GRAPHICAL MODELS FOR DISCRETE OBSERVATION DISTRIBUTIONS IN SPEECH RECOGNITION

Antonio Miguel, Eduardo Lleida, Alfonso Ortega

Communication Technologies Group (GTC), I3A, University of Zaragoza, Spain.

ABSTRACT

Speech recognition has been traditionally associated to continuous random variables. The most successful models have been the HMM with mixture of Gaussians in the states to generate/capture observations. In this work we show how the graphical models can be used to extract the joint information of more than two features, which is not modeled with full covariance matrices. This is possible if we previously quantize the speech features to a small number of levels and work with discrete random variables. It is shown a method to estimate a constrained number of parents subset of the directed acyclic graph based model framework. Some experimental results are obtained with this method compared to baseline systems of full and diagonal covariance matrices. Additionally, it is shown that it is possible to improve the information of the discrete random variable with qualitative features, such us voicing class or pitch information.

1. INTRODUCTION

The modeling of discrete variable probability distributions is a very interesting problem specially when dealing with high dimensional variables and all their complex interactions [1]. Nowadays, it is a very active field of research in the scope of pattern recognition, machine learning and intelligent systems.

In this paper we propose the use of the graphical model approach to discover underlying dependency structures in the process of generation of observations in the states of the HMM (Hidden Markov Model). Usually the joint distribution of the random feature vector which are the observations of a HMM are modeled to follow a GMM (Gaussian Mixture Model) with diagonal covariance matrix. When more accuracy is required and enough training data are available the covariance matrices are assumed to be not diagonal. In this case, the most complex relationship considered between components of the vector is a pair of components.

In this work it is proposed to quantize the components of the speech feature vector. The resulting quantized random vector can be described now with a disc rete random variable joint distribution. The techniques proposed in the paper try to approximate the joint distribution of all the components by taking approximations. It will be shown that for a certain level of complexity, a good approximation is given by a factorization described by a directed acyclic graph with a constrained number of parents per node.

Many authors have previously contributed to this line of research from different areas, [2], but the application of these tools in acoustic modeling is limited. There have been applications of graphical models or Bayesian networks applied as an alternative to the HMM independence assumption, to build language models or spoken automatic dialog systems. These have been more natural fields to develop techniques based on discrete probabilities since, stochastic approaches to language modeling and dialog systems use discrete variables referring to words or, dialog acts over limited size sets, such us vocabularies. These kind of applications suit perfectly the graphical model ability to exploit the potential of intricate hidden dependencies among large number of variables.

This paper is organized as follows. In Section 1 there is an introduction. In Section 2 the quantization of the speech is presented. In Section 3 the factorization is described. In Section 4, the parameter estimation is derived. Experimental results are shown in Section 5 and finally conclusions are in Section 6.

2. QUANTIZED SPEECH FEATURES

Most of today's systems for speech recognition are based on statistical approaches for modeling the process of emission of observations in the HMM. From the statistical point of view the speech signal can be considered a very complex process. Not only the non-stationary nature of the signal but also the variability of observations or measurements we can get have a wide range across speakers, environments, or even for a same individual. The exact modeling of all of these sources of uncertainty in a brute force approach is not affordable since it would require astronomic sizes of models, training data and computing time. It is important to build affordable systems to provide mechanisms able of making approximations and generalize the knowledge. In this work discrete variables are used to model the speech signal, so that we are able to learn joint distributions of the speech features.

In order to perform the quantization process, a very simple process is proposed for this work. First we take the complete training corpus and, after extracting all the feature vectors, evaluate some simple statistics as the histogram. Then, we find a number of areas with an approximate probability mass, same percentile. The limits between these areas will serve to build the quantizer. This process can be seen equivalent to construct a histogram equalization transformation function with an uniform target distribution and quantize uniformly. the objective is to build a quantizer so that each quantized level represents the same amount of mass of probability in the input signal.

We should note that the process here described for building the quantizer is simple and more optimum solutions can be proposed, since once it has been obtained from the training set it remains unaltered for all the experiments. There also exist the potential future possibility of incorporating a real time implementation of a histogram equalization system, which can be interesting under mismatch of signal and models because of different training and testing conditions. In that case the histogram could be estimated based on a temporal window around the current feature vector.

3. FACTORIZED PROBABILITY DENSITY FUNCTIONS

To define the distribution associated in general to a D dimensional random vector $\mathbf{x} = (x_1, \ldots, x_d, \ldots, x_D)$, where each component of the vector z_d is a discrete variable with outcomes $x_d \in \{1, \ldots, M\}$, where M is the number of levels after the quantization and D is the dimension of the feature space. The joint

This work has been supported by the national project TIN 2005-08660-C04-01.

distribution can be expressed mathematically in the following way, without loss of generality:

$$p(\mathbf{x}) = p(x_1) \prod_{d=2}^{D} p(x_d|x_1, ..., x_{d-1}), \quad (1)$$

where we apply the Bayes theorem recursively, and about the notation $p(\cdot)$ is used for density function and $P(\cdot)$ stands for probability of event.

When the size of the feature vectors grows, the joint model is intractable, since we would need to estimate $|\Theta| = M^D - 1$ parameters. The naive approximation consists on ignoring all the dependencies in the general term of the previous expression.

The objective in the graphical model approach is to find a way of describing the interactions and independencies between the variables of a probabilistic model, and represent as much useful information from data into models as possible. This information can be conveniently represented in the form of graphs [2, 1].

The pdf of interest for us is, as we have noted before, is the pdf of the generic probabilistic process in the state of a HMM. Also it can be a component in a mixture for each state, instead of a simple pdf. The exact pdf to model the observations or feature vectors is the joint pdf (1), but, as we have said, in most cases is intractable.

In [3] was proposed a method for storaging pdfs based on a convenient "factorization" of the exact pdf. The process consisted on the selection of an appropriate order for the index of the variables, the factorization as in (1) following this order and the approximation of the conditioned distributions by more simpler ones. In [3] the number of dependencies of each variable did not exceed one.

The combination of the order and the approximations to factorize a joint pdf can be explained with a DAG (Directed Acyclic Graph) [2, 1]. The probability structure described by the graph is also called Bayesian Network. To build a a factorization model, each variable in the model x_d is associated to a node in the graph v, therefore the size of the graph is $V = D$. The pdf given by a graph can be expressed as:

$$p(\mathbf{x}) \simeq \prod_{v=1}^{V} p(x_v|\boldsymbol{\pi}(v)), \quad (2)$$

where the expression $\boldsymbol{\pi}(v)$ denotes the dependencies associated to node v, which are the set of parent of the node v in the graph. The naive Bayes models are the case of $\boldsymbol{\pi}(\cdot) = \emptyset$ for all the variables.

3.1. Constrained order dependency models

For a given node, the number parent nodes defines the dependencies of the variable with respect to other variables in the graph. If this number of dependencies is high, the number of parameters in the model is large and more data are necessary to estimate accurately those parameters. The kind of model we propose is a subset in the factorizations provided by the directed acyclic graphs, where the complexity of the target factorization is controlled as a parameter of the model. The objective is to find the best graph so that the number of parameters remains low and the approximation to the joint distribution is good enough and keeps most of the information.

The proposed approach, the Constrained Directed Acyclic Graph (CDAG), is a DAG with a limited number of parents. The order r is the maximum number of parents in the graph.

As a first approach we can express the model pdf of a CDAG(r) as follows:

$$p(\mathbf{x}) \simeq \prod_{v=1}^{V} p(x_v|\boldsymbol{\pi}(v)) = \prod_{v=1}^{V} p(x_v|\pi_1(v), \ldots, \pi_r(v)), \quad (3)$$

where $\pi_1(v)$ is the first component of the parent set (of size r) for the node v. We have to note that if the order the model, r, is set to zero, then we have the naive Bayes model, CDAG(0), and if the order is set to one, then we have the [3] tree model, CDAG(1). For simplicity in the notation we are going to express the model probability and estimate the parameters for an order $r = 2$, but the method is also applicable to larger orders.

We propose the adjacency matrix of the graph, $\mathbf{A}$, to establish a more convenient notation. The term $p(x_v|\pi_1(v), \pi_2(v))$ for $r = 2$ in (3), can be expressed as:

$$p(x_v|\pi_1(v), \pi_2(v)) = \prod_{v'} \prod_{v''} [p(x_v|x_{v'}, x_{v''})]^{(a_{v',v} \cdot a_{v'',v})}, \quad (4)$$

where $a_{v',v}$ is a component of adjacency matrix which is equal to one if the node v' is a parent of node v.

3.2. CDAG model likelihood

In order to express the model probability using the adjacency matrix notation, we have to define an indicator vector $\mathbf{b}$, with only one element equal to one, which is the index of the node without parents. The value of the components of $\mathbf{b}$ can be expressed as:

$$b_v = \begin{cases} 1 & \text{if } \sum_{v'=1}^{V} a_{v',v} = 0, \\ 0 & \text{cc.} \end{cases} \quad (5)$$

The expression (3) can be written using the adjacency matrix, $\mathbf{A}$, and the vector $\mathbf{b}$ as follows:

$$p(\mathbf{x}) \simeq \prod_{v} [p(x_v)]^{b_v} \cdot \prod_{v'} \prod_{v''} [p(x_v|x_{v'}, x_{v''})]^{(a_{v',v} \cdot a_{v'',v})} \quad (6)$$

where if $a_{v',v}$ and $a_{v'',v}$ are equal to one, the pair of edges (v', v) and (v'', v) are in the graph, and the factor $p(x_v|x_{v'}, x_{v''})$ contributes to the product .

In order to achieve a more compact notation and simpler estimation derivation, we augment the information represented in the adjacency matrix to a matrix $\mathbf{R} = \mathbf{A} + \mathbf{b} \cdot \mathbf{I}$. Introducing the augmented matrix notation, the expression (6) can be now written as:

$$p(\mathbf{x}) \simeq \prod_{v} \prod_{v'} \prod_{v''} [p(x_v|x_{v'}, x_{v''})]^{(r_{v',v} \cdot r_{v'',v})}, \quad (7)$$

where this representation is also more compact because we consider the special cases:

$$p(x_v|x_{v'}, x_{v''}) = \begin{cases} p(x_v) & v = v', v = v'' \\ p(x_v|x_{v'}) & v = v'' \\ p(x_v|x_{v''}) & v = v' \\ p(x_v|x_{v'}, x_{v''}) & cc \end{cases}. \quad (8)$$

We can express the previous expression (7) as:

$$p(\mathbf{x}) \simeq \prod_{v,v',v''} \prod_{m',m''} [p(x_v|x_{v'}=m', x_{v''}=m'')]^{(r_{v',v} r_{v'',v})(\delta_{x_{v'},m'} \delta_{x_{v''},m''})}, \quad (9)$$

In the previous expression, we can identify the distributions in the factors as Multinomials:

$$x_v|_{x_{v'}=m', x_{v''}=m''} \sim Mult_M(1, \mathbf{p}_{v,v',v'',m',m''}). \quad (10)$$

where $\mathbf{p}_{v,v',v'',m',m''}$ is the prototype vector, i.e. the histogram which gives us the probability of the M possible values of the conditioned variable. The components of the prototype vector can be expressed as:

$$p_{v,v',v'',m,m',m''} = P(x_v = m|x_{v'} = m', x_{v''} = m'') \quad (11)$$

Then we can express the conditioned variable distribution as:

$$p(x_v|x_{v'}=m',x_{v''}=m'',\mathbf{P})=\prod_m[p_{v,v',v'',m,m',m''}]^{\delta_{x_v,m}}, \tag{12}$$

with the constraint $\sum_{m=1}^{M} p_{v,v',v'',m,m',m''}=1$ for all $v,v',v''=1,\ldots,V$ and $m',m''=1,\ldots,M$.

For a set of parameters $\mathbf{\Theta}=(\mathbf{P},\mathbf{R})$, the log likelihood function for a training set, $\mathbf{X}=\{\mathbf{x}_1,\ldots,\mathbf{x}_n\}$ is:

$$L(\mathbf{\Theta};\mathbf{X})=\sum_n\sum_{v,v',v''}\sum_{m,m',m''}(r_{v',v}r_{v'',v})\times \\ \times\left(\delta_{x_{nv},m}\delta_{x_{nv'},m'}\delta_{x_{nv''},m''}\right)\cdot\log p_{v,v',v'',m,m',m''} \tag{13}$$

4. CDAG PARAMETER ESTIMATION

In order to estimate the optimum set of parameters to maximize the log likelihood function we have to solve the following optimization:

$$\{\hat{\mathbf{P}},\hat{\mathbf{R}}\}=\arg\max_{\mathbf{P},\mathbf{R}} L(\mathbf{P},\mathbf{R};\mathbf{X}), \tag{14}$$

subject to $\sum_{m=1}^{M} p_{v,v',v'',m,m',m''}$ for all $v,v',v''=1,\ldots,V$ and $m',m''=1,\ldots,M$, and subject to $\mathbf{R}\in$CDAG(2).

It is possible to show that the optimum parameter subset $\hat{\mathbf{P}}$, can be solved independently of the topology of the graph as:

$$\hat{p}_{v,v',v'',m,m',m''}=\frac{\sum_n\delta_{x_v,m}\delta_{x_{v'},m'}\delta_{x_{v''},m''}}{\sum_n\delta_{x_{v'},m'}\delta_{x_{v''},m''}}. \tag{15}$$

where in the numerator we have the number of feature vector examples in the training data whose v component is equal to the value m, the component v' is equal to m' and the component v'' is equal to m''. The numerator can be interpreted in this sense too, and together we can see that the parameter $\hat{p}_{v,v',v'',m,m',m''}$ estimates the probability $\hat{p}(x_v=m|x_{v'}=m',x_{v''}=m'')$.

The optimum set of parameters $\hat{\mathbf{R}}$, provides the edge set and the graph will be fully characterized. Once we have found the set of parameters $\hat{\mathbf{P}}$, we can obtain with a convenient manipulation an optimization similar to [3]:

$$\hat{\mathbf{R}}=\arg\max_{\mathbf{R}}\sum_{v,v',v''}(r_{v',v}r_{v'',v})\hat{I}(x_v||x_{v'},x_{v''}), \tag{16}$$

subject to $\mathbf{R}\in$CDAG(2). Where, with the notation $\hat{I}(x_v||x_{v'},x_{v''})$ we refer to the mutual information:

$$\hat{I}(x_v||x_{v'},x_{v''})= \\ \sum_{\forall x_v,x_{v'},x_{v''}}\hat{p}(x_v,x_{v'},x_{v''})\log\frac{\hat{p}(x_v,x_{v'},x_{v''})}{\hat{p}(x_v)\hat{p}(x_{v'},x_{v''})}, \tag{17}$$

which is also the Kullback-Leibler divergence between the distributions $\hat{p}(x_v,x_{v'},x_{v''})$ and $\hat{p}(x_v)\cdot\hat{p}(x_{v'},x_{v''})$, i.e. joint and approximated respectively.

It is interesting that as in [3] we obtain the same solutions (15) and (16) by minimizing the Kullback-Leibler divergence between the approximate and the exact model $D(p(\mathbf{x})||\hat{p}(\mathbf{x}))$.

4.1. Approximated algorithm for graph building

The exact algorithm to find the best graph from this expression is a hard problem, but a fast but approximate algorithm to estimate the best graph is proposed in this paper. The objective is to find an algorithm to obtain the best graph in terms of maximum likelihood,

Algorithm 1 Approximate optimum graph to obtain a CDAG(2)

Input: Random samples $\mathbf{X}$
Output: The graph $\hat{\mathbf{R}}$ of the class CDAG(2)
1. Initialization
Initiate graph matrix:
$\hat{\mathbf{R}}\leftarrow\mathbf{0}$
Estimate all $\hat{p}(x_v,x_{v'},x_{v''})$
Calculate all $\hat{I}(x_v||x_{v'},x_{v''})$
Initiate set of non assigned nodes, $\mathcal{N}$:
$\mathcal{N}\leftarrow\{x_1,\ldots,x_V\}$
Order decreasingly all $\hat{I}(x_v||x_{v'},x_{v''})$ so that:
$\hat{I}(x_{m_{1,1}}||x_{m_{1,2}},x_{m_{1,3}})>\hat{I}(x_{m_{2,1}}||x_{m_{2,2}},x_{m_{2,3}})>\ldots$
2. Search edges
$k\leftarrow 1$
while $|\mathcal{N}|>1$ **do**
  $v\leftarrow m_{k,1}$, $v'\leftarrow m_{k,2}$, $v''\leftarrow m_{k,3}$
  if $x_v\in\mathcal{N}$ **then**
    $\mathbf{R}'\leftarrow\hat{\mathbf{R}}$
    Add edges (v',v) and (v'',v) to $\mathbf{R}'$:
    $r'_{v',v}\leftarrow 1$, $r'_{v'',v}\leftarrow 1$
    if $|\mathbf{I}-\mathbf{R}'|\neq 0$ **then**
      $\hat{r}_{v',v}\leftarrow 1$, $\hat{r}_{v'',v}\leftarrow 1$
      $\mathcal{N}\leftarrow\mathcal{N}\setminus x_v$
    end
  end
  $k\leftarrow k+1$
end
Assign the last variable $x_v\in\mathcal{N}$:
$\hat{r}_{v,v}\leftarrow 1$

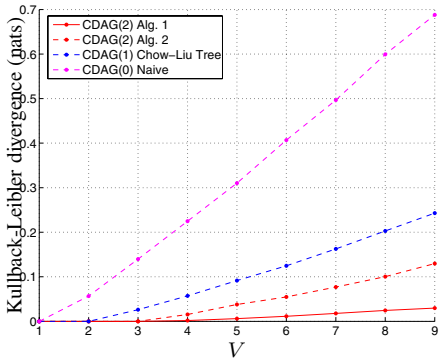
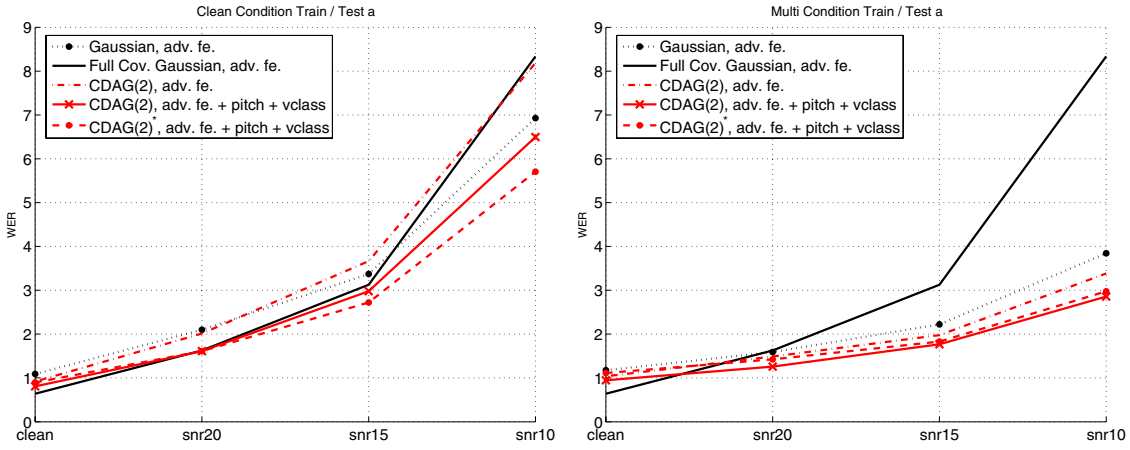


Figure 1. *DKL vs V for artificially generated data.*

which is equivalent to find the set of edges $\hat{\mathbf{R}}$ that maximize expression (16). This problem is trivial for order $r=0$,(the naive Bayes model), and can be solved exactly in polinomial time for order $r=1$, where we will obtain a special kind of graph, a tree, which is shown in [3]. For higher orders such us $r=2$, the problem becomes intractable, we can not subdivide the problem into smaller independent problems and the problem cannot be solved efficiently by dynamic programming approaches.

The approximate Algorithm 1, can be explained as follows. First the joint distributions $\hat{p}(x_v,x_{v'},x_{v''})$ and the Kullback Leibler divergences $\hat{I}(x_v||x_{v'},x_{v''})$ have to be calculated in a initializing phase. Then, the values of the Kullback-Leibler divergences are ordered and the indices of the variables v, v' and v'' are stored in the auxiliary variables $m(k,1)$, $m(k,2)$ and $m(k,3)$ respectively. The next step is the approximate search of the edges to construct the matrix $\hat{\mathbf{R}}$ with a maximum value of the sum of partial Kullback-Leibler divergences for all the edges in the graph, while keeping the graph acyclic. This is done in a loop by adding consecutively pairs of edges (v',v) and (v'',v) following



(a) (b)

Figure 2. Experimental results for Aurora2, test set a, for moderate noisy conditions. a) clean condition train, b) multi condition train.

the previous descending order. Before adding them to the solution, it is checked that the addition of both pairs does not form a cycle. In the last step we have only one variable x_v in $\mathcal{N}$. This variable has no parents, which is marked with $b_v = 1$ or $\hat{r}_{v,v} = 1$.

There is an operation in the search process which can be computationally expensive. It is the determinant calculation, to check if the new graph resulting after the addition of the current edges is acyclic. More efficient searches can be performed if this part is substituted by an incremental check of the acycliness.

The estimation process shown here can be incorporated to a hidden variable structure as a HMM or HMM with mixtures in the states. The results in the experimental section use an EM estimation of the parameters which can be derived as in [4], where is done for Bernouilli mixtures.

There exists an approach [5] to discover conditional independences or the I-map in data sets. In order to do so, there is a first step which involves the computation of terms $I(x_v||x_{v'})$ to construct a preliminary graph, which is an approximation with respect to the more exact mutual information measures the ones used in this work. In later passes of that algorithm CI (conditional independence tests) are performed. The CI test consists in the computation if the mutual information of two variables x_v and $x_{v'}$, given a cutset $\mathcal{C}$, is above certain threshold. The CI test step is carried without restrictions of the order of the joint pdfs involved. Another difference is that our approach is not intended to discover to true underlying graph with any number of parents in the nodes, but a constrained order graph.

5. EXPERIMENTS

A preliminary experiment is show in Figure 1. The performance of the Algorithm 1, compared to the naive Bayes approach, the Chow tree [3], or the approximations performed by algorithms similar to [5] (which is referred to Alg. 2) is shown. Since it is an experiment based on artificial data, we can compute the Kullback-Leibler divergence of the different models with the exact model, the joint distribution. We can see that the CDAG(2) model with Algorithm 1 behaves quite accurately in the experiment.

The proposal in this paper has also been evaluated on the Aurora 2 task [6] which is a connected digit strings recognizing task in different noise environments. The feature set are the extended adv ETSI front-end features [7], and the baseline system has been trained with HMM word models of 14 states and 3 component Gaussian mixtures for the digits, a 1 state with 6 components model for the inter-word silence unit and a 3 state with 6 components model for the begin-end silence unit. The models were trained with 20 iterations of the EM algorithm.

In Figure 2 we can see experiments with Aurora corpus. There are two baseline systems with Gaussian mixtures, but in a case there are diagonal covariance matrices and in the other full covariance matrices. Results for the discrete feature vector systems we obtained. The number of quantization levels was left to $M = 5$. We can see in Figure 2 the results obtained for all the discrete random variable approaches. We also can see that additional WER reduction can be obtained with the addition of more qualitative features such us the voicing class or the pith given by the extended ETSI front-end. The mean WER (word error rate) reduction for the best case in the test set a for the clean, snr20 to snr05 conditions (moderate noise) is a 13.5%.

6. CONCLUSIONS

In this work it has been shown a method to model high dimensional discrete distributions which is based on the assumption of a model of dependencies with a limited number of them. The generalization ability of these factorizations had been previously shown in previous works. We have adapted a previous solution for a constrained order of one to larger orders. The result has been a very interesting class of model with a good accuracy, specially in noise conditions and benefits such us a low transmission rate for the features which we will continue to enhance in future works.

7. REFERENCES

[1] J. Pearl, *Causality: Models, Reasoning, and Inference.*, Cambridge Univ. Press., 2000.

[2] S. Lauritzen, *Graphical Models*, Oxford Univ. Press, 1996.

[3] C. K. Chow y C. N. Liu, "Approximating discrete probability distributions with dependence trees," *IEEE Trans. on Information Theory*, vol. 14 (3), pp. 462–467, 1968.

[4] A. Juan y E. Vidal, "On the use of Bernoulli mixture models for text classification," *Pattern Recognition*, vol. 35, no. 12, pp. 2705–2710, December 2002.

[5] J. Cheng, R. Greiner, J. Kelly, D. A. Bell, y W. Liu, "Learning bayesian networks from data: an information-theory based approach," *Artificial Intelligence*, vol. 137, pp. 43–90.

[6] H. G. Hirsch y D. Pearce, "The AURORA experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *ISCA ITRW ASR2000*, Paris, France, September 2000, pp. 18–20.

[7] "ETSI ES 202 050 v1.1.1 distributed speech recognition; advanced front-end feature extraction algorithm; compression algorithms," July 2002.