
C++/MATLAB code to estimate the IBM and long term SNRs

Written by Arun Narayanan, 2012
The Ohio State University, Perception and Neurodynamics Lab (PNL)

Reference:

Narayanan A. and Wang D.L. (2012): A CASA based system for long-term SNR estimation. *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, pp. 2518-2527.

To extract the files, use the following command:
tar -xzf longTermSNR.tar.gz

Input

sig		The signal. Can be the filename (in MSWAV format), or the actual signal itself. The signal is assumed to be sampled at 16 kHz.	(required)
params*		A structure with multiple fields	(optional)
	params.ebm	The estimated mask or the filename containing the mask. If this parameter is set, the mask is not estimated. The mask should be saved as an ASCII file, with each row containing data for one channel ([channel x frames] format in MATLAB terminology). As of now, the program assumes a 128 channel mask (hard-coded). Note that the mask can contain real values, not just binary values.	(optional)
	params.ebm_ke	The mask estimated using the CASA algorithm (Hu&Wang, IEEE-TASLP, 2011). Could be the filename or the mask itself.	(optional)
	params.broadband	This field should exist for broadband SNR estimation.	(optional)
	params.subband	This field should exist for subband SNR estimation.	(optional)
	params.tmpDir	The directory to store temporary files. If not passed, the current directory (`pwd`) will be used instead. Ideal to set it somewhere on your /scratch directory.	(optional)
	params.numChan	Number of frequency channels, defaults to 128. You can also use 64. The results in the paper use the 128 channel mask for SNR estimation.	(optional)

*If the argument params is omitted, only the 128-channel mask gets estimated.

Output

masks		The estimated masks. Contains c128 and c64 fields for 128 channel and 64 channel estimated IBMs, respectively. 64 channel mask is created by sub-sampling the 128 channel mask if numChan is set to 128. If numChan is 64, this field will not contain a 128 channel mask.
snr		The estimated SNRs. May contain the following fields
	snr.filtered	The filtered SNR (SNR in the cochleagram domain)
	snr.broadband	The broadband SNR (SNR in the time domain)
	snr.subband	The subband SNRs (SNR across 64 sub-bands)

Following assumptions are made:
1. The subband SNRs are estimated across 64 subbands defined by the gammatone filterbank.
2. 20 msec window, a sampling rate of 16 kHz and a frequency range of 50--8000 Hz.

Example usage (in MATLAB):

```
% First read the signal:
sig = dlmread('Examples/sx364.val');

% Next, cd to the directory containing the code. Note that this is important as function calls use 'relative' directory name. You should NOT add the directory to your path and then call snrEst function.

% 1. If only the mask needs to estimated:
clear params;
curDir=pwd;
params.tmpDir = [curDir,'/Examples/Tmp/'];
[masks,snrs] = snrEst(sig,params);

% 2. If you have already estimated the mask using an alternative strategy, and want to estimate only the broadband SNR:
clear params;
curDir=pwd;
params.tmpDir = [curDir,'/Examples/Tmp/'];
params.ebm = '<Mask directory>/mask.ibm';
params.broadband = '';
[masks,snrs] = snrEst(sig,params);

% 3. If you want to estimate the masks and the SNRs:
clear params;
curDir=pwd;
params.tmpDir = [curDir,'/Examples/Tmp/'];
params.broadband='';
params.subband='';
[masks,snrs] = snrEst(sig,params);

% 4. If you have a CASA mask, and want to improve it by adding the speech enhancement based mask:
clear params;
curDir=pwd;
params.tmpDir = [curDir,'/Examples/Tmp/'];
params.ebm_ke = '<CASA mask directory>/<CASA mask filename>';
[masks,snrs] = snrEst(sig,params);

% 5. For the 64 channel version, simply add the following member to params:
params.numChan=64;
% The rest of the settings remain unchanged.
```

Additional Comments:

---

1. Broadband SNR estimation results of the proposed system from the paper:

Noise	-10 dB	-5 dB	0 dB	5 dB	10 dB	15 dB	Average
White	1.68	0.80	0.52	0.55	0.73	1.10	0.89
Car	0.01	0.00	0.00	0.00	0.01	0.06	0.01
Babble	1.82	0.72	0.41	0.41	0.57	0.90	0.81
Factory	1.16	0.77	0.44	0.36	0.59	1.01	0.72
Average	1.17	0.57	0.34	0.33	0.47	0.77	0.61

Filtered SNR estimation results of the proposed system from the paper:

Noise	-10 dB	-5 dB	0 dB	5 dB	10 dB	15 dB	Average
White	1.65	0.78	0.46	0.54	0.71	1.09	0.87
Car	0.35	0.07	0.01	0.01	0.06	0.24	0.12
Babble	1.96	0.86	0.40	0.50	0.60	0.93	0.88
Factory	1.30	0.97	0.54	0.50	0.72	1.13	0.86
Average	1.31	0.67	0.35	0.39	0.52	0.85	0.68

2. This version uses a considerably faster implementation of the Tandem algorithm (Hu&Wang (2011); credits for a faster implementation goes to Ke Hu) with an additional segmentation stage. The results are slightly different from those reported in the paper (on average, 0.02 dB worse in broadband SNR estimation performance and 0.05 dB worse in filtered SNR estimation performance) when the same number of frequency channels are used (128) as used originally. For the 64-channel version, the performance is worse, on average, by 0.09 dB and 0.21 dB, respectively, for broadband and filtered SNR estimation. This may be because the parameters were specifically tuned for the 128-channel version (for e.g., the SNR threshold used while estimating the speech enhancement based mask). It has been noted by the author of the tandem algorithm that the two versions produce similar performance in terms of mask estimation. So with some tuning, the 64 channel version's SNR estimation results may turn out to be at least as good as the 128-channel results.

The results obtained using the faster implementation are as follows:

A. The 128 channel version:
Broadband SNR estimation:

Noise	-10 dB	-5 dB	0 dB	5 dB	10 dB	15 dB	Average
White	1.46	0.71	0.52	0.59	0.83	1.20	0.89
Car	0.01	0.00	0.00	0.00	0.01	0.07	0.01
Babble	2.21	0.73	0.42	0.45	0.68	1.05	0.93
Factory	1.06	0.76	0.42	0.38	0.58	1.05	0.71
Average	1.18	0.55	0.34	0.36	0.52	0.84	0.63

Filtered SNR estimation:

Noise	-10 dB	-5 dB	0 dB	5 dB	10 dB	15 dB	Average
White	1.39	0.65	0.51	0.57	0.78	1.15	0.84
Car	0.56	0.11	0.02	0.01	0.07	0.36	0.19
Babble	2.41	0.82	0.47	0.52	0.67	1.18	1.01
Factory	1.19	0.87	0.53	0.52	0.77	1.26	0.85
Average	1.40	0.61	0.38	0.40	0.57	0.99	0.73

B. The 64 channel version:
Broadband SNR estimation:

Noise	-10 dB	-5 dB	0 dB	5 dB	10 dB	15 dB	Average
White	1.24	0.74	0.66	0.72	0.95	1.43	0.96
Car	0.01	0.00	0.00	0.01	0.04	0.23	0.05
Babble	2.90	1.13	0.58	0.46	0.61	0.92	1.10
Factory	1.24	0.66	0.32	0.34	0.54	1.01	0.69
Average	1.35	0.63	0.39	0.38	0.54	0.90	0.70

Filtered SNR estimation:

Noise	-10 dB	-5 dB	0 dB	5 dB	10 dB	15 dB	Average
White	1.21	0.64	0.64	0.72	0.95	1.38	0.92
Car	0.44	0.31	0.50	0.70	0.79	0.92	0.61
Babble	3.16	1.21	0.63	0.56	0.65	1.03	1.20
Factory	1.32	0.79	0.48	0.49	0.76	1.20	0.84
Average	1.53	0.74	0.56	0.62	0.79	1.13	0.89

Acknowledgements

The authors would like to thank G. Hu, K. Hu and K. Han for providing software implementations of the works described in Hu&Wang, IEEE-TASLP, 2010 and Hu&Wang, IEEE-TASLP, 2011; R. Hendriks, R. Heusdens, J. Jensen and J. Erkelens for sharing MATLAB code for the works described in Erkelens *et al.*, IEEE-TASLP, 2007 and Hendriks *et al.*, IEEE-ICASSP, 2010.