# Speech Separation By Cost-Sensitive Deep Learning

Xiao-Lei Zhang*
* Center for Intelligent Acoustics and Immersive Communications, School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an, China
E-mail: xiaolei.zhang@nwpu.edu.cn, xiaolei.zhang9@gmail.com

***Abstract*—Deep learning based speech separation has demonstrated good performance in adverse environments. Recent study shows that multi-condition training, which trains a model with several noise scenarios, shows good generalization in test. However, treating all noise scenarios with the same training cost is usually not a good choice: A common problem is that, when training data contain a wide range of SNR, the data in low SNR environments suffer from large training loss, which results in a performance drop when test SNRs are low. In this paper, we propose three cost-sensitive deep learning methods to improve the performance of speech separation methods at low SNRs, which are the methods of (i) learning with a cost-sensitive objective, (ii) learning with cost-sensitive oversampling of training data, and (iii) learning with cost-sensitive undersampling of training data. We also propose to aggregate the three methods to a cost-sensitive deep ensemble learning method. Experimental results demonstrate the effectiveness of the proposed methods.**

## I. Introduction

Speech separation aims to separate target speech signals from background noises or interfering speech signals, where the problem of separating speech from background noises is also known as speech enhancement. This paper focuses on separating speech signals of two speakers in a single channel. Theoretically, this problem is an ill-posed one, which is usually solved by putting additional constraints, objective assumptions, or side information on the problem. One kind of promising techniques is supervised speech separation.

Among the supervised speech separation methods, Wang and Wang first proposed deep learning based speech separation/enhancement [1], which has shown significantly better performance than conventional methods when separating speech from background noises. It learns a mapping function, which is a deep neural network (DNN) that contains multiple layers of nonlinear transforms, from a mixed signal to a predefined training target. Later on, many deep learning based methods have been developed [2]–[9], see [10] for an overview. Based on the training targets, the methods can be generally categorized to two classes—masking-based or mapping-based. Masking-based methods learn a mapping function from a mixed signal to an ideal time-frequency (T-F) mask in the training stage, and then uses the estimated mask produced in the test stage to mask the test signal [1], [2], [7], [8]. Mapping-based methods learn a mapping function from a mixed signal to a target speech signal [4], [6].

Multi-condition training, which trains a DNN model with multiple noise environments, is important towards the real-world applications of DNN-based speech separation. Specifically, the first result in [4] shows that multi-condition training yields better performance than conventional methods in unseen noise scenarios. Later on, results in [11], [12] show that, when training with a vast amount of noise types and a wide range of SNR levels, DNN-based methods in unseen test scenarios are as good as the DNNs whose training and test environments are matching. However, a weakness of multi-condition training is that it evaluates the training loss of all noise scenarios by the same loss function, which may not fit specific applications. This weakness is common when training data contains a wide range of SNR levels [9]. Due to an equal evaluation of training costs at all SNR levels, the training data at low SNR levels, which have low magnitude energy, cannot influence the overall training error much. As a result, multi-condition training is biased against the noise scenarios with low SNR levels. This is unfavorable, since that a main advantage of deep learning based separation is its good performance in extremely low SNR environments.

In this paper, motivated by [13], we aim to improve the performance of multi-condition training in low SNR scenarios by cost-sensitive learning. The key idea is to reweight the training loss of the low SNR scenarios. Here we propose three cost-sensitive deep learning methods, which are the methods of learning with a cost-sensitive objective, with the cost-sensitive oversampling of training data, and with the cost-sensitive undersampling of training data. We also propose to aggregate the above cost-sensitive learning methods. Our experimental results showed the effectiveness of the proposed methods.

## II. Problem formulation

Suppose there is a training speech corpus $\{\mathbf{y}_m, \mathbf{d}_m\}_{m=1}^M$ where $\mathbf{y}_m$ is the acoustic feature of the $m$-th noisy speech frame and $\mathbf{d}_m$ is the acoustic feature of the corresponding clean speech. The general training objective of a supervised speech separation method, e.g. DNN, can be formulated as:

$$\min_{\boldsymbol{\alpha}} \sum_{m=1}^{M} \ell(\mathbf{x}_m, \hat{\mathbf{x}}_m) \tag{1}$$

where $\boldsymbol{\alpha}$ is the parameter of a speech separation algorithm $f_{\boldsymbol{\alpha}}(\cdot)$, $\ell(\cdot)$ measures training loss e.g. squared loss $\|\cdot\|_2^2$, $\mathbf{x}_m$ represents the desired output at frame $m$, and $\hat{\mathbf{x}}_m$, which is produced from $f_{\boldsymbol{\alpha}}(\cdot)$, is the estimation of $\mathbf{x}_m$. Two common cases of Equation (1) are the direct mapping [4] and ideal raio masking (IRM) [2], which specifies $\mathbf{x}_m$ as clean speech $\mathbf{d}_m$ or a mask $\mathbf{d}_m/(\mathbf{d}_m + \mathbf{n}_m)$ respectively, where $\mathbf{n}_m$ is the interference.

To make supervised speech separation generalize well in unseen scenarios, a common method is to train the model with multiple scenarios jointly. Specifically, suppose we have a set of training data with multiple noise scenarios, denoted as $\{\{(\mathbf{y}_{s,m}, \mathbf{d}_{s,m})\}_{m=1}^{M_s}\}_{s=1}^{S}$, problem (1) can be rewritten as:

$$\sum_{s=1}^{S}\left(\min_{\boldsymbol{\alpha}}\sum_{m=1}^{M_s}\ell(\mathbf{x}_{s,m},\hat{\mathbf{x}}_{s,m})\right). \tag{2}$$

This optimization objective treats all scenarios equally important in respect of training errors, so that the scenarios where the target speech has relatively large magnitude values or a large number of training data tend to benefit much from the training algorithm, e.g. backpropagation of DNN.

However, in practice, some scenarios where the target speech has small magnitude values or a small number of training data should be emphasized, e.g. speech in very low SNR levels or a sudden pulse traffic noise. Eventually, the training errors of different scenarios may not be given an equal weight.

## III. Cost-sensitive learning

### A. A cost-sensitive objective

To solve the aforementioned problem, one way is to change problem (2) to a cost-sensitive objective:

$$\sum_{s=1}^{S}\left(\min_{\boldsymbol{\alpha}}\sum_{m=1}^{M_s}\ell_s(\mathbf{x}_{s,m},\hat{\mathbf{x}}_{s,m})\right) \tag{3}$$

where $\ell_s(\cdot)$ measures the training loss of the $s$-th training scenario, and $M_s$ represents the size of the training data of the $s$-th scenario.

An apparent idea of specifying problem (3) is to reweight the cost function of each training scenario by $\ell_s(\cdot) = w_s\ell(\cdot)$ which derives the following objective:

$$\sum_{s=1}^{S}\left(w_s\min_{\boldsymbol{\alpha}}\sum_{m=1}^{M_s}\ell(\mathbf{x}_{s,m},\hat{\mathbf{x}}_{s,m})\right) \tag{4}$$

where $\{w_s\}_{s=1}^{S}$ are handcrafted weights.

In this paper, we aim at enhancing the performance of deep learning based separation methods at low SNR levels. For this purpose, a simple implementation of $\{w_s\}_{s=1}^{S}$ is as follows:

$$w_s = \frac{10^{-(\sigma t_s)/20}}{\sum_{k=1}^{S}10^{-(\sigma t_k)/20}} \tag{5}$$

where $t_s$ is the SNR level of the $s$-th noise scenario, $10^{-t_s/20}$ is the average magnitude ratio of speech and noise, and $\sigma$ is a free parameter. The idea behind Equation (5) is simply as follows. The training loss evaluated by $\ell(\cdot)$ is proportional to the SNR (in magnitude) $E(d_s)/E(n_s) = 10^{t_s/20}$, hence, a direct way to compensate the training loss at a low SNR level is to amplify the training loss with a weight $w'_s = E(n_s)/E(d_s) = 10^{-t_s/20}$, where $E(\cdot)$ represents the expectation operator. Because Equation (5) only considers the influence of SNR on performance, it may be also helpful to tune $w'_s$ for compensating other factors, e.g. the size of training data, which results in $w'_s = 10^{-(\sigma t_s)/20}$. To make problem (3) more controllable in practice, we further normalize $w'_s$ by a denominator $\sum_{k=1}^{S}10^{-(\sigma t_k)/20}$ which derives Equation (5).

For optimizing the above cost-sensitive objective by DNN, we simply multiply the gradient of the original DNN objective with the scaling factor $w_s$.

### B. Oversampling

Oversampling changes the training data distribution by randomly resampling the *speech frames*[1] of each noise scenario *with replacement*[2].

For the $s$-th noise scenario, the number of training speech frames $M_s^*$ after oversampling is defined by

$$M_s^* = \begin{cases} \left\lfloor \frac{w_s}{w_\lambda}M_\lambda \right\rfloor, & \text{if } \left\lfloor \frac{w_s}{w_\lambda}M_\lambda \right\rfloor > M_s \\ M_s, & \text{otherwise} \end{cases} \tag{6}$$

where $\lambda$ indicates a noise scenario that is identified by the following equation:

$$\lambda = \arg\min_{j=1,\ldots,S}\frac{w_j}{M_j}. \tag{7}$$

If $M_s^* > M_s$, then the $s$-th noise scenario should be added with $(M_s^* - M_s)$ resampled speech frames. For example, if $\mathcal{X}_s = \{\mathbf{x}_{s,1}, \mathbf{x}_{s,2}, \mathbf{x}_{s,3}\}$ with $M_s = 3$ and $M_s^* = 6$, then an over-sample of $\mathcal{X}_s$, denoted by $\mathcal{X}_s^*$, may be $\mathcal{X}_s^* = \mathcal{X}_s \cup \{\mathbf{x}_{s,4}^* = \mathbf{x}_{s,2}, \mathbf{x}_{s,5}^* = \mathbf{x}_{s,2}, \mathbf{x}_{s,6}^* = \mathbf{x}_{s,1}\}$ where $\mathbf{x}_{s,m}$ with $m = 1, 2, 3$ are speech frames.

After oversampling, the learning machine is trained with the resampled data and the original cost-insensitive objective, i.e. Equation (1).

### C. Undersampling

Undersampling changes the training data distribution by randomly eliminating speech frames from each noise scenario. For the $s$-th noise scenario, the number of training speech frames $M_s^*$ after undersampling is defined by

$$M_s^* = \begin{cases} \left\lfloor \frac{w_s}{w_\lambda}M_\lambda \right\rfloor, & \text{if } \left\lfloor \frac{w_s}{w_\lambda}M_\lambda \right\rfloor < M_s \\ M_s, & \text{otherwise} \end{cases} \tag{8}$$

where $\lambda$ is defined by

$$\lambda = \arg\max_{j=1,\ldots,S}\frac{w_j}{M_j}. \tag{9}$$

If $M_s^* < M_s$, then we simply eliminate $(M_s - M_s^*)$ speech frames randomly from the training data of the $s$-th noise scenario. Note that eliminating speech frames randomly is not an optimal way. Here we just present the idea and possibility of the undersampling, leaving the methods on how to eliminate speech frames effectively as future work.

[1]The term 'speech frames' are the basic training unit of a learning machine. Note that, for speech separation (including enhancement), we usually expand a speech frame with contextual frames in the time axis, in this case, the term means the "expanded speech frames".

[2]The term 'with replacement' means that a random selection of a speech frame from an original corpus does not change the corpus, so that a speech frame in the original corpus may be duplicated in the new corpus.

## IV. Cost-sensitive deep ensemble learning

Ensemble learning aggregates a set of base learners to reach performance that is better than any of the base learners, as if (i) the base learners are stronger than random guess and (ii) they are diverse from each other in terms of errors. There are four common ways to enlarge the diversity between the base learners, which are the methods of manipulating input features, output targets, training data, and hyperparameters of base learners. The three cost-sensitive learning methods in Section III manipulate output targets and training data, which can yield many aggregation methods. Here we simply average the output of all base learners. The method is named *cost-sensitive deep ensemble learning*.

## V. Experiments

In this section, we evaluate the proposed methods over the 4 possible gender pairs, where the first speaker of a gender pair is the target speaker and the other one interfering speaker.

### A. Experimental settings

The experimental setting follows that in [9]. Specifically, we used the predefined training corpus of the speech separation challenge (SSC) [14] dataset. The training corpus contains 34 speakers, each of which has 500 clean utterances. We resampled all corpora to 8 kHz, and extracted the STFT features with the frame length set to 25 ms and the frame shift set to 10 ms. We randomly picked 2 pairs of speakers for each gender pair, which generated 8 tasks in total. We used the first 450 clean utterances of a speaker for constructing the training data, and the last 50 clean utterances of the speaker for testing. Each task had 7 SNR levels ranging from $\{-12, -9, -6, -3, 0, 3, 6\}$ dB. For each training SNR level of a separation task, we constructed 1000 training mixtures. The two components of a training mixture were randomly selected from the two speakers of the separation task, respectively. For each test SNR level of a task, we constructed 50 test mixtures by mixing the last 50 clean utterances of the two speakers respectively.

For each separation task, 7 IRM-based speech separation methods were tested:

- **Single-condition training (SCT_1)**: A model is trained with the speech data at an SNR level of $-12$ dB.
- **Single-condition training (SCT_2)**: A model is trained with the speech data at an SNR level that is the same as the test SNR level.
- **Multi-condition training (MCT)**: A model is trained with the data from all training SNR levels.
- **Cost-sensitive learning with the cost-sensitive objective (CSL_w)**: A model is trained by the method in Section III-A with the data at all SNR levels. Parameter $\sigma$ was set to $[0.5, 1, 2]$ respectively.
- **Cost-sensitive learning with oversampling (CSL_o)**: A model is trained by the method in Section III-B with the data at all SNR levels. Parameter $\sigma$ was set to 1.

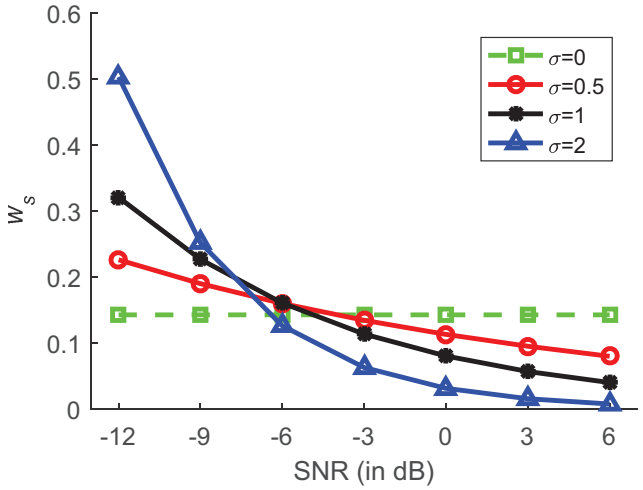


Fig. 1: Cost-sensitive weights.

- **Cost-sensitive learning with undersampling (CSL_u)**: A model is trained by the method in Section III-C with the data at all SNR levels. Parameter $\sigma$ was set to 1.
- **Cost-sensitive ensemble learning (CSL_e)**: CSL_e aggregates 6 models which are 3 CSL_w models with $\sigma$ set to $\{0.5, 1, 2\}$ respectively, 1 CSL_o model with $\sigma = 1$, 1 CSL_u model with $\sigma = 1$, and the MCT model.

Each method was evaluated in all 7 test environments. All comparison methods aimed at enhancing the noisy magnitude spectrograms, leaving the noisy phase spectrograms unprocessed. The enhanced magnitude spectrograms and the noisy phase spectrograms were used together to recover the speech signals in time domain by inverse STFT. All comparison methods used DNN as the learning machine. The parameter setting of DNN followed that in [9]. The short-time objective intelligibility (STOI) is adopted as the evaluation metric.

### B. Results

We first draw the values of $w_s$ with $\sigma = \{0, 0.5, 1, 2\}$ respectively in Fig. 1, where $\sigma = 0$ corresponds to MCT. From the figure, we can see that the CSL models with different $\sigma$ emphasize (or suppress) different SNR levels. Specifically, in the case of $\sigma = \{0.5, 1\}$, the environments with SNR $t_s \leq -6$ dB will be emphasized; in the case of $\sigma = 2$, the environments with $t_s \leq -9$ dB will be emphasized.

We report the average performance of the 8 separation tasks in Table I. From the table, we observe the following experimental phenomena. (i) The relative performance variation between CSL and MCT has an exact one-to-one correspondence with the pattern of $w_s$ in Fig. 1. For example, when $t_s = -9$ dB, the CSL_w with $\sigma = 1$ is more effective than MCT and the CSL_w with $\sigma = 0.5$, but is less effective than the CSL_w with $\sigma = 2$. Hence, in our future work, we just need to focus on designing $w_s$. (ii) CSL_o and CSL_u are less effective than CSL_w. CSL_o is even weaker than MCT, which might be caused by overfitting. This overfitting problem has also been observed in the machine learning community. (iii) CSL_e is more effective than any of its base learners.

TABLE I: STOI (in percent) comparison between speech separation methods.

| | −12 dB | −9 dB | −6 dB | −3 dB | 0 dB | 3 dB | 6 dB |
|---|---|---|---|---|---|---|---|
| Noisy | 47.09 | 52.88 | 59.40 | 66.34 | 73.28 | 79.75 | 85.35 |
| SCT_1 | 75.00 | 78.73 | 80.45 | 80.93 | 82.12 | 84.62 | 87.77 |
| SCT_2 | 75.00 | 79.27 | 83.51 | 86.78 | 89.80 | 92.44 | **94.57** |
| MCT | 75.03 | 80.52 | 84.86 | 88.27 | 90.93 | 92.99 | 94.56 |
| CSL_w ($\sigma = 0.5$) | 75.24 | 80.67 | 84.93 | 88.24 | 90.83 | 92.84 | 94.39 |
| CSL_w ($\sigma = 1$) | 75.48 | 80.79 | 84.90 | 88.07 | 90.56 | 92.51 | 94.05 |
| CSL_w ($\sigma = 2$) | 75.81 | 80.86 | 84.70 | 87.62 | 89.88 | 91.70 | 93.23 |
| CSL_o ($\sigma = 1$) | 74.02 | 79.79 | 84.32 | 87.77 | 90.42 | 92.48 | 94.07 |
| CSL_u ($\sigma = 1$) | 75.47 | 80.55 | 84.51 | 87.62 | 90.07 | 92.04 | 93.64 |
| CSL_e | **77.58** | **82.34** | **86.05** | **88.95** | **91.23** | **93.05** | 94.49 |

TABLE II: A study of cost-sensitive deep ensemble learning with respect to its base learners in STOI (in percent).

| | −12 dB | −9 dB | −6 dB | −3 dB | 0 dB | 3 dB | 6 dB |
|---|---|---|---|---|---|---|---|
| CSL_w ($\sigma = 1$) | 75.48 | 80.79 | 84.90 | 88.07 | 90.56 | 92.51 | 94.05 |
| CSL_e1 | 76.27 | 81.45 | 85.47 | 88.56 | 90.97 | 92.86 | 94.35 |
| CSL_e2 | 77.00 | 81.91 | 85.73 | 88.68 | 91.00 | 92.85 | 94.32 |
| CSL_e3 | 77.26 | 82.12 | 85.90 | 88.85 | 91.17 | 93.02 | 94.47 |
| CSL_e4 | 77.52 | 82.28 | 85.97 | 88.84 | 91.10 | 92.91 | 94.35 |
| CSL_e5 | 77.58 | 82.34 | 86.05 | 88.95 | 91.23 | 93.05 | 94.49 |

To study how the performance of CSL_e varies with the number of base learners, we constructed 5 CSL_e algorithms, denoted as CSL_e$k$ with $k = 1, \ldots, 5$. CSL_e$(k+1)$ was constructed by adding a new base learner to CSL_e$k$. The sequence of adding new learners to CSL_e started with the CSL_w with $\sigma = 1$, followed by CSL_o, CSL_u, the CSL_w with $\sigma = 0.5$, the CSL_w with $\sigma = 2$, and MCT respectively. Results are listed in Table II. From the table, we observe that aggregating new base learners to CSL_e improves performance consistently, even when the ineffective CSL_o is added.

## VI. Conclusions

In this paper, we have proposed cost-sensitive learning to offset the biased estimation of the multi-condition training based speech separation against low SNR environments. We have discussed three methods of cost-sensitive learning. The first method defines a cost-sensitive objective that assigns high training costs to the low SNR environments. The second method oversamples the data of the noise scenarios that should have high training costs. The third method undersamples the data of the noise scenarios that should have low training costs. We have also proposed to aggregate the above cost-sensitive learning methods to a cost-sensitive deep ensemble learning method. Experimental results demonstrate the effectiveness of the proposed methods.

## Acknowledgment

This work was supported in part by National Natural Science Foundation of China under Grant 61671381.

## References

[1] Y. Wang and D. L. Wang, "Towards scaling up classification-based speech separation," *IEEE TASLP*, vol. 21, no. 7, pp. 1381–1390, 2013.

[2] Y. Wang, A. Narayanan, and D. L. Wang, "On training targets for supervised speech separation," *IEEE/ACM TASLP*, vol. 22, no. 12, pp. 1849–1858, 2014.

[3] J. Chen, Y. Wang, and D. L. Wang, "A feature study for classification-based speech separation at very low signal-to-noise ratio," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 22, no. 12, pp. 1993–2002, 2014.

[4] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "An experimental study on speech enhancement based on deep neural networks," *IEEE SPL*, vol. 21, no. 1, pp. 65–68, 2014.

[5] F. Weninger, J. R. Hershey, J. Le Roux, and B. Schuller, "Discriminatively trained recurrent neural networks for single-channel speech separation," in *GlobalSIP*, 2014, pp. 577–581.

[6] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM TASLP*, vol. 23, no. 1, pp. 7–19, 2015.

[7] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Joint optimization of masks and deep recurrent neural networks for monaural source separation," *IEEE/ACM TASLP*, vol. 23, no. 12, pp. 2136–2147, 2015.

[8] D. S. Williamson, Y. Wang, and D. L. Wang, "Complex ratio masking for monaural speech separation," *IEEE/ACM TASLP*, vol. 24, no. 3, pp. 483–492, 2016.

[9] X.-L. Zhang and D. L. Wang, "A deep ensemble learning method for monaural speech separation," *IEEE/ACM TASLP*, vol. 24, no. 5, pp. 967–977, 2016.

[10] D. L. Wang and J. Chen, "Supervised speech separation based on deep learning: An overview," *arXiv preprint arXiv:1708.07524*, 2017.

[11] Y. Wang, J. Chen, and D. L. Wang, "Deep neural network based supervised speech segregation generalizes to novel noises through large-scale training," *Tech. Rep. OSU-CISRC-3/15-TR02, Dept. of Comput. Sci. and Eng., The Ohio State Univ., Columbus, OH, USA*, 2015.

[12] X.-L. Zhang and D. L. Wang, "Boosting contextual information for deep neural network based voice activity detection," *IEEE/ACM TASLP*, vol. 24, no. 2, pp. 252–264, 2016.

[13] Z.-H. Zhou and X.-Y. Liu, "Training cost-sensitive neural networks with methods addressing the class imbalance problem," *IEEE TKDE*, vol. 18, no. 1, pp. 63–77, 2006.

[14] M. Cooke and T.-W. Lee, "Speech separation challenge," http://staffwww.dcs.shef.ac.uk/people/M.Cooke/SpeechSeparationChallenge.htm, 2006.