a dictionary matrix $\mathbf{A}$. As an equation this is:

$$\mathbf{y} \approx \sum_{m=1}^{M} \mathbf{a}_m x_m = \mathbf{A}\mathbf{x} \ , \qquad (1)$$

where x_m is a non-negative weight or activation of the m:th exemplar, and $\mathbf{x}$ is a vector containing all the activations. $\mathbf{A}$ holds in total M exemplars from both speech and noise, which allows representing noisy speech. The details of the representation are described in [3].

Activation values are obtained with a Non-negative Matrix Factorization (NMF) algorithm as in [3]. The algorithm min-

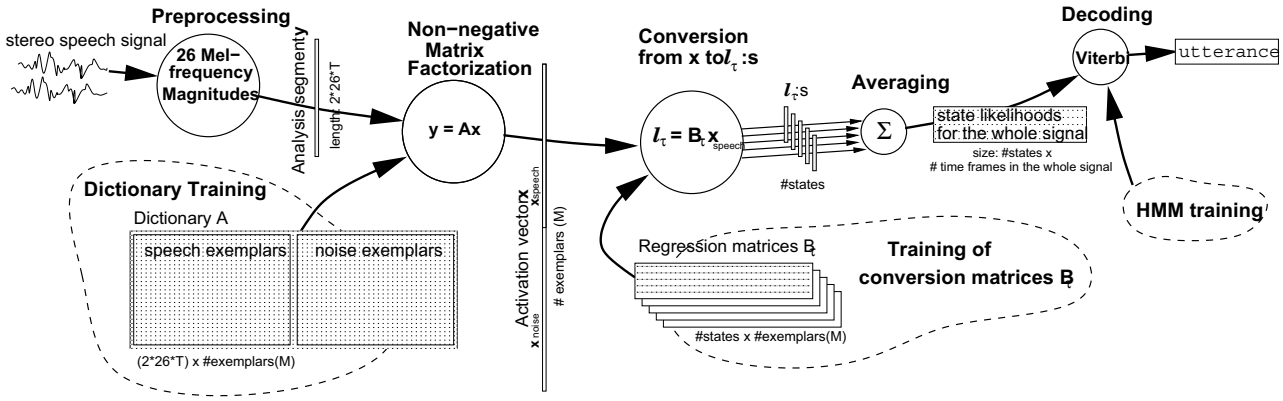

Figure 1: Speech recognition framework.

imizes the Kullback-Leibler divergence between the observations and the model plus L_1 norm sparsity promoting penalty using multiplicative update rules. The system processes long utterances by applying the sparse representation in overlapping, fixed-length segments, whose length is T frames. Shift of one frame is used between overlapping segments.

Our system does speech recognition by using a set of states, which are conceptually similar to the states in conventional HMM-based recognizers. However, in our system the likelihoods of the states are obtained differently. In the baseline system [3], the activations in $\mathbf{x}$ are converted to word state likelihoods by using labeled state lists of dictionary speech exemplars, which are obtained by forced alignment. Element x_m from an activation vector $\mathbf{x}$ is copied as a likelihood value to state likelihood vectors $l_\tau, \tau = 1...T$, which account to T frames inside the analysis segment. x_m is copied into l_τ for the state that is found as the τ:th element of $\mathbf{a}_m$:s state list.

In the end, the overlapping state likelihood vectors for each signal frame from consecutive analysis segments are averaged and combined to make up a state likelihood matrix $\mathcal{L}$ for the whole utterance. From $\mathcal{L}$, the most probable state sequence is tracked with a Viterbi decoder.

3. Proposed mapping from activations to state likelihoods

The goal of the proposed method is to find a mapping from exemplar activation vector $\mathbf{x}$ to state likelihood vector $\boldsymbol{l}_\tau$ in each frame $\tau = 1...T$ of a speech segment. For simplicity, we restrict ourselves to linear mappings. The mapping is given as

$$\boldsymbol{l}_\tau = \mathbf{B}_\tau \mathbf{x} \ , \tag{2}$$

where $\mathbf{B}_\tau$ is the mapping matrix for likelihoods in frame τ. The mapping is found by using training data consisting of activations and corresponding target state likelihood vectors, which are described in more detain in Section 3.1.

In our case the input space is very high dimensional, which makes the use of conventional methods, such as linear discriminant analysis, problematic. There are methods such as regularized discriminant analysis and shrinkage discriminant analysis that can be used with high-dimensional inputs. In this study, however, we explored two regression algorithms for the mapping, namely Ordinary Least Squares (OLS) and Partial Least Squares (PLS) [6]. PLS was chosen since it is known to produce good results in cases where the input data has a large number of dimensions that are highly collinear.

3.1. Training the regression matrices

In our case, the input space of regression matrices is of size 5000. It consists of truncated training data activation vectors $\mathbf{x}_S$, which only have the activations that correspond to speech exemplars of the dictionary $\mathbf{A}$. The target space consists of 250 states. Each training data segment has a labeled length T sequence of forced alignment target states obtained from its canonical transcription. Target vectors $\mathbf{l}_\tau, \tau = 1...T$ for the training are binary, having value 1 for the state that is labeled for the τ:th frame in the said training segment and value zero for all the other states. All the truncated training data activation vectors are collected into columns of matrix $\mathbf{X}$, and all the training data target vectors are gathered into columns of matrices $\mathbf{L}_\tau, \tau = 1...T$. The regression matrix training with the above explained input and output data is then performed with OLS and PLS algorithms.

3.2. Ordinary Least Squares

The *Ordinary Least Squares* (OLS) method finds a linear model for mapping the input space to the output space so that minimizes the total L_2 error of the model and target values. The solution of OLS regression with our variables becomes:

$$\mathbf{B}_\tau = \mathbf{L}_\tau \mathbf{X}^T (\mathbf{X}\mathbf{X}^T)^{-1} \ . \tag{3}$$

The problem with the OLS formula above is that the inverse of a matrix $\mathbf{X}\mathbf{X}^T$ must be calculated, which may be singular or nearly singular and thus infeasible. To avoid such a situation, Tikhonov regularization with $\mathbf{\Gamma} = \alpha\mathbf{I}$ will be used to stabilize the inverse matrix, changing the formula to

$$\mathbf{B}_\tau = \mathbf{L}_\tau \mathbf{X}^T (\mathbf{X}\mathbf{X}^T + \alpha\mathbf{I})^{-1} \ . \tag{4}$$

3.3. Partial Least Squares

The *Partial Least Squares* (PLS) method [6], also called Projection to Latent Structures, is a regression method designed for input data with high number of dimensions and with high collinearity. PLS does not use the input vectors as such, but constructs another set of basis vectors to do the linear regression in a new space. PLS represents the input and output data as matrix decompositions: $\mathbf{X} = \mathbf{V}_X \mathbf{S}_X$ and $\mathbf{L}_\tau = \mathbf{V}_{L\tau} \mathbf{S}_{L\tau}$. $\mathbf{V}_X$ and $\mathbf{V}_{L\tau}$ hold the new spanning vectors for input and target data, respectively. $\mathbf{S}_X$ and $\mathbf{S}_{L\tau}$ give the coordinate values, often called scores, of the input and target data with respect to their new coordinate axes. The number of spanning vectors used

to construct a new space is a dimensionality parameter for PLS, and it determines the rank of the model.

After iteratively rotating the spanning vectors of the new bases, $\mathbf{V}_X$ and $\mathbf{V}_{L\tau}$, PLS finds such directions, that the score matrices $\mathbf{S}_X$ and $\mathbf{S}_{L\tau}$ are as identical as possible. Then the conversion $\mathbf{D}$ from $\mathbf{S}_X$ to $\mathbf{S}_{L\tau}$,and from that, the conversion matrix $\mathbf{B}_\tau$ for the original space can be obtained by

$$\begin{aligned} \mathbf{D}_\tau &= \mathbf{S}_{L\tau}\mathbf{S}_X^T(\mathbf{S}_X\mathbf{S}_X^T)^{-1} \quad \text{and} \\ \mathbf{B}_\tau &= \mathbf{V}_{L\tau}\mathbf{D}\mathbf{V}_X^T(\mathbf{V}_X\mathbf{V}_X^T)^{-1} \ . \end{aligned}$$

In this work, PLS has been implemented with Statistically Inspired Modification of PLS, namely the SIMPLS-algorithm [7].

4. Experiments

4.1. Experimental setup

To compare the recognition quality of our previous state label based mapping and the mapping with learned regression matrices, we used the CHiME challenge database [8]. CHiME is based on the GRID corpus, where 34 speakers read aloud simple command sentences, consisting of linear grammar and a vocabulary of 51 words [9]. The GRID sentence structure is *verb-color-preposition-letter-digit-adverb*. There are 25 different letters and 10 digits. Other classes have four word options each. In CHiME database, the utterances are reverberated with a room response, and then mixed into stereo background noise sampled from a real living room. The task is to recognize words from 'letter' and 'digit' classes in 600 test utterances at six SNR levels: +9, +6, +3, 0, -3 and -6 dB. Test score is the number of correctly recognized keywords in the 1200 word instances at each SNR.

In our test setup, we used features consisting of 26 Mel-scale spectral magnitudes for each stereo channel, sampled at 16 kHz. Frame length was 25 ms and frame shift 10 ms. Segment lengths of $T = 10$, 20 and 30 consecutive frames were used, thus the total length of a concatenated segment vector was $2 * 26 * T$ (520–1560).

4.2. Dictionaries and factorization

Two different speech dictionary types were generated. For *speaker-dependent* dictionaries, 60% of the noiseless training set of each speaker was converted into partially overlapping exemplar-segments by going through the set with a random shift of 4–8 frames. The full dictionaries of approximately 10000-17000 exemplars were reduced to 5000 exemplars for each speaker / segment length combination, while maximising the flatness of included word distribution. In addition, a *speaker-independent* speech dictionary of 5000 exemplars was generated similarly for each segment length by combining 147–148 exemplars from each speaker, again with an attempt for maximally flat coverage of words and speakers. The remaining 40% of training utterances were used for learning the regression matrices. Speaker-dependent dictionaries were trained with utterances of the same speaker. Independent training used the combined utterances of all speakers.

In the factorization phase, we extracted a noise dictionary of 5000 exemplars for each utterance by picking partially overlapping pure noise segments from the immediate neighborhood of the utterance to be recognized. Speech and noise dictionaries were combined, and then re-weighted together to equal Euclidean norms over Mel bands and exemplars. The same band weights were applied to the utterance features. Factorization was performed with 300 NMF iterations, either in double precision CPU or single precision GPU computing. The difference between these was found negligible.

4.3. Decoding

The activations were converted into state likelihoods by using the label- and regression-based methods explained in sections 3.2 and 3.3. For the Tichonov regularization in OLS we used values 10^{-2}, 10^{-7} and 10^{-12}. PLS-regression was tested with dimensionalities 500, 650 and 800.

Finally, the likelihood matrices were decoded using a modified HVite binary from the HTK toolkit. Apart from the externally calculated state likelihoods, the original CHiME models were used 'as is' in decoding. Scoring of letter and digit keywords was performed with the standard CHiME scripts.

5. Results and discussion

The results of our recognition experiments are summarized in Table 1. Pane (a) shows the results for speaker-independent recognition, and pane (b) for speaker-dependent dictionaries. For each segment length T we show recognition rates for the previous binary label system, OLS regression and PLS regression. In addition, the baseline rates from CHiME documentation are shown on the topmost row [8]. The baseline system is standard HMM-based recognition using speaker-dependent GMMs. Similar speaker-independent baseline results were not available. For regression results, OLS regularization parameter 10^{-2} and PLS dimension 800 were selected. In OLS, the differences between parameters were minimal. In PLS, 800 dimensions yielded the best results with very few exceptions. The rate increased approximately by 0–1% (absolute) for each 150-dimension step. Especially for segment lengths 20 and 30, it is possible that even higher values could improve the results further.

In speaker-independent recognition, we notice that regression provides improvements of 4.3–14.1% (absolute) in results over binary label. In general, a relative reduction of approximately 20% is present in the error rates. The likely reasons for this are twofold. First, the speaker-independent dictionaries are fairly small for this task. 5000 exemplars could suffice for covering the phonetic variation within the set. However, in the word-based labelling system, some speaker-state combinations may be underrepresented, thus similar phonetic features with different word labels may get activated. Trained regression matrices manage to overcome this problem by activating several potential states at once. The second reason for the success of regression in this case is that the matrices have been trained from combined utterances of all speakers. In other words, there is 34 times more training data than in the speaker-dependent case. Possibly due to this abundance of training data, OLS yields the highest recognition rate of the tested methods.

Overall, speaker-independent recognition does not seem to perform very well. It should be noted, though, that the baseline scores were obtained using speaker-dependent acoustic models. In CHiME data, a lot of the background noise consists of people speaking in the living room, or speech coming from television. Therefore a speaker-independent model will easily pick up inaccurate speech segments from these external sources.

The overall results of speaker-dependent recognition are substantially better. In high SNRs, approximately 90% of all keywords are recognised correctly. This is impressive, because especially the letters are easily confused even by human listen-

Table 1: CHiME test results using exemplar-based factorization and three likelihood generation methods. For each SNR and segment length T, keyword recognition percentages are shown using binary labels, OLS (10^{-2} regularization) and PLS (800 dimensions). The baseline system is CHiME reference decoder, which uses mono MFCC features and speaker-dependent GMMs.

(a) Speaker-independent results

SNR	(dB)	9	6	3	0	-3	-6
CHiME	baseline	82.4	75.0	62.9	49.5	35.4	30.3
T=10	labels	69.9	66.0	58.7	52.4	42.9	37.8
	OLS	**84.3**	77.8	71.4	65.3	56.4	48.6
	PLS	82.1	77.1	71.0	64.0	57.0	49.3
T=20	labels	77.3	72.8	68.2	62.7	51.1	44.0
	OLS	**85.2**	**80.5**	**78.7**	**71.1**	60.2	51.5
	PLS	82.9	78.8	74.8	70.1	59.5	50.6
T=30	labels	76.0	73.5	68.2	61.8	52.7	44.7
	OLS	82.8	**80.5**	76.3	70.7	**62.1**	**54.4**
	PLS	81.1	77.8	74.3	68.8	61.1	52.4

(b) Speaker-dependent results

SNR	(dB)	9	6	3	0	-3	-6
CHiME	baseline	82.4	75.0	62.9	49.5	35.4	30.3
T=10	labels	91.3	88.3	85.8	80.8	71.4	62.3
	OLS	89.8	86.8	85.0	79.7	70.1	62.7
	PLS	90.5	87.8	84.5	80.2	71.3	63.7
T=20	labels	91.6	89.2	87.6	84.2	74.7	68.0
	OLS	91.1	90.0	**88.5**	**85.2**	77.6	69.2
	PLS	**91.9**	**89.3**	88.2	85.0	**78.6**	**69.6**
T=30	labels	88.8	88.1	86.3	82.9	75.1	68.3
	OLS	88.8	86.0	86.4	83.3	76.1	69.2
	PLS	89.1	85.7	84.8	82.4	77.2	68.8

ers [9]. Noisy results are also convincing, with an increase of $\approx$ 30–40% (absolute) over the baseline at lower SNRs. Segment length 20 appears optimal in its combination of initial recognition rate and noise robustness.

In this scenario, the results of regression-based likelihood conversion are mixed. For this test set size, the differences between binary labels, OLS and PLS cannot be considered significant with sufficient confidence. The original labeling system performs well, because the speech dictionary only covers one speaker at a time and thus can contain a close approximation of almost every speech pattern of the current speaker. After sufficiently many NMF iterations, a reliable estimate of the underlying word is usually discovered, regardless of partial phonetic ambiguity. As the conversion errors are few to begin with, there is little to gain with regression. Still, we notice that both algorithms appear to produce slight improvements in the noisy end. The performance of OLS and PLS is mostly similar. It should be noted that OLS results were generally identical for all parameter sizes, while PLS performance depended on the dimension parameter. More careful tuning of it for different segment lengths would probably make it superior to OLS.

The training data set available for regression matrix learning was notably small in speaker-dependent recognition. After dictionary generation, only 200 training utterances were available per speaker. Therefore especially letter likelihoods had to be learned from only a few word instances. Comparing the results to the speaker-independent case, we can theorize that the advantages of PLS are higher, when training data is scarce. If this is not the case, OLS may be a better choice due to its lower computational cost for similar or higher quality.

6. Conclusions

A new, learning-based method was proposed for mapping speech exemplar activations into state likelihoods in automatic speech recognition. By training the conversion matrices with regression algorithms, it is possible to automatically handle phonetic ambiguity and resulting labeling problems of dictionary exemplars. Furthermore, the algorithms allow use of learned or synthetic exemplars without previous knowledge of their linguistic content.

The methods were tested using noisy speech utterances from the CHiME challenge corpus. In speaker-independent recognition, where the original state labeling was unreliable while regression training data was plentiful, all results improved by 4.3–14.1% in comparison to the previous labeling system. In speaker-dependent recognition, no significant increase or decrease was present. We conclude that automatically learned mapping can match or surpass the recognition quality of explicitly assigned state labels. Ordinary Least Squares regression was found straightforward and reliable for the purpose. Partial Least Squares requires careful parameter selection, but it may yield higher results especially for limited training data.

7. References

[1] L. Deng and H. Strik, "Structure-based and template-based automatic speech recognition - comparing parametric and non-parametric approaches," in *Proc. INTERSPEECH*, 2007.

[2] T. N. Sainath, A. Carmi, D. Kanevsky and B. Ramabhadran, "Bayesian Compressive Sensing for Phonetic Classification," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing, Dallas, U.S.A., 2010.*

[3] J. F. Gemmeke, T. Virtanen and A. Hurmalainen, "Exemplar-based sparse representations for noise robust automatic speech recognition," accepted for publication in *IEEE Transactions on Audio, Speech and Language processing*, 2011.

[4] G. S. V. S. Sivaram, S. K. Nemala, M. Elhilali, T. D. Tran and H. Hermansky, "Sparse Coding for Speech Recognition," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing, Dallas, U.S.A., 2010.*

[5] J. F. Gemmeke, T. Virtanen and A. Hurmalainen, "An exemplar-based framework for noise-robust automatic speech recognition," in *Proc. INTERSPEECH 2010*, Makuhari, Japan, 2010.

[6] P. Geladi and B. R. Kowalski, "Partial Least-Squares Regression: a Tutorial," in *Analytica Chimica Acta*, vol. 185, no. 1, 1986.

[7] S. de Jong, "SIMPLS: An alternative approach to partial least squares regression," in *Chemometrics and Intelligent Laboratory Systems*, vol. 18, issue 3, Mar. 1993.

[8] H. Christensen, J. Barker, N. Ma and P. Green, "The CHiME corpus: a resource and a challenge for Computational Hearing in Multisource Environments," in *Proc. INTERSPEECH 2010*, Makuhari, Japan, 2010.

[9] M. Cooke, J. Barker, S. Cunningham and X. Shao, "An audio-visual corpus for speech perception and automatic speech recognition," in *Journal of the Acoustical Society of America*, 120(5), 2006.