Towards Scaling Up Classification-Based Speech Separation

Yuxuan Wang and DeLiang Wang, *Fellow, IEEE*

***Abstract*—Formulating speech separation as a binary classification problem has been shown to be effective. While good separation performance is achieved in matched test conditions using kernel support vector machines (SVMs), separation in unmatched conditions involving new speakers and environments remains a big challenge. A simple yet effective method to cope with the mismatch is to include many different acoustic conditions into the training set. However, large-scale training is almost intractable for kernel machines due to computational complexity. To enable training on relatively large datasets, we propose to learn more linearly separable and discriminative features from raw acoustic features and train linear SVMs, which are much easier and faster to train than kernel SVMs. For feature learning, we employ standard pre-trained deep neural networks (DNNs). The proposed DNN-SVM system is trained on a variety of acoustic conditions within a reasonable amount of time. Experiments on various test mixtures demonstrate good generalization to unseen speakers and background noises.**

***Index Terms*—Computational auditory scene analysis (CASA), deep belief networks, feature learning, monaural speech separation, support vector machines.**

I. Introduction

SPEECH separation has many important real-world applications such as hearing aids design and robust automatic speech recognition (ASR). However, separation performance in general acoustic environments is far from being satisfactory. Monaural speech separation is particularly difficult as one has access only to a single-channel noisy signal. In this case, intrinsic speech or noise properties need to be exploited for effective separation. In this paper, we focus on monaural speech separation from nonspeech background interference.

Spectral subtraction (e.g., [3]) is a classical method for noise reduction, which subtracts an estimate of the noise spectrum from the mixture spectrum. Wiener filtering and mean-squared error estimation methods (e.g., [12], [19]) are also widely used in the speech enhancement community. However, assumptions regarding the statistical properties of noise are crucial to speech enhancement methods, such as stationarity which is hard to satisfy for general acoustic environments. Recent model-based methods separate target speech by estimating Wiener gains (e.g., [15], [36]), but statistical source models are usually required or need to be adapted. Inspired by human auditory processing, computational auditory scene analysis (CASA) [45] has the potential to deal with more general kinds of interference by utilizing auditory-based grouping cues. However, existing CASA systems have limited capability, especially in dealing with unvoiced speech which lacks harmonic structure.

The ideal binary mask (IBM) has been suggested as a primary computational goal for CASA algorithms [44]. The IBM is a time-frequency (T-F) mask constructed from pre-mixed speech and noise. For each T-F unit, if the signal-to-noise (SNR) ratio is greater than a local SNR criterion (LC), we call it target-dominant and the corresponding mask element in the IBM is set to 1. Otherwise, the mask element is set to 0 and we call the unit interference-dominant. Quantitatively, the IBM is defined as:

$$IBM(t,f) = \begin{cases} 1, & \text{if } SNR(t,f) > LC \\ 0, & \text{otherwise} \end{cases}$$

where $SNR(t,f)$ denotes the local SNR (in decibels) within the T-F unit at time t and frequency f. It has been shown that large speech intelligibility gains can be achieved by IBM processing, even for mixtures with very low SNR [6], [34]. It has also been shown that if the IBM is well estimated, separation algorithms can indeed improve speech intelligibility [29], [38]. The effectiveness of IBM estimation has also been demonstrated for robust ASR [18], [39].

Since binary decisions are made for IBM estimation, it is natural to cast speech separation as a binary classification problem [45]. Substantial advances have been made along this line [17], [30], [31], [38], [39], [46]. Following this line of research, our task in this study is to estimate the IBM through binary classification. For classification, both features and classifiers are important. For the choice of classifier, our previous work [17] has shown Gaussian-kernel support vector machines (SVMs) perform better than Gaussian mixture models (GMMs). In [46], we have also identified a set of T-F unit-level complementary features that performs very well in matched test conditions. However, we still observe a significant performance gap between matched and unmatched test conditions.

The issue of generalization is critical for classification as a form of supervised learning. When feature distributions of the test set differ significantly from those in the training set, the

Manuscript received June 07, 2012; revised September 26, 2012; accepted February 21, 2013. Date of publication March 07, 2013; date of current version March 22, 2013. This work was supported in part by the Air Force Office of Scientific Research (AFOSR) under Grant (FA9550-12-1-0130), in part by an STTR subcontract from Kuzer, and in part by the Ohio Supercomputer Center. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Bryan Pardo.

Y. Wang is with the Department of Computer Science and Engineering, The Ohio State University, Columbus, OH 43210 USA (e-mail: wangyuxu@cse.ohio-state.edu).

D. Wang is with the Department of Computer Science and Engineering and the Center for Cognitive Science, The Ohio State University, Columbus, OH 43210 USA (e-mail: dwang@cse.ohio-state.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASL.2013.2250961

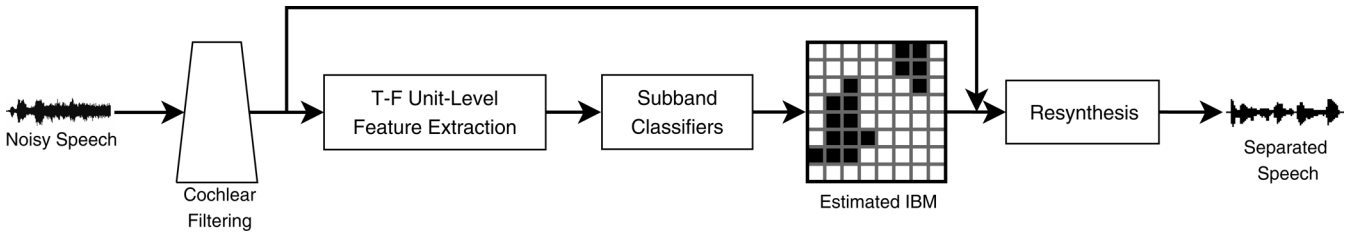


Fig. 1. Schematic diagram of a typical classification-based speech separation system.

learned decision boundary may no longer be discriminative, leading to poor classification performance. Factors causing poor generalization could be many. Different speakers, background noises, input SNRs, room reverberations and channel distortions can all introduce severe mismatches between training and test conditions. Nevertheless, good generalization is key to a speech separation system; otherwise real-world deployment would be problematic. To cope with the mismatch problem, model adaptation could potentially be helpful, but adaptation of kernel SVMs is nontrivial. A straightforward solution is to include a variety of acoustic conditions into the training set to sufficiently cover different kinds of variation. This would dramatically increase the size of the training set. Kernel SVMs cannot handle large datasets due to the expensive quadratic programming. The overall complexity of a conventional kernel SVM is usually between $\mathcal{O}(N^2)$ and $\mathcal{O}(N^3)$ [4], where N is the number of training samples. It is hard to train such a classifier even on hundreds of thousands of samples with reasonably short time. Approximate training methods exist, but their performance are usually significantly worse.

The objective of this paper is to alleviate the generalization issue by training with a large variety of acoustic conditions coupled with the use of linear SVMs [14], [40], which scale well with the size of the training set and can easily handle millions of training samples. To employ linear SVMs, acoustic features for classification need to be linearly separable, which is not the case for unit-level acoustic features. To address this issue, we propose to discriminatively learn new features from raw acoustic features using feedforward multilayer neural networks. The last hidden layer representations of such networks are more linearly separable and are therefore taken as the features for training linear SVMs. We want to point out that this study is not about scaling up deep neural networks or SVMs, which is an important but different research topic.

To enable better and more robust feature learning, these feature learning neural networks are pre-trained using restricted Boltzmann machines (RBMs), which are generative models and serve as pre-training for the recently proposed deep belief networks (DBNs) [21], [22]. Neural networks with many hidden layers can be viewed as hierarchical feature detectors that capture higher-order correlations between raw features. However, prior to DBN, training deep neural networks using the back-propagation algorithm was considered nearly impossible due to problems such as vanishing gradients and pathological objective function landscapes. DBNs pre-train each layer generatively using RBMs. This way of initializing network weights has empirically been proven effective [13], and there is an increasing number of successful applications of DBNs (or its way of network initialization), first in visual processing (e.g., [33]) and more recently, in speech processing [11], [35].

The rest of the paper is organized as follows. We first describe a typical classification-based speech separation system, and illustrate the generalization issue in Section II. We then introduce the proposed DNN-SVM speech separation system in Section III, and present a series of pilot experiments in Section IV. The resulting system is trained on a relatively large dataset, and experimental results are presented in Section V. Discussions and conclusions are provided in Section VI.

II. Speech Separation as Binary Classification

A. Framework

As mentioned before, we aim to estimate the IBM via binary classification. Fig. 1 shows the framework of formulating speech separation as binary classification. A sound mixture with the 16 kHz sampling rate is passed through a 64-channel gammatone filterbank with center frequencies spanning from 50 Hz to 8000 Hz on the equivalent rectangular bandwidth rate scale. The output from each channel is divided into 20-ms frames with 10-ms frame shift, producing a T-F representation called cochleagram [45], which consists of a matrix of T-F units. To estimate the IBM, we classify each T-F unit in the cochleagram as either target-dominant or interference-dominant through supervised training. Due to different spectral properties across frequency, a binary classifier, e.g., an SVM, is trained for each filter channel (subband classifier), where the training labels are provided by the IBM. Since a binary decision needs to be made for each T-F unit, features for classification are extracted from each T-F unit as described in [46], where a complementary feature set was also identified. The feature set consists of amplitude modulation spectrogram (AMS), relative spectral transform and perceptual linear prediction (RASTA-PLP), mel-frequency cepstral coefficients (MFCC) and pitch-based features. RASTA-PLP and pitch-based features are important for generalization to unseen conditions. In training, ground truth pitch is extracted from clean speech using PRAAT [2]. In testing, the pitch estimated from a recent multi-pitch tracker [28] is used to initialize the tandem algorithm [24], which produces the final estimated pitch points. The classification results from the 64 subband classifiers yield an estimated IBM. By binary weighting of the cochleagram using the estimated IBM (i.e., retain the target-dominant T-F units and discard the rest in the cochleagram), the target speech is separated from the sound mixture in a resynthesis step [45].

Since the task is classification, it is straightforward to measure the performance using classification accuracy. However,

TABLE I
HIT-FA RESULTS FOR TWO CLASSIFIERS TRAINED ON DIFFERENT NUMBERS OF NOISES

Classifier	Matched-noise condition			Unmatched-noise condition		
	HIT	FA	HIT−FA	HIT	FA	HIT−FA
S50N3	85.0%	7.4%	77.6%	82.6%	20.8%	61.8%
S50N12	81.6%	7.4%	74.2%	78.3%	11.6%	66.7%

simply using accuracy as the evaluation criterion may not be appropriate, as miss and false-alarm errors are treated equally. Speech intelligibility studies [30], [34] have shown that false-alarm (FA) errors are far more detrimental to human speech intelligibility than miss errors. Their difference, the HIT-FA rate, has been shown to be well correlated to intelligibility by Kim *et al.* [30]. The HIT rate is the percent of correctly classified target-dominant (1's) T-F units in the IBM. The FA rate is the percent of wrongly classified interference-dominant (0's) T-F units in the IBM. Therefore, we use HIT-FA as our main evaluation criterion for assessing classification-based speech separation systems.

B. Generalization Issue

A recently proposed classification-based separation system [17] adopts Gaussian-kernel SVMs as subband classifiers (see Fig. 1). We show that such a system has limited generalization to unseen environments if it is only trained on small datasets. We examine the generalization issue with respect to two dimensions: different noises and different speakers. We train kernel SVMs using the IEEE corpus [26] and a subset of 100 environmental noises [25] for the following proof-of-concept experiments. We use a 0-dB LC value for all the experiments in this paper.

First, we train two Gaussian-kernel SVMs on 50 IEEE female utterances mixed with first 3 noises ($N_1 - N_3$) and then 12 noises (including $N_1 - N_3$) at 0 dB. These two classifiers, which we call S50N3 and S50N12 respectively, are tested in two test conditions. Ten new IEEE female utterances (same speaker) are mixed with $N_1 - N_3$ to create a matched-noise test condition, and 5 unseen noises to create an unmatched-noise test condition, all at 0 dB. Table I presents the overall HIT-FA rates for the two classifiers. S50N3 outperforms S50N12 in the matched-noise condition due to higher HIT rates, because it is exclusively trained on $N_1 - N_3$. However, S50N12 significantly outperforms S50N3 in the unmatched-noise test condition due to much lower FA rates. One might question if the improvement of S50N12 in the unmatched-noise test condition is simply due to an increase in the number of training samples. Our experiments with a classifier trained on 200 IEEE female utterances mixed with $N_1 - N_3$ indicate that increasing the number of training utterances only leads to improved performance in the matched-noise condition but not the unmatched-noise condition. To conclude, increasing the number of training noises clearly improves the generalization to unseen noises.

Next, we examine the situation when the test speaker differs from the training one. We train three classifiers for comparisons. The first and second are trained on the IEEE female and male utterances respectively, while the third is trained on both. Five noises are randomly chosen to mix with the training utterances at 0 dB to create the training set. The test and training noises are the same but the mixtures of both genders are tested by the three classifiers. Fig. 2 shows the HIT-FA rates. We can see that while the first two classifiers perform well in matched-speaker scenarios, their performance significantly degrades when tested on a new speaker. Different speakers, especially different genders, may have different energy distributions across frequency channels, hence posing difficulties for classifiers that are insufficiently trained. In contrast, the behavior of the third classifier suggests the effectiveness of training on multiple speakers.

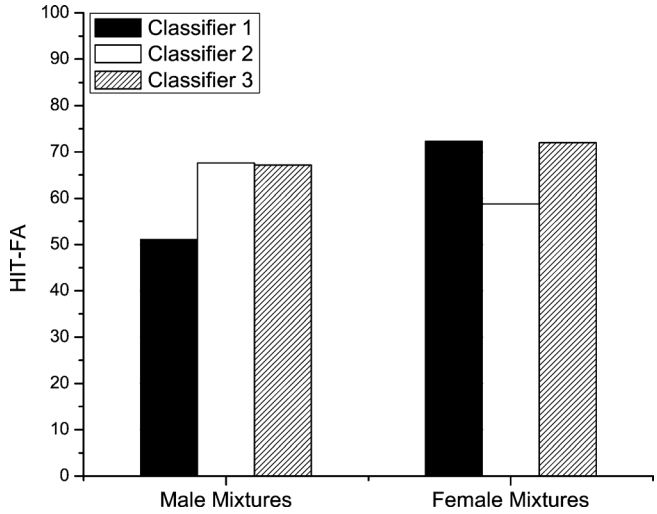


Fig. 2. HIT-FA results when tested on different speakers. The first and second classifiers are trained on the IEEE female and male utterances, respectively. The third classifier is trained on both.

In conclusion, we have shown that classification-based speech separation has to address the generalization issue, and the issue may be alleviated by expanded training on more acoustic conditions. Even when generalization is not an issue, e.g., when the system is deployed in a matched environment, increasing the number of training utterances could still be helpful [29], [46]. Therefore large-scale training is a promising direction for handling generalization of classification-based separation systems. On the other hand, the high complexity of kernel SVMs makes large-scale training prohibitive. This motivates us to study alternative subband classifiers that have both good performance and scalability.

III. DNN-SVM System for Speech Separation

A. Restricted Boltzmann Machines

Boltzmann machines are probabilistic, generative models, which can be used to find regularities (features) hidden in raw input. Restricted Boltzmann machines (RBMs) [21] are two-layer neural networks with a visible layer and a hidden layer. RBMs simplify Boltzmann machines by allowing connections only between the visible and hidden layer. An RBM has an energy function defining joint probability:

$$p(\boldsymbol{v}, \boldsymbol{h}) = \frac{e^{-E(\boldsymbol{v}, \boldsymbol{h})}}{Z}, \tag{1}$$

where $\boldsymbol{v}$ and $\boldsymbol{h}$ denote a visible and hidden layer configuration, respectively. Z is called the partition function to ensure $p(\boldsymbol{v}, \boldsymbol{h})$

is a valid probability distribution. The hidden layer is binary and hidden units are Bernoulli random variables. But the visible layer $\boldsymbol{v}$ can be either binary or real-valued, the latter being more suitable for modeling acoustic features. If we assume visible units are Gaussian random variables with unit variance, we can define the energy function E for this Gaussian-Bernoulli RBM as:

$$E(\boldsymbol{v},\boldsymbol{h}) = \frac{1}{2}\sum_i (v_i - a_i)^2 - \sum_j b_j h_j - \sum_{i,j} w_{ij} v_i h_j, \quad (2)$$

where v_i and h_j are the ith and jth units of $\boldsymbol{v}$ and $\boldsymbol{h}$, a_i and b_i are the biases for v_i and h_j, respectively, and w_{ij} is the symmetric weight between h_j and v_i.

The fact that an RBM is bipartite makes inference of $\boldsymbol{h}$ easy as conditional distribution $p(\boldsymbol{h}|\boldsymbol{v})$ and $p(\boldsymbol{v}|\boldsymbol{h})$ factorize to $p(\boldsymbol{h}|\boldsymbol{v}) = \prod_j p(h_j|\boldsymbol{v})$ and $p(\boldsymbol{v}|\boldsymbol{h}) = \prod_i p(v_i|\boldsymbol{h})$, and,

$$p(h_j = 1|\boldsymbol{v}) = \sigma(b_j + \sum_i w_{ji} v_i) \quad (3)$$

$$p(v_i|\boldsymbol{h}) = \mathcal{N}(v_i; a_i + \sum_j w_{ij} h_j, 1), \quad (4)$$

where $\sigma(x) = 1/(1+e^{-x})$ is the standard logistic sigmoid function and $\mathcal{N}$ denotes the normal distribution.

To train an RBM, we need to calculate the gradient of the log likelihood. Let θ denote any parameter in E, given a training sample $\boldsymbol{v_o}$, the gradient is the difference between expectation under empirical distribution and expectation under model distribution [32]:

$$\frac{\partial(\log p(\boldsymbol{v_o}))}{\partial\theta} = E_{p(\boldsymbol{h}|\boldsymbol{v_o})}\left[\frac{\partial E(\boldsymbol{v_o},\boldsymbol{h})}{\partial\theta}\right] - E_{p(\boldsymbol{v},\boldsymbol{h})}\left[\frac{\partial E(\boldsymbol{v},\boldsymbol{h})}{\partial\theta}\right]. \quad (5)$$

While inference is easy, exact learning is still hard in RBMs as the calculation of the second term in (6) is intractable. Hinton [20] suggests to use contrastive divergence to approximate the gradient. For example, when optimizing the network weights, the derivative can be approximated as:

$$\frac{\partial(\log p(\boldsymbol{v_o}))}{\partial w_{ij}} \approx \langle v_i h_j\rangle_{data} - \langle v_i h_j\rangle_{reconstruction}. \quad (6)$$

Here $\langle\cdot\rangle$ means correlation. The first term in (6) can be easily calculated after doing a forward pass. The second term measures the correlation between v_i and h_j whose activities are generated (reconstructed) by the model, i.e., they are generated by alternatively applying (3) and (4). This is essentially Gibbs sampling, and in practice it is found that one full step of Gibbs sampling often works reasonably well. Two important details need to be pointed out. First, in the reconstruction step of a Gaussian-Bernoulli RBM, we never directly sample from (4). Instead, we only take the mean value as the reconstruction. Second, the Gibbs sampler should start from $\boldsymbol{v}_0$ rather than a random state. Mini-batch or stochastic gradient descent (SGD) is usually used to perform optimization following the approximated gradient. The training of Bernoulli-Bernoulli RBMs is similar, where the main difference is that the visible units are Bernoulli random variables now (also the energy function is slightly different, see e.g., [32]). RBMs have been successfully applied as building blocks for the DBN [21], which is a powerful multilayer generative model. For more technical discussions and implementation details, we refer the interested readers to [21], [32] and [35].

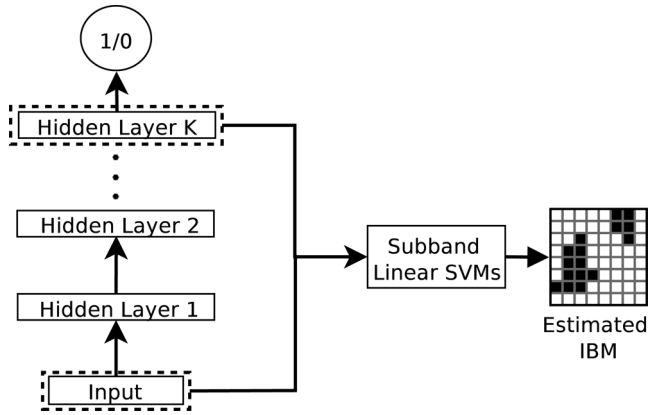


Fig. 3. Schematic diagram of the proposed DNN-SVM system for IBM estimation. Both feature learning and linear SVM training are carried out for each filter channel (i.e., DNN-SVM serves as the subband classifier).

B. DNN-SVM: Architecture

Fig. 3 illustrates the architecture of the proposed DNN-SVM speech separation system, where DNN-SVM serves as the subband classifier. The first stage of the system involves training a feedforward neural network to learn feature encoding. To overcome issues in training multilayer perceptrons (MLPs), the network is unsupervisedly pre-trained using RBMs in a greedy layerwise fashion. Raw acoustic features are used as training data to train the first RBM, whose hidden activations are then treated as the new training data for the second RBM, and so on. We use a Gaussian-Bernoulli RBM for the first layer and Bernoulli-Bernoulli RBMs for all the layers above. The weights resulting from training a stack of RBMs are used to initialize a feedforward neural network. This way of initialization has empirically been found to aid the subsequent backpropagation training and provide a measure of regularization [13]. The advantage of RBM pre-training remains even when a large number of training samples are used [13], and it is often critical for training a deep network having many hidden layers [13], [32]. To make internal representations discriminative, the whole network is then supervisedly fined-tuned using the backpropagation algorithm with a logistic output layer. We choose the limited-memory Broyden-Fletcher-Goldfarb-Shanno algorithm (L-BFGS) as the optimizer for backpropagation learning.

We choose the last hidden layer activations as the learned features after the network is sufficiently fine-tuned. The weights from the last hidden layer to the output layer would essentially define a linear classifier, hence the last hidden layer activations are more amenable to linear classification. While it is true that DNN outputs already form an estimated IBM, in practice we find that concatenating the learned features with the raw features could result in HIT-FA improvements. Considering the previous success of using SVMs for speech separation, we train subband linear SVMs on the concatenated features for final IBM estimation. The estimated IBM can be further enhanced by using auditory segmentation [17].

Quite a few algorithms, such as sparse coding, local coordinate coding, and k-means, can be used for feature learning (e.g., [9], [10]), but we choose neural networks for several reasons:

- Deep neural networks can be viewed as hierarchical feature detectors, which can potentially capture higher-order correlations between raw features better than shallow methods [1]. Unlike some existing methods, discriminative feature learning is conveniently handled by the backpropagation algorithm.
- Relatively speaking, gradient descent based training does not have large complexity and has good scalability. Compared to kernel SVMs, backpropagation training is much more scalable especially when using mini-batch SGD, which is both theoretically and practically suitable for large-scale learning [5] and naturally permit online learning. This is important as we do not want the feature learning stage to become a bottleneck for the overall training.
- Feature decoding is extremely fast in feedforward networks. The learned features are obtained by passing the raw data through the network. This is important for both efficient training and real-time deployment. This is, however, not always the case for other feature learning algorithms. For example, sparse coding sometimes requires solving a new optimization problem to get the learned features [10].

IV. Pilot Experiments on DNN-SVM

A number of design choices have to be made before training DNN-SVM on large datasets. Here, we present some pilot studies for DNN-SVM using a relatively small corpus, created by mixing 50 IEEE female utterances with 12 randomly chosen noises at 0 dB. The test set is created by mixing 10 new utterances with 12 seen noises (matched-noise test condition) and 10 unseen noises (unmatched-noise test condition) at 0 dB. The training set consists of about 150,000 samples for each channel.

We first study the raw features used for feature learning. All the features considered here are from the complementary feature set [46], denoted as COMB. We use 15-D AMS, 13-D RASTA-PLP, 31-D MFCC, and 6-D pitch-based features. COMB or individual features can be used for feature learning and the learned features are always combined with COMB for training linear SVMs. Fig. 4 shows the overall HIT-FA rates using $\mathrm{MFCC}\Delta$, $\mathrm{RASTA\text{-}PLP}\Delta$ and COMB for feature learning, where Δ denotes the first-order delta features. In the matched-noise condition, $\mathrm{MFCC}\Delta$ and $\mathrm{RASTA\text{-}PLP}\Delta$ perform on par and are slightly better than COMB.The COMB features seem to exhibit slight overfitting, possibly due to insufficient samples for unsupervised RBM pre-training. But in unmatched-test condition, $\mathrm{RASTA\text{-}PLP}\Delta$ and COMB hold up well and are much better than $\mathrm{MFCC}\Delta$. This is consistent with our previous conclusion [46] that MFCC does not generalize well. On this small corpus, RASTA-PLP shows a clear advantage to the other features.

Next, we show performance of DNN-SVM with and without unsupervised RBM pre-training. We use mini-batch gradient descent with a batch size of 256 for RBM pre-training. Without pre-training, DNN is essentially the same as MLP. Fig. 5 shows

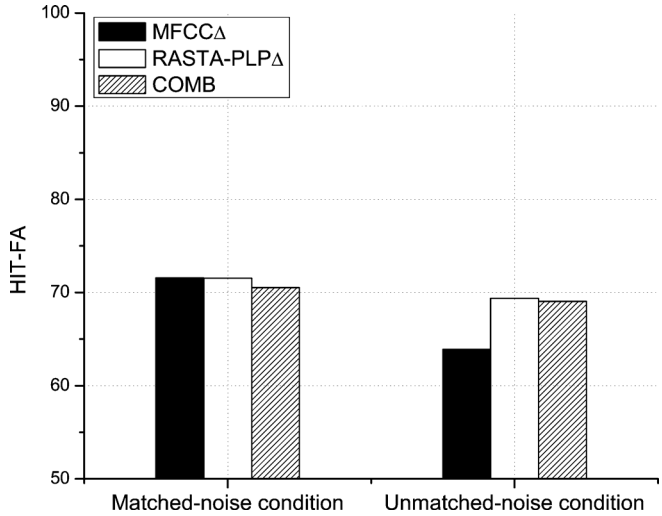


Fig. 4. HIT-FA results in two test conditions. Different raw features are used in feature learning.

the comparisons as a function of the number of hidden layers. First, 39-D $\mathrm{RASTA\text{-}PLP}\Delta\Delta$ (delta and acceleration) is used as the input for feature learning, and 50 units are used for each hidden layer. From Fig. 5(a) and (b), we can see that RBM pre-training consistently and significantly improves the performance in both matched and unmatched conditions. Adding a second hidden layer improves the results over using a single one, but the improvement is less significant with more hidden layers added. Note that without RBM pre-training, the performance tends to degrade with more hidden layers, especially in the unmatched-noise condition. In our experiments (not shown), we also found that RBM pre-training adds stability to overall training; such a benefit of reducing test error variance by using RBM pre-training is also found in [13]. For more complex networks, using RBM pre-training is more demanding. We employ a network with 100 units for each hidden layer with the COMB input for feature learning. From Fig. 5(c) and (d), we can see that the performance gap between RBM pre-training and no pre-training becomes more significant for both test conditions. This is true even when only one hidden layer is used (i.e., a shallow network). We observe significant overfitting when no pre-training is used, and RBM pre-training seems to alleviate overfitting significantly, which could be attributed to its regularization effect [13]. It is interesting to note that although pre-training is important even for shallow networks, the improvement of using two or more hidden layers over a single one is relatively small. This may be due to the ceiling effects-it is difficult to further improve the already-good performance with a single hidden layer on this particular corpus. To test this possibility, we use a more challenging corpus in which speech utterances are mixed with the speech-shaped and babble noises at -5 dB. Feature learning is carried out on the COMB feature set. From the results shown in Fig. 6, it is clear that using two hidden layers significantly outperforms using one hidden layer, given that the network is pre-trained using RBM. Although using three or more hidden layers does not improve the performance significantly, the situation may be different for other demanding corpora. To conclude, we found that RBM pre-training is important for DNN-SVM, and two hidden layers seem to be a good choice.

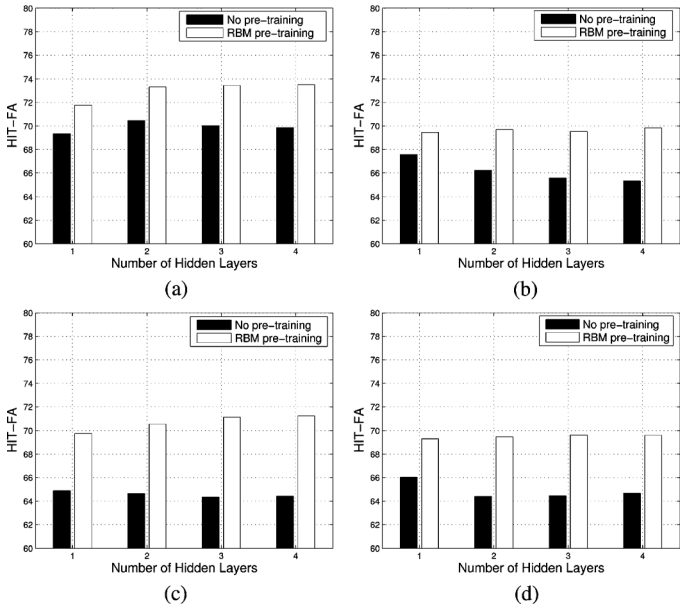


Fig. 5. HIT-FA results with and without RBM pre-training. (a)–(b) Features are learned from RASTA-PLP$\Delta\Delta$. (c)–(d) Features are learned from the COMB feature set. (a) Matched-noise condition. (b) Unmatched-noise condition. (c) Matched-noise condition. (d) Unmatched-noise condition.

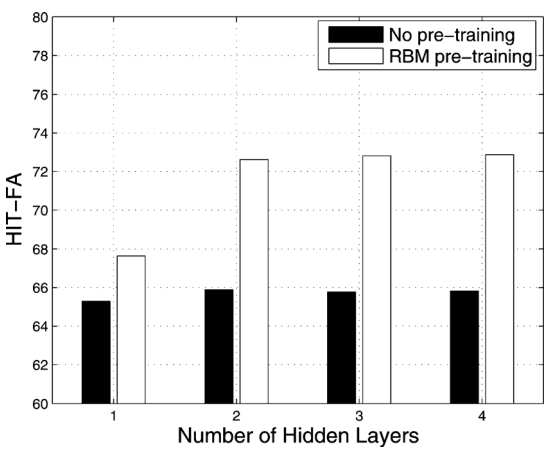


Fig. 6. HIT-FA results with and without RBM pre-training on a more challenging corpus where speech utterances are mixed with the speech-shaped and babble noises at -5 dB.

Finally, to validate the effectiveness of the proposed system, we compare DNN-SVM with linear SVMs and Gaussian-kernel SVMs on the above IEEE training and test set. Linear SVMs and Gaussian-kernel SVMs are trained using the COMB feature set. We employ a two hidden layer DNN with 50 units for each hidden layer to learn features from RASTA-PLP$\Delta\Delta$. We use 100 epochs of mini-batch gradient descent for RBM pre-training and 500 epochs of L-BFGS for network fine-tuning. We document 3 kinds of HIT-FA: voiced intervals, unvoiced intervals, and overall. Voicing boundaries are determined from ground truth pitch. As comparisons, we also include results from a DNN-gSVM system, which is exactly the same as DNN-SVM but with linear SVMs replaced by Gaussian-kernel SVMs. From Table II, we can see that linear SVMs should not be directly used as raw features are not linearly separable. Linear SVM is 16.7% worse than DNN-SVM in terms of overall HIT-FA

TABLE II
HIT-FA RESULTS OF CLASSIFICATION-BASED SPEECH SEPARATION SYSTEMS ON A SMALL-SCALE CORPUS

System	Matched-noise condition			Unmatched-noise condition		
	Overall	Voiced	Unvoiced	Overall	Voiced	Unvoiced
Linear SVM	56.5%	63.0%	34.5%	64.0%	69.1%	41.7%
Gaussian SVM	68.7%	73.4%	51.5%	68.2%	72.4%	48.9%
DNN-SVM	73.2%	75.3%	64.6%	69.8%	72.1%	58.5%
DNN-gSVM	74.3%	76.5%	66.0%	70.3%	72.5%	59.9%

in the matched-noise condition. In our experience, DNN-SVM training is orders of magnitudes faster than kernel SVMs even when kernel cache is turned on [7]. The test time of DNN-SVM is also much less than that of kernel SVMs. Encouragingly, the performance of DNN-SVM is also significantly better than kernel SVMs in the matched-noise condition, with 4.5% and 13.1% HIT-FA improvements in overall and unvoiced intervals, respectively. It is worth noting that unvoiced speech separation is more difficult since unvoiced speech lacks harmonics and has weak energy [23]. The performance gap between DNN-SVM and DNN-gSVM is marginal, indicating that the learned features are indeed amenable to linear classification.

V. RESULTS

A. Experimental Settings

We now scale up DNN-SVM training to a larger dataset. To create the training set, we randomly choose 100 male utterances and 100 female utterances from the TIMIT [16] training part across 8 dialect regions. These 200 utterances are mixed with 100 environmental noises [25] at 0 dB, producing about 6 million, fully dense training samples for each channel (64 channels in total). To create the test set, 20 utterances from different unseen speakers of both genders are randomly chosen from the TIMIT test part. These utterances are mixed with 20 new nonspeech noises[1] compiled from the test noises used by the tandem algorithm [24], the NOISEX corpus [43], and short snippets of nonspeech noises from a corpus [8]. To further evaluate generalization of our system, we create another test set by mixing 10 IEEE female utterances and 10 IEEE male utterances with the above 20 unseen noises. The total number of test samples is about 210,000 for each channel between the two test sets.

Considering performance and computational complexity (see Figs. 5 and 6), we use relatively small DNNs with two hidden layers. The small number of tunable network parameters facilitates fast and scalable training with reasonably good performance. We use 100 epochs of mini-batch gradient descent for RBM pre-training, and 500 epochs of L-BFGS for fine-tuning the whole network. We use a learning rate of 0.001 for the first Gaussian-Bernoulli RBM, and 0.01 for the above Bernoulli-Bernoulli RBM. All the data are variance normalized as assumed by (2).

[1]The 20 unseen noises are N_1: white noise, N_2: cocktail party, N_3: crow noise, N_4: traffic, N_5: playground, N_6: crowd yelling, N_7: crowd laugh, N_8: bird chirp, N_9: strong wind, N_{10}: rain, N_{11}: factory noise 1, N_{12}: speech-shaped noise, N_{13}: F-16, N_{14}: destroyer, N_{15}: factory noise 2, N_{16}: machine operation, N_{17}: electric fan, N_{18}: washer, N_{19}: footstep, and N_{20}: child playing.

TABLE III
HIT-FA RESULTS ON THE 0 dB TIMIT AND IEEE TEST SET. FEATURES ARE LEARNED FROM RASTA-PLP$\Delta\Delta$

System	TIMIT			IEEE Female			IEEE Male		
	Overall	Voiced	Unvoiced	Overall	Voiced	Unvoiced	Overall	Voiced	Unvoiced
Linear SVM	53.7%	62.3%	24.9%	58.5%	63.4%	30.7%	54.0%	59.4%	29.0%
Ideal-tandem	n/a	65.5%	n/a	n/a	66.5%	n/a	n/a	66.0%	n/a
Tandem	n/a	57.7%	n/a	n/a	59.8%	n/a	n/a	65.6%	n/a
DNN-SVM	62.8%	68.3%	44.3%	66.5%	69.4%	49.6%	63.4%	67.3%	45.3%
DNN-SVM-SEG	63.8%	70.0%	44.0%	68.3%	71.7%	49.0%	65.5%	70.0%	46.0%

TABLE IV
HIT-FA RESULTS ON THE -5 dB TIMIT AND IEEE TEST SET. FEATURES ARE LEARNED FROM RASTA-PLP$\Delta\Delta$

System	TIMIT			IEEE Female			IEEE Male		
	Overall	Voiced	Unvoiced	Overall	Voiced	Unvoiced	Overall	Voiced	Unvoiced
Linear SVM	52.2%	59.2%	28.3%	54.3%	58.0%	33.6%	51.2%	55.7%	32.0%
Ideal-tandem	n/a	60.7%	n/a	n/a	59.2%	n/a	n/a	60.4%	n/a
Tandem	n/a	53.4%	n/a	n/a	54.2%	n/a	n/a	60.0%	n/a
DNN-SVM	60.1%	64.5%	45.2%	61.6%	63.8%	49.6%	59.8%	63.0%	44.8%
DNN-SVM-SEG	61.4%	66.3%	45.1%	64.1%	67.0%	49.0%	62.6%	66.3%	45.7%

B. Feature Learning From RASTA-PLP

Features are learned from RASTA-PLP$\Delta\Delta$ with 50 units per hidden layer. The overall training was parallelized to a cluster of computing nodes as the training for each channel is independent. The binary masks are further refined by cross-channel correlation based auditory segmentation [17] and the resulting system is denoted by DNN-SVM-SEG. To put the performance of DNN-SVM systems in perspective, we compare with the tandem algorithm [24], a recent CASA system that generalizes well to unseen scenarios by jointly estimating pitch contours and associated voiced masks. We compare with two versions of the algorithm, named as ideal-tandem and tandem. The first one uses ideal sequential grouping and thus represents the selling performance of the tandem algorithm, while the second one uses pitch-based grouping, which removes pitch contours that are out of the plausible pitch range and selects the longer one if two pitch contours overlap.

Table III reports the HIT-FA rates on 0 dB mixtures. The DNN-SVM system performs significantly better than linear SVMs that are trained using the COMB feature set, indicating that discriminatively learning more linearly separable features is indeed needed. This is especially true for unvoiced speech separation in which pitch-based features can not be used. The DNN-SVM system also outperforms the tandem algorithm for voiced speech separation even with ideal sequential grouping, and is much better than with actual sequential grouping. Comparing DNN-SVM and DNN-SVM-SEG, we can see that auditory segmentation offers some improvement. Although the DNN-SVM system is trained on TIMIT utterances, generalization to other corpora does not seem to be a problem as demonstrated by the results on the IEEE corpus. We have also used the trained models to estimate the IBM for -5 dB mixtures. HIT-FA rates are reported in Table IV. As expected, the results are worse than in Table III but the degradation is not severe. We expect improved results if the systems are also trained on -5 dB mixtures.

It would be interesting to see HIT-FA performance as a function of the number of training noises and utterances. Fig. 7(a) shows the effect of progressively training with more noises (mixed with 200 utterances) on the 0 dB TIMIT test set. The performance increases with the number of training noises, but the overall HIT-FA peaks at 50, which seems enough for the TIMIT test set. It is possible that the performance peaks at other numbers for different test sets. We point out that SVMs are optimized in terms of classification accuracy rather than the HIT-FA rate. In fact, as the number of training noises increases, we observe a monotonically improving trend in terms of classification accuracy. On the other hand, since high accuracy correlates with high HIT-FA, we expect new performance peaks beyond using 100 training noises. Fig. 7(b) shows the effect of progressively training with more utterances (mixed with 20 noises) on the 0 dB TIMIT test set. The performance keeps increasing until 150 utterances.

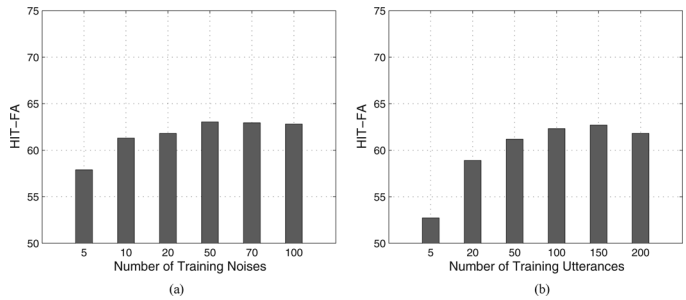


Fig. 7. HIT-FA results on the 0 dB TIMIT test set as a function of (a) the number of training noises (mixed with 200 utterances), and (b) the number of training utterances (mixed with 20 noises).

C. Distance Analysis for Feature Learning

The above experiments suggest that the discriminatively learned features not only enhance linear separability but also improve classification performance, e.g., comparing DNN-SVM with Gaussian-kernel SVMs in Table II. To analyze the effect of discriminative feature learning, we carry out a distance analysis between raw RASTA-PLP$\Delta\Delta$ features and the learned features in a representative channel on the IEEE test set. The distance from the $+1$ class (target-dominant) to the -1 class (interference-dominant) within a feature set could be a quantitative measure of class separability [41]. We employ the constrained minimum (CM) distance [42] as our metric, which has been previously used to study the robustness

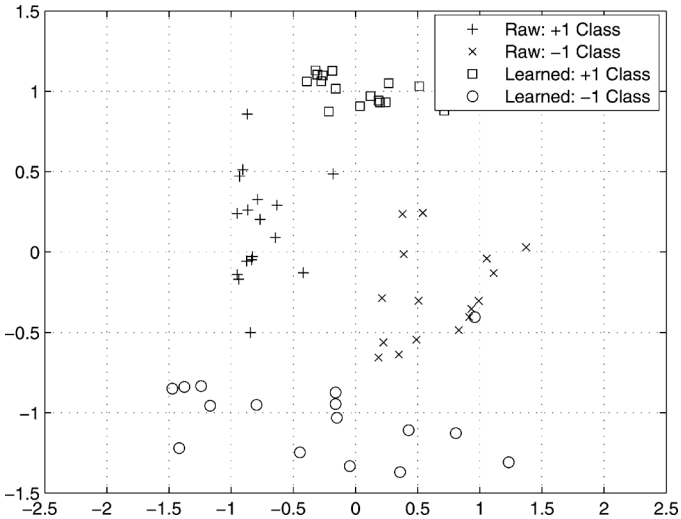


Fig. 8. An MDS distance analysis for the features learned from RASTA-PLP$\Delta\Delta$ in a representative channel. The analysis is carried out on the IEEE test set of a female speaker. The distance between embedded points is a measure of feature separability, i.e., the larger the distance is, the more separable the features are.

of pitch-based features [27]. The CM distance compares the summary statistics between feature sets and is of Mahalanobis type:

$$d_{CM}(D_1, D_2|S)^2 = (\mu_1 - \mu_2)^T cov^{-1}[S](\mu_1 - \mu_2), \quad (7)$$

where μ_1 and μ_2 are the means of the datasets D_1 and D_2, respectively. S is the underlying feature distribution function, which we estimate from the datasets. To visualize the class distribution on a 2-D plane, we calculate the pairwise CM distance between the raw features and the learned features of each noise, and carry out a metric multidimensional scaling (MDS) afterwards. We visualize the 2-D MDS embeddings in Fig. 8, in which each point corresponds to the embedding of a test noise (a few points are excluded for better visualization). The Euclidean distances between the embedded points approximate the original CM distances. We can see that the distance between the $+1$ and -1 class of the learned features is clearly larger than that of the raw features, indicating larger separability brought about by feature learning.

D. Re-Evaluating Feature Learning

We have shown that RASTA-PLP and its variants such as RASTA-PLP$\Delta\Delta$ are more suitable for feature learning than the other features on a small-scale corpus. However, it is possible that the same trend no longer holds when more samples are included in the training set. On the 0 dB TIMIT test set, we re-evaluate each feature's performance as a function of the number of training noises (mixed with 200 utterances). We use 100 units per hidden layer for MFCC$\Delta\Delta$ and COMB. Interestingly, the trend indeed changes as shown in Fig. 9. When trained on 5 and 10 noises, the overall HIT-FA rates of COMB and MFCC$\Delta\Delta$ are significantly lower than that of RASTA-PLP$\Delta\Delta$. However, both of them start to catch up and then outperform RASTA-PLP$\Delta\Delta$ when trained on more than 20 noises. The performance improvement achieved by learning features from COMB is significant. In our previous

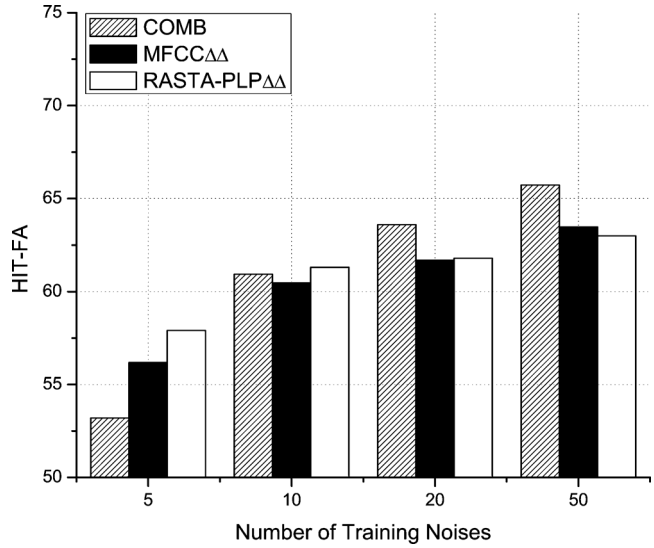


Fig. 9. Overall HIT-FA results of DNN-SVM as a function of the number of training noises (mixed with 200 utterances) on the 0 dB TIMIT test set. Features are learned from RASTA-PLP$\Delta\Delta$, MFCC$\Delta\Delta$ and COMB.

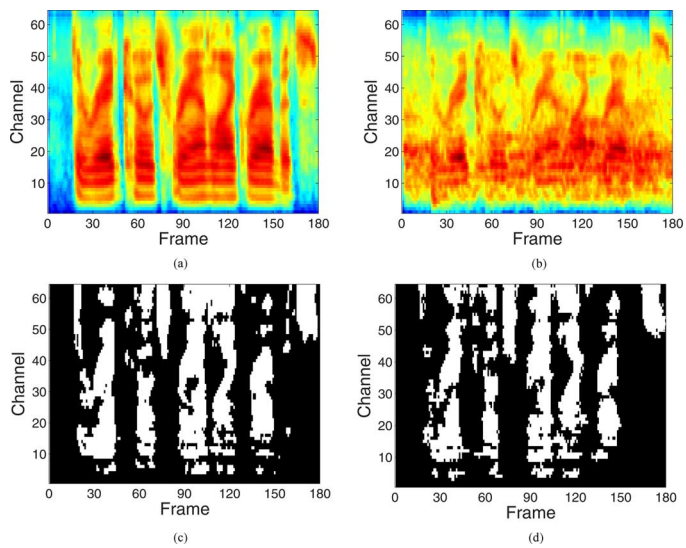


Fig. 10. Separation illustration for a TIMIT utterance mixed with a cocktail party noise. (a) Cochleagram of the utterance. (b) Cochleagram of the mixture. (c) Ideal binary mask. (d) Estimated IBM.

work [46], we showed that COMB outperforms RASTA-PLP in both matched and unmatched test conditions, but MFCC does not generalize as well as RASTA-PLP. The reason why MFCC$\Delta\Delta$ becomes better is likely because the difficulty of generalization diminishes when the feature space is sufficiently covered by the large training set. As the empirical distribution converges to the true distribution, the performance in matched test conditions is indicative of generalization. Besides, there seems to be another reason leading to significantly better feature learning using COMB. As we observed, COMB is more vulnerable to overfitting when using neural networks. But when the training set becomes larger, two things can help. First, unsupervised RBM pre-training is likely more effective given sufficient unlabeled data. Second, the use of more data tends to alleviate overfitting.

The COMB feature set is used for feature learning in our final system. We present the HIT-FA results in Table V. The final

TABLE V
HIT-FA RESULTS OF THE FINAL DNN-SVM-SEG SYSTEM

Mixture SNR	TIMIT			IEEE Female			IEEE Male		
	Overall	Voiced	Unvoiced	Overall	Voiced	Unvoiced	Overall	Voiced	Unvoiced
0 dB	66.9%	72.1%	50.3%	70.9%	73.9%	55.0%	68.1%	71.8%	51.9%
-5 dB	63.8%	67.8%	50.8%	65.9%	68.2%	54.0%	64.7%	67.5%	52.5%

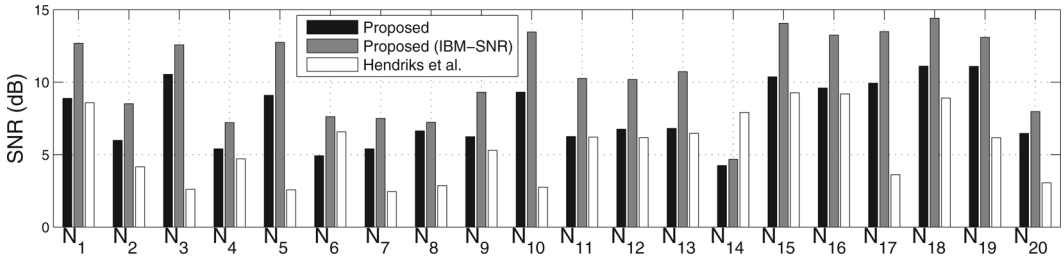


Fig. 11. SNR comparison between the final DNN-SVM-SEG system and Hendriks et al.'s algorithm [19] on the 0 dB TIMIT test set. "IBM-SNR" stands for the IBM-modulated SNR.

DNN-SVM-SEG system achieves promising results in terms of generalization to new noises and speakers. Fig. 10 illustrates the separation results for a TIMIT test utterance mixed with a cocktail party noise at 0 dB.

As a final comparison, we compare with a state-of-the-art speech enhancement algorithm [19]. Since speech enhancement does not aim to estimate the IBM, we compare waveforms directly by measuring the SNR of the separated speech. Aside from the traditional SNR (using clean speech as the ground truth), we also present the IBM-modulated SNR for the proposed system, which uses the target speech resynthesized from the IBM as the ground truth. The IBM-modulated SNR is considered a more appropriate measure [24], as the IBM represents the ground truth of classification. We show the SNR comparisons on the 0 dB TIMIT test set in Fig. 11. Our system significantly outperforms the speech enhancement algorithm on most of the noises. On average, our system obtains 10.5 dB IBM-modulated SNR gain and 7.9 dB SNR gain, while the speech enhancement algorithm obtains 5.4 dB SNR gain.

VI. CONCLUDING REMARKS

We have described our first attempt towards scaling up classification-based speech separation systems. Conventional systems are usually trained on small datasets. This has been shown to be problematic in terms of generalization. Even if T-F unit features used for classification are robust to changing background noises, generalization to new speakers and SNRs is still an issue. We showed that the mismatch problem could be significantly alleviated by training on more acoustic conditions. However, the resulting large training set poses a big challenge to conventional kernel SVMs, which have huge complexity and poor scalability. We have proposed to learn more linearly separable features from raw acoustic features. Linear SVMs are then trained on the combination of learned and raw features to estimate the IBM. We choose neural networks for feature learning due to their scalability and flexibility. With the goal of estimating the IBM, we have shown that a set of small, standard, RBM pre-trained neural networks coupled with linear SVMs can be practically trained on a variety of speakers and noises, and the resulting classification performance is clearly better than Gaussian-kernel SVMs and outperforms related separation systems. To our knowledge, this is the first study that employs supervised deep neural networks for speech separation. The final DNN-SVM-SEG system discriminatively learns features from a complementary feature set, and produces promising generalization results. We note that the DNN-SVM system can also generate ratio or soft masks either by using the probabilities from the logistic output layer of DNN or by mapping SVM outputs to posterior probabilities [37]. How to train such a system to estimate a Wiener filter (a ratio mask) is an interesting topic for future study.

Further improvements lie in new advances in learning algorithms and feature extraction. For example, our system partly relies on pitch-based features, and with better pitch tracking in noisy environments, the overall classification performance is expected to improve. In fact, the current system can be trained on much larger datasets by using graphics processing units (GPUs) and switching the optimizer to stochastic (sub)gradient descent for both DNN and SVM [40]. For massive datasets, however, methods for parallelizing first and (quasi) second-order optimization methods are needed. Finally, we point out that the context information of T-F units could be better utilized in future work. Exploiting the spectrotemporal structure within the deep learning framework is promising [47].

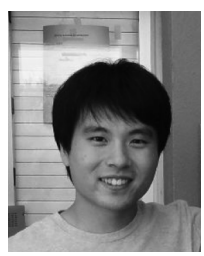
ACKNOWLEDGMENT

The authors would like to thank M. Belkin for early discussions.

REFERENCES

[1] Y. Bengio, "Learning deep architectures for AI," *Foundat. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.
[2] P. Boersma and D. Weenink, Praat: Doing Phonetics by Computer (Version 4.3.14) 2005 [Online]. Available: http://www.fon.hum.uva.nl/praat
[3] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. ASSP-27, no. 2, pp. 113–120, Apr. 1979.
[4] A. Bordes, S. Ertekin, J. Weston, and L. Bottou, "Fast kernel classifiers with online and active learning," *J. Mach. Learn. Res.*, vol. 6, pp. 1579–1619, 2005.
[5] L. Bottou and O. Bousquet, "The tradeoffs of large scale learning," in *Adv. Neural Inf. Process. Syst. 20*, 2008, pp. 161–168.

[6] D. Brungart, P. Chang, B. Simpson, and D. Wang, "Isolating the energetic component of speech-on-speech masking with ideal time-frequency segregation," *J. Acoust. Soc. Amer.*, vol. 120, pp. 4007–4018, 2006.
[7] C. Chang and C. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 27–27, 2011.
[8] H. Christensen, J. Barker, N. Ma, and P. Green, "The CHiME corpus: A resource and a challenge for computational hearing in multisource environments," in *Proc. Interspeech*, 2010.
[9] A. Coates, H. Lee, and A. Ng, "An analysis of single-layer networks in unsupervised feature learning," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011.
[10] A. Coates and A. Ng, "The importance of encoding versus training with sparse coding and vector quantization," in *Proc. 28th Int. Conf. Mach. Learn.*, 2011.
[11] G. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large vocabulary speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 1, pp. 30–42, Jan. 2012.
[12] Y. Ephraim and D. Malah, "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. ASSP-32, no. 6, pp. 1109–1121, Dec. 1984.
[13] D. Erhan, Y. Bengio, A. Courville, P. Manzagol, P. Vincent, and S. Bengio, "Why does unsupervised pre-training help deep learning?," *J. Mach. Learn. Res.*, vol. 11, pp. 625–660, 2010.
[14] R. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin, "LIBLINEAR: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, 2008.
[15] C. Févotte, N. Bertin, and J.-L. Durrieu, "Nonnegative matrix factorization with the itakura-saito divergence: With application to music analysis," *Neural Comput.*, vol. 21, no. 3, pp. 793–830, 2009.
[16] J. Garofolo*, DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus*. Gaithersburg, MD, USA: National Inst. of Standards and Technology, 1993.
[17] K. Han and D. Wang, "A classification approach to speech segregation," *J. Acoust. Soc. Amer.*, vol. 132, pp. 3475–3483, 2012.
[18] W. Hartmann and E. Fosler-Lussier, "Investigations into the incorporation of the ideal binary mask in ASR," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2011, pp. 4804–4807.
[19] R. Hendriks, R. Heusdens, and J. Jensen, "MMSE based noise PSD tracking with low complexity," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process.*, 2010, pp. 4266–4269.
[20] G. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Comput.*, vol. 14, no. 8, pp. 1771–1800, 2002.
[21] G. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.
[22] G. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–504, 2006.
[23] G. Hu and D. Wang, "Segregation of unvoiced speech from nonspeech interference," *J. Acoust. Soc. Amer.*, vol. 124, pp. 1306–1319, 2008.
[24] G. Hu and D. Wang, "A tandem algorithm for pitch estimation and voiced speech segregation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 8, pp. 2067–2079, Nov. 2010.
[25] G. Hu, 100 nonspeech environmental sounds 2004 [Online]. Available: http://www.cse.ohio-state.edu/pnl/corpus/HuCorpus.html
[26] "IEEE recommended practice for speech quality measurements," *IEEE Trans. Audio Electroacoust.*, vol. 17, no. 3, pp. 225–246, Sep. 1969.
[27] Z. Jin and D. Wang, "A supervised learning approach to monaural segregation of reverberant speech," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 17, no. 4, pp. 625–638, May 2009.
[28] Z. Jin and D. Wang, "HMM-based multipitch tracking for noisy and reverberant speech," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 5, pp. 1091–1102, Jul. 2011.
[29] G. Kim and P. Loizou, "Improving speech intelligibility in noise using environment-optimized algorithms," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 8, pp. 2080–2090, Nov. 2010.
[30] G. Kim, Y. Lu, Y. Hu, and P. Loizou, "An algorithm that improves speech intelligibility in noise for normal-hearing listeners," *J. Acoust. Soc. Amer.*, vol. 126, pp. 1486–1494, 2009.
[31] W. Kim and R. Stern, "Mask classification for missing-feature reconstruction for robust speech recognition with unknown background noise," *Speech Commun.*, vol. 53, no. 1, pp. 1–11, 2011.
[32] H. Larochelle, Y. Bengio, J. Louradour, and P. Lamblin, "Exploring strategies for training deep neural networks," *J. Mach. Learn. Res.*, vol. 10, pp. 1–40, 2009.
[33] H. Lee, R. Grosse, R. Ranganath, and A. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proc. 26th Int. Conf. Mach. Learn.*, 2009, pp. 609–616.
[34] N. Li and P. Loizou, "Factors influencing intelligibility of ideal binary-masked speech: Implications for noise reduction," *J. Acoust. Soc. Amer.*, vol. 123, no. 3, pp. 1673–1682, 2008.
[35] A. Mohamed, G. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 1, pp. 14–21, Jan. 2012.
[36] A. Ozerov, E. Vincent, and F. Bimbot, "A general flexible framework for the handling of prior information in audio source separation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 4, pp. 1118–1133, May 2012.
[37] J. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Adv. Large Margin Classifiers*, pp. 61–74, 1999.
[38] N. Roman, D. Wang, and G. Brown, "Speech segregation based on sound localization," *J. Acoust. Soc. Amer.*, vol. 114, pp. 2236–2252, 2003.
[39] M. Seltzer, B. Raj, and R. Stern, "A Bayesian classifier for spectrographic mask estimation for missing feature speech recognition," *Speech Commun.*, vol. 43, no. 4, pp. 379–393, 2004.
[40] S. Shalev-Shwartz, Y. Singer, and N. Srebro, "Pegasos: Primal estimated sub-gradient solver for SVM," in *Proc. 24th Int. Conf. Mach. learn.*, 2007, pp. 807–814.
[41] S. Singh, "Multiresolution estimates of classification complexity," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 12, pp. 1534–1539, Dec. 2003.
[42] N. Tatti, "Distances between data sets based on summary statistic," *J. Mach. Learn. Res.*, vol. 8, pp. 131–154, 2007.
[43] A. Varga and H. Steeneken, "Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems," *Speech Commun.*, vol. 12, pp. 247–251, 1993.
[44] D. Wang, "On ideal binary mask as the computational goal of auditory scene analysis," in *Speech Separation by Humans and Machines*, P. Divenyi, Ed. Norwell, MA, USA: Kluwer, 2005, pp. 181–197.
[45] , D. Wang and G. Brown, Eds.*, Computational Auditory Scene Analysis: Principles, Algorithms and Applications*. Hoboken, NJ, USA: Wiley-IEEE Press, 2006.
[46] Y. Wang, K. Han, and D. Wang, "Exploring monaural features for classification-based speech segregation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 2, pp. 270–279, Feb. 2013.
[47] Y. Wang and D. Wang, "Cocktail party processing via structured prediction," in *Adv. Neural Inf. Process. Syst. 25*, 2012, pp. 224–232.



Yuxuan Wang received his B.E. degree in network engineering from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2009. He is currently pursuing his Ph.D. degree at The Ohio State University. He is interested in machine learning, optimization, speech separation, and computational neuroscience.

DeLiang Wang, (F'04) photograph and biography not available at the time of publication.