INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Kernel Machines Beat Deep Neural Networks on Mask-based Single-channel Speech Enhancement

Like Hui, Siyuan Ma, Mikhail Belkin

Department of Computer Science and Engineering, The Ohio State University, USA

hui.87@osu.edu, {masi, mbelkin}@cse.ohio-state.edu

Abstract

We apply a fast kernel method for mask-based single-channel speech enhancement. Specifically, our method solves a kernel regression problem associated to a non-smooth kernel function (exponential power kernel) with a highly efficient iterative method (EigenPro). Due to the simplicity of this method, its hyper-parameters such as kernel bandwidth can be automatically and efficiently selected using line search with subsamples of training data. We observe an empirical correlation between the regression loss (mean square error) and regular metrics for speech enhancement. This observation justifies our training target and motivates us to achieve lower regression loss by training separate kernel models for different frequency subbands. We compare our method with the state-of-the-art deep neural networks on mask-based HINT and TIMIT. Experimental results show that our kernel method consistently outperforms deep neural networks while requiring less training time.

Index Terms: large-scale kernel machines, deep neural networks, speech enhancement, exponential power kernel, automatic hyper-parameter selection

1. Introduction

Speech enhancement aims at reducing noise from target speech. The challenging problem of this task has received significant attention in research and applications. In recent years the dominant methodology for addressing single-channel speech enhancement has been based on neural networks of different architectures [1, 2]. Deep Neural Networks (DNNs) present an attractive learning paradigm due to their empirical success on a range of problems and efficient optimization.

In this paper, we demonstrate that modern large-scale kernel machines are a powerful alternative to DNNs, capable of matching and surpassing their performance while utilizing less computational resources in training. Specifically, we take the approach to speech enhancement based on the Ideal Binary Mask (IBM) and Ideal Ratio Mask (IRM) methodology. The first application of DNNs to this problem was presented in [3], which used a DNN-SVM (support vector machine) system to solve the classification problem corresponding to estimating the IBM. [4] compared different training targets including IRM. [5] proposed a regression-based approach to estimate speech log power spectrum. Recently, [6] applies recurrent neural networks to similar mask-based tasks and [7] applies convolutional networks to the spectrum-based tasks.

Kernel-based shallow models (which can be interpreted as two-layer neural networks with a fixed first layer), were also proposed to deal with speech tasks. The time complexity of exact kernel methods is quadratic to the number of training samples at training time [8]. This makes it difficult to scale to large datasets.

Based on random Fourier feature approximations [9], [10, 11, 12] developed kernel methods for speech recognition tasks. However, according to [13], random Fourier feature approximations consistently lead to performance inferior to that of exact kernel methods.

In this work, based on a recently developed highly efficient kernel optimization method EigenPro [14], we propose the first kernel practice on speech enhancement task, with a non-smooth kernel function and an automatic hyper-parameter selection algorithm. This allows kernel machines to handle large datasets without approximation loss.

We conduct experiments on standard datasets using mask-based training target. Our results show that, with EigenPro iteration, kernel methods can consistently outperform the performance of DNN in terms of the target mean square error (MSE) as well as the commonly used speech quality evaluation metrics including perceptual evaluation of speech quality (PESQ) [15] and short-time objective intelligibility (STOI) [16]. Our methods also require significantly less resource time than comparable training for DNNs.

The contributions of our paper are as follows:

1. Using modern kernel algorithms we show performance on mask-based speech enhancement surpassing that of neural networks and requiring less training time.
2. To achieve the best performance, we use exponential power kernel, which, to the best of our knowledge, has not been used for regression or classification tasks.
3. The simplicity of our approach allows us to develop a nearly automatic hyper-parameter selection procedure for target speech frequency subbands.

The rest of the paper is organized as follows. Section 2 introduces our proposed kernel-based speech enhancement system. Experimental results and time complexity comparisons are discussed in Section 3. Section 4 gives the conclusion.

2. Kernel-based speech enhancement

The standard kernel methods for classification/regression denote a function f that minimizes the discrepancy between $f(\boldsymbol{x}_j)$ and y_j, given labeled samples $(\boldsymbol{x}_j, y_j)_{j=1,\ldots,n}$ where $\boldsymbol{x}_j \in \mathbb{R}^d$ is a feature vector and $y_j \in \mathbb{R}$ is its label.

Specifically, the space of f is a Reproducing Kernel Hilbert Space $\mathcal{H}$ associated to a positive-definite kernel function $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. We typically seek a function $f^* \in \mathcal{H}$ for the following optimization problem:

$$f^\star = \text{argmin}_{f(j)=y_j, j=1,2,\ldots,n} \|f\|_{\mathcal{H}} \tag{1}$$

According to the Representer Theorem [17], f^* has the form

$$f(\boldsymbol{x}) = \sum_{j=1}^{n} \alpha_j k(\boldsymbol{x}, \boldsymbol{x}_j) \tag{2}$$

http://dx.doi.org/10.21437/Interspeech.2019-1344

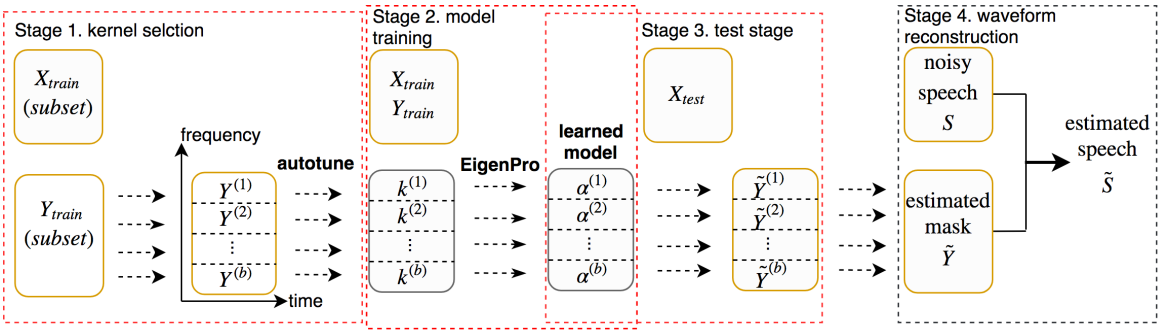

Figure 1: *Kernel-based speech enhancement framework*

To compute f^* is equivalent to solve the linear system,

$$K\boldsymbol{\alpha} = (y_1, \cdots, y_n)^T \tag{3}$$

where the kernel matrix K has entry $[K]_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $\boldsymbol{\alpha} \triangleq (\alpha_1, \cdots, \alpha_n)^T$ is the representation of f under basis $\{k(\cdot, \boldsymbol{x}_1), \cdots, k(\cdot, \boldsymbol{x}_n)\}$.

Next, we show how to apply this kernel method to the speech enhancement task. Section 2.1 gives the framework of the proposed system. The specifics of the kernel type we use in our experiments is in Section 2.2. Section 2.3 gives the details of our automatic hyper-parameter selection algorithm.

2.1. The framework of the proposed system

Our proposed kernel-based speech enhancement system is as depicted in Fig. 1. We split the training targets of all frequency subbands into b subband blocks along the frequency axis, and train one kernel machine for per frequency subband block. Here b is the number of subband blocks, and in our experiments b is set to be 4. We will refer to frequency subband block as *block* in the remainder of the paper.

As in Fig. 1, in stage 1, we use a subset of training data X_{train} (feature), Y_{train} (target/label) to do kernel selection, that is, to get the optimal kernels $k^{(1)}, k^{(2)}, ..., k^{(b)}$ for b blocks $Y^{(1)}, Y^{(2)}, ..., Y^{(b)}$.

With optimal kernels, $k^{(1)}, k^{(2)}, ..., k^{(b)}$, stage 2 is to train linear systems $(\alpha^{(1)}, \alpha^{(2)}, ..., \alpha^{(b)})$ using all training data X_{train}, Y_{train}, and EigenPro iteration proposed in [14] for optimization. In stage 3, feed test data X_{test} into the learned models $\alpha^{(1)}, \alpha^{(2)}, ..., \alpha^{(b)}$, we get estimated mask $\tilde{Y}^{(1)}, \tilde{Y}^{(2)}, ..., \tilde{Y}^{(b)}$ of b blocks, and by combining them together, we get the estimated mask $\tilde{Y}$. Stage 4 is waveform reconstruction, we apply $\tilde{Y}$ to the noisy speech, which produces the estimated clean speech $\tilde{S}$.

For i-th block, the framework learns one model $f^{(i)}$ related to a kernel $k^{(i)}$ with parameters automatically tuned for this block,

$$f^{(i)}(\boldsymbol{x}) = \sum_{j=1}^{n} \alpha_j^{(i)} k^{(i)}(\boldsymbol{x}, \boldsymbol{x}_j) \tag{4}$$

Finally, our kernel machine is to learn an approximate solution $\alpha^{(i)}$ (or $f^{(i)}$) for the optimization problem in Eq. (1), and it is then formed by $\{f^{(1)}, \cdots, f^{(b)}\}$.

2.2. Exponential power kernel

We use an exponential power kernel of the form

$$k_{\gamma,\sigma}(\boldsymbol{x}, \boldsymbol{z}) = \exp(-\frac{\|\boldsymbol{x} - \boldsymbol{z}\|^\gamma}{\sigma}) \tag{5}$$

for our kernel machine, where σ is the kernel bandwidth and γ is often called shape parameter. [18] shows that the exponential power kernel is positive definite, hence a valid reproducing kernel. This kernel also covers a large family of reproducing kernels including Gaussian kernel ($\gamma = 2$) and Laplacian kernel ($\gamma = 1$). We observe that in many noise settings of speech enhancement, the best performance is achieved using this kernel with shape parameter $\gamma \leq 1$, which is highly non-smooth.

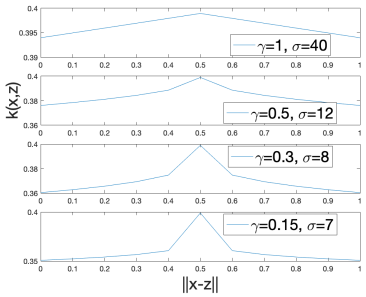

Figure 2: *Exponential power kernel with different parameters, smaller γ, σ lead to less smooth kernel*

We have not seen any application of this kernel (with $\gamma < 1$) in supervised learning literature. In Fig. 2, we plot this kernel function with 4 groups parameter settings (for the 4 blocks) which gives the best kernel performance in our experiments.

2.3. Automatic subbands adaptive kernels

We develop an automatic or nearly automatic hyper-parameter selection algorithm to get the optimal kernel for each block, as shown in Algorithm 1. Note that kernel machines only have two hyper-parameters, namely shape parameter γ and kernel bandwidth σ. With our automatic hyper-parameter selection algorithm, what needs to be done is to give a set of γ and a range of σ. The algorithm will output the optimal γ (γ_{opt}) and optimal σ (σ_{opt}). Also, using a subset of training data X_{train}, Y_{train} (we use one fourth of the training data in our experiments) is enough to get the optimal/near-optimal hyper-parameters and thus consumes less resource time.

3. Experimental results

We use kernel machines with 4 blocks (frequency subband block) for speech enhancement. For fair comparison, we train both kernel machines and DNNs from scratch using the same

Algorithm 1 Automatic hyper-parameter selection[1]

Input: $\boldsymbol{D}_{train}, \boldsymbol{D}_{val}$: training and validation data, Γ: a set of γ for the exponential power kernel, σ_l, σ_h: smallest and largest bandwidth
Output: $\gamma_{opt}, \sigma_{opt}$ for D_{train}
procedure autotune($\boldsymbol{D}_{train}, \boldsymbol{D}_{val}, \Gamma, \sigma_l, \sigma_h$)
  define **subprocedure** cross-validate(γ, σ) as: train one kernel model with $k_{\gamma,\sigma}$ on $\boldsymbol{D}_{train}$ using EigenPro iteration, return its loss on D_{val}.
  for γ in Γ **do**
    $\sigma_\gamma =$ search(cross-validate($\gamma, \cdot$), σ_l, σ_h)
  $\gamma_{opt}, \sigma_{opt} \leftarrow \text{argmin}_{\gamma\in\Gamma,\sigma_\gamma}$ cross-validate(γ, σ_γ)
  return $\gamma_{opt}, \sigma_{opt}$
procedure search(f, σ_l, σ_h)
  if ($\sigma_h - \sigma_l \leq 2$) **then**
    return σ_l
  select $\sigma_{m1}, \sigma_{m2} \in (\sigma_l, \sigma_h)$
  compute $f(\sigma_l), f(\sigma_{m1}), f(\sigma_{m2}), f(\sigma_h)$
  switch min$\{f(\sigma_l), f(\sigma_{m1}), f(\sigma_{m2}), f(\sigma_h)\}$ **do**
    case $f(\sigma_l)$: **return** search(f, σ_l, σ_{m1})
    case $f(\sigma_{m1})$: **return** search(f, σ_l, σ_{m2})
    case $f(\sigma_{m2})$: **return** search(f, σ_{m1}, σ_h)
    case $f(\sigma_h)$: **return** search(f, σ_{m2}, σ_h)

features and targets. We halt the training for any model when error on validation set stops decreasing. Experiments are run on a server with 128GB main memory, two Intel Xeon(R) E5-2620 CPUs, and one GTX Titan Xp (Pascal) GPU.

3.1. Classification task

We train kernel machines and DNNs for a speech enhancement task in [19], which is based on HINT dataset and adopts binary masks (IBM) as targets. The training set contains 100 sentences and subjects are tested using 160 different sentences. The SSN and multi-talker babble are two noise backgrounds employed in this classification setting. Noises are mixed to the sentences at different Signal-Noise-Ratio (SNR), with SNR at -2, -5, -8dB for SSN and babble at 0, -2, -5dB. A sample of SSN or babble having a randomly determined start point within the looped noise is mixed to every sentence from training or testing set. We employ the same fixed set of complementary features as in [19] for both DNNs and kernel machines.

As our kernel machine is designed for regression task, we use a threshold 0.5 to map its real-valued prediction to binary target $\{0, 1\}$.

Table 1: *Kernel & DNN on HINT*

Metrics	Model	Babble			SSN		
		0dB	-2dB	-5dB	-2dB	-5dB	-8dB
Accuracy	DNN	0.90	0.91	0.90	0.91	**0.91**	**0.92**
	Kernel	**0.92**	**0.92**	**0.91**	**0.92**	0.90	0.89
STOI	DNN	0.83	0.80	0.76	0.79	**0.76**	**0.74**
	Kernel	**0.86**	**0.83**	**0.78**	**0.81**	0.75	0.71

In Table 1, we compare the classification accuracy and STOI of kernel machines and DNNs under different noise settings. We see that our kernel machines outperform DNNs on 4 out of 6 noise settings (the other two are -5dB and -8dB SSN noise settings). In terms of accuracy, DNNs perform similar for different SNRs, while kernel machines give clear decreasing value from large SNR to small SNR noise settings. Both DNNs and kernel machines have smaller STOI for smaller SNR noise conditions.

In all, the proposed kernel machines at least match the performance of DNNs on this classification task.

3.2. Regression task

We compare kernel machines and DNNs on a speech enhancement task described in [4], which is based on TIMIT corpus [20] and uses real-valued masks (IRM). As in [4], our training sentences contain 2000 randomly chosen sentences from the training set of TIMIT, and our test set contains 192 sentences from TIMIT core test set. SSN, babble, a factory noise (factory1), a destroyer engine room noise (engine), an operation room noise (oproom) are five noises used in this task. First halves of the noises are used to mix with training utterances and second halves are used for test utterances at -5, 0, 5dB.

We follow the description in [4] for data preprocessing and DNN construction/training. Table 3 reports the MSE, STOI, and PESQ on test set for kernel machines and DNNs. We also present the STOI and PESQ of the noisy speech without enhancement.

For all noise settings, as shown in Table 3, we see that kernel machines consistently produce better MSE, which is the training objective for both models, with at most 35.7% relative improvement (Engine -5dB and oproom -5dB: kernel: $1.17 \cdot 10^{-2}$, DNNs: $1.82 \cdot 10^{-2}$). We also see that STOI and PESQ of kernel machines are consistently better than or comparable to that from DNNs with only one exception (Factory1 0dB). Hence, we see that kernels match/outperform DNNs regards to MSE, STOI and PESQ.

Table 2: *Comparison of kernel machines with 1 block and 4 blocks*

Noise setting	Metrics	Kernel (1 block)	Kernel (4 blocks)	DNN
SSN 0 dB	MSE ($\cdot 10^{-2}$)	1.60	**1.48**	1.67
	STOI	0.81	**0.82**	**0.82**
	PESQ	2.35	**2.36**	2.32
SSN -5 dB	MSE ($\cdot 10^{-2}$)	1.67	**1.60**	1.76
	STOI	0.73	**0.74**	**0.74**
	PESQ	2.01	**2.03**	2.00
Factory1 -5 dB	MSE ($\cdot 10^{-2}$)	2.76	**2.71**	2.77
	STOI	0.67	**0.68**	**0.68**
	PESQ	1.78	**1.79**	1.77

3.3. Single kernel and subbands adaptive kernels

The results of kernels in Table 1 and Table 3 are from our default 4 blocks adaptive kernel machines. As the signal distribution is quite different along different frequencies, combining different kernels adapt to different frequencies can utilize more local information. Note that this adaptive kernel approach can be adopted to other domains which also have local prior knowledge. We show the improvement given by using adaptive kernels in Table 2. In Table 2, kernel (1 block) means training only one kernel for all frequency subbands, and kernel (4 blocks) means the default 4 blocks kernels. Compared with kernel (1

[1]We apply memoization technique for computing cross-validate($\cdot, \cdot$). We first attempt to set σ_{m1}, σ_{m2} as a value that is already used in (σ_l, σ_h), then we choose them to split (σ_l, σ_h) into three parts as equal as possible.

Table 3: *Kernel & DNN on TIMIT: (MSE: lowest is best, STOI and PESQ: highest is best. Best results bolded.)*

Noise type	Metrics	5 dB			0 dB			-5 dB		
		Kernel	DNN	Noisy	Kernel	DNN	Noisy	Kernel	DNN	Noisy
Engine	MSE ($\cdot 10^{-2}$)	**1.10**	1.41	-	**1.34**	1.86	-	**1.17**	1.82	-
	STOI	**0.91**	0.90	0.80	**0.86**	0.85	0.68	**0.80**	0.77	0.57
	PESQ	**2.77**	**2.77**	1.97	**2.51**	2.45	1.66	**2.19**	2.16	1.41
Babble	MSE ($\cdot 10^{-2}$)	**3.34**	3.49	-	**4.18**	4.37	-	**4.94**	5.43	-
	STOI	**0.86**	**0.86**	0.77	**0.77**	**0.77**	0.66	**0.64**	**0.64**	0.55
	PESQ	**2.54**	2.52	2.08	**2.12**	2.10	1.73	**1.70**	1.61	1.42
SSN	MSE ($\cdot 10^{-2}$)	**1.35**	1.53	-	**1.48**	1.67	-	**1.60**	1.76	-
	STOI	**0.88**	**0.88**	0.81	**0.82**	**0.82**	0.69	**0.74**	**0.74**	0.57
	PESQ	**2.68**	2.66	2.05	**2.36**	2.32	1.75	**2.03**	2.00	1.48
Oproom	MSE ($\cdot 10^{-2}$)	**1.44**	1.85	-	**1.34**	1.86	-	**1.17**	1.82	-
	STOI	**0.88**	**0.88**	0.79	**0.84**	0.83	0.70	**0.79**	0.76	0.59
	PESQ	**2.80**	2.79	2.16	**2.50**	2.47	1.78	**2.23**	2.12	1.40
Factory1	MSE ($\cdot 10^{-2}$)	**2.51**	2.53	-	**2.52**	2.55	-	**2.71**	2.77	-
	STOI	**0.86**	**0.86**	0.77	0.78	**0.79**	0.65	**0.68**	**0.68**	0.54
	PESQ	**2.56**	2.51	1.99	2.20	**2.23**	1.62	**1.79**	1.77	1.29

block), kernel (4 blocks) enables MSE to decrease further, and it also brings 0.01 absolute STOI improvement.

We reveals a correlation between optimization metric (MSE) and task desired metric (STOI/PESQ) associated with frequency subbands. From the results in Table 2, and also our experiments using kernel (1 block), we find that smaller regression loss does not always result in better STOI, and [21, 22] have similar observations.

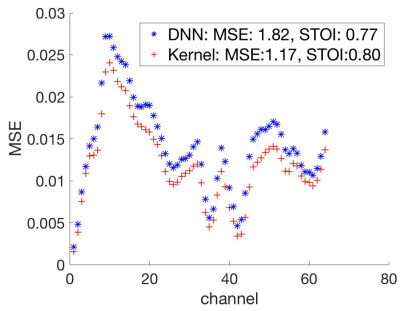

Figure 3: *Engine -5 dB: MSE along per frequency subband*

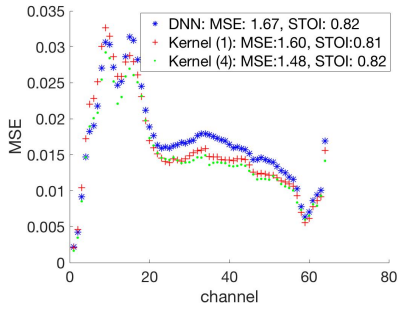

Figure 4: *SSN 0 dB: MSE along per frequency subband*

To fully understand the gap between regression loss and enhancement metrics, we first compared the MSE of every frequency subband of DNNs and kernels. As shown in Fig. 3, for noise settings that kernels have much smaller overall MSE and also smaller MSE on each frequency subband, kernels also achieve better STOI. For settings like SSN 0dB, as shown in Fig. 4, even though single kernel (1 block) has smaller overall MSE, its STOI is not as good as DNNs. Multiple kernels (4 blocks) decrease MSE further and also achieve better STOI. These results show that having smaller MSE along all frequency subbands leads to better STOI.

3.4. Time complexity

In Table 4, we compare the training time of DNNs and kernel machine on both HINT and TIMIT. Note that the training of kernel machines in all experiments typically completes in no more than 10 epochs, significantly less than the number of epochs needed by DNNs. Furthermore, the training time of kernel machines is also less than that of DNNs. Notably, training kernel machine with 1 block takes much less time than DNNs.

Table 4: *Running time/epochs of Kernel & DNN*

Dataset	Time (minutes)			Epochs	
	Kernel		DNN	Kernel	DNN
	1 block	4 blocks			
HINT	**0.8**	3.2	6.6	**10**	50
TIMIT	**18**	65	124	**5**	93

4. Conclusion

In this paper, we have shown that kernel machines using exponential power kernels show strong performance on speech enhancement problems. Notably, our method needs no parameter tuning for optimization and employs nearly automatic tuning for kernel hyper-parameter selection. Moreover, we show that the training time and computational requirement of our method are comparable or less than those needed to train neural networks. We expect that this highly efficient kernel method will be useful for other problems in speech and signal processing.

5. Acknowledgements

We would like to thank Deliang Wang for valuable comments, and Yuxuan Wang, Jitong Chen and Ashutosh Pandey for helpful discussions. We thank NSF for financial support, and we thank NVIDIA for their donations.

6. References

[1] D. Wang and J. Chen, "Supervised speech separation based on deep learning: An overview," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 10, pp. 1702–1726., 2018.

[2] Z. Zhang, J. Geiger, J. Pohjalainen, A. E.-D. Mousa, W. Jin, and B. Schuller, "Deep learning for environmentally robust speech recognition: An overview of recent developments," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 9, no. 5, p. 49, 2018.

[3] Y. Wang and D. Wang, "Towards scaling up classification-based speech separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 7, pp. 1381–1390, 2013.

[4] Y. Wang, A. Narayanan, and D. Wang, "On training targets for supervised speech separation," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 22, no. 12, pp. 1849–1858, 2014.

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

[6] Z.-Q. Wang and D. Wang, "Recurrent deep stacking networks for supervised speech separation," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 71–75.

[7] A. Pandey and D. Wang, "A new framework for supervised speech enhancement in the time domain," *Proc. Interspeech 2018*, pp. 1136–1140, 2018.

[8] D. Achlioptas, F. McSherry, and B. Schölkopf, "Sampling techniques for kernel methods," in *Advances in neural information processing systems*, 2002, pp. 335–342.

[9] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Advances in neural information processing systems*, 2008, pp. 1177–1184.

[10] P.-S. Huang, H. Avron, T. N. Sainath, V. Sindhwani, and B. Ramabhadran, "Kernel methods match deep neural networks on timit." in *Acoustics, Speech and Signal Processing (ICASSP), 2014*, 2014, pp. 205–209.

[11] J. Chen, L. Wu, K. Audhkhasi, B. Kingsbury, and B. Ramabhadrari, "Efficient one-vs-one kernel ridge regression for speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 2454–2458.

[12] Z. Lu, D. Quo, A. B. Garakani, K. Liu, A. May, A. Bellet, L. Fan, M. Collins, B. Kingsbury, M. Picheny, and F. Sha, "A comparison between deep neural nets and kernel acoustic models for speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5070–5074.

[13] M. Belkin, D. Hsu, S. Ma, and S. Mandal, "Reconciling modern machine learning and the bias-variance trade-off," *arXiv preprint arXiv:1812.11118*, 2018.

[14] S. Ma and M. Belkin, "Learning kernels that adapt to gpu," *arXiv preprint arXiv:1806.06144*, 2018.

[15] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221)*, vol. 2. IEEE, 2001, pp. 749–752.

[16] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "An algorithm for intelligibility prediction of time–frequency weighted noisy speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2125–2136, 2011.

[17] B. Schölkopf, A. J. Smola, F. Bach *et al.*, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[18] B. Giraud and R. Peschanski, "On positive functions with positive fourier transforms," *Acta Physica Polonica B*, vol. 37, p. 331, 2006.

[19] E. W. Healy, S. E. Yoho, Y. Wang, and D. Wang, "An algorithm to improve speech recognition in noise for hearing-impaired listeners," *The Journal of the Acoustical Society of America*, vol. 134, no. 4, pp. 3029–3038, 2013.

[20] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "Darpa timit acoustic-phonetic continous speech corpus cd-rom," *NIST speech disc*, vol. 1-1.1, 1993.

[21] S. Venkataramani, R. Higa, and P. Smaragdis, "Performance based cost functions for end-to-end speech separation," *arXiv preprint arXiv:1806.00511*, 2018.

[22] H. Zhang, X. Zhang, and G. Gao, "Training supervised speech separation system to improve stoi and pesq directly," in *Acoustics, Speech and Signal Processing (ICASSP), 2018 IEEE International Conference on. IEEE,*, 2018, pp. 5374–5378.