Fusion of parametric and non-parametric approaches to noise-robust ASR

Yang Sun\textsuperscript{a,}\textsuperscript{*}, Jort F. Gemmeke\textsuperscript{b}, Bert Cranen\textsuperscript{a}, Louis ten Bosch\textsuperscript{a}, Lou Boves\textsuperscript{a}

\textsuperscript{a}Centre for Language and Speech Technology, Radboud University Nijmegen, 6525 HT Nijmegen, The Netherlands
\textsuperscript{b}Department of Electrical Engineering (ESAT), KU Leuven, B-3001 Heverlee, Belgium

Received 24 September 2012; received in revised form 9 July 2013; accepted 16 July 2013
Available online 25 July 2013

Abstract

In this paper we present a principled method for the fusion of independent estimates of the state likelihood in a Dynamic Bayesian Network (DBN) by means of the Virtual Evidence option for improving speech recognition in the AURORA-2 task. A first estimate is derived from a conventional parametric Gaussian Mixture Model; a second estimate is obtained from a non-parametric Sparse Classification (SC) system. During training the parameters pertaining to the input streams can be optimized independently, but also jointly, provided that all streams represent true probability functions. During decoding the weights of the streams can be varied much more freely. It appeared that the state likelihoods in the GMM and SC streams are very different, and that this makes it necessary to apply different weights to the streams in decoding. When using optimal weights, the dual-input system can outperform the individual GMM or the SC systems for all SNR levels in test sets A and B in the AURORA-2 task.

\section{1. Introduction}

Parametric models, such as Gaussian Mixture Models (GMMs), have been used successfully in a wide range of pattern recognition problems. For example, acoustic models based on GMMs of Mel-frequency Cepstrum coefficients (MFCC) in Hidden Markov Models (HMMs) have dominated Automatic Speech Recognition (ASR) for the last 30 years (Bourlard et al., 1996). Modeling speech features as Gaussian mixtures has proved to be a powerful approach for clean speech. In noisy conditions, however, the performance of GMM-based recognizers is known to degrade dramatically. Basically, this is because it is difficult and expensive to model speech in noise sufficiently accurately using GMMs if one wants to account for all potentially relevant (non-stationary) noises. As a consequence, the parameters characterizing observed noisy speech signals often do not match the distributions derived from the training material which has been recorded in noise-free conditions or in conditions with only a small number of noise types.

In the past, several different approaches have been proposed to make GMM-based HMM systems more robust against noises that were not represented in the training data. One approach, exemplified by ETSI (2007), consists of trying to remove the noise from the signal. By doing so, the mismatch between the trained distributions and the observed signal is reduced. Another approach, which comes in several different flavors, is known as \textit{Missing Data Theory} (Raj and Stern, 2005; Seltzer et al., 2004; Cooke et al., 2001). Basically, this class of approaches aims to determine the acoustic features that are not dominated by noise, and to base decoding on that subset of the features. Yet another set of approaches, known as model...
compensation, exemplified by van Dalen and Gales (2011), aim at adapting the trained distributions to the characteristics of the noise. All approaches mentioned above have in common that they can improve recognition performance in signal to noise ratios (SNR) between 20 and 0 dB substantially, although mostly at the cost of some degradation of the performance in clean conditions.

Recently, a new approach, named Sparse Classification (SC) (Gemmeke et al., 2011b), was introduced to the ASR field, which holds the promise of producing robust estimates of the posterior probabilities of phones or states, even in SNR < 0 dB conditions. Because SC makes no assumptions about the distributions of the acoustic features, nor of the shapes of the classes and the boundaries between these, the new approach can be regarded as non-parametric. Using a dictionary of speech and noise segments, called exemplars, represented in the form of Mel-scaled magnitude spectrograms, clean and noisy speech can be approximated as a linear combination of a small number of such exemplars. By only using the linear combination of the speech exemplars in the approximation as a basis for decoding (and discarding the selected noise exemplars), it is possible to improve recognition performance in the lower SNR conditions, even in the −5 dB condition. However, for clean speech the performance of this (non-parametric) SC approach falls well below the best conventional (parametric) GMM-based systems.

In this paper, we investigate a dual-input ASR system that fuses the state likelihoods obtained from parametric GMMs and the posterior probabilities from a non-parametric SC system, in such a manner that the dual-input system can harness the power of the GMMs for accurately modeling speech in clean conditions, and at the same time profit from the performance of the SC system in noisy conditions.

Multiple methods for combining information streams in ASR have been proposed. For instance, there have been several attempts to augment acoustic features such as MFCC or PLP coefficients by different types of information that can be derived from the speech signals, such as articulatory features (Rasipuram and Magimai, 2011) or state posterior probabilities computed by means of MLPs or SVM-based classifiers (Aradilla, 2008; Valente, 2010; Misra, 2005; Misra et al., 2003). These approaches have in common that they append the additional features to the original acoustic features, or that they use the alternative features instead of the original acoustic features. More recently, Conditional Random Field approaches have been used for merging evidence from qualitatively different sources (Morris and Fosler-Lussier, 2008). Besides the early fusion approaches mentioned above, there are also late fusion approaches, such as ROVER (Fiscus, 1997), which fuse the output of multiple independent recognition systems.

Our approach is similar to the fusion of probabilities at the HMM-state level applied in studies such as Wu et al. (1998a), Wu et al. (1998b), Kirchhoff and Bilmes (2000), Kirchhoff et al. (2000), Ellis (2000) and Wölmer et al. (2012). However, rather than trying to find optimal procedures for combining independent state posterior probability estimates obtained from SVM, MLP and GMM systems during decoding (e.g., by weighted multiplication or addition), we explore whether the concept of Virtual Evidence (VE) in a Dynamic Bayesian Network (DBN) (Pearl, 1988) may bring an additional advantage. The VE-concept in DBNs provides a mathematically coherent framework that makes it possible to jointly train all parameters of the DBN, such as GMMs and Conditional Probability Tables (CPTs). Moreover, the VE-concept is not limited to combining feature streams at the state level, but makes it possible to insert external evidence at all levels in a network. However, before attempting fusion above the state level, we first want to fully understand the fundamental issues related to this approach to fusion at the state level.

In previous papers we used a trial-and-error approach for finding the best way to combine the evidence from GMMs and an SC-system. In Sun et al. (2010) we started exploring the effect of a weighted combination of GMM and SC streams. In Sun et al. (2011b) the SC stream was represented in the form of posterior probabilities of all HMM-states, and in Sun et al. (2011a) we investigated the impact of keeping only the most likely HMM-states provided by SC as Virtual Evidence. In this paper we develop a framework that unifies our earlier experiments, and that provides a principled understanding of how the optimal weights of the streams are determined by the distributions of the input streams. This framework not only allows us to explain and interpret the commonalities and differences in our earlier experiments, it is also instrumental in setting directions for future research.

The long-term goal of our research is to improve the noise-robustness of ASR systems. As a first step in that direction, we investigate the recognition performance that can be obtained in the AURORA-2 connected digit recognition task (Hirsch and Pearce, 2000). For our research we used the Graphical Modeling Toolkit (GMTK) (Bilmes, 2002), because GMTK provides a flexible platform to investigate the use of VE in a DBN. More particularly, GMTK provides easy access to all GMMs and CPTs that are formed during training. We use this feature to systematically investigate the degree to which different model components and different training scenarios affect the decoding results.

The rest of the paper is organized as follows. In Section 2 we review the basics of the SC and DBN systems and introduce our dual-input DBN, followed by a description of the experimental settings in Section 3. We report and discuss the results of our experiments on AURORA-2 in Section 4. Conclusions and suggestions for future work are presented in Section 5.

2. Model description

In this paper, we use a dual-input DBN to fuse likelihoods obtained from GMMs with the state posterior
2.1. State probability estimation using Sparse Classification

In our SC approach audio signals (speech as well as noise) are represented in the form of the magnitudes of 23 band-pass filters, equally spaced on a Mel-frequency scale, and sampled at 100 frames/s. For the experiments in this paper each frame of the clean speech in the AURORA-2 training database was labeled with the state-id it pertained to. The state labels were obtained by means of a forced alignment, using a conventional HMM system with 16-state word models for the eleven digit words, a 3-state forced alignment, using a conventional HMM. This greatly facilitates the extension of existing models and, more importantly, the exploration of novel ideas (Çetin, 2005; Saenko et al., 2009).

The noise used to corrupt the speech in the multi-condition training database was reconstructed by subtracting the spectrogram (of which three of the 11 digits ‘zero’ to ‘nine’ (cardinality 13) are discarded) are also meaningful for describing the relative likelihood of the corresponding state labels. This enables us to use the speech exemplar weights together with the stored exemplar-state mapping to calculate the likelihood of all states for each frame in the window as a weighted sum of state occupancies. This yields a Q \times T dimensional state likelihood matrix \textbf{L}_w for each window, with \( Q = 179 \) states.

Since we apply a sliding window approach, each frame in the utterance is associated with multiple overlapping state likelihood matrices (cf. Fig. 1). By summing (and normalizing) the relevant columns of the state likelihood matrices, we obtain the state posterior probability estimate denoted \( p(q_t | \text{SC}_t) \), a 179 dimensional vector at every 10ms frame in which each component corresponds to a probability estimate for each state \( q_t \) used in the 16-state word models. For a more in-depth explanation we refer to Gemmeke et al. (2011b).

2.2. Dynamic Bayesian Networks

Dynamic Bayesian Networks (DBN) are a subset of graphical models that encompass many existing algorithms for ASR (Bilmes, 2001). The DBN framework allows one to make explicit assumptions about relationships between variables in a model that are difficult to express in a conventional HMM. This greatly facilitates the extension of existing models and, more importantly, the exploration of novel ideas (Çetin, 2005; Saenko et al., 2009).

2.2.1. DBN baseline

The DBN baseline architecture used in this study is taken from the AURORA-2 tutorial that comes with the GMTK distribution (Bilmes, 2002). Denoting the sequence of values that a variable assumes in subsequent frames \( t \) during the interval \([1, T]\) as \((\cdot)_1^T\), the single input DBN (of which three of the 7 frames are depicted in Fig. 2) complies with the following algebraic factorisation of the joint probability:

\[
p(w_1^{T}, w_2^{T}, w_3^{T}, q_1^{T}, q_2^{T}, q_3^{T}, y_1^{T})
= \prod_{t=1}^{T} (p(y_t | q_t)) f(q_t | w_t^{T}, w_{t-1}^{T}) f(w_t^{T} | w_t^{T}, w_{t-1}^{T}, q_t^{T}) p(q_t | q_{t-1}^{T})
\]

\[
f(w_t^{T}) p(w_t) \prod_{i=2}^{T} \left\{ p(w_t | w_{t-1}, w_{t-1}) f(w_{t-1}^{T} | q_t, w_{t-1}^{T}, w_{t-1}^{T}) \right\}
\]

in which \( y_t \) is the acoustic observation at time \( t \), \( f(\cdot) \) indicates deterministic CPTs, \( p(y_t | q_t) \) represents a continuous probability density function, and the other factors \( p(\cdot) \) represent discrete density CPTs.

As in Bilmes et al. (2001), the variable \( w \) (cardinality 13) represents a linguistic “word” unit (11 digits ‘zero’ to ‘nine’

\[1 \text{ http://ssli.ee.washington.edu/~bilmes/gmtk/auroraTutorial.tar.gz.} \]
and ‘oh’), ‘silence’ or ‘short pause’; \(w^w\) (cardinality 16, 3 or 1 for digits, silence and short pause, respectively) keeps track of the state position within a “word” unit; \(q^f\) and \(w^w\) (both having cardinality 2) represent state and word transitions, respectively; the short pause consists of a single state which is tied to the middle state of the silence; \(y\) denotes the observed MFCC vector; \(q\) represents the state-id and has cardinality \(S = 11 \times 16 + 3 = 179\).

White symbols in Fig. 2 represent hidden variables, while observed variables are shaded; discrete variables are represented by squares and continuous variables by circles. Furthermore, straight lines represent deterministic relations, while zigzagged lines indicate probabilistic relations. Relations between continuous and discrete variables are modelled using GMMs; relations between discrete variables are described in terms of discrete conditional probability tables (CPTs). Dashed lines correspond to a switching parent dependency.

2.2.2. Dual-input DBN

In our approach we prefer to combine the likelihoods from the GMMs with the SC state posteriors in the form of virtual evidence (VE). In contrast to the other approaches discussed in the introduction, no additional transformations or dimensionality reduction procedures are required and the external information provided by SC can be used as is. Perhaps more importantly, the VE approach makes it possible to train the GMMs taking into account the beliefs of the external knowledge source, in a way that complies with the Bayesian framework.

As explained in Subramanya et al. (2007) and Bilmes (2004), external, probabilistic evidence about the value of a variable in the network can be incorporated by introducing an observed variable \(VE\), which is a child of the variables for which one has evidence (in our case \(q\)) and by setting \(p(VE = 1 | q) = h(q)\), where \(h(.)\) is a valid probability density function that represents the available probabilistic evidence. In our case, \(h(q)\) is set equal to the probability estimates \(p(q_i | SC)\) obtained from the SC system. To avoid numerical problems during the computation of log-probs (the format required to insert the SC input into the DBN), the (many) zeros in \(p(q_i | SC)\) are substituted by a floor value of \(10^{-30}\).

The input stage of the dual-input DBN that we created for combining the MFCC input and the state probability estimates from the SC system is shown in Fig. 3. As before, the observed variable \(y\) denotes the MFCC feature vector at time frame \(t\), and the dependency between \(q\) and \(y\) is modeled by GMMs. In parallel to \(y\), the SC input is inserted as a second input stream in the form of VE. Furthermore, we introduce weights for both streams, \(x\) and \(\beta\) respectively, which allow us to control the impact of either input. The role of these stream weights will be discussed in more detail later. Both the \(y\) and the VE input are sampled at a rate of 100 observations per second; the two input streams are strictly synchronized.

Due to the fact that the GMM and the SC system use intrinsically different classification procedures, it is unlikely that the estimates of the parallel streams are always in full agreement. To handle possible disagreements, we introduced a hidden node \(\hat{q}\). Disagreements between the SC posterior estimates and the state sequence that is optimal in the presence of all other evidence is modeled by a \(179 \times 179\) CPT (indicated as SC-CPT in the remainder of this paper). Thus, at the state level, the network will see the VE input in the form of the product of the frame-vectors provided by the SC system and the SC-CPT, i.e.,

\[
p(VE_t | q_t) = \sum_{q_{t-1}} p(VE_t | \hat{q}_t, q_t) \cdot p(\hat{q}_t | q_t).
\]

Since the mechanisms underlying the GMM-based and the SC-based classifiers are sufficiently different, we treat the likelihoods obtained from the GMM and the SC streams as if they are conditionally independent. Thus, we assume that the joint likelihood \(p(y_t, VE_t | q_t)\) in Fig. 3 is equal to the product of the likelihoods of the individual inputs:

\[
p(y_t, VE_t | q_t) = p(y_t | q_t)p(VE_t | q_t)
\]  

(2) where \(p(y_t | q_t)\) and \(p(VE_t | q_t)\) are the contribution of GMM and SC, respectively. For the joint probability in Eq. (1) the addition of the VE input boils down to replacing the term \(p(y_t | q_t)\) by \(p(y_t | q_t)p(VE_t | q_t)\).

![Fig. 2. Architecture of the Dynamic Bayesian Network which is taken as a starting point in this paper.](image1)

![Fig. 3. Observation and state layer of the dual-input DBN.](image2)
The SC system yields state posterior probability estimates, rather than state likelihoods. Since the dual-input DBN requires its inputs in the form of likelihoods, \( p(q_t|SC_t) \) must be converted to the likelihood \( p(SC_t|q_t) \) through division by the state priors \( p(q_t) \) (Bilmes, 2004). It appears that in the AURORA-2 task all state priors are virtually identical, so that division by the prior is (nearly) equivalent to scaling the posteriori probabilities with a frame dependent factor. Scaling all components in the likelihood vector does not influence the decoding result, since all hypotheses in the search will be penalised or boosted by the same amount, keeping the competition between these hypothesis intact. Therefore, we can treat the posterior probability estimates \( p(q_t|SC_t) \) as if these were scaled likelihoods. As a result, the stream merging to be discussed in this paper takes place in the log-likelihood domain.

After converting the factorized joint likelihood to the log domain, the state sequence \( Q_{1:T} \) returned by a Viterbi decoding which maximizes the joint log-likelihood \( LL \) can be expressed as:

\[
Q_{1:T} = \arg\max_{q_t} \{ LL \} = \arg\max_{q_t} \left\{ \sum_t \log(p(y_t|q_t)) \right. \\
+ \beta \sum_t \log(p(VE_t|q_t)) + \sum_t \log(p(Rest_t)) \right\} \tag{3}
\]

The term \( \sum_t \log(p(Rest_t)) \) in Eq. (3) summarizes the contribution of all the remaining nodes “above” the state level in the DBN in Fig. 2, i.e., the contribution of the state transition probabilities and the language model to the scores of the best path.

The coefficients \( \alpha \) and \( \beta \), the weights assigned to the GMM and SC inputs in Fig. 3, make it possible to vary the contributions of the parallel inputs. However, these coefficients have a different status during training and decoding. During training it is essential that all probability distributions in the network represent true probability functions (i.e. sum to unity). Otherwise, stable training results cannot be guaranteed. This requirement can only be met if \( \alpha = \beta = 1 \) during training with two parallel inputs (Bilmes, 2002, p. 22). During decoding, however, we have more freedom, because the Viterbi search that maximizes Eq. (3) can yield consistent results for arbitrary values of \( \alpha \geq 0 \) and \( \beta \geq 0 \). It is not required to impose \( \alpha = \beta = 1 \), nor to impose a limitation on the sum \( \alpha + \beta \), to guarantee consistency.

From Eq. (3) it can be seen that using values \( \alpha + \beta \neq 1 \) affects the balance between the first two terms on the one hand and the third term on the other. This is reminiscent of the language model factor that is present (and must be optimized) in conventional ASR systems. In the AURORA-2 task not only the prior probabilities of the states are almost constant, but the same holds for the (non-zero) state transition probabilities. Therefore, it seems to be safe to assume that the impact of the third factor in Eq. (3) can be ignored in the experiments. Doing this will simplify the design of experiments aimed at finding the optimal values of \( \alpha \) and \( \beta \). We will come back to this issue in the Discussion section.

### 3. Set-up of the experiments

#### 3.1. Database

In our experiments, we use the multi-condition training set in the AURORA-2 database (Hirsch and Pearce, 2000) for training the GMMs and CPTs in the DBN. This set contains 8440 connected digit utterances from the TIDIGITS database, spoken by 55 male and 55 female speakers.

The utterances are artificially corrupted with four noise types (subway, babble, car, and exhibition hall), with SNRs ranging from clean to SNR = 5 dB.

For testing we used test set ‘A’ (utterances corrupted by the same noise types as in the multi-condition training set) and test set ‘B’, containing utterances corrupted by four other noise types (viz. restaurant, street, airport, train station), which are not comprised in the training materials of the GMMs and which are also not covered by the noise dictionary employed in the SC system. Both test set ‘A’ and ‘B’ contain 4004 utterances consisting of a sequence of one to seven digits, 1001 utterances for each noise type. All utterances occur in seven noise levels, viz. clean, and SNR = 20, 15, 10, 5, 0, and −5 dB.

#### 3.2. Features

The MFCC input to the DBN consisted of 39 dimensional vectors containing 12 cepstral features plus a separate log-energy coefficient, as well as the corresponding first and second order delta coefficients. They were based on a 23 band Mel-frequency spectrum, using a Hamming analysis window of 25 ms and a frame shift of 10 ms. Subsequently, all coefficients were mean and variance normalized for each utterance.

As described in Section 2.1, for the SC input we used the likelihood (scaled posterior probability) estimates that were produced by the system described in Gemmeke et al. (2011b).

#### 3.3. DBN training

Training of the DBN amounts to learning the CPTs (connecting the discrete nodes) and the GMMs (connecting the continuous input \( y_t \) to the discrete node \( q_t \) in Figs. 2 and 3) that maximize the likelihood of the training data. Thus, it involves the simultaneous estimation of all functions which describe the probabilistic relations corresponding to the edges in the network. In our experiments we focus on training the “acoustic” models, viz. (1) the GMMs that characterize the relations between \( y_t \) and \( q_t \), and (2) the SC-CPT used to map the state probability estimates from the SC system (i.e., \( p(VE_t|\tilde{q}_t) \)) to \( q_t \).

During training the GMMs, Gaussians were split once the difference of the log-likelihoods between two iterations

---

Y. Sun et al. / Speech Communication 56 (2014) 49–62
did not differ more than 2%: our final GMMs consisted of up to 64 mixture components. For our experiments we used three slightly different DBNs which involved two sets of GMMs and two SC-CPTs.

1. The first set of GMMs was trained without the SC input being present (by setting \( x = 1 \) \&\& \( \beta = 0 \) during training), which effectively defaults to using the network in Fig. 2.

2. The second set of GMMs was trained with the SC input being present \( x = 1 \) \&\& \( \beta = 1 \), and using the identity matrix as CPT.

3. For training the SC-CPT in the presence of GMMs \((x = 1 \&\& \beta = 1)\), a 179 \times 179 matrix was initialised uniformly and training continued until the likelihood of generating the training data in successive iterations did not differ more than 2%.

The CPTs gathered in the term \( \sum \log(p(Rest_i)) \) in Eq. (3) were trained for each of the three DBNs that contained a different combination of the two sets of GMM models and the SC-CPT. This resulted in three (slightly) different sets of CPTs, each of which will be used in the proper context in decoding.

3.4. Design of the experiments

Using the two sets of GMMs and the SC-CPTs we conducted a number of experiments in which we varied the fusion of the two input streams during decoding. In the first experiment we aimed to verify the correctness of the DBN implementation and to set a baseline for the performance of a GMM-only and an SC-only decoder (cf. Section 4.1). In the second experiment we compared the effect of simply fusing the two baseline systems with the effect of jointly training the GMMs and the SC-CPT. The results of this experiment triggered an in-depth analysis of the distributions of the GMM and VE likelihoods in the two parallel input streams (cf. Section 4.2). Next, we carried out a set of experiments in which we varied \( x \) and \( \beta \) (cf. Section 4.4). In a final set of experiments (cf. Section 4.5) we manipulated the SC input vectors in addition to changing the weights \( x \) and \( \beta \) during decoding. A more detailed motivation for the latter two sets of experiments will be presented in the corresponding subsections.

4. Results

4.1. Baselines

To set a baseline and to verify the correctness of the DBN implementation we created two single-input baseline systems. The first baseline system uses the GMMs that were trained without the presence of the SC input. The second baseline system only uses the likelihood estimates from the SC-system as its input, in combination with the identity matrix \( I \) for the SC-CPT. The word error rates obtained with these systems, averaged over the four noise types in test set A and B respectively, are shown in the top panel of Table 1. The results for the GMM-only system are in the row labeled \( G(\text{base}) \); the row labeled \( S(\text{base}) \) contains the results of the SC-only system.

The performance of the GMM-only system is comparable to state-of-the-art HMM systems (Chen and Bilmes, 2007). The performance of the SC-only system is virtually identical to the system in Gemmeke et al. (2011b). It can be seen that the \( G(\text{base}) \) system outperforms \( S(\text{base}) \) in almost all conditions (both for test set A and B). The exception is at \( \text{SNR}= -5\text{dB} \) in test set A, where \( S(\text{base}) \) performs substantially and significantly better than \( G(\text{base}) \).

4.2. Combination of individually and jointly trained models

Next, we investigated to what extent the two different input streams provide complementary information. We first combined the independently trained baseline systems in a straightforward manner, by creating a dual-input decoder that uses the GMM models that were trained without the presence of the SC-input in combination with the SC-input and the identity matrix for the SC-CPT, and setting \( x = \beta = 1 \). The word error rates obtained with this system are shown in the row \( G/S(\text{indiv}) \) in the second panel of Table 1. The row \( G/S(\text{joint}) \) in that panel shows the results obtained for the dual-input system in which the GMM

<table>
<thead>
<tr>
<th>( x )</th>
<th>( \beta )</th>
<th>system</th>
<th>SNR(dB) ( \backslash )</th>
<th>test set A</th>
<th>test set B</th>
</tr>
</thead>
<tbody>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>( G(\text{base}) )</td>
<td>0.63</td>
<td>0.77</td>
<td>1.19</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>( S(\text{base}) )</td>
<td>7.92</td>
<td>8.91</td>
<td>9.55</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>( G/S(\text{indiv}) )</td>
<td>0.54</td>
<td>0.78</td>
<td>1.08</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>( G/S(\text{joint}) )</td>
<td>0.55</td>
<td>0.77</td>
<td>1.08</td>
</tr>
<tr>
<td>0.4</td>
<td>0.2</td>
<td>( G/S(\text{sel_w}) )</td>
<td>0.48</td>
<td>0.84</td>
<td>1.17</td>
</tr>
<tr>
<td>95% conf. interval</td>
<td>0.14</td>
<td>0.15</td>
<td>0.19</td>
<td>0.27</td>
<td>0.41</td>
</tr>
</tbody>
</table>
models are trained with the SC-input present, and where the SC-input is used in combination with the trained SC-CPT. Again, during decoding we set $\alpha = \beta = 1$.

From Table 1 it can be seen that the straightforward combination of the two input streams ($G(\text{base})$) improves most results over the individual systems. Joint training has a small, additional advantage over the straightforward combination of the two input streams. Thus, it appears that the two streams do contain complementary information and that the system is able to learn the systematic differences between the state assignments $\tilde{q}$ of the SC system and the eventual state assignments $q$. A closer inspection of the trained SC-CPT indeed showed that it deviates somewhat from the identity matrix. The main observation is that the state alignments of the SC system differ from the reference forced alignment: the SC system assigns an appreciable part of the probability mass to both neighboring states, and sometimes one of the neighbors obtains the highest posterior. To compensate for the different alignments, the trained SC-CPT is a narrow (roughly 3 states wide) banded matrix, rather than a true diagonal identity matrix. No other systematic discrepancies were observed.

Another interesting observation that can be made from Table 1 is that combining GMM and SC reduces the 0.63% error rate of $G(\text{base})$ in clean speech to 0.55%. Despite the fact that the error rate of the $S(\text{base})$ system in the clean condition is as high as 7.92%, the SC stream apparently can compensate for some of the errors that results from the GMM-only system. However, there is also one condition, i.e., SNR = $-5$ dB in test set A, where even the best dual-input system performs worse than $S(\text{base})$. Although the difference in WER of 45.14% vs. 43.95% only approaches statistical significance, this finding still calls for an explanation.

4.3. Asymmetrical effects of GMMs and SC

To explain the asymmetrical effects of the fusion of the two input streams ($S(\text{base})$ helping $G(\text{base})$ in the clean conditions, and $G(\text{base})$ hindering $S(\text{base})$ in the most noisy condition) we need to consider the distributions of the likelihoods $p(y_t|q_t)$ and $p(VE_t|q_t)$. From Eq. (3) it can be inferred that the relative impact of $p(y_t|q_t)$ and $p(VE_t|q_t)$ is determined by the average shape of these likelihood vectors: if one of the distributions tends to divide the total probability mass over a large number of states, while the other concentrates the probability mass in a small number of states, the latter is likely to have a much stronger impact than the former.

To investigate commonalities and differences between the distributions obtained from the GMMs and the SC system, we computed the statistics of the maximum values in $p(y_t|q_t)$ and $p(VE_t|q_t)$ for the utterances in test set A in four SNRs, viz. clean, and SNR=10, 0 and $-5$ dB. For each speech frame we computed the likelihood $p(y_t|q_t)$ by computing the likelihoods of the observed MFCC vector for each of the 179 trained GMMs and normalizing them in such a way that they sum up to one. The VE likelihood $p(VE_t|q_t)$ was obtained by multiplying the observed state likelihood vectors $p(VE_t|\tilde{q}_t)$ from the external SC-system by the DBN-internal SC-CPT $p(\tilde{q}_t|q_t)$. Subsequently, we computed the statistics of the maximum values in $p(y_t|q_t)$ with both sets of GMM models, and the maxima in $p(VE_t|q_t)$ with the identity matrix $I$ and the trained SC-CPT. If the value of the maximum averaged over a complete test set is large, most of the probability mass is concentrated in a single state $q_t$, which corresponds to a very sharp distribution.

The results are shown in Table 2. The most striking observation from this table is that in $p(y_t|q_t)$ most of the probability mass gets assigned to one state (out of 179). Although slightly less, this bias towards a single state also exists with the GMMs trained in the presence of the SC input. Moreover, it can be observed that the GMMs retain this tendency even in noisy speech. In the distributions of $p(VE_t|q_t)$ the probability mass is spread over multiple states. Especially in low SNR-conditions the most probable state accounts for less than 20% of the probability mass.

The effect of the different shapes of the distributions of $p(y_t|q_t)$ and $p(VE_t|q_t)$ is also illustrated in Fig. 4, which shows the probability estimates in one utterance (five-five-zero) in clean and SNR = $-5$ dB subway noise. The horizontal axis represents time, while the vertical axis represents the states; between each pair of horizontal grid lines there are 16 states for each digit. In the $p(y_t|q_t)$ estimates (top), there is usually only one candidate state for each time frame, in both the clean and SNR = $-5$ dB condition. In the $p(VE_t|q_t)$ estimates (bottom), several neighbouring states, as well as states from other digits (which presumably show some resemblance with the exemplars of the current digit) receive substantial probability values.

To better understand the findings in Fig. 4 and Table 2, we computed the per-frame entropy of the likelihoods. The results confirmed that, on average, the SC-based VE likelihoods form a flatter distribution than the GMM-based state-likelihoods. Therefore, the finding that at high SNRs the inferior SC input can help the superior GMM input, and that at SNR = $-5$ dB the inferior GMM input hurts the superior SC input, is probably due to the difference in impact between the terms $\log(p(y_t|q_t))$ and $\log(p(VE_t|q_t))$ in Eq. (3) when $\alpha = \beta = 1$.

<table>
<thead>
<tr>
<th>SNR(dB)</th>
<th>clean (%)</th>
<th>10 (%)</th>
<th>0 (%)</th>
<th>-5 (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMM(indiv)</td>
<td>93.47</td>
<td>92.21</td>
<td>91.43</td>
<td>90.96</td>
</tr>
<tr>
<td>SC(indiv)</td>
<td>25.02</td>
<td>20.28</td>
<td>17.00</td>
<td>14.59</td>
</tr>
<tr>
<td>GMM(joint)</td>
<td>92.51</td>
<td>90.39</td>
<td>89.28</td>
<td>88.48</td>
</tr>
<tr>
<td>SC(joint)</td>
<td>28.30</td>
<td>23.58</td>
<td>19.77</td>
<td>17.49</td>
</tr>
</tbody>
</table>

Table 2

The average values of the maxima in the likelihoods $p(y_t|q_t)$ and $p(VE_t|q_t)$ obtained for four SNR conditions in test set A. GMM(indiv): using the GMMs trained in the absence of SC; SC(indiv): using the identity matrix as SC-CPT; GMM(joint): using the GMMs trained in the presence of SC; SC(joint): using the trained SC-CPT.
4.4. Balancing the weights of the two input streams

Since it appeared that the two input streams may have different impacts, we want to investigate whether the performance of the dual-input system can be improved by optimizing the weights of the two input streams during decoding. This can be done by varying \( \alpha \) and \( \beta \). In the linear (probability) domain, \( \alpha \) and \( \beta \) act as exponents that affect the shape of the distributions. If \( p(\cdot) \) denotes a probability distribution, then \( p^\alpha(\cdot) \) has a flatter shape for values of \( \alpha < 1 \), while the shape becomes sharper for \( \alpha > 1 \). In the log-domain, \( \alpha \) and \( \beta \) serve as multiplication factors that affect the dynamic range of the log-prob scores of the two streams and thereby their relative impact on the eventual recognition result.

Using a grid search in which we explored 165 combinations of \( \alpha \) and \( \beta \), we evaluated the word accuracy for both test set A and B at four SNR conditions: clean, 10 dB, 0 dB and \(-5\) dB. In this grid search 11 values for \( \alpha \) were chosen in the interval \([0, 1]\), using a step size of 0.1. For \( \beta \) we used 15 values divided into two subranges. The first subrange for \( \beta \) was \([1, 5]\), using a step size 1; this corresponds to increasing the relative difference between ‘large’ and ‘small’ probability estimates in the SC output, which should counter the tendency of the GMM stream to concentrate most of the probability mass in a single state. Additionally, we explored \( \beta \) values in the subrange \([0, 0.9]\), with a step size 0.1, to keep correspondence with the values of \( \alpha \) used for the GMM-stream. It should be noted that a setting \((\alpha < 1) \land (\beta < 1)\) effectively decreases the contribution of the first two terms in Eq. (3) relative to the third term, while \((\alpha > 1) \land (\beta > 1)\) has the opposite effect.

Fig. 5 shows the word error rates as a function of \( \alpha \) (along the vertical axis) and \( \beta \) (along the horizontal axis) in the form of filled contour plots. The (red) dots indicate the \((\alpha, \beta)\)-combinations for which the word error rates were in the bottom 5-percentile of the values obtained across the 165 grid points; this bottom 5-percentile word error rate is shown above each subplot. Successive contour lines represent the distance from these 5-percentile best performance levels: Going from white to black the red contour levels correspond to the \( p \) = 84%, 95%, 97.5%, 99%, 99.9% and 99.999% (one-sided) confidence intervals. In the black areas the white contour lines show increasing WERs (increase of 1% absolute per contour).

From Fig. 5 we can make a number of observations. First, stream weighting is not only important for a dual input system. In a single-input GMM system, represented by the accuracies on the \( \beta = 0 \) axis, the curvature of the white contour lines indicates that the best performance for the GMM-only system is typically obtained for a value \( \alpha \approx 0.4 \). At the same time, it is clear that this flattening of the GMM pdf’s does not yield a performance that can compete with the dual input system. The additional SC stream \((\beta \neq 0)\) helps to improve recognition results in all \{test set, SNR\}-conditions. For \( \beta = 0 \) the WERs are...
Fig. 5. Ranges of $\alpha$ and $\beta$ for which statistically similar word error rates are obtained (using the full dimensional SC input). Different contour lines represent distances from the top 5-percentile performance levels in terms of confidence intervals (see text). The (red) dots represent ($\alpha$, $\beta$)settings which result in word error rates below the bottom 5-percentile level (this level is denoted above each subplot). For a further explanation see the text. (For interpretation of the references to colour in this figure caption, the reader is referred to the web version of this article.)

significantly ($p \leq 0.05$) higher in all conditions, except for a small range of $\alpha = [0.2, 0.3]$ in the clean condition. From the largely horizontal patterns in all sub-figures it can be inferred that in all (test set, SNR)-conditions the performance of the dual-input system is far more sensitive to $\alpha$ than to $\beta$. In the two cleanest conditions, there is a wide range of $\beta$ values within which the performance does not vary significantly once a proper value for $\alpha$ is chosen. In the more noisy conditions the range of $\beta$ values within which performance does not change significantly is (slightly) more restricted. In addition, it can be seen that, especially in the cleaner conditions, larger values of $\beta$ make the system less sensitive to the value of $\alpha$.

The bottom panel of Table 1 shows the performance obtained at the grid point ($\alpha, \beta$)=(0.4,0.2). This grid point was chosen based on the low average WER level on test
set A. It can be seen that only between 10 dB and 20 dB SNR for test set A the performance is slightly worse than $G/S(joint)$; in all other (test set, SNR)-conditions the performance is better than $G/S(joint)$, with $(\alpha, \beta) = (1.0, 1.0)$. Most importantly, in test set A the performance in the SNR = −5 dB-condition now exceeds the performance of $S(base)$; the difference is significant at the $p \leq 0.05$ level. This shows that the performance of the dual-input system can be improved by proper weighting of the contributions of the two input streams during decoding. For test set B, the chosen weight combination also gives better performance than $G/S(joint)$ with $(\alpha, \beta) = (1.0, 1.0)$ for all SNRs. This suggests that the beneficial effect of a proper stream weighting generalizes to noise types that were not seen during training.

A comparison of the various SNR conditions in Fig. 5 suggests that the value of $x$ must be reduced as the SNR level decreases. Also, it appears that most of the close-to-optimal results are obtained with values $\beta < 1$. If both $x$ and $\beta$ are smaller than one, the relative weight of the information encapsulated in the $p(Rest)$ term in Eq. (3) becomes more important. The 10 dB SNR condition in test set A might seem to be an exception, because here the best results are obtained with $x = 1$. This can be explained by noting that this condition gives the best average match between the test data and the multi-condition training data. The finding that the best performance on clean speech of the GMM-only system ($\beta = 0$) is obtained with with values $x < 1$ is due to a mismatch between the clean test data and the multi-condition training data.

In the $(x, \beta)$-region explored in the grid search, there is an interaction between $x$ and $\beta$: the optimal $\beta$ values vary with $x$. This relation is not monotonous: Fig. 5 shows that in five out of seven conditions close-to-optimal results can also be obtained with values of $\beta > 1$. In the SNR = 10, 0 and −5 dB conditions of test set B (of which the noises have been seen neither by the GMMs nor by the SC system), the grid points where the top 5% performances are obtained are located in two disjoint regions, viz. $x \approx 0.3$ and $\beta \approx 0.2$ on the one hand, and $x \approx 0.4 − 0.5$ and $\beta \approx 4 − 5$, on the other. In the former area, the information of the GMM stream dominates the decisions on the most likely state sequence, while the information of the SC stream has more impact in the latter area. This corroborates the conclusion that the two streams do carry different evidence. It also suggests that it is not possible to find a unique set of weights that is optimal for all SNR conditions.

4.5. Reducing the support of SC

Values of $\beta > 1$ emphasize the states that are considered most likely by the SC-system and de-emphasizes the less likely ones. At very high values of $\beta$ the de-emphasis becomes equivalent to discarding the lowest state probabilities in the SC vector. In previous research on using DBNs for combining SC and GMMs in speech recognition, we retained only a limited number of non-zero SC-coefficients by successively removing the smallest coefficients and subsequently renormalizing the remaining coefficients to sum to one (Sun et al., 2011a; Sun et al., 2011b). This procedure was dubbed “reducing the support of the SC vectors”. The results suggested that truncating the SC-vector did improve the word error rate in a system with uniform weights for the two streams.

The results in Section 4.4 showed that values $\beta > 1$, which also emphasizes the largest coefficients in the SC vectors, improve WERs in the lower SNR conditions. Therefore, we analyze the relation between the two mechanisms for reshaping the SC vectors in more detail. We will refer to the number of non-zero coefficients that are retained as SC-Dim. We investigated the interaction between $x$ and SC-Dim, while keeping $\beta = 1$. We varied $x$ in the interval $[0.1, 1]$, using a step size 0.1. For SC-Dim we selected eight values: {1, 2, 5, 10, 20, 50, 100, 179}. For each value of SC-Dim, we replaced the vector elements with a rank $\geq$ SC-Dim by the floor value $10^{-30}$, after which the vectors were renormalized.

Fig. 6 shows the recognition performance as a function of $x$ (on the vertical axis) and SC-Dim (on the horizontal axis) by means of filled contour plots. The sub-plots show the word error rates for test set A and B in four different SNR conditions. Again, starting from the white area, the contour levels correspond to the $p=84\%$, 95\%, 97.5\%, 99\%, 99.9\% and 99.999\% (one-sided) confidence intervals relative to the performance at the 5-percentile point. The white contour lines in the black areas correspond to ever larger WERs (increase of 1\% absolute per contour line).

The first observation that can be made from Fig. 6 is the striking similarity between the left hand side of the sub-plots, where SC-Dim > 50, and the right hand side of the plots in Fig. 5, where $\beta > 1$. This indicates that the net effects of reducing the support of SC and increasing the influence of the SC stream by the stream weight $\beta$ are very similar. As a result, we can draw many of the same conclusions as in Section 4.4. For example, from the slopes of the contour lines in the black areas in the sub-plots for clean and SNR = 10 dB it can be seen that emphasizing the largest entries in the SC vector requires a higher value of $x$, which corresponds to a higher weight of the GMM estimates relative to the SC estimates.

The fairly sharp transition to higher WERs for SC-Dim < 50 suggests that the 50 coefficients with the lowest rank/ highest value all contain some relevant information, except perhaps in the −5 dB condition in test set A, where an alternative optimum is present at very low values of SC-Dim. This is the single condition in which the $S(base)$ system clearly outperforms the $G(base)$ system. Setting entries in the SC vector to the floor value $10^{-30}$ makes paths through the corresponding states in the Viterbi search (cf. Eq. (2)) very costly. Apparently, the few states that are still licensed when SC-Dim is very small are often on the correct path. But it is also clear that the GMMs still contribute useful information for choosing between the small number of candidates that are left.
In contrast to the fairly abrupt changes in WER – when decreasing SC-dim from 179 down to 1 – that are evident from Fig. 6, in Sun et al. (2011a,b) we found that the WERs changed relatively gradually. Typically, the lowest WERs were found for values of SC-Dim at the lower end of the 1–179 range. This seeming discrepancy can be explained by noting that we always used $\alpha = 1$ in our previous studies. This corresponds to the top horizontal line in Fig. 6. Along this line the differences in WER are less outspoken and occur much more smoothly than for lower values of $\alpha$. As in the previous studies, we see that with decreasing SNR the lowest WERs are found for lower values of SC-Dim. The detailed analysis in this paper shows that it is essential to have a complete picture, that uncovers...
the impact of the statistical properties of the streams that are combined. An analysis that is limited to part of the space spanned by the parameters investigated in this paper, which ignores the potential effects of different distributions of SC and GMM likelihoods, can give rise to misinterpretations.

In Section 4.4 it was found that the best results were obtained with values \( \beta < 1 \), which de-emphasize the largest coefficients in the SC vector. Therefore, we investigated the effect of using SC-Dim < 179 in combination with stream weights that differ from one. In Table 3 we show the WER results obtained with the previously selected ‘optimal’ values \((x, \beta) = (0.4, 0.2)\), for several values of SC-Dim. The WERs indicate that the results obtained with SC-Dim > 20 or maybe even > 5 do not differ significantly from those obtained with SC-Dim = 179. The fact that in the lowest SNR conditions the best results are obtained with very small values of SC-Dim is in accordance with the previous finding that in these conditions close to optimal results can be obtained with values \( \beta \approx 5 \). Taken together, these results suggest that the contributions of the coefficients in the SC vectors with ranks between 20 and 50 are marginal, and that most of the time the coefficients with ranks \( \leq 5 \) indicate the correct state. Because the SC state likelihoods are computed from a sparse combination of exemplars, this does not come as a surprise.

5. Discussion and conclusions

In this paper we tried to develop a principled explanation for the results of previous experiments in which we observed improvements in WER for the AURORA-2 tasks by means of several ways of combining likelihood scores obtained from a GMM with independently obtained state likelihoods in the form of Virtual Evidence in a DBN. The reason for wanting to combine GMMs and SC is that GMMs perform very well in the high SNR conditions, while SC shows superior performance in the low SNR conditions. One of the reasons for using the VE option in a DBN is that this makes it possible to simultaneously train GMMs and a CPT that inserts SC estimates into the DBN. Since unexpected results may occur when the conditional probability functions on the edges of the DBN are not proper probability distributions, it is imperative to keep the weights of the parallel inputs equal to one during training. However, in decoding the stream weights can be optimized with the only constraint that the weights must be \( \geq 0 \). Thus, a large part of the research focused on optimizing these weights during decoding.

From our results in Table 1 it can be seen that the dual-input system with weights equal to one for both inputs outperformed both individual systems in all SNR conditions in test sets A and B, except in the SNR = −5 dB (worst) condition in test set A. It can also be seen that joint training of the GMMs and the CPTs in the dual input DBN only resulted in a marginal improvement over fusing the individually trained systems. Importantly, joint training did not remove the inferior performance of the dual-input system in the −5 dB condition in test set A. Somewhat surprisingly, a small (although not statistically significant) performance gain was observed for the dual-input system in the clean condition, in which the GMM-system outperformed the SC-system by a wide margin. Apparently, the SC-stream can occasionally help the GMM stream, even if it (as an individual stream) leads to inferior performance.

The asymmetric behavior of the fusing of the GMM and SC systems in the clean and −5 dB SNR condition in test set A gave rise to an in-depth analysis of the statistical properties of the GMM state likelihood scores and the SC state posterior probability estimates. It appeared that these properties are very different: while the GMM estimates tend to concentrate the lion’s share of the total likelihood in a single state, the SC estimates always attribute similar probabilities to several acoustically similar states. These intrinsically different properties make it necessary to assign different weights to the two streams so that they can make optimal contributions during decoding. By optimizing these weights it is possible to construct a dual-input system that outperforms the best individual system in all conditions, including the −5 dB SNR condition in test set A. However, we did not succeed in finding a unique set of weights that provide optimal results in all conditions. We also found that it can be advantageous to ‘flatten’ the vector of state likelihoods obtained from the GMMs.

<table>
<thead>
<tr>
<th>SC-dim</th>
<th>system</th>
<th>SNR(dB)</th>
<th>test set A</th>
<th>test set B</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>clean</td>
<td>20</td>
<td>15</td>
</tr>
<tr>
<td>179</td>
<td>G/S(179dim)</td>
<td>0.48</td>
<td>0.84</td>
<td>1.17</td>
</tr>
<tr>
<td>100</td>
<td>G/S(100dim)</td>
<td>0.48</td>
<td>0.83</td>
<td>1.14</td>
</tr>
<tr>
<td>50</td>
<td>G/S(50dim)</td>
<td>0.50</td>
<td>0.85</td>
<td>1.14</td>
</tr>
<tr>
<td>20</td>
<td>G/S(20dim)</td>
<td>0.50</td>
<td>0.85</td>
<td>1.17</td>
</tr>
<tr>
<td>5</td>
<td>G/S(5dim)</td>
<td>0.51</td>
<td>0.90</td>
<td>1.36</td>
</tr>
<tr>
<td>2</td>
<td>G/S(2dim)</td>
<td>0.67</td>
<td>0.93</td>
<td>1.56</td>
</tr>
<tr>
<td>1</td>
<td>G/S(1dim)</td>
<td>0.60</td>
<td>0.95</td>
<td>1.47</td>
</tr>
<tr>
<td>95% conf. interval</td>
<td>0.11</td>
<td>0.15</td>
<td>0.18</td>
<td>0.25</td>
</tr>
</tbody>
</table>
which – everything else being equal – corresponds to increasing the weight of the non-acoustical part of the DBN during decoding.

In previous experiments with fusing parallel input streams at the state probability level using state estimates obtained from MLPs or GMMs there was no need for optimizing stream weights during decoding (Wu et al., 1998a; Wu et al., 1998b; Kirchhoff and Bilmes, 2000; Kirchhoff et al., 2000; Ellis, 2000). Therefore, it might be argued that the need for optimizing the weights is an unfortunate side effect of the way in which the SC system computes state probability estimates and that the results in this paper do not generalize. However, the emergence of novel classifiers in the sparse representation framework and in other machine learning frameworks is likely to introduce additional systems that show promising performance in adverse conditions, while still producing probability vectors with an entropy that is much higher than what is usually obtained from MLP and GMM systems. The need for fusing systems with widely different average entropy outputs is also present in other application domains. We believe that the results presented in this paper can help guide future efforts in fusing such differing systems.

As explained in Section 2.2.2, joint training with the input streams can only be guaranteed to yield consistent results with stream weights $\alpha = \beta = 1$. However, the finding that reducing the support of the SC stream is tantamount to using a value $\beta > 1$ in decoding opens the possibility for also changing the weights of the streams during joint training, without jeopardizing the stability of the results. After re-normalization a truncated SC input is still a valid VE input. Joint training in a condition in which the statistical properties of the two input streams are more similar might well be more effective in identifying the useful information in the joint streams. Therefore, concluding that joint training is of little added value would be premature.

Several other avenues for future research exist. Obviously, integrating the recent improvements of the SC approach reported in Gemmeke et al. (2011a); Gemmeke and Van hamme (2011) and Hurmalainen et al. (2011) are expected to improve the performance of the dual-input recognizer. A more fundamental line of research focuses on the effective combination of streams. As we already discussed in Section 4.4, choosing $\alpha < 1 / \beta < 1$ amounts to reducing the acoustic evidence relative to the information encapsulated in the model topology. The topology used in our experiments makes it possible to assign different weights to the streams, and the combined effect of these weights determines the relative impact of the acoustic and non-acoustical probability distributions in the network. In future experiments it might be advantageous to introduce a separate control mechanism for adjusting the relative input stream weights on the one hand, and the weights assigned to acoustic evidence relative to the weight of the word model topology and language model on the other.

The experiments described in this paper have shown that it is not possible to find a unique set of parameters that yields superior results in all SNR conditions in both test set A and B. This strongly suggests that it is necessary to develop adaptive procedures that can find locally optimal values of the parameters. This, too, will be a topic of future research.

Acknowledgement

The research leading to these results has received funding from the European Community’s Seventh Framework Programme FP7/2007-2013 under Grant agreement n° 213850 – SCALE. The research of Jort F. Gemmeke was funded by IWT-SBO project ALADIN contract 100049. Louis ten Bosch received funding from the FP7-SME project OPTI-FOX, project reference 262266.

References


ETSI, 2007. ETSI standard doc.: Speech processing, transmission and quality aspects (STQ); distributed speech recognition; advanced front-end feature extraction algorithm; ES 202 050 V1.1.5.


