# Stream fusion for multi-stream automatic speech recognition

**Hesam Sagha, Feipeng Li, Ehsan Variani, José del R Millán, Ricardo Chavarriaga, Björn Schuller**

**Angaben zur Veröffentlichung / Publication details:**

Sagha, Hesam, Feipeng Li, Ehsan Variani, José del R Millán, Ricardo Chavarriaga, and Björn Schuller. 2016. "Stream fusion for multi-stream automatic speech recognition." *International Journal of Speech Technology* 19 (4): 669–75. https://doi.org/10.1007/s10772-016-9357-1.

# Stream fusion for multi-stream automatic speech recognition

**Hesam Sagha[1] · Feipeng Li[2,3] · Ehsan Variani[2,4] · José del R. Millán[5] · Ricardo Chavarriaga[5] · Björn Schuller[1,6]**

**Abstract** Multi-stream automatic speech recognition (MS-ASR) has been confirmed to boost the recognition performance in noisy conditions. In this system, the generation and the fusion of the streams are the essential parts and need to be designed in such a way to reduce the effect of noise on the final decision. This paper shows how to improve the performance of the MS-ASR by targeting two questions; (1) How many streams are to be combined, and (2) how to combine them. First, we propose a novel approach based on stream reliability to select the number of streams to be fused. Second, a fusion method based on Parallel Hidden Markov Models is introduced. Applying the method on two datasets (TIMIT and RATS) with different noises, we show an improvement of MS-ASR.

**Keywords** Multi-stream speech recognition · Performance monitor · Classifier ensemble creation and fusion

Hesam Sagha
hesam.sagha@uni-passau.de

[1] Chair of Complex & Intelligent Systems, University of Passau, Passau, Germany

[2] Center of Language and Speech Processing, Johns Hopkins University, Baltimore, USA

[3] Present Address: Apple Inc, San Francisco Bay Area, CA, USA

[4] Present Address: Google, San Francisco Bay Area, CA, USA

[5] Defitech Chair in Brain-Machine Interface, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland

[6] Department of Computing, Imperial College, London, UK

## 1 Introduction

How to build an automatic speech recognition (ASR) system that is robust in noisy environments has been a long-lasting research topic. Previous studies can be generally classified into two categories, i. e., feature based and model based noise robustness. Feature based methods try to produce features that are invariant to noise [e. g., applying Phase Auto Correlation (Ikbal et al. 2012), RASTA filtering (Hermansky and Morgan 1994) and deep long short term memory networks (LSTM) (Weninger et al. 2014)], while model based methods try to build classifiers that are robust to the variability of the signals, for example by applying multi-condition training (Furui 1992), LSTM networks (Geiger et al. 2014), non-negative matrix factorisation (Wöllmer et al. 2013), and a multi-stream speech recognition system (MS-ASR) (Hermansky and Morgan 1994; Sharma 1999; Mesgarani et al. 2011; Bourlard et al.. 1997; Hermansky 2013).

Our approach is toward the MS-ASR, which emulates the parallel processing in the human auditory system. Studies on human speech perception (Fletcher 1953) indicate that the phoneme error rate of fullband speech, $e$, can be approximated by

$$e = e_1 e_2 \ldots e_{20}, \tag{1}$$

where $e_i$ is the phoneme error rate of the ith subband speech signal (Allen 1994). This approximation suggests that multiple frequency subbands form independent channels for speech communication, and the total recognition error are dominated by the channel that gives the lowest error rate. Inspired by this study, Hermansky et al. proposed a MS-ASR with 2 and 7 frequency bands (Hermansky et al. 1996). Each of the subband classifiers is a phoneme-based multi-layer perceptron/hidden Markov model (MLP/HMM) hybrid recognizer using subband perceptual linear prediction (PLP) features as inputs and phoneme likelihood as outputs. They used a linear (weighted sum of likelihoods) and non-linear MLP fusion to combine classifier decisions. In turn, Bourlard and Dupont proposed a MS-ASR system with 4 frequency bands (Bourlard and Dupont 1997). A hybrid MLP/HMM system is built for each subband, and the state likelihoods are combined to give a more reliable decision. In a following study, a 7-band system is developed with a MLP trained for stream fusion (Tibrewala and Hermansky 1997). It was shown that, the multi-stream system outperforms conventional single-stream systems for moderate noise levels. In another study, the 7-band system was extended to produce all possible combinations of 7 subbands giving 127 streams and combine them with a majority voting (Sharma 1999). However, in the situations where the test data is prone to noise, the performance of the system may be degraded due to the inclusion of the noisy streams. Therefore, in these situations, for each test instance, the most efficient streams out of all the streams should be selected dynamically for the fusion [similar to Giacinto and Roli (2000)]. A dynamic selection approach is proposed in (Variani et al. 2013) to exclude the subbands that are corrupted by noise. They used a 7-band system and generated all the 127 combinations (processing streams) of them. Then, the 127 streams are ranked based on a measure called Mean temporal distance ($M$ measure) (Hermansky et al. 2013). Finally, the top $N$ (default $N = 10$) classifiers are fused to generate the final hypothesis. In another study, the combination of the streams are done through weightning each stream (Mallidi and Hermansky 2016). The weights are obtained based on the reconstruction errors of DNN activations using auto-encoders.

In this study, we aim to improve the MS-ASR system from (Variani et al. 2013) by proposing solutions to the following two questions: (a) What is the optimum number of processing streams $N$ to be selected; and (b) How to fuse the decisions of the $N$ selected classifiers.

## 2 Method

The proposed MS-ASR is comprised of four main components (refer to Fig. 1): Generation of the streams, monitoring performance of each stream, selection of the number of the streams to be combined ($N$), and the fusion of the $N$ best streams. In this case, each stream is a classifier (trained on certain frequency subbands) providing decisions (phonemes' likelihood). Here, we are dealing with the particular implementation proposed in (Variani et al. 2013) to create the streams and monitor the stream performance. The novelty in this paper is the automatic and dynamic selection of the number of streams ($N$) to be fused at each time interval (Sect. 2.3) and the fusion of the streams by a Parallel HMM (Sect. 2.4) in order to improve the phoneme recognition performance.

### 2.1 Stream formation

Frequency domain linear prediction (FDLP) features which characterise the temporal evolution of the signal's Hilbert envelope in a frequency subband over 200 ms are extracted (Ganapathy and Hermansky 2012). Each subband is created by bandpass filtering of the frequency spectrum. Features of neighbouring bands are combined to form *subband streams*. These streams separately feed one deep neural network (DNN) classifier, each. Each classifier $i$ provides a posterior probability vector reflecting the level of uncertainty at which the data can be assigned to any of the target phonemes. Then, the logarithm is taken from the posteriors to produce a normal distribution and we reduce the dimensions into a 25-d vector, $o_i$, using Karhunen–Loeve transform (KLT) in order to reduce the number of parameters at the second stage. Next, all possible combinations of the classifiers at the first stage are created $\{[o_1], \ldots, [o_1 o_2], \ldots, [o_1 \ldots o_k]\}$, and for each of them, the concatenation of $o_i$s serves as the input of the DNNs at the second stage. For example, in case there are 7 classifiers at the first stage, there will be $2^7 - 1 = 127$ DNN classifiers (*processing streams*) at the second stage, each of them may have a different input vector length considering how many classifiers of the first stage are combined. At the second stage, again, each classifier produces a probability vector of the same size of the phoneme list. Finally, we need to measure the performance of each classifier and select the ones which are more informative and fuse their decisions. The selection process is done by monitoring performance of each stream (classifier outputs) and is described in the next section.

### 2.2 Performance monitoring

We assume that, noise is not stationary and at each window of time we need to select dynamically the best streams (dynamic ensemble selection). Out of all available classifiers at the 2nd stage, not all of them have the same performance; Some channels may be exposed to noise or some may not have high performance inherently. Hence, a set of $N$ promising streams should be selected and fused. In this section, we describe a performance monitoring method based on the posteriorgram analysis (Hermansky et al. 2013). The method provides a measure, $M$, indicating that a classifier is performing well if (a) it is able to produce a high posterior probability for most of the phonemes, (b) it

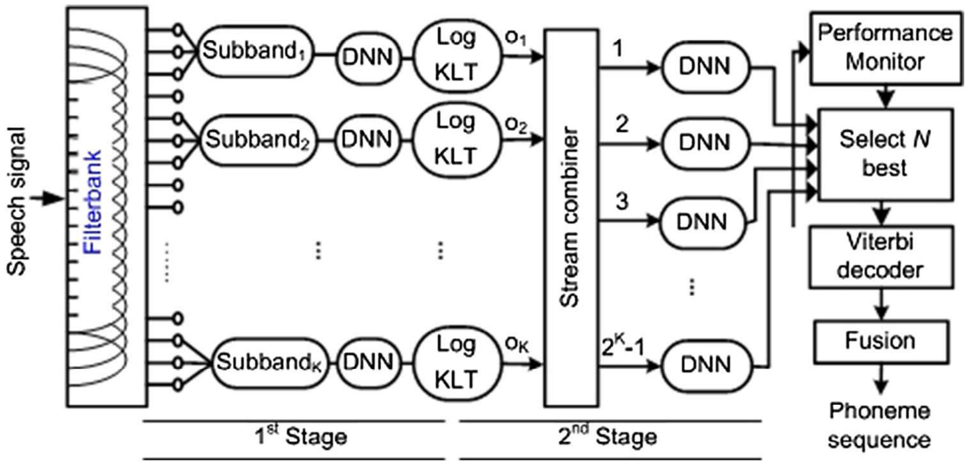

**Fig. 1** Block diagram of the multi-stream phoneme recognition system

can produce posterior probabilities which are sufficiently different for a time span larger than the phoneme articulation. This could be obtained by measuring the distance between posterior probability vectors at different times, considering that in an intelligible speech, we expect phonemes to change fast ( ~70 ms) and variety of phonemes are uttered at each time interval (10–15 phonemes per second). Given a posteriorgram, the distance between two posterior vectors at time $t$ and $t + \Delta t$ is defined as a symmetric KL-divergence:

$$D(P_t, P_{t+\Delta t}) = \sum_i P_t(i)\ \ln\left(\frac{P_t(i)}{P_{t+\Delta t}(i)}\right) + \sum_t P_{t+\Delta t}(i)\ \ln\left(\frac{P_{t+\Delta t}(i)}{P_t(i)}\right), \qquad (2)$$

where $P_t(i)$ is the ith element (phoneme) of the posterior probability vector $P$ at time $t$. Finally, $M$ is defined as the average of distance over time $t$:

$$M(\Delta t) = \frac{\sum_{t=0}^{T-\Delta t} D(P_t, P_{t+\Delta t})}{T - \Delta t}, \qquad (3)$$

where $T$ is the length of the utterance. For the results shown later, since the utterances are already segmented, $T$ is the length of the sentence utterance. Setting $\Delta t$ to 250 ms is reasonable, because the effect of coarticulation is low after this duration. Hence, for all experiments we set it to this value. $M$ implicitly measures the following criteria:

*Classifier accuracy* If a classfier is able to produce sharp decisions (high value of one posterior element and low values for others) and able to produce suffciently different posteriors for different phonemes, then the classifier is most probably accurate and M is high.

*Classifier bias* If a classifier is biased, only some of the elements of the posterior vector are active and the rest are close to zero most of the times. Consequently, the distance between two posteriors at the interval $$\Delta$$t is low.

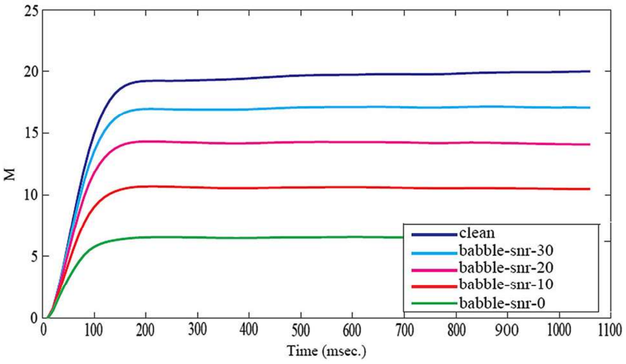

**Fig. 2** The effect of the noise level on the $M$ measure. Averaged over 1344 test utterances. $X$ axis is the $\Delta t$

*Phonemes per second* An intelligible speech comprises about 15−20 phonemes per second. Having a posteriorgram which presents a low phoneme rate, decreases the value of M because the interval $$\Delta$$t may not be large enough to capture two consecutive distinct phonemes.

*Level of noise* If a classifier is fed with noisy data, we found that the posteriors tend toward uniform distribution. This effect results in similar posteriors for different phonemes, and consequently M degrades. Figure 2 shows an example of M measure when the signal is exposed to Babble noise with different SNRs[13].

To rank the classifiers, $M$ is extracted from the posteriorgram of all the classifiers, and the classifiers with the highest values are selected.

### 2.3 Ensemble creation

How many classifiers are needed to get the most out of the fusion? Selecting only the classifier ($N = 1$) with the highest $M$ may not be sufficient and more classifiers are needed to collect more evidence. On the other hand, if we combine the decisions of all the classifiers at the second stage ($2^K - 1$), the effect of noise and low performance classifiers dominates and degrades the final performance. Furthermore, since we assume the noise is not stationary, at each time interval, different numbers of classifiers are to be fused to yield a high accuracy. Therefore, we propose an approach for dynamic selection of $N$ instead of choosing a static number. As we observed, the level of $M$ is correlated with the performance, and noise degrades it. Obviously, in noisy conditions we need to gather more evidence to achieve higher reliability, while in the clean condition a lower number of classifiers could be sufficient to carry enough information. This expectation brings the idea of setting a threshold, *TH*, on the cumulative summation of sorted $M$ values:

$$N = max(X)\,\text{suchthat} \quad \left.\sum_{i=1}^{X} M_i^s\right) < TH, \tag{4}$$

where $M_i^s$ is the ith value of the sorted $M$ vector in descending order, $M^s$. *TH* can be obtained by cross validation on training data, or by using a validation set.

### 2.4 Stream fusion

Once the classifiers are ranked and $N$ is set, we combine the decisions of the $N$ best classifiers. First, we use a phoneme-level HMM model to decode the posteriors. This model is trained on the transcribed speech training data. The hidden states correspond to the phonemes, and at each state only a specific phoneme is observable. This is because for the training we only have the sequence of phoneme labels, and there is no observation. However, while decoding, we assign the observation probability by the classifier posteriors. The transition probability between phonemes $i$ and $j$ are estimated from the transcribed speech in the training data:

$$a_{ij} = P(v_t = j | v_{t-1} = i), \tag{5}$$

where $v_t$ and $v_{t-1}$ are two consequent phonemes. This model is copied for all the selected classifiers to obtain the posteriors and to extract the Viterbi path on each posteriorgram. The forward probabilities are assigned as:

$$\hat{P}_t^n(j) = \alpha P_t^n(j) \sum_i \hat{P}_{t-1}^n(i) a_{ij}, \tag{6}$$

where $\alpha$ is the normalization factor and $P_t^n(j)$ is the posterior probability of phoneme $j$ of classifier $n$ at time $t$, and $\hat{P}_0^n = P_0^n$. The sequence of phonemes for each posteriorgram is extracted using Viterbi decoding. The Viterbi path can be obtained by backtracking from time $t = T$ (end of utterance or observing window) and starting with the phoneme with the highest likelihood $\hat{P}_T^n$. Once the Viterbi path is extracted for each of the selected classifiers, the majority voting at each time $t$ is taken as the final decision.

## 3 Experiments

We tested the methods on two datasets. The first database, TIMIT, consists of transcribed speech of different speakers and sex for English language (Garofolo et al. 1988). The average utterance length is about 4.5 s, equivalent to 450 decisions. Data are partitioned into 3000 utterances for training, 696 for validation (and setting Threshold *TH*), and 1344 for testing. Different types of noise are added to the test utterances to emulate possible distortions in the speech in different environments. They are Car noise at 0 dB, Exhibition hall noise at 5 dB, Factory noise at 10 dB, Restaurant noise at 10 dB, Street noise at 5dB. The Car noise is a band-limited noise concentrated around low frequencies, while the other noises are scattered over all the frequencies.

The second database, RATS, was created by the Linguistic Data Consortium for the DARPA Robust Automatic Transcription System (RATS) project. It contains about 40 h of Levantine Arabic. The whole database includes a clean channel *src* and eight distorted channels A–H created to simulate various communication channels under adversary environments. The narrow-band speech has been downsampled to 8 kHz. For this dataset, we used the data of channel G for the test and designed a 5-band MS-ASR, so there are 31 deep neural netowrks (DNNs) at the second stage. The dataset is partitioned into 60,741 utterances for training and 30,371 utterances for test.

All DNNs in both stages have 3 layers of 1000 nodes in the hidden layer and 40 nodes in their output layer corresponding to the 40 phonemes to be recognized.[1] The input layer in the first stage has 84 nodes corresponding to the number of features, and in the second layer they are different depending on the combination of the classifiers made on the first stage (varies between 25–175 for TIMIT and 25–125 for RATS).

## 4 Results

In Table 1, the phoneme error rate for conventional *fullband*, *7-Stream*, and *best oracle* are provided. The *fullband* approach trains one DNN on all the FDLP features while the *7-Stream* approach trains 7 DNNs on 7 subbands and combine them with another DNN (Tibrewala and Hermansky 1997). Finally, *best oracle* is a supervised hand-picked selection of the best stream for each utterance. The latter one is not applicable online, since the labels are not available. On the clean condition and band-limited noise

[1] We used the Quicknet toolbox developed at the International Computer Science Institute (http://www1.icsi.berkeley.edu/Speech/qn.html).

**Table 1** The phoneme error rate of the conventional full-band and multi-band methods beside *best oracle*

| TIMIT | Fullband | 7-Stream (Tibrewala and Hermansky 1997) | Best oracle |
|---|---|---|---|
| Clean | 31.35 | 31.27 | 23.78 |
| Car (0 dB) | 54.32 | 48.76 | 34.30 |
| Exhall (5 dB) | 70.67 | 71.36 | 61.85 |
| Factory (10 dB) | 68.91 | 69.87 | 59.91 |
| Restaurant (10 dB) | 63.14 | 65.03 | 55.18 |
| Street (5 dB) | 67.26 | 68.47 | 58.08 |
| RATS | Fullband | 5-Stream | Best oracle |
| Channel G | 57.55 | 58.71 | 52.08 |

Supervised selection of the best stream on each utterance

**Table 2** Comparing phoneme error rate with static and dynamic number of classifiers and with the two methods; averaging posteriorgram (AVG) or Parallel HMM (PHMM)

| Fusion | Static N | | | | | | Dynamic N | |
|---|---|---|---|---|---|---|---|---|
| TIMIT | AVG<br>1 | PHMM | AVG (Variani et al 2013)<br>10 | PHMM | AVG (Sharma 1999)<br>127 | PHMM | AVG (N)<br>TH = 180 | PHMM |
| Clean | 32.32 | 29.89 | 29.76 | **28.81** | 31.42 | 28.88 | 29.79 (10.12 ± 1.26) | 28.82 |
| Car (0 dB) | 45.00 | 42.44 | 39.71 | 39.15 | 44.49 | 44.41 | 39.84 (12.66 ± 1.61) | **39.11** |
| Exhall (5 dB) | 73.24 | 69.92 | 68.67 | 68.49 | 69.10 | 68.62 | 68.19 (21.21 ± 3.33) | **67.76** |
| Factory (10 dB) | 70.24 | 68.23 | 67.22 | 66.58 | 68.49 | 67.63 | 67.11 (19.58 ± 2.98) | **66.33** |
| Restaurant (10 dB) | 65.83 | 63.14 | 61.80 | 61.01 | 63.12 | 62.42 | 61.56 (17.72 ± 2.58) | **60.59** |
| Street (5 dB) | 68.64 | 68.23 | 65.43 | 64.65 | 66.73 | 65.97 | 65.17 (19.80 ± 4.50) | **64.53** |
| RATS | 1 | | 5 | | 31 | | TH = 80 | |
| Chanel G | 59.99 | 58.48 | 58.13 | 57.53 | 58.77 | 58.21 | 57.86 (10.72 ± 4.46) | **57.31** |

The minimum phoneme error rate for each corpus is bold typed

**Table 3** Effect of TH on phoneme error rate for different noises on TIMIT dataset

| TH | 120 | 150 | 180 | 210 | 240 | 270 | 300 |
|---|---|---|---|---|---|---|---|
| Validation | 26.91 | 26.77 | 26.71 | 26.73 | **26.69** | **29.69** | **26.69** |
| Clean | 28.88 | 28.91 | 28.82 | 28.81 | 28.81 | **28.80** | 28.81 |
| Car (0 dB) | 39.03 | **38.98** | 39.11 | 39.11 | 39.17 | 39.34 | 38.43 |
| Exhall (5 dB) | 68.04 | 67.78 | 67.76 | **67.61** | 67.71 | 67.66 | **67.61** |
| Factory (10 dB) | 66.61 | 66.44 | 66.32 | 66.36 | 66.36 | **66.30** | 66.40 |
| Restaurant (10 dB) | 60.85 | 60.70 | **60.59** | 60.64 | 60.65 | 60.68 | 60.75 |
| Street (5 dB) | 64.66 | 64.54 | 64.53 | 64.52 | **64.46** | 64.47 | 64.64 |

The minimum phoneme error rate for each corpus is bold typed

(Car), the error rate is lower for *7-Stream* than for *Fullband*. However, for the other noises *7-Stream* is slightly worse.

Table 2 provides the comparison of phoneme error rate for static and dynamic choice of $N$ using the proposed approach. Furthermore, we compare the PHMM decision fusion with the averaging of the posteriors (AVG). We chose $TH = 180$ for TIMIT and $TH = 80$ for the RATS databases. The effect of $TH$ will be discussed later. When the noise is band-limited, such as Car noise, there is about 15 % absolute improvement with respect to the conventional *Fullband* and about 9 % with respect to the *7-Stream* approaches. However, for wide-band noise, the relative improvement is around 3 % absolute with respect to the conventional *Fullband* and about 4 % with respect to the *7-Stream*.

Furthermore, dynamic $N$ performs slightly better than static $N$ in the noisy conditions and is performing the same in the clean condition. As expected, on the noisy conditions more classifiers ($\sim 19$) are selected (due to the low $M$ values) while for the clean condition and band-limited car noise the value $N$ is low ($\sim 10$) close to the preselected value (chosen on the validation set), which is justifiable since many classifiers are still unaffected by the noise. Moreover, comparing the PHMM and AVG fusion approaches, on average the PHMM produces a 1.14 % lower phoneme error rate.

The effect of the chosen $TH$ on the IMIT database is provided in Table 3. In general, a high $TH$ results in a larger $N$, and consequently, a lower performance on noisy data. According to the error rates on the validation set, the value of $TH = 240$ appears reasonable. Nevertheless, not a specific value is optimal in all the conditions, suggesting for further improvement. However, setting any value between 120 and 300 for $TH$ yields lower phoneme error rate with respect to the other approaches (cf. Table 2).

## 5 Discussion

In this work, we improved the performance of MS-ASR by using an automatic selection of the number of the classifiers and a phoneme-level language model.

Note that, in our experiments, we have used DNNs as the base classifiers without applying enhancement techniques such as dropout to achieve higher accuracy. Therefore, the base performance on TIMIT is not as good as state-of-the-art methods [e.g., using Deep Belief Net (Mohamed et al. 2012)].

In terms of compputational cost, MS-ASR can be run in a parallel mode, where each stream is processed on one computation node. The additional cost of MS-ASR appeals during the training phase which is done offline. In addition, the dynamic or static selection of the number of classifiers helps to reduce the computational cost of the fusion of the posteriorgrams (fusing a limited number of posteriorgrams instead of all of them).

The future work is to evaluate the approach on other datasets as well as to enhance the performance of the base classifiers. On the other hand, as it is observed in Tables 1 and 2, there is still a gap between the obtained results and the oracle selection. This suggests more investigation on the performance monitoring to improve it. To do so, one way is to improve $M$ by weighting according to some statistics on the training data (such as probability of being chosen as the best stream). Also, we found that one-point $M$ (i. e., only with $\Delta t = 250$ ms) is not robust and therefore, not sufficiently reliable. Taking an average of $M$s in a window (e. g., 200–800 ms) could yield higher performance. Furthermore, an improved version of $M$ proposed in (Mallidi et al. 2015) can be used for more robust ranking. Alternatively, we may improve the fusion of streams after extracting the most likely path by choosing another combination method such as naïve Bayesian fusion rather than simple majority voting or by using language models for each selected classifier.

## 6 Conclusion

In this paper, we proposed an approach to select the best $N$ classifiers from an ensemble of classifiers which are trained on subbands of speech signals for improved ASR. The approach is based on the '$M$ value' which represents the performance of a classifier at runtime. Additionally, a bigram phoneme model was used to obtain the phonemes in an utterance for each classifier output (posteriorgram), and we combine the results through majority voting. We applied our approach on two speech datasets (TIMIT and RATS) and we yield improvements comparing to the existing MS-ASR systems (Tibrewala and Hermansky 1997; Variani et al. 2013) as well as conventional Fullband ASR.

**Acknowledgments** The authors would like to thank Professor Hynek Hermansky for his valuable comments.

## References

Allen, J. (1994). How do humans process and recognize speech? *IEEE Transactions on Speech and Audio Processing*, *2*(4), 567–577.

Bourlard, H. & Dupont, S. (1997). Subband-based speech recognition. In *22nd International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol 2 (pp 1251–1254). Munich, Germany

Bourlard, H., Dupont , S., Ris, C. (1997). Multi-stream speech recognition. Tech. Rep. IDIAP-RR 96-07, IDIAP

Fletcher, H. (1953). *Speech and hearing in communication*. New York: Krieger.

Furui, S. (1992). Towards robust speech recognition under adverse conditions. In *ESCA Workshop on Speech Processing in Adverse Conditions* (pp. 31–41)

Ganapathy, S., & Hermansky, H. (2012). Temporal resolution analysis in frequency domain linear prediction. *The Journal of the Acoustical Society of America*, *132*(5), 436–442.

Garofolo, J. S., et al. (1988). Getting started with the darpa timit cd-rom: An acoustic phonetic continuous speech database. National Institute of Standards and Technology (NIST), Gaithersburgh, MD, p. 107

Geiger, J. T., Zhang, Z., Weninger, F., Schuller, B., Rigoll, G. (2014). Robust speech recognition using long short-term memory recurrent neural networks for hybrid acoustic modelling. In: *Proceedings of 15th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, ISCA, Singapore, Singapore

Giacinto, G., Roli, F. (2000). Dynamic classifier selection. In *Multiple Classifier Systems* (pp. 177–189). Springer

Hermansky, H. (2013). Multistream recognition of speech: Dealing with unknown unknowns. *IEEE Proceedings*, *101*(5), 1076–1088.

Hermansky, H., & Morgan, N. (1994). Rasta processing of speech. *IEEE Transactions on Speech and Audio Processing*, *2*(4), 578–589.

Hermansky, H., Tibrewala, S., Pavel, M. (1996). Towards ASR on partially corrupted speech. In *Fourth International Conference on Spoken Language (ICSLP)*, vol 1 (pp. 462–465). IEEE, Philadelphia, PA, USA

Hermansky, H., Variani, E., Peddinti, V. (2013). Mean temporal distance: Predicting ASR error from temporal properties of speech signal. In *38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, Vancouver, Canada

Ikbal, S., Misra, H., Hermansky, H., & Magimai-Doss, M. (2012). Phase autocorrelation (PAC) features for noise robust speech recognition. *Speech Communication*, *54*(7), 867–880.

Mallidi, S. H., & Hermansky, H. (2016). Novel neural network based fusion for multistream ASR. In *41st International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 5680–5684). Shanghai, China: IEEE.

Mallidi, S. H., Ogawa, T., & Hermansky, H. (2015). Uncertainty estimation of dnn classifiers. In *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)* (pp. 283–288). USA: Arizona.

Mesgarani, N., Thomas, S., Hermansky, H. (2011). Adaptive stream fusion in multistream recognition of speech. In *12th Annual Conference of the International Speech Communication Association (InterSpeech)*. Portland, Oregon

Mohamed, A., Dahl, G., & Hinton, G. (2012). Acoustic modeling using deep belief networks. *IEEE Transactions on Audio Speech and Language Processing*, *20*(1), 14–22.

Sharma, S. R. (1999). Multi-stream approach to robust speech recognition. PhD thesis

Tibrewala, S., Hermansky, H. (1997). Sub-band based recognition of noisy speech. In *22nd IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol 2 (pp. 1255–1258). Munich, Germany,

Variani, E., Li, F., Hermansky, H. (2013). Multi-stream recognition of noisy speech with performance monitoring. In *14th Annual Conference of the International Speech Communication Association (InterSpeech)*. Lyon, France

Weninger, F., Geiger, J., Wöllmer, M., Schuller, B., & Rigoll, G. (2014). Feature enhancement by deep LSTM networks for ASR in reverberant multisource environments. *Computer Speech and Language*, *28*(4), 888–902.

Wöllmer, M., Weninger, F., Geiger, J., Schuller, B., & Rigoll, G. (2013). Noise robust ASR in reverberated multisource environments applying convolutive NMF and long short-term memory. *Computer Speech and Language*, *27*(3), 780–797.