ENSEMBLE LEARNING FOR SPEECH ENHANCEMENT

Jonathan Le Roux, Shinji Watanabe, John R. Hershey

Mitsubishi Electric Research Laboratories (MERL), 201 Broadway, Cambridge, MA 02139, USA,
{leroux,watanabe,hershey}@merl.com

ABSTRACT

Over the years, countless algorithms have been proposed to solve the problem of speech enhancement from a noisy mixture. Many have succeeded in improving at least parts of the signal, while often deteriorating others. Based on the assumption that different algorithms are likely to enjoy different qualities and suffer from different flaws, we investigate the possibility of combining the strengths of multiple speech enhancement algorithms, formulating the problem in an ensemble learning framework. As a first example of such a system, we consider the prediction of a time-frequency mask obtained from the clean speech, based on the outputs of various algorithms applied on the noisy mixture. We consider several approaches involving various notions of context and various machine learning algorithms for classification, in the case of binary masks, and regression, in the case of continuous masks. We show that combining several algorithms in this way can lead to an improvement in enhancement performance, while simple averaging or voting techniques fail to do so.

***Index Terms*—** Ensemble learning, Speech enhancement, Time-frequency mask, Classification, Stacking

1. INTRODUCTION

Speech enhancement methods attempt to improve the quality and intelligibility of speech that has been degraded by interfering noise or other processes. One thing that makes this problem difficult is that the interference can come in many different varieties. To further complicate matters, often the operational constraints on computation and latency preclude the use of complex models that can represent and adapt to many different noise types. As it is difficult for a simple algorithm to accommodate the variety of conditions, some assumptions about the statistical properties of the target and interference signals have to be made. Over the years, many different algorithms have been proposed, each having different explicit or implicit assumptions about the nature of the speech and interference [1]. Assuming that the strengths and weaknesses of a set of algorithms differ, it would be desirable to combine them in a way that takes advantage of all their strengths.

Ensemble machine learning methods aim at combining different models, and exploit the independence of the errors made by each classifier to reduce the estimation variance, and hence the error rate. These methods range from simple *voting* procedures, where the quantities inferred by each model are averaged together, to *stacking*, in which a secondary model is trained to perform the combination in a way that is tuned to training data. An advantage of voting methods is that they can be applied without consideration of the test conditions. However, stacking methods can learn more complex combination functions, potentially leading to better performance.

Ensemble methods have been used extensively in automatic speech recognition (ASR) to fuse speech recognition hypotheses of different recognizers via voting procedures such as recognizer output voting error reduction (ROVER) [2]. Particularly relevant to our work here are ensemble ASR methods in which the recognizers differ according to the enhancement or robustness algorithms used in their front end [3]. A chief advantage of ensemble methods is that they can build upon a variety of existing algorithms to improve performance.

To make use of ensemble learning in the speech enhancement paradigm, we consider a more direct integration of the enhancement algorithms. We compute the time-frequency masking functions that, when applied to the noisy spectrogram, yield the spectrum of the enhanced signals. The result of their combination is to produce an ensemble time-frequency masking function. Here, for simplicity, we primarily focus on the estimation of binary masking functions, and only touch upon the estimation of continuous masking functions. We investigate both simple voting as well as stacking, in which a variety of classification algorithms, such as support vector machines (SVM) [4], naive Bayes classifiers (NB) [5], decision trees (DT) [6], and random forests (RF) [7], are used to infer the binary masking function. Estimation of binary masks for enhancement and separation has been considered in a machine learning context before [8, 9, 10, 11], but not in an ensemble learning framework.

In experiments with difficult interference conditions, we show that a combination of several enhancement algorithms using stacking can lead to an improvement in enhancement performance, whereas simple averaging or voting techniques fail to do so.

2. GENERAL FRAMEWORK

We assume an ensemble of speech enhancement algorithms that are to be treated as "black boxes" in the sense that we only use the enhanced signals for combination. It would also be reasonable to combine enhancement algorithms at the "decision" level using some internal representations. However, we would like to allow the use of arbitrary models and avoid the use of heterogeneous features.

We thus perform the combination in a domain that is independent of the particular formulation of each enhancement algorithm. A good choice for such a domain is the short-time power spectrum, which is widely used in signal processing because of its relative insensitivity to phase and its ability to reveal time and frequency patterns in the signal. Regardless of the internal representation they use, speech enhancement algorithms take as input a noisy signal $y[t]$ in the time domain and transform it to an enhanced estimate $\hat{x}[t]$ of the clean signal. In the short-time power spectrum domain, this enhancement process can be approximated as applying a time-frequency masking function to the spectrogram of the noisy input signal. If the optimal masking function were known, the speech sig-

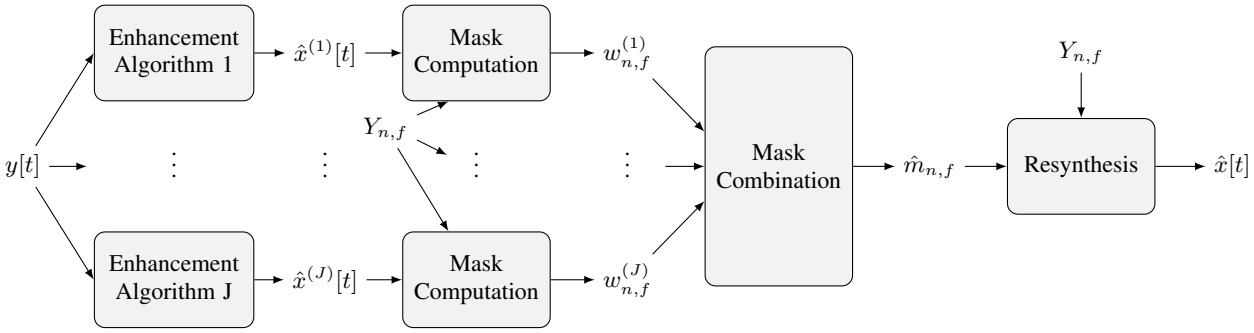

Figure 1: Overview of the framework.

nal could be reconstructed almost perfectly by applying the masking function to the noisy power spectrum and inverting the representation. Our method is thus to combine time-frequency masking functions obtained from the enhancement algorithms, in order to estimate an optimal masking function to better reconstruct the speech.

For a given enhancement algorithm j in our ensemble, we define an *equivalent continuous masking function*, $w_{n,f}^{(j)}$, for time frame n and frequency f. We also formulate a target masking function $w_{n,f}^{*}$ as that which transforms the noisy spectrum into the clean spectrum. For simplicity, the masking functions can be approximated as *binary masking functions*, $m_{n,f}^{(j)}$, and $m_{n,f}^{*}$.

The binary target mask $m_{n,f}^{*}$ is convenient in that the ensemble inference problem can be posed as binary classification, where a classifier computes a binary mask estimate $\hat{m}_{n,f}$ using as input the masking functions $\{w_{n,f}^{(1)}, ..., w_{n,f}^{(J)}\}$, or their binary counterparts, derived from each of the enhancement algorithms.

Simple voting or averaging procedures on the input signals or their masking functions could be used, but here we also investigate stacking approaches in which the method of combination is learned from training data. In the context of stacking, we can also consider the temporal and frequency context in the neighborhood of each masking function value to be estimated.

Once the combined mask is inferred, it can be applied to the noisy signal spectrum, and combined with noisy phases to produce the estimated speech signal, $\hat{x}[t]$, via an inverse transform such as the overlap-add procedure for the short-time Fourier transforms (STFT). The overall system architecture is shown in Fig. 1.

3. TARGETS

Time-frequency masking functions estimated from the noisy mixture have often been used as a means to perform source separation or speech enhancement [12]. Time-frequency masks apply a weight to each bin of a time-frequency representation of the acoustic input, such as cochleograms, short-time Fourier transforms, wavelet transforms, and so on, to emphasize regions which are dominated by the target source and suppress regions which are dominated by other sources. The weight values can be either binary or continuous. Continuous values can be interpreted as the energy ratio between the target and the mixture, as in a Wiener filter, or as the probability that the corresponding bin belongs to the target source.

Restricting the mask to take only binary values has been shown to be a reasonable proxy for the optimal masking function in general conditions [13]. Binary masks have the disadvantage that they cannot account for cancellation effects and may introduce strong artifacts depending on the interfering noise. However, advantages in our setting include the ease of estimation of the two possible values instead of a continuum, as well as their potential for computational savings. We thus here mainly focus on the binarized continuous mask obtained from the clean speech as the target for our method, and only touch upon the use of continuous masks in a regression framework.

4. INPUTS

As mentioned above, each enhancement algorithm may be processing the noisy input signal in various domains, whether directly in the time domain or more likely in some time-frequency representation such as the STFT or a Gammatone-based transform, with various filterbank settings. Instead of directly attempting to combine these inner representations, we choose here to use the final outputs, the enhanced time-domain signals $\hat{x}^{(j)}, j = 1, \ldots, J$. This enables us to consider any speech enhancement algorithm as a potential input to our system, regardless of its implementational details.

From these enhanced signals, we could consider deriving any type of features for combination. For convenience and simplicity, we consider here re-analysing all enhanced signals using the same common time-frequency representation used to derive the target. This enables us to have a direct correspondence between the time-frequency bins of the input features and those of the target.

To avoid feature-scaling issues, we do not directly use features such as the power spectrogram or log-power spectrogram, but define an equivalent continuous mask $w^{(j)}$ for each algorithm as the ratio of the power spectrogram of the enhanced signal $\hat{X}^{(j)}$ to that of the noisy mixture Y:

$$w_{n,f}^{(j)} = \hat{X}_{n,f}^{(j)} / Y_{n,f}, \tag{1}$$

and similarly for $w_{n,f}^{*}$. This approximates each algorithm as a reweighting method in a common time-frequency representation.

Finally, we also derive binary mask features $m^{(j)}$ from the continuous masks: $m_{n,f}^{(j)} = [w_{n,f}^{(j)} > 0.5]$, and $m_{n,f}^{*} = [w_{n,f}^{*} > 0.5]$, where $[a > b] = 1$ if $a > b$ and 0 otherwise. The motivation for considering binary masks as inputs is two-fold: they may lead to more robust estimators; and their use can reduce the computational cost with regard to the continuous masks, for example with support vector machines and decision trees.

5. INFERENCE ALGORITHMS

5.1. Voting

Voting or averaging is an ensemble combination strategy that simply combines outputs of the models by taking an average of their

values. In the case of classification, the output is usually the mode of the distribution over classes, whereas in regression, the output would be the mean or some other average of the output values. Uncertainty within each model can also be considered, but here since we derive the mask values from an ensemble of arbitrary enhancement methods, we do not consider the uncertainty within each enhancement algorithm.

In voting, continuous or binary mask values for all algorithms at time-frequency bin (n, f) are used to estimate the target mask (either $w^*_{n,f}$ or $m^*_{n,f}$) at the same bin. The input feature vectors are thus typically $z_{n,f} = (w^{(1)}_{n,f}, ..., w^{(J)}_{n,f})^T$ for the continuous masks and $z_{n,f} = (m^{(1)}_{n,f}, ..., m^{(J)}_{n,f})^T$ for the binary masks.

For the continuous masking function inputs, we consider the mean of the masking values as a continuous mask estimate, which corresponds to averaging the original power spectrum estimates. We also consider the median in a similar way.

For the binary masking function inputs, voting considers the mode of the masking value distribution:

$$\hat{m}^{\text{voting}}_{n,f} = [\frac{1}{J}\sum m^{(j)}_{n,f} > 0.5]. \quad (2)$$

Since there are no learned parameters, voting methods cannot over-fit the training data. To the extent that the masking values make uncorrelated errors, then voting and averaging procedures tend to recover from these errors. In other words, the variance across classifiers can be reduced by the voting procedure. However, whenever the errors are correlated, the averaging just reinforces the errors, so the classifier can remain biased.

5.2. Stacking

Stacking is an ensemble learning strategy in which multiple estimation algorithms for the same task are used as input into a final algorithm that is trained on data to combine their results. This procedure can reduce the bias even when the outputs of the ensemble are correlated; however, the learning may also over-fit the training data. The case of binary mask targets allows us to use simple binary classifiers to produce mask estimates. One can also use different forms of regression to produce continuous mask estimates, but here we mainly focus on a classification-based approach. We investigated a variety of classifiers, such as SVM, NB, DT, and RF.

We here consider separate classifiers $\mathcal{C}^f_{\Theta^f}$ for each frequency f, with parameters Θ^f. At each frame n, given an input vector $i_{n,f}$, the classifier produces a mask estimate $\hat{m}_{n,f} = \mathcal{C}^f_{\Theta^f}(i_{n,f})$. We first learn the parameters Θ^f so that they minimize a loss function $\mathcal{L}$ with respect to the target mask $m^*_{n,f}$ on training data $\mathcal{T}$:

$$\bar{\Theta}^f = \underset{\Theta^f}{\text{argmin}}\, \mathcal{L}[(\mathcal{C}^f_{\Theta^f}(i_{n,f}), m^*_{n,f}), n \in \mathcal{T}], \quad \forall f. \quad (3)$$

At test time, we estimate the mask using the learned parameters $\bar{\Theta}^f$:

$$\hat{m}_{n,f} = \mathcal{C}^f_{\bar{\Theta}^f}(i_{n,f}), \quad \forall n, f. \quad (4)$$

The loss function $\mathcal{L}$ is determined by the classifier type.

In the framework of stacking, we can consider including time and/or frequency context information into the input feature vectors. Here, we extend the features in the time direction by $c^{(n)}$ frames to the left and to the right, and in the frequency direction by $c^{(f)}$ frequency bins below and above. The input feature vector to estimate $m^*_{n,f}$ thus consists of the concatenation of time-frequency patches with $(2c^{(n)} + 1) \times (2c^{(f)} + 1)$ elements in the neighborhood of the bin (n, f) for each algorithm. The boundary cases in both directions are handled appropriately.

6. EVALUATION

6.1. Setup

We used audio data from the medium vocabulary task (Track 2) of the 2nd CHiME Speech Separation and Recognition Challenge [14]. The speech is taken from the Wall Street Journal (WSJ0) 5k vocabulary read speech corpus, and convolved with binaural room impulse responses before mixing with binaural recordings of a noisy domestic environment. The RT60 of the room is 300 ms. The noise excerpts are selected as to obtain input signal-to-noise ratio (SNR) ranges of -6, -3, 0, 3, 6, and 9 dB without rescaling. Noises are highly non-stationary, such as speech by other speakers, home noises, or music, making the denoising task very challenging.

As we need parallel data to train our system as well as to evaluate its enhancement performance, we randomly sample utterances across all input SNRs from the development set data (`si_dt_05`), for which scaled reverberated clean speech data are provided on top of the noisy mixture data, to build a training set of 100 utterances with random input SNR, and an evaluation set of 600 utterances, 100 for each input SNR.

The sampling rate was 16 kHz. The common time-frequency representation for the target and all enhancement output signals was obtained using the short-time Fourier transform with a frame length of 640 samples, 50% overlap and a sine window for analysis and re-synthesis.

Performance was evaluated in terms of averaged signal-to-distortion ratio (SDR), using the `bss_eval` toolbox [15]. The SDR averaged over the noisy mixtures was 1.85 dB. Resynthesizing the clean speech from its equivalent continuous mask w^* led to an average SDR of 17.54 dB, and the binarized continuous mask m^* led to 17.01 dB, which represents the ideal performance that could be expected from this implementation of our method.

We considered the following speech enhancement algorithms, which constitute a varied set of techniques, including state-of-the-art methods: vector-Taylor series (VTS) [16], indirect VTS [17], OMLSA-IMCRA [18], as well as implementations of the classical MMSE and log-MMSE algorithms taken from [1]. Results for these input algorithms are shown in Table 1. CMask and BMask denote the result of applying respectively the equivalent continuous mask and its binarized version to the noisy complex spectrum, and resynthesizing to the time domain. The continuous mask was truncated to values between 0 and 1. Note that, differing from [17], VTS and OMLSA performed better than indirect VTS on this data.

6.2. Results

We first investigate the performance of averaging on the input continuous masks, both using mean and median, and of voting on the binarized masks. As shown in Table 2, none of these methods led to improvements compared to the input algorithms, the performance actually decreasing with respect to the best ones. While voting in particular is known to help when combining complex systems such as in ASR, the poor performance here could be due to the fact that the combination is impacted by the poorly performing algorithms in a direct way, while processing by complex systems may still lead to interesting hypotheses prior to combination.

Table 1: SDR (dB) for each input speech enhancement algorithm

SDR	OMLSA	logMMSE	MMSE	indirectVTS	VTS
Original	5.20	2.93	2.96	4.32	1.61*
CMask	5.22	2.90	2.95	4.40	5.61
BMask	5.03	2.68	2.71	4.41	5.37

*VTS did not use the truncation explained in [17].

Table 2: SDR (dB) for averaging and voting methods

	Mean	Median	Voting
SDR	3.94	4.42	4.76

We now turn to ensemble learning methods. First, Table 3 investigates the performance of each classifier (linear SVM [19], DT: decision tree, RF: random forest, NB: naive Bayes) under the simple experimental condition "B $\rightarrow$ B", i.e., where both input features and output targets are binary masks. In this preliminary experiment, we did not attempt to tune the regularization parameters of the classifiers. Every approach was comparable to or outperformed the best performing single speech enhancement algorithm, here VTS. Although the random forest achieved the best performance (SDR = 5.92 dB) without considering time-frequency contexts, adding them did not seem to improve performance for DT, RF and NB. As it also drastically increased computational cost for RF and NB, we did not consider large contexts for these classifiers. On the other hand, we found that the performance of SVM improved consistently when the feature dimensionality increased by considering contexts. This is reasonable, since SVM can make use of redundant features to set accurate and robust classification bounds, while the other classifiers face over-training problems. Based on the results of Table 3, the subsequent experiments focus on SVM classifier results using the whole frame as frequency context in the input features.

Table 4 compares the SVM results with binary masks (B $\rightarrow$ B) and continuous masks (C $\rightarrow$ B) as features to estimate binary masks. We also estimated continuous masks in the output by using support vector regression with continuous mask features (C $\rightarrow$ C). Table 4 shows that the continuous feature case outperformed the binary feature case by up to 1.32 dB, which indicates that the continuous values are informative features to combine speech enhancement algorithms. The result of the continuous mask estimation did not outperform that of the binary mask estimation in this setting, although other regression methods may lead to better results.

Finally, we scaled up the experiments by increasing the size of the training data (100 $\rightarrow$ 1260 utterances), as shown in Table 4. We finally obtained **7.97** dB, which improved from the best single system (VTS) by 2.36 dB, and from the voting method by 3.21 dB. This confirms the effectiveness of our system combination approach based on ensemble learning.

Table 3: SDR (dB) for each classifier (B $\rightarrow$ B, i.e., Input feature: binary feature. Output target: binary mask)

Context (time $c^{(n)}$, bin $c^{(f)}$)	SVM	DT	RF	NB
No context (0, 0)	5.40	5.87	5.92	5.38
(1, 0)	5.60	5.86	5.89	5.11
(2, 0)	5.62	5.72	5.81	5.00
(0, 1)	5.64	5.90	5.90	5.09
(0, 2)	5.65	5.74	5.87	5.00
(0, all)	**6.21**	4.04	N/A	N/A

Table 4: SDR (dB) for various types of inputs/outputs using SVM

SVM	B$\rightarrow$B	C$\rightarrow$B	C$\rightarrow$B (1260 utt.)	C$\rightarrow$C
(0, all)	6.21	7.53	**7.97**	6.37

7. CONCLUSION

We presented an ensemble learning approach to speech enhancement. By learning how to combine the outputs of multiple enhancement algorithms, we were able to significantly outperform the original algorithms. Future work will investigate further the use of regression to estimate continuous masking functions, as well as the influence of the proposed system on ASR performance.

8. REFERENCES

[1] P. C. Loizou, *Speech Enhancement, Theory and Practice*. Boca Raton, FL: CRC Press, 2007.

[2] J. G. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER)," in *Proc. ASRU*, 1997, pp. 347–354.

[3] J. Barker, E. Vincent, N. Ma, H. Christensen, and P. Green, "The PASCAL CHiME speech separation and recognition challenge," *Computer Speech & Language*, 2012.

[4] C. Cortes and V. Vapnik, "Support vector machine," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[5] D. D. Lewis, "Naive (Bayes) at forty: The independence assumption in information retrieval," in *Proc. ECML*, 1998, pp. 4–15.

[6] L. Olshen, J. H. Breiman, R. A. Friedman, and C. J. Stone, *Classification and Regression Trees*. Wadsworth International Group, 1984.

[7] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[8] F. R. Bach and M. I. Jordan, "Learning spectral clustering, with application to speech separation," *JMLR*, vol. 7, pp. 1963–2001, 2006.

[9] R. J. Weiss and D. P. W. Ellis, "Estimating single-channel source separation masks: Relevance vector machine classifiers vs. pitch-based masking," in *Proc. SAPA*, 2006, pp. 31–36.

[10] G. Kim, Y. Lu, Y. Hu, and P. C. Loizou, "An algorithm that improves speech intelligibility in noise for normal-hearing listeners," *J. Acoust. Soc. Am.*, vol. 126, no. 3, pp. 1486–1494, 2009.

[11] K. Han and D. Wang, "A classification based approach to speech segregation," *J. Acoust. Soc. Am.*, vol. 132, no. 5, pp. 3475–3483, 2012.

[12] D. Wang and G. J. Brown, Eds., *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications*. Wiley, 2006.

[13] D. Wang, "On ideal binary mask as the computational goal of auditory scene analysis," in *Speech separation by humans and machines*, P. Divenyi, Ed. Kluwer Academic Publishers, 2005, ch. 12, pp. 181–197.

[14] E. Vincent, J. Barker, S. Watanabe, J. Le Roux, F. Nesta, and M. Matassoni, "The second CHiME speech separation and recognition challenge: Datasets, tasks and baselines," in *Proc. ICASSP*, May 2013.

[15] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE Trans. ASLP*, vol. 14, no. 4, pp. 1462–1469, 2006.

[16] P. J. Moreno, B. Raj, and R. M. Stern, "A vector Taylor series approach for environment-independent speech recognition," in *Proc. ICASSP*, vol. 2, May 1996, pp. 733–736.

[17] J. Le Roux and J. R. Hershey, "Indirect model-based speech enhancement," in *Proc. ICASSP*, Mar. 2012, pp. 4045–4048.

[18] I. Cohen, "Noise spectrum estimation in adverse environments: Improved minima controlled recursive averaging," *IEEE Trans. SAP*, vol. 11, no. 5, pp. 466–475, 2003.

[19] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *JMLR*, vol. 9, pp. 1871–1874, 2008.