

Soft Margin Estimation on Improving Environment Structures for Ensemble Speaker and Speaking Environment Modeling

Yu Tsao

National Institute of Information and
Communications Technology
Kyoto, Japan
+81-90-6827-2764
yu.tsao@nict.go.jp

Jinyu Li

Microsoft
One Microsoft Way
Redmond, WA, USA
+1-404-861-9633
jinyuli@microsoft.com

Chin-Hui Lee

School of ECE
Georgia Institute of Technology
Atlanta, GA, USA
+1-404-894-7468
chl@ece.gatech.edu

Satoshi Nakamura

National Institute of Information and
Communications Technology
Kyoto, Japan
+81-774-63-6224
satoshi.nakamura@nict.go.jp

ABSTRACT

Recently, we proposed an ensemble speaker and speaking environment modeling (ESSEM) approach to enhance the robustness of automatic speech recognition (ASR) under adverse conditions. The ESSEM framework comprises two phases, offline and online phases. In the offline phase, we prepare an environment structure that is formed by multiple sets of hidden Markov models (HMMs). Each HMM set represents a particular speaker and speaking environment. In the online phase, ESSEM estimates a mapping function to transform the prepared environment structure to a set of HMMs for the unknown testing condition. In this study, we incorporate the soft margin estimation (SME) to increase the discriminative power of the environment structure in the offline stage and therefore enhance the overall ESSEM performance. We evaluated the performance on the Aurora-2 connected digit database. With the SME refined environment structure, ESSEM provides better performance than the original framework. By using our best online mapping function, ESSEM achieves a word error rate (WER) of 4.62%, corresponding to 14.60% relative WER reduction (from 5.41% to 4.62%) over the best baseline performance of 5.41% WER.

Categories and Subject Descriptors

I.2.7 Natural Language Processing

General Terms

Algorithms, Experimentation, Languages, Theory.

Keywords

ASR, noise robustness, ESSEM, SME, model adaptation.

1. INTRODUCTION

For an automatic speech recognition (ASR) system, robustness under adverse environments is a key issue to its success. The difficulty of handling this issue is that a testing condition usually contains multiple mismatch sources, which may come from speaker variations and speaking environment noises. Although some parametric functions can specify particular distortions well, an unknown combination of multiple distortions can be very complex and hard to characterize. Until now, many approaches have been proposed to increase the ASR robustness under adverse conditions. Among them, a class of approaches attempts to handle the mismatches in the model space [1, 2]. These approaches can be classified into two groups. The first group intends to prepare a set of acoustic models that is robust to environmental changes in the offline. One good direction is to collect speech data from a wide range of different speaker and speaking environments for training the acoustic models. Multi-style training scheme is a successful example [3]. Another direction is to adopt a good model-training method, such as discriminative training. Effective methods include minimum classification error (MCE) [4] and soft margin estimation (SME) [5, 6]. It is well known that the discriminative training can refine the maximum likelihood (ML) trained acoustic models to achieve better ASR performance. The second group is to adapt the acoustic models to match the testing condition in the online. Successful examples include MAP [7-8], MLLR [9], and stochastic matching [1, 2]. Mostly, these two groups of approaches can be combined to achieve better overall performance, i.e., developing a decent initial acoustic model in the offline and performing a model adaptation in the online.

More recently, we extended the stochastic matching algorithm to an ensemble speaker and speaking environment modeling (ESSEM) algorithm to improve the ASR robustness under adverse conditions [10-12]. Different to the multi-style training method

that estimates a set of acoustic models collectively, ESSEM uses training data to prepare multiple sets of environment-specific acoustic models, with each set characterizing its corresponding environment more precisely. In the offline, ESSEM uses these multiple sets of acoustic models to establish an environment structure; this environment structure provides prior knowledge for the testing conditions. In the online, ESSEM estimates a mapping function to transform the environment structure into one HMM set. With the environment structure providing good prior information, ESSEM can achieve good performance even with some simple mapping functions [11]. In this study, we incorporate the SME algorithm [5, 6] in the ESSEM framework to increase the discriminative ability of the environment structure. The ESSEM environment structure is first prepared using the maximum likelihood (ML) training, followed by an SME refining procedure. In the online, ESSEM estimates a mapping function to transform the SME-refined environment structure for the target HMM set.

2. ESSEM FRAMEWORK

First, we review the two phases of ESSEM—offline environment space construction and online mapping function estimation.

2.1 Offline Environment Space Construction

In the offline, we collect P sets of speech data, with each set of speech data representing a particular speaker and speaking environment. For the real-world implementation, it might be too prohibitive to collect speech data for a wide range of different combinations of adverse conditions and noise levels so that we propose to artificially simulate the data at specific distortions and signal-to-noise (SNR) levels [10]. With the P sets of speech data, we accordingly train P sets of HMMs, Λ_p , $p=1\dots P$. Next, the entire set of mean parameters in a set of HMMs is concatenated into a super-vector, V_p , $p=1, \dots, P$. These P super-vectors form an ensemble speaker and speaking environment (ESS) structure, Ω_V , where $\Omega_V = \{V_1 V_2 \dots V_P\}$.

Our previous study has introduced several approaches to enhance the ESS structure in the offline stage [10]. To improve the structure of the ESS space, we proposed environment clustering (EC) and environment partitioning (EP) approaches; to increase the discriminative ability of the ESS space, we derived the MCE-based intra-environment (intraEnv) training and inter-environment (interEnv) training. Figure 1 illustrates the intraEnv and interEnv training for enhancing discrimination of an ESS space [10]. Each of the above offline approaches can provide further improvement individually, and the combination of them can give the best overall performance [10]. In this paper, we propose an SME-based intraEnv training to further enhance the discriminative capability of parameters in the ESS space over the MCE-based intraEnv training and therefore achieve better ESSEM performance.

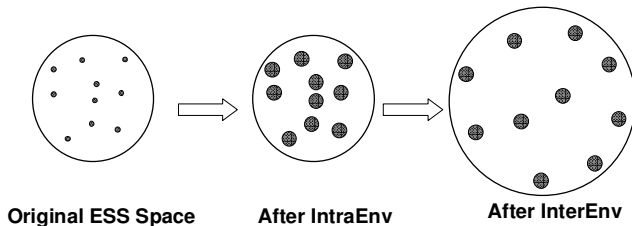


Figure 1. IntraEnv and interEnv training.

2.2 Online Mapping Function Estimation

In the ESSEM online process, we estimate the target super-vector, V_Y , for the testing environment through a mapping function, G_ϕ :

$$V_Y = G_\phi(\Omega_V). \quad (1)$$

The form of G_ϕ depends on the amount of adaptation data and distortion types. We can estimate the nuisance parameters $\hat{\phi}$ in G_ϕ based on the ML criterion:

$$\hat{\phi} = \underset{\phi}{\operatorname{argmax}} P(F_Y | \Omega_V, \phi, W), \quad (2)$$

where W is the transcription corresponding to the testing utterances, F_Y . With the estimated target super-vector, V_Y , we can build the set of acoustic models for the testing condition. In the following description, we will present four types of online mapping functions, including best first (BF), linear combination (LC), linear combination with a correction bias (LCB) and multiple cluster matching (MCM).

2.2.1 Best First (BF)

With the prepared environment structure, the BF function determines the super-vector that best matches to the testing condition based on a maximum likelihood (ML) criterion:

$$V_Y = \underset{p}{\operatorname{argmax}} P(F_Y | V_p), p=1, 2, \dots, P. \quad (3)$$

In the implementation, we can use a parallel decoding scheme or a tree structure to facilitate the BF process.

2.2.2 Linear Combination (LC)

When using LC as the online mapping function, the target super-vector is estimated based on a linear combination of the super-vectors prepared in the ESS space.

$$V_Y = \sum_{p=1}^P \hat{w}_p V_p, \quad (4)$$

where \hat{w}_p is the p -th weighting coefficient in the linear combination function. Similarly, we estimate the set of weighting coefficients based on the ML criterion:

$$\{\hat{w}_p\}_{p=1}^P = \underset{\{w_p\}_{p=1}^P}{\operatorname{argmax}} P(F_Y | \sum_{p=1}^P w_p V_p). \quad (5)$$

2.2.3 Linear Combination with a Correction Bias (LCB)

The BF and LC mapping functions can enable ESSEM to well characterize distortions that are prepared in the training set. However, the performance is limited when dealing with new distortion types that are not collected in the training set. Therefore, we derived more complex mapping functions. First, we improve LC in Eq-(4) by incorporating a correction bias \hat{b} :

$$V_Y = \sum_{p=1}^P \hat{w}_p V_p + \hat{b}. \quad (6)$$

Similarly, the weighting coefficients and correction bias can be estimated base on the ML criterion:

$$\{\{\hat{w}_p\}_{p=1}^P; \hat{b}\} = \underset{\{\{w_p\}_{p=1}^P; b\}}{\operatorname{argmax}} P(F_Y | \sum_{p=1}^P w_p V_p + b). \quad (7)$$

2.2.4 Multiple Cluster Matching (MCM)

We derived the MCM mapping function based on the ensemble estimator (EE) algorithm [11, 13]. The MCM mapping process consists of two steps. In the first step, a mapping function transforms the original ESS space to a new environment structure, Ω_{V_E} . This environment structure has a better coverage and resolution to characterize the testing condition. In the second step, another mapping function, \mathbf{G}_{φ_E} transforms the new environment structure to the target super-vector, V_Y :

$$V_Y = \mathbf{G}_{\varphi_E}(\Omega_{V_E}), \quad (8)$$

where

$$\varphi_E = \underset{\varphi_E}{\operatorname{argmax}} P(F_Y | \varphi_E, \Omega_{V_E}), \quad (9)$$

This particular mapping function provides the best performance in our ESSEM evaluations. More details on the above four mapping functions have been provided in our previous study [11, 12].

3. SME ON REFINING ESS STRUCTURES

In this section, we first introduce the SME algorithm; then we discuss applying SME for intraEnv training to increase the discriminative ability of ESS spaces.

3.1 SME Algorithm

Originated from the statistical learning theory [14], SME considers the test risk to be bounded by two terms, an empirical risk and a generalization term (generalization term is bounded by a decreasing function of margin [14]). During optimization, SME not only minimizes the empirical risk but also maximizes the margin. Therefore, the objective function for SME is defined as:

$$L^{SME}(\rho, \Lambda) = \frac{\lambda}{\rho} + R_{emp}(\Lambda) = \frac{\lambda}{\rho} + \frac{1}{U} \sum_{u=1}^U l(F^u, \Lambda), \quad (10)$$

where Λ denotes HMM parameters, $l(F^u, \Lambda)$ is a loss function for the u -th utterance F^u , U is the number of training utterances, ρ is the soft margin, and λ is a coefficient to balance the soft margin maximization and the empirical risk minimization. The loss function is defined by a hinge loss function ($(x)_+ = \max(x, 0)$) as:

$$l(F^u, \Lambda) = \begin{cases} \rho - d(F^u, \Lambda) \\ \rho - d(F^u, \Lambda), & \text{if } \rho - d(F^u, \Lambda) > 0 \\ 0, & \text{otherwise} \end{cases}, \quad (11)$$

with the separation measure d defined as:

$$d(F^u, \Lambda) = \frac{1}{n_u} \sum_r \log \left[\frac{P_\Lambda(F^{ur} | S_u)}{P_\Lambda(F^{ur} | \hat{S}_u)} \right] I(F^{ur} \in D_u), \quad (12)$$

where D_u is the frame set in which the frames have different labels in the competing strings; n_u is the number of frames in D_u ; $I(\cdot)$ is an indicator function; F^{ur} is the r -th frame of utterance F^u ; $P_\Lambda(F^{ur} | S_u)$ and $P_\Lambda(F^{ur} | \hat{S}_u)$ are the likelihood scores for the target string S_u and the most competing string \hat{S}_u , respectively. By plugging Eq-(11) and Eq-(12) into Eq-(10), the final objective function to minimize for the SME algorithm becomes:

$$L^{SME}(\rho, \Lambda) = \frac{\lambda}{\rho} + \frac{1}{U} \sum_{u=1}^U [\rho - d(F^u, \Lambda)]_+. \quad (13)$$

3.2 SME for Refining Environment Structures

In this study, we introduce the SME-based intraEnv training to refine the ESS space. Similar to the MCE-based intraEnv training [10], each environment-specific HMM set is first trained on ML and then refined by SME. By only considering mean parameters, we derive the following objective function for the SME refinement:

$$L^{SME}(\rho, V_p) = \frac{\lambda}{\rho} + \frac{1}{U} \sum_{u=1}^U [\rho - d(F_p^u, V_p)]_+, p = 1, \dots, P, \quad (14)$$

where F_p^u is the u -th training utterances in the p -th environment.

In this paper, we limit our discussion on applying SME to enhance the discriminative ability of each individual super-vector. Incorporating SME for interEnv training to increase the separation among different super-vectors will be studied in the future.

4. EXPERIMENTS

This section presents experimental setup and results. As reported in our previous study, after MCE-based intraEnv and interEnv training, the ESS space possesses better discriminative power and enables ESSEM to provide better performance than the original ML-trained ESS space [10]. In the following experiments, we will compare the ESSEM performance achieved by SME-based and by the original MCE-based intraEnv training.

4.1 Experimental Setup

The ESSEM performance was evaluated on the Aurora-2 connected digit database [15]. We used the multi-condition training set to build environment-specific HMMs and to construct the ESS space. This training set includes the same four types of noise as in test set A (Subway, Babble, Car, Exhibition noises) sampled at four SNR levels (20 dB, 15 dB, 10 dB, and 5 dB) along with clean data. Therefore we have speech data for 17 (4×4+1) different speaking environments. We further divided the training set by each speaker's gender identity, and finally, we can have speech data for 34 (17×2) different speaker and speaking environments. We used a modified ETSI advanced front-end (AFE) for feature extraction [16] and the complex back-end configuration for HMM topology [17]. All digits were modeled by 16-state whole word models with each state characterized by 20 Gaussian mixture components. The silence and the short pause were modeled by 3 states and 1 state, respectively, with each state characterized by 36 Gaussian mixture components. The full evaluation set of Aurora-2 was used to test performance, and we only reported the testing results from 0dB to 20dB in this paper. In the training stage, two gender-dependent (GD) HMM sets were first trained by using the ML estimation. Then, 17 environment-specific HMM sets for each gender were obtained by adapting (we used the MAP algorithm) mean vectors from that GD HMM set. Therefore, two ESS spaces corresponding to the two genders were prepared. We have included more details on the experimental setup in our previous study [10-12]. In the following, we denote the results of ESSEM using this original ESS space as "ML" results. Next, we refined the ML-based ESS space with MCE-based intraEnv training and SME-based intraEnv training and obtained two refined ESS spaces; both ESS spaces were then retrained by MCE-based interEnv training. For simplicity, we call the ESSEM results of using SME-based intraEnv and MCE-based intraEnv training as "SME" and "MCE" results, respectively.

In addition to the *intraEnv* and *interEnv* training, we implemented the EC approach to improve the overall performance. In the offline, we constructed an EC tree, with each node consisting of a group of environments. In the online, a cluster selection (CS) process was performed to select the group of environments that best matches to the testing condition. The selected environments then form a new ESS space for the online transformation process. More details about EC can be found in our previous study [10].

4.2 Experimental Results

In this section, we first compare parameter separations of the ESS spaces trained with ML, MCE and SME. Next, we present the ESSEM recognition performance on the Aurora-2 task.

4.2.1 Parameter Separation

We adopted an accumulated divergence distance to quantitatively measure the parameter separation in individual super-vector [6]. In this paper, we chose the environment of “Exhibition noise, 10dB SNR, female speakers” as a representative and illustrated its corresponding accumulated divergence distances for “ML”, “MCE”, and “SME” in Table 1. The comparison between “ML” versus “MCE” and “SME” *intraEnv* training corresponds to the left versus the middle panels in Figure 1. Please note that for the *intraEnv* training, we only adjust mean parameters. Therefore, the same variance parameters were used when calculating the accumulated divergence distances.

From Table 1, it is clear that after MCE-based *intraEnv* training, parameter separation in the HMM set is increased over that without *intraEnv* training (the “ML” results). Additionally, it is clear that SME-based *intraEnv* training can further increase the separation in the HMM set over MCE-based *intraEnv* training. These observations indicate that SME has better capability to enhancing discrimination among model parameters than MCE and ML, which is actually consistent to our previous study [6].

Table 1. Divergence distances for different training methods

Training Method	ML	MCE	SME
Accumulated Divergence	67.18	68.09	68.83

4.2.2 Recognition Performance

In the previous section, we demonstrated the parameter separations of environment structures estimated by “ML”, “MCE” and “SME” training criteria. However, the separation measure is not the only indicator of accuracy. In this section, we present the recognition results, in average word error rates (WER), of ESSEM using ESS spaces estimated with the three different training methods. We evaluated the ESSEM recognition performance in a per-utterance unsupervised self-adaptation mode on a gender dependent (GD) system [10-12]. Each testing utterance was first decoded into an *N*-best list (*N*=8) and then used for ESSEM adaptation. The two GD HMM sets were used for an automatic gender identification (AGI) process to determine every speaker’s gender. During testing, we used every incoming testing utterance to: 1) identify speaker’s gender and select the corresponding gender-specific HMMs; 2) perform the CS process to locate the most suitable EC-clustered ESS space; 3) implement the ESSEM adaptation in an unsupervised self-adaptation manner; 4) test recognition with the ESSEM-adapted acoustic models.

4.2.2.1 Baseline

First, we show the baseline results of the three systems in Table 2. Because the two sets of GD HMMs were also refined by *intraEnv* training, we can see that the baselines of “SME” and “MCE” give better overall performance than that of “ML”. However a close-up investigation reveals that when comparing to “ML”, although “MCE” achieves better performance in SetA and SetB, it gives worse performance in SetC (which contains additional channel distortion). It should be the natural limitation of the MCE training that aims at increasing distance among modeling units only according to the available training data [4, 6]. On the other hand, we find that “SME” can provide better performance than “ML” for all the three testing sets. These results confirm the outstanding generalization capability of SME training for HMMs. In the following, we will further investigate the ESSEM performance using different forms of mapping function.

Table 2. Baseline: WER (in %) from 0dB to 20dB

	SetA	SetB	SetC	Overall
Baseline (ML)	5.11	5.51	6.42	5.53
Baseline (MCE)	5.11	5.38	6.56	5.51
Baseline (SME)	5.05	5.31	6.31	5.41

4.2.2.2 LC mapping function

In this section, we evaluated the ESSEM results using LC (in Eq-(4)) as the mapping function and listed the results in Table 3. From Table 3, we find similar results to the baseline results in Table 2. First, “MCE” achieves better performance than “ML” in SetA and SetB, while worse in SetC. Meanwhile, “SME” gives the best performance among the three training methods for all the three test sets. These results again verify that the SME training has a promising capability to increase the generalization of the ESS space to handle different types of distortion.

Table 3. ESSEM with LC: WER (in %) from 0dB to 20dB

	SetA	SetB	Set C	Overall
LC (ML)	4.71	5.21	5.60	5.09
LC (MCE)	4.64	4.99	5.64	4.98
LC (SME)	4.52	4.87	5.48	4.85

4.2.2.3 Other forms of mapping function

We further compared the SME-based and MCE-based *intraEnv* training using other forms of mapping function. Table 4 summarizes the performance of ESSEM using BF, LCB, and MCM. From Table 4, it is clear that SME training gives better performance than MCE training when using a same type of mapping function. For example when using LCB, SME-based training achieves 4.72% WER, which is clearly better than 4.85% WER achieved by MCE-based training. Moreover, we can see that for both MCE-based and SME-based training, ESSEM provides better overall recognition performance when a more complex mapping function is used. In this paper, the combination of SME-based *intraEnv* training with the MCM mapping function achieves the best performance and provides 4.62% WER on the Aurora-2 task. In Table 5, we include the detailed results of “SME+MCM” for the 50 different testing conditions in Aurora-2.

Table 4. ESSEM with different mapping functions

	SetA	SetB	SetC	Overall
BF (MCE)	4.98	5.22	6.38	5.35
LCB (MCE)	4.62	4.95	5.13	4.85
MCM (MCE)	4.48	4.95	5.00	4.77
BF (SME)	5.00	5.12	6.22	5.29
LCB (SME)	4.48	4.85	4.96	4.72
MCM (SME)	4.38	4.71	4.90	4.62

5. CONCLUSION

In this paper, we incorporate the SME algorithm to perform intraEnv training so as to refine the environment structure in the ESSEM framework. From the experimental results, we first verify that SME can refine the environment structure by increasing the discriminative power. Moreover, by using the SME refined environment structure, ESSEM can achieve better performance than by using either an ML-based or MCE-based trained environment structure. When combined with our best online mapping function, multiple cluster matching (MCM), ESSEM provides a 4.62% WER, which corresponds to 14.60% WER reduction over our best SME baseline result as shown in Table 2. Based on the success of SME-based intraEnv training, our first future work is to derive an SME-based interEnv training scheme. We believe that by maximizing the margins between different speaker and speaking environments, the discriminative power and coverage of the environment structure will be further enhanced. Next, we will research on applying the SME algorithm to refine the online mapping function estimation in the future.

6. REFERENCES

- [1] Sankar, A. and Lee, C.-H., "A maximum-likelihood approach to stochastic matching for robust speech recognition", IEEE Trans. on Speech Audio Proc., vol. 4, pp.190-202, 1996.
- [2] Suredran, A. C., Lee, C.-H., and Rahim, M., "Nonlinear compensation for stochastic matching", IEEE Trans. on Speech Audio Processing, vol. 7, pp.643-655, Nov.1999.
- [3] Lippmann, R. P., Martin, E. A., and Paul, D. B., "Multi-style training for robust isolated-word speech recognition", Proc. ICASSP 1987, Dallas, TX, pp. 705-708, Apr. 1987.
- [4] Juang, B.-H., Chou, W., and Lee, C.-H., "Minimum classification error rate methods for speech recognition", IEEE Trans. on Speech and Audio Proc., pp. 257-265, 1997.
- [5] Li, J., Yuan, M., and Lee, C.-H., "Approximate test risk bound minimization through soft margin estimation", IEEE Trans. on Audio, Speech, and Language Processing, vol. 15, no. 8, pp. 2393-2404, 2007.
- [6] Li, J., 2008. Soft margin estimation for automatic speech recognition, Ph.D. Dissertation, School of ECE, Georgia Institute of Technology.
- [7] Lee, C.-H. and Huo, Q., "On adaptive decision rules and decision parameter adaptation for automatic speech recognition", Proc. IEEE, vol. 88, pp. 1241-1269, 2000.
- [8] Gauvain, J.-L. and Lee, C.-H., "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains", IEEE Trans. on Speech Audio Proc., vol. 2, pp. 291-99, 1994.
- [9] Leggetter, C. J. and Woodland, P. C., "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models", Computer Speech and Language, vol. 9, pp.171-185, 1995.
- [10] Tsao, Y. and Lee, C.-H., "An ensemble speaker and speaking environment modeling approach to robust speech recognition", IEEE Trans. on Audio, Speech, and Language Proc., vol. 17, pp.1025-1037, 2009.
- [11] Tsao, Y. and Lee, C.-H., "Improving the ensemble speaker and speaking environment modeling approach by enhancing the precision of the online estimation process", Proc. Interspeech, pp. 1265-1268, 2008.
- [12] Tsao, Y., Li, J., and Lee, C.-H., "Ensemble speaker and speaking environment modeling approach with advanced online estimation process", Proc. ICASSP, pp. 3833-3836, 2009.
- [13] A. Bruce, H. Y. Gao, and W. Stuetzle, "Wavelet denoising: a comparison of subset-selection and ensemble methods", Statistica Sinica, vol. 9, pp. 167-182, 1999.
- [14] Vapnik, V. 1995. The Nature of Statistical Learning Theory, Springer-Verlag, New York.
- [15] Hirsh, H. G. and Pearce, D., "The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions", ISCA ITRW ASR 2000, Paris, 2000.
- [16] Wu, J. and Huo, Q., "Several HKU approaches for robust speech recognition and their evaluation on Aurora connected digit recognition tasks", Proc. Eurospeech, pp. 21-24, 2003.
- [17] Macho, D., Mauuary, L., Noe, B., Cheng, Y. M., Ealey, D., Jouver, D., Kelleher, H., Pearce, D., and Saadoun, F., "Evaluation of a noise-robust DSR front-end on Aurora databases", Proc. ICSLP, pp. 17-20, 2002.

Table 5. ESSEM with the SME-based intraEnv training and MCM online mapping function

	Set A					Set B					Set C			Overall
	Subway	Babble	Car	Exhibition	Average	Restaurant	Street	Airport	Station	Average	Subway M	Street M	Average	
20dB	99.66	99.49	99.58	99.48	99.55	99.72	99.40	99.49	99.75	99.59	99.69	99.40	99.55	99.57
15dB	99.51	99.18	99.43	99.14	99.32	99.39	99.09	99.55	99.51	99.39	99.54	99.12	99.33	99.35
10dB	98.74	98.49	98.72	97.84	98.45	98.37	97.76	98.39	98.52	98.26	98.31	97.58	97.95	98.27
5dB	96.47	94.98	96.81	94.35	95.65	95.00	94.62	95.62	95.87	95.28	96.01	94.29	95.15	95.40
0dB	87.35	79.87	88.61	84.63	85.12	81.21	82.98	86.13	85.37	83.92	85.14	81.95	83.55	84.32
Average	96.35	94.40	96.63	95.09	95.62	94.74	94.77	95.84	95.80	95.29	95.74	94.47	95.10	95.38