Monaural Speech Separation on Many Integrated Core Architecture

Wang He[1(✉)], Xu Weixia[1], Guan Naiyang[2], and Yang Canqun[2]

[1] Institute of Computers, College of Computer,
National University of Defence Technology, Changsha 410073, Hunan, China
kuubun@163.com
[2] Institute of Software, College of Computer,
National University of Defence Technology, Changsha 410073, Hunan, China

Abstract. Monaural speech separation is a challenging problem in practical audio analysis applications. Non-negative matrix factorization (NMF) is one of the most effective methods to solve this problem because it can learn meaningful features from a speech dataset in a supervised manner. Recently, a semi-supervised method, i.e., transductive NMF (TNMF), has shown great power to separate speeches from different individuals by incorporating both training and testing data in learning the dictionary. However, both NMF-based and TNMF-based monaural speech separation approaches have high computational complexity, and prohibit them from real-time processing. In this paper, we implement TNMF-based monaural speech separation on many integrated core (MIC) architecture to meet the requirement of real-time speech separation. This approach conducts parallelism based on the OpenMP technology, and performs the computing intensitive matrix manipulations on a MIC coprocessor. The experimental results confirm the efficiency of our implementation of monaural speech separation on MIC architecture.

Keywords: Monaural speech separation · Intel many integrated core architecture · Non-negative matrix factorization

1 Introduction

Speech separation plays an important role in many practical applications, e.g., noise reduction and speech recognition [1,2], singing voice separation [3,4], etc. Monaural speech separation aims to recover the source speeches from a single channel signal, which makes this problem even more challengeable. Traditional methods include non-negative matrix factorization (NMF) [5–7] and deep neural network (DNN) [8,9].

NMF is a linear model with non-negativity constraint incorporated. Before applying NMF to speech separation problem, the signals should be transformed to frequency-domain. The operation is based on the matrix constructed with modulus of the frequency-domain speech signals. In the training stage, NMF based speech separation first learns phonemic features on the training speech

W. Xu et al. (Eds.): NCCET 2016, CCIS 666, pp. 149–156, 2016.
DOI: 10.1007/978-981-10-3159-5_14

signals, and these phonemic features take the form of spectral bases. Then in the testing stage, after learning the test mixture speech signals, the speeches of each speaker can be recovered according to the spectral bases. Another widely used technique to speech separation is DNN, which is a nonlinear model. DNN constructs deep neural networks to learn features of speech signals and has shown its efficiency in many practical tasks [10–12].

In particular, both NMF-based methods and DNN-based methods improve the separation accuracy significantly, but the computational complexities increases quickly with the increase of data. So it is difficult to apply these methods to real-time speech separation, as real-time speech separation requires the feedback time as short as possible. One useful method to solve this problem is to make use of accelerators to implement these algorithms. Nowadays, various accelerators developed quickly, including Intel MIC coprocessor [13,14] and general purpose computation on graphics processing units (GPGPU) [15,16], etc. With the help of accelerators, high parallel computations can be accelerated easily. Previous work on accelerating NMF include using graphics processing units (GPUs) to accelerate NMF [17,18], and parallel version of NMF on multicore architecture [19]. In order to meet the requirement of real-time speech separation applications, this paper proposes to implement the most effective monaural speech separation method called transductive NMF (TNMF) on Intel Many Integrated Core (MIC) architecture to reduce the execution time. As traditional NMF methods cannot utilize the training data when learning features, Guan et al. [20] proposed a semi-supervised variation of NMF, i.e., TNMF, to perform monaural speech separation. The MIC-based algorithm greatly accelerates the TNMF-based monaural speech separation with the help of Intel MIC coprocessor. Experiments on TIMIT dataset confirm its efficiency.

The remainder of the paper is organized as follows. Section 2 introduces related work, including speech separation with NMF and the Intel Many Integrated Core architecture. Section 3 introduces how to use MIC to accelerate NMF based monaural speech separation and Sect. 4 shows the experimental results. In Sect. 5, we conclude the paper.

2 Background

2.1 Monaural Speech Separation with NMF

Assuming $V \in R_+^{m \times n}$ is a non-negative matrix, NMF aims to find non-negative approximation of V, i.e.,

$$V \approx W \times H. \tag{1}$$

where $W \in R_+^{m \times r}$ and $H \in R_+^{r \times n}$, and r is usually smaller than m and n.

In particular, W and H can be optimized by solving the following problem:

$$\min_{W \geq 0, H \geq 0} ||V - WH||_F^2, \tag{2}$$

where $||\cdot||_F$ denotes the Frobenius norm. The objective function of this optimization problem measures the divergence between V and WH. As NMF incorporates

non-negativity constraints over factor matrices, when applied to speech separation tasks, the original speech signals should be transformed from time-domain to frequency-domain, as time-domain signals has negative entries. The transformation can be implemented by using short-time Fourier transform (STFT).

Assuming we have p speakers and their corresponding speeches, let $V_k \in R_+^{m \times n_k}$ denote the matrix constructed with modulus of frequency-domain signal of k-th speaker. In order to conduct speech separation, we use NMF to factorize each V_k independently in the training stage, i.e.,

$$V_k \approx W_k H_k, \tag{3}$$

where $W_k \in R_+^{m \times r}$ denotes the learned phonemic features of k-th speaker, namely spectral bases, and $H_k \in R_+^{r \times n}$ denotes the activations corresponding to W_k. Let $V^m \in R_+^{m \times n}$ denotes the matrix constructed with modulus of frequency-domain mixture signals.. In testing stage, we should decompose V^m with NMF as follows:

$$V^m \approx W^m H^m, \tag{4}$$

where $W^m = [W_1, \cdots, W_p]$ is constructed by the spectral bases of corresponding speakers, $H^m \in R_+^{rp \times n}$ denotes the obtained activations. To separate the mixture speech signal, we should decompose H^m into $H^m = [H_1^{mT}, \cdots, H_p^{mT}]^T$ according to W^m. So the separated speech is

$$V_k^m \approx W_k H_k^m. \tag{5}$$

However, the NMF algorithms based on objective function (2) cannot utilize mixture signals in the training stage. To solve this problem and to improve the accuracy, a semi-supervised algorithm called transductive non-negative matrix factorization (TNMF) was presented [20]. The objective function of TNMF is

$$\min_{\forall 1 \le k \le p, W_k \ge 0, H_k \ge 0, H^m \ge 0} \{\sum_{k=1}^{p} ||V_k - W_k H_k||_F^2 + \lambda ||V^m - W^m H^m||_F^2\}. \tag{6}$$

where λ is the trade-off parameter to balance the influence of two parts of the objective function.

The TNMF model can be solved by using the multiplicative update rule (MUR) [20,21] as follows:

$$W_k \leftarrow W_k \cdot \frac{V_k H_k^T + \lambda V^m H_k^{mT}}{W_k H_k H_k^T + \lambda W^m H^m H_k^{mT}}, \tag{7}$$

$$H_k \leftarrow H_k \cdot \frac{W_k^T V_k}{W_k^T W_k H_k}, \tag{8}$$

$$H^m \leftarrow H^m \cdot \frac{W^{mT} V^m}{W^{mT} W^m H^m}. \tag{9}$$

Based on the obtained solution, i.e., W_k, H_k and H^m, we can easily separate the mixture speech according to [20].

2.2 Intel Many Integrated Core Architecture

The Intel many integrated core (MIC) architecture aims to accelerate highly parallel and computationally intensive programs. The Intel Xeon Phi coprocessor based on the Intel MIC architecture consists of 61 cores. Each core has two level caches, includes a 32 KB L1 data cache and L1 instruction cache, and a 512 KB private L2 cache. And the Intel Xeon phi coprocessor also has a 512 bits vector processor unit (VPU) with Single Instruction Multiple Data (SIMD) architecture [22]. An important advantage of Intel MIC architecture is its compatibility with original programs, which makes it easy for developers to accelerate their programs. The Intel coprocessor is supported by a variety of numerous libraries, compilers and tuning tools, etc.

To utilize the Intel Xeon Phi coprocessor, we have both the offload mode and the native mode. The machine executes main program on the processor and offloads the selected sections to coprocessor in offload mode. The program is executed in both processor and coprocessor locally in native mode. The offload mode can further be divided into two modes, namely pragma offload mode and shared virtual memory model mode.

3 Parallel TNMF Algorithm for MIC Architecture

This paper pays attention to the TNMF-based monaural speech separation algorithm. To accelerate the TNMF-based algorithm, the main work is to parallel the matrix manipulation in Eqs. (7) to (9). In practical, we use Intel math kernel library (MKL) [24] to perform the matrix manipulation. MKL has rich functions and can compute the operations with high-efficiency.

We use OpenMP technology to perform the parallel operations on multicores. The data needed in the algorithm are constructed as matrix, so we can utilize the computational ability of coprocessor's VPU. **Algorithm 1** summarizes the MIC-based monaural speech separation algorithm.

Algorithm 1. MIC-based monaural speech separation algorithm

Input : Training speech signals S_1 and S_2,
Testing mixture speech signal S^m
Output: Recovered speech signals S_1^m and S_2^m

1 Transform the original speech signals from time-domain to frequency-domain, and get the signals' modulus:V_1,V_2,V^m
2 Offload V_1, V_2, V^m from CPU to coprocessor
3 Update W_k, H_k and H^m until converge with TNMF algorithm's update rules
4 Offload W_k, H_k and H^m from coprocessor to CPU
5 Compute the recovered signals: $V_k^m \approx W_k H_k^m$
6 Recover the signals to time-domain speech signals, obtain S_1^m and S_2^m.

The input of this algorithm is speech signals of training speeches and testing mixture speech. The first step of the algorithm is to transform the signals

from time-domain to frequency-domain by STFT. Then we can get the modulus of transformed signals, as V_1, V_2 and V^m. To accelerate the computation, we offload the matrices to coprocessor, and update them in coprocessor. MKL has two offload modes, including automatic and compiler assisted. We use compiler assisted mode here to offload the data to coprocessor automatically. When the update stage finished, the obtained matrice W_k, H_k and H^m will be offloaded to CPU. The magnitude spectrogram of each recovered speech can be obtained by $V_k^m \approx W_k H_k^m$. Then we can easily get the time-domain signals of the recovered signals.

4 Experiments

To verify the performance of the accelerated algorithm, we compare the execution time between MIC-based monaural speech separation algorithm and original non-accelerated one. The dataset utilized in this experiment is the TIMIT dataset [23]. We randomly choose some speech segments from two speakers. The testing speech signal is generated by summing two segments from different speakers. Another two segments from the corresponding two speakers are chosen as the training data. The training speech segments is about 23 s long and the testing speech segments is about 3 s long. These speech segments are all sampled at a rate of 16 kHz. In experiments, the FFT size in all examples is set to 1024, the trade-off parameter λ is set to 0.1. All the experiment results are averages of 10 replications.

Figure 1 gives the original speech signals, which generate the testing speech segments, and the recovered speech signals by MIC-based monaural speech separation. The first row shows the time-domain signals, while the second row shows the frequency-domain signals. In particular, column (a) and column (c) represent the same speech signal while column (b) and column (c) represent the same speech signal. The original speech signals are listed in column (a) and (b) while the recovered speech signals are listed in column (c) and (d).

To verify the computational efficiency of MIC-based algorithm and the original non-accelerated one, we compare the execution time between the two algorithms in different parameters. In experiments, two parameters have significant influence on execution time, which are spectral bases number and iteration number. When spectral bases number increases, we can learn the features of speech signals more accurately, and when the iteration number increases, the decomposition result will be more accurate. Meanwhile, the computational complexity increases along with the increase of the two parameters. We set the spectral bases numbers and iteration number to different values respectively, and record the execution time of MIC-based algorithm and original algorithm. In MIC-based algorithm, the thread number is set to 8. In experiments, when the spectral bases number was set to different values, the iteration number is set to 1000. When the iteration number was set to different values, the spectral bases number is set to 120. Figure 2 presents the results.

It is obvious that MIC-based algorithm costs much less execution time. And more importantly, when the computational complexity increases, the execution

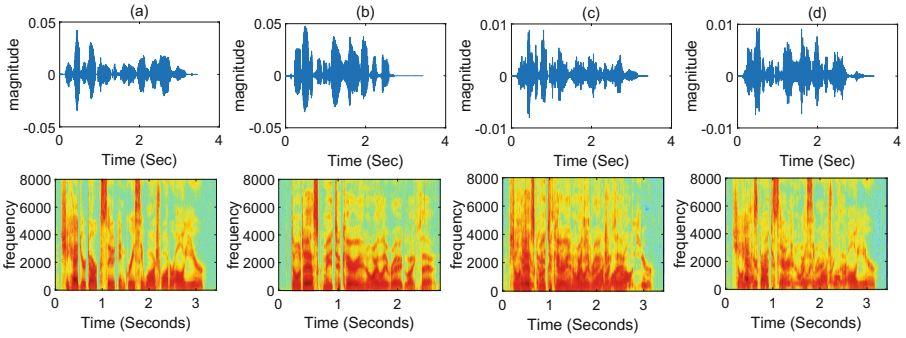

Fig. 1. The top and bottom rows represent time-domain and frequency-domain signals. Column (a) and (b) represent original signals, column (c) and (d) represent recovered signals. Columns (a) and (c) represent the same speech signals while column (b) and (d) represent the same speech signals.

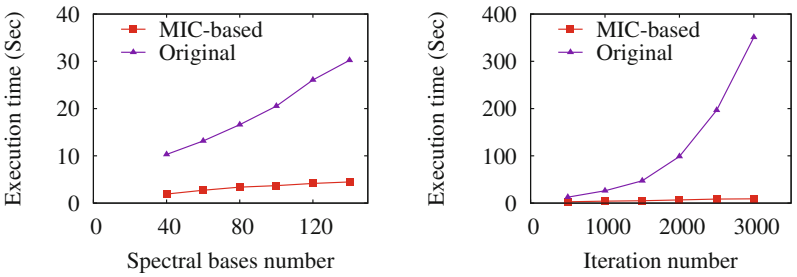

Fig. 2. Execution time with different parameter values.

time of MIC-based algorithm increases more slowly compared with the original non-accelerated algorithm.

Then, we set the thread number from 2 to 16 to test the influence of multi-thread parallelization on MIC-based algorithm. The spectral bases number is set to 120 while the iteration number is set to 1000. Table 1 shows the results.

Table 1. Execution time with different thread number

Thread number	2	4	8	16
MIC-based (Sec)	6.337	3.477	2.857	3.044

With the increase of thread number, the execution time of MIC-based algorithm has a trend of decrease. However, as the thread number is set to 16, it has a longer execution time than the thread number is set to 8. In practical, it is important to choose the correct thread number to get the best performance according to different situations.

In experiments, the MIC architecture shows its great effectiveness in NMF-based monaural speech separation. The intensitive matrix manipulations are offloaded to coprocessor and performed parallel, which makes the MIC-based monaural speech separation has a much shorter execution time than original non-accelerated algorithm. The thread number should also be chosen correctly to get a better performance. In summary, MIC-based monaural speech separation is more suitable for real-time speech separation applications.

5 Conclusion

This paper presented a MIC-based monaural speech separation. This algorithm is a parallel version of TNMF-based monaural speech separation algorithm. In practical, MIC architecture shows its power on accelerating highly parallel workloads. To verify the effectiveness of the MIC-based algorithm, we conduct experiments on TIMIT dataset to separate mixture speech signals. We compare the execution time of MIC-based algorithm and original non-accelerated algorithm under different conditions. The experiment results confirm that MIC-based monaural speech separation has much less execution time than original non-accelerated algorithm, so it is more suitable for real-time speech separation applications.

Acknowledgments. This work was supported by National High Technology Research and Development Program "863" Program) of China (under grant No. 2015AA01A301) and National Natural Science Foundation of China (under grant No. 61502515).

References

1. Vinyals, O., Ravuri, S.V., Povey, D.: Revisiting recurrent neural networks for robust ASR. In: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 4085–4088 (2012)
2. Maas, A., Le, Q.V., Oneil, T.M., Vinyals, O., Nguyen, P., Ng, A.Y.: Recurrent neural networks for noise reduction in robust ASR (2012)
3. Huang, P.S., Chen, S.D., Smaragdis, P., Hasegawa-Johnson, M.: Singing-voice separation from monaural recordings using robust principal component analysis. In: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 57–60. IEEE (2012)
4. Huang, P.S., Kim, M., Hasegawa-Johnson, M., Smaragdis, P.: Singing-voice separation from monaural recordings using deep recurrent neural networks. In: ISMIR, pp. 477–482 (2014)
5. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature **401**(6755), 788–791 (1999)
6. Wang, Z., Sha, F.: Discriminative non-negative matrix factorization for single-channel speech separation. In: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3749–3753. IEEE (2014)
7. Weninger, F., Le Roux, J., Hershey, J.R., Watanabe, S.: Discriminative nmf and its application to single-channel source separation. In: INTERSPEECH, pp. 865–869 (2014)

8. Huang, P.-S., Kim, M., Hasegawa-Johnson, M., Smaragdis, P.: Deep learning for monaural speech separation. In: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1562–1566. IEEE (2014)
9. Weninger, F., Hershey, J.R., Le Roux, J., Schuller, B.: Discriminatively trained recurrent neural networks for single-channel speech separation. In: 2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP), pp. 577–581. IEEE (2014)
10. Erdogan, H., Hershey, J.R., Watanabe, S., Le Roux, J.: Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks. In: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 708–712. IEEE (2015)
11. Weninger, F., Eyben, F., Schuller, B.: Single-channel speech separation with memory-enhanced recurrent neural networks. In: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3709–3713. IEEE (2014)
12. Zhang, X.-L., Wang, D.: A deep ensemble learning method for monaural speech separation. IEEE/ACM Trans. Audio Speech Lang. Process. **24**(5), 967–977 (2016)
13. Duran, A., Klemm, M.: The intel many integrated core architecture. In: 2012 International Conference on High Performance Computing and Simulation (HPCS), pp. 365–366. IEEE (2012)
14. Jeffers, J., Reinders, J.: Intel Xeon Phi coprocessor high-performance programming. Newnes (2013)
15. Tarditi, D., Puri, S., Oglesby, J.: Accelerator: using data parallelism to program gpus for general-purpose uses. In: ACM SIGARCH Computer Architecture News, vol. 34, no. 5, pp. 325–335. ACM (2006)
16. Lee, S., Min, S.-J., Eigenmann, R.: OpenMP to GPGPU: a compiler framework for automatic translation and optimization. ACM Sigplan Not. **44**(4), 101–110 (2009)
17. Platoš, J., Gajdoš, P., Krömer, P., Snášel, V.: Non-negative matrix factorization on GPU. In: Zavoral, F., Yaghob, J., Pichappan, P., El-Qawasmeh, E. (eds.) Networked Digital Technologies. Communications in Computer and Information Science, vol. 87, pp. 21–30. Springer, Heidelberg (2010)
18. Mejía-Roa, E., Tabas-Madrid, D., Setoain, J., García, C., Tirado, F., Pascual-Montano, A.: NMF-mGPU: non-negative matrix factorization on multi-GPU systems. BMC Bioinf. **16**(1), 1 (2015)
19. Alonso, P., García, V., Martínez-Zaldívar, F.J., Salazar, A., Vergara, L., Vidal, A.M.: Parallel approach to NNMF on multicore architecture. J. Supercomput. **70**(2), 564–576 (2014)
20. Guan, N., Lan, L., Tao, D., Luo, Z., Yang, X.: Transductive nonnegative matrix factorization for semi-supervised high-performance speech separation. In: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2534–2538. IEEE (2014)
21. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: Advances in Neural Information Processing Systems, pp. 556–562 (2001)
22. Chrysos, G.: Intel xeon phi coprocessor-the architecture, Intel Whitepaper (2014)
23. Garofolo, J.S., Lamel, L.F., Fisher, W.M., Fiscus, J.G., Pallett, D.S.: DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1. NASA STI/Recon technical report n, vol. 93 (1993)
24. Intel, M.: Intel math kernel library (2007)