Matrix Factorization for Speech Enhancement

Peter Li
Peter.Li@nyu.edu

Yijun Xiao
ryjxiao@nyu.edu

1 Introduction

In this report, we explore techniques for speech enhancement using matrix factorization. We focus on enhancing speech signals corrupted with non-stationary environmental noise. Formally, the problem is as follows:

Given observations,X_{noisy}, of speech corrupted with additive noise,

$$X_{noisy}(t) = X_{speech}(t) + X_{noise}(t) \tag{1}$$

we try to recover X_{speech}. The goal is to recover a signal that has better perceptual quality compared to the original noisy signal. This has uses in many applications including telecommunications, voice recognition, and hearing aids Benesty and Makino [2005].

Preliminaries

Feature Representation

Matrix factorization methods for speech enhancement operate on time/frequency representations of audio signals. As a first step, we compute the short-time Fourier Transform (STFT) of our signal.

$$W = STFT(X_{noisy}) \tag{2}$$

Next, we compute the magnitude and phase of each entry in W:

$$\mathbf{M} = |\mathbf{W}|, \mathbf{P} = \mathbf{angle}(\mathbf{W}) \tag{3}$$

$\mathbf{M}$ is then used in our matrix factorization algorithms to produce an enhanced version, $\mathbf{M_{enhanced}}$. To reconstruct a time domain signal, we use the inverse STFT proposed in Griffin and Lim [1984]. In this reconstruction, we use the phase estimated from the original, noisy signal.

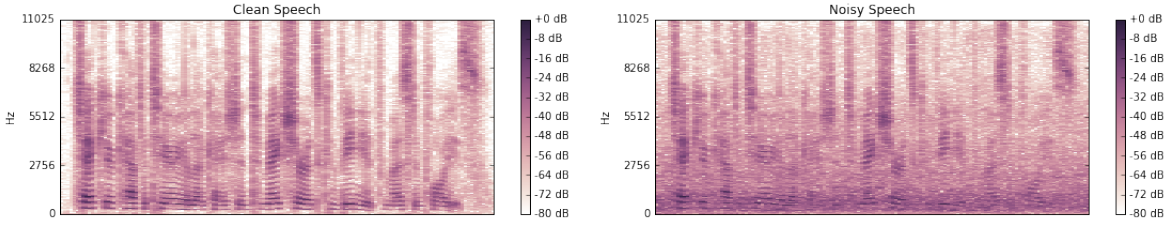

Figure 1: Magnitude Spectrogram of Clean and Noisy Speech. Notice that there is significant overlap in speech and noise frequencies. Because of this, simple filtering techniques are not adequate for speech enhancement

2 Methods

Matrix factorization methods for speech enhancement factor the spectrogram of a noisy signal, $\mathbf{M}$ as a product of a dictionary matrix, $\mathbf{D}$ and a code matrix $\mathbf{C}$. Next, entries of $\mathbf{D}$ and $\mathbf{C}$ are assigned to either speech or noise.

$$\begin{aligned}\mathbf{M} &= \mathbf{DC} \\ &= [\mathbf{D_{speech}} \mathbf{D_{noise}}] \begin{bmatrix} \mathbf{C_{speech}} \\ \mathbf{C_{noise}} \end{bmatrix}\end{aligned} \tag{4}$$

The noise components are then discarded to get the enhanced signal:

$$\mathbf{M_{enhanced}} = \mathbf{D_{speech}} \mathbf{C_{speech}} \tag{5}$$

In the following two sections, we present a survey of methods to first compute the factorization of $\mathbf{M}$ and then assign dictionary atoms and codes to particular sources.

2.1 Unsupervised Speech Enhancement

Consider the case where we only have a single observed mixed audio, denoising has to be done in an unsupervised fashion. We adopted a matrix factorization based method in this project to decompose the mixed signal in the time-frequency domain and try to group the basis/atoms into speech and noise. We then reconstruct the spectrogram using only the basis/atoms from the speech group.

Matrix factorization methods, especially non-negative matrix factorization, have been proved to be effective in audio source separation tasks (Smaragdis et al. [2014], Wilson et al. [2008]). We experiment three different matrix decomposition methods on their abilities to reconstruct the original signals.

Non-negative Matrix Factorization. Non-negative matrix factorization is a method that solves the following optimization problem:

$$\begin{aligned} \underset{\mathbf{D},\mathbf{C}}{\text{minimize}} \quad & \|\mathbf{M} - \mathbf{DC}\|_2^2 \\ \text{subject to} \quad & \mathbf{D}, \mathbf{C} \geq 0 \end{aligned}$$

$\mathbf{D}, \mathbf{C} \geq 0$ represents the elements in $\mathbf{D}, \mathbf{C}$ are non-negative. Note that we use the squared Euclidean distance here to measure the difference between the reconstructed matrix $\mathbf{DC}$ from the original matrix $\mathbf{M}$. Other popular options for the distance measure include the Kullback-Leibler (KL) divergence (Lee and Seung) and Itakura-Saito (IS) divergence (Fvotte et al. [2009]). One can add Frobenius norm and element-wise L1 norm to regularize and induce sparsity in both dictionary and code matrix.

Sparse PCA. Sparse PCA (Zou et al. [2006]) is an extension to conventional PCA. It tries to find principle components that are linear combinations of a subset of the input variables which provides superior interpretability. Compared to the sparse coding algorithm, the dictionary matrix is sparse whereas in sparse coding the code matrix is sparse. In the setting of decomposition, the problem is formulated as follows:

$$\begin{aligned} \underset{\mathbf{D},\mathbf{C}}{\text{minimize}} \quad & \frac{1}{2}\|\mathbf{M} - \mathbf{DC}\|_2^2 + \lambda \|\mathbf{D}\|_1 \\ \text{subject to} \quad & \|\mathbf{C}_i\|_2 = 1, \forall i \end{aligned}$$

where $\mathbf{C}_i$'s are the rows in the code matrix.

Dictionary Learning / Sparse Coding. Dictionary learning and sparse coding are actually two separate processes to solve an unified problem.

$$\begin{aligned} \underset{\mathbf{D},\mathbf{C}}{\text{minimize}} \quad & \frac{1}{2}\|\mathbf{M}-\mathbf{D}\mathbf{C}\|_2^2+\lambda\|\mathbf{C}\|_1 \\ \text{subject to} \quad & \|\mathbf{D}_i\|_2=1, \forall i \end{aligned}$$

where $\mathbf{D}_i$'s are the atoms (columns) in the dictionary matrix. We often refer to dictionary learning as the process of finding the optimal dictionary matrix $\mathbf{D}$ and sparse coding as the process of encoding given the dictionary matrix $\mathbf{D}$. The element-wise L1 penalty term induces sparsity in the code matrix. In other words, this model is best suited for problems where data can be encoded using only a small subset of atoms in the dictionary.

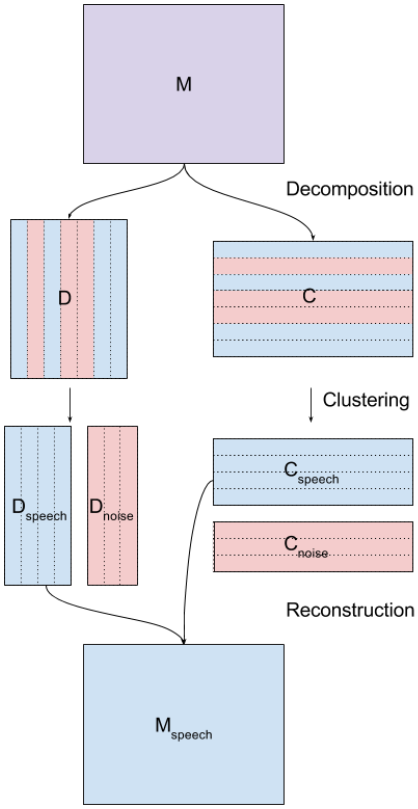

Figure 2: Diagram of the unsupervised approach

Basis/Activation Clustering After the decomposition of the spectrogram of the mixed signal, we need to group the atoms in the dictionary and their corresponding activations in the code matrix into speech and noise. To accomplish this, we decide to use clustering algorithms on the atoms/basis as well as the activations. Apparently, this approach requires a strong assumption that the basis or the activations for speech and noise are significantly different.

The ideal scenario is the atoms of the dictionary perfectly cluster into two groups, one corresponds to speech and the other corresponds to noise. But this method breaks if atoms form clusters within speech or noise groups. The number of clusters therefore is not restricted to two in our case. After k clusters are determined, we loop through every combination of the k clusters and find the combination that produce the reconstruction with the maximum signal-to-distortion ratio (SDR).

The diagram for the unsupervised approach is depicted in Figure 2. After the spectrogram is reconstructed, we perform inverse Short-Time Fourier Transformation (ISTFT) to obtain a denoised speech.

2.2 "Semi-supervised" Speech Enhancement

In this second approach, we present a "semi-supervised" method for speech enhancement. This closely follows the approach described in Schmidt et al. [2007].

One weakness of purely unsupervised methods is that there is no good way to determine which dictionary atoms belong to each source. Traditional, supervised methods in speech enhancement sidestep this problem by learning $\mathbf{D_{noise}}$ and $\mathbf{D_{speech}}$ on separated training speech and noise data. The problem is then simplified to learning the code matrix, $\mathbf{C}$. Unfortunately, this method typically relies on learning speaker dependent $\mathbf{D_{speech}}$.

In the "Semi-supervised" approach of Schmidt et al. [2007], the authors take an intermediate approach. The authors present a speaker independent speech enhancement model where $\mathbf{D_{noise}}$ is learned from training data, but $\mathbf{D_{speech}}$ is not. The problem them becomes estimating $\mathbf{D_{speech}}, \mathbf{C_{speech}}$, and $\mathbf{C_{noise}}$. Since $\mathbf{D_{noise}}$ is fixed, we have no problem assigning assigning atoms and codes to their respective sources. The factorization is computed using Non-negative Sparse Coding (NMF + sparsity promoting penalty on the codes) Hoyer [2004].

Formally, we solve the following optimization problems:

1. Learn $\mathbf{D_{noise}}$ on training data

$$\begin{aligned} \underset{D_n}{\text{minimize}} \quad & \|\mathbf{M_{train}} - \mathbf{D_n C}\|_2^2 + \gamma \sum_{ij} \mathbf{C_{ij}} \\ \text{subject to} \quad & \mathbf{D_n, C} \geq \mathbf{0} \end{aligned}$$

2. Estimate $\mathbf{D_s}, \mathbf{C_s}$, and $\mathbf{C_n}$

$$\begin{aligned} \underset{\mathbf{D_s}, \mathbf{C_s}, \mathbf{C_n}}{\text{minimize}} \quad & \left\| \mathbf{M_{train}} - [\mathbf{D_s D_n}] \begin{bmatrix} \mathbf{C_s} \\ \mathbf{C_n} \end{bmatrix} \right\|_2^2 + \lambda_s \sum_{ij} \mathbf{C_{ij}^s} + \lambda_n \sum_{ij} \mathbf{C_{ij}^n} \\ \text{subject to} \quad & \mathbf{D_s, C_s, C_n} \geq \mathbf{0} \end{aligned}$$

3 Experiments and Results

3.1 Data

For this project, we synthesized the mixed audio by combining speech signals from the TIMIT corpus (Garofolo et al. [1993]) with background noises collected from `freesound.org`. TIMIT contains recordings of 630 speakers reading ten phonetically rich sentences. Signal length range from 2s to 7s. Noise audio is a 30 min long recording of the background noise in a large hall. To generate the noisy audio, we mix clean speech signals with a random segment of noise of the same duration. The mixing weight is chosen such that we get a signal to noise ratio of 5dB.

STFT were computed with the following specifications:

1. FFT window size: 2048
2. Stride: 512 (overlap of 0.75 window)
3. window function: asymmetric Hann window

Model performance was evaluated using signal to distortion ratio (SDR) (Vincent et al. [2006]). This is a measure of signal quality that measures ratio of the power of the clean signal to power of the noise contained in the enhanced signal.

3.2 Experiment 1: Validity of Using Matrix Factorization Methods

To demonstrate that matrix factorization is a viable way to tackle speech enhancement task, we first evaluate its ability to reconstruct clean speech and noise separately. It would be impossible to enhance the mixed audio if clean speech itself cannot be reconstructed properly.

Method	# Components	$\lambda = 0$	$\lambda = 0.1$	$\lambda = 0.3$	$\lambda = 1$
NMF	8	11.22 (0.88)	11.25 (0.89)	11.24 (0.90)	11.24 (0.87)
	16	16.72 (1.57)	16.84 (1.56)	16.69 (1.58)	16.57 (1.48)
	32	24.12 (2.39)	24.18 (2.28)	23.65 (2.05)	21.99 (1.33)
	64	34.63 (3.67)	32.86 (1.91)	29.40 (1.39)	23.34 (1.26)
Sparse PCA	8	9.98 (0.91)	10.73 (0.86)	10.70 (1.06)	9.39 (0.96)
	16	14.46 (1.37)	15.67 (1.13)	14.67 (0.98)	11.94 (1.04)
	32	20.15 (4.43)	20.13 (1.17)	17.98 (0.83)	13.52 (1.18)
	64	32.88 (4.13)	23.01 (0.71)	20.00 (0.81)	14.99 (1.34)
D.L.	8	11.01 (1.00)	11.39 (0.96)	11.64 (0.91)	11.58 (0.88)
	16	16.23 (1.28)	17.07 (1.57)	17.43 (1.54)	16.60 (1.40)
	32	22.92 (2.47)	24.31 (2.43)	24.10 (1.92)	22.63 (1.97)
	64	35.37 (6.40)	34.54 (3.30)	33.88 (3.46)	30.70 (2.37)

Table 1: Reconstruction signal-to-distortion ratios (SDR) for different number of components and sparsity parameter. Both mean and standard deviation are reported. All methods are evaluated on the same 10 samples. λ is the sparsity parameter used in decomposition.

For clean speech, we decompose the unmixed speech signal in the time-frequency domain and reconstruct itself from the learned atoms. Number of components, sparsity parameter are tuned for all three models. Each set of hyper-parameter was experimented on the same 10 mixed audio samples and we report the mean and standard error of the resulting SDRs. The results are reported in Table 1

From Table 1, we can conclude that the more dictionary components we use, the less sparsity we impose, the higher SDR we get. This makes perfect sense in this "in-sample" experiment as imposing sparsity using an L1 penalty term does not decrease the training reconstruction errors. However if the data is sparse-coded in nature, dictionary learning can be more generalizable. We also observe a superior performance of dictionary learning methods over sparse PCA and NMF when the number of components is large. Sparse PCA has a significantly lower SDR when sparsity on the basis/atoms is imposed which indicates that the basis/atoms for the spectrogram are not sparse in nature.

The purpose of this experiment is to validate that matrix decomposition methods are viable approaches to reconstruct signals. Judging from the final SDRs, this assumption is true (even SDR=10 is a relatively clean signal).

3.3 Experiment 2: Unsupervised Speech Enhancement

For the unsupervised approach, we use dictionary learning approach to perform decomposition because of its superior reconstruction ability. Two different clustering methods, K-means and Spectral clustering are used and compared in the basis clustering process. Different number of clusters from 5 to 12 are experimented.

As we do not know which cluster corresponds to speech and which to noise, every possible combinations of the atom clusters are considered to reconstruct the original signal. We choose the best possible reconstruction with respect to the SDR measure. The results are listed in Table 2.

Number of Clusters	SDR Improvement K-means	Spectral Clustering
5	0.18 (0.32)	0.20 (0.38)
6	0.10 (0.17)	0.57 (0.90)
7	0.30 (0.48)	0.29 (0.52)
8	0.11 (0.20)	0.39 (0.62)
9	0.10 (0.22)	0.51 (0.69)
10	0.31 (0.47)	0.59 (0.69)
11	0.19 (0.35)	0.64 (0.93)
12	0.30 (0.45)	0.35 (0.45)

Table 2: Improvement in SDR for different number of clusters in the unsupervised approach. Mean and standard deviation are both reported.

From the results, we can see that pure unsupervised approach fails to give consistent improvements over the baseline. In fact, one to three out of the ten samples have significant improvements on SDR (around 1.0-2.0) whereas most other samples have no improvements. This is why we have a larger standard deviation than mean improvement.

Spectral clustering works somewhat better than K-means. But both have unpredictable performances depending on initialization and the sample at hand. In conclusion, pure unsupervised approach based on basis/activations clustering is hard to make it perform consistently.

3.4 Experiment 3: Noise Reconstruction

In this experiment, we investigate whether we can actually use a pre-trained noise dictionary to reconstruct new noise samples.

First a noise dictionary was trained on 30s of noise audio. Then this dictionary was used to encode 5s segments of new noise.

Number of Dictionary Components	$\gamma = 0.01$	$\gamma = 0.1$	$\gamma = 1$
20	10.24	10.08	8.11
50	13.41	13.04	8.78
100	16.32	15.69	8.97

Table 3: Reconstruction signal-to-distortion ratios (SDR) for different number of components and sparsity parameters, reconstructing noise using a pre-trained noise dictionary.

From these results, we see that we can reconstruct noise to a reasonable degree of accuracy using a pre-trained dictionary. The power of our reconstructed signal is at least 8 dB higher than the power of the error. Perceptually, these reconstructions sound reasonable.

3.5 Experiment 4: "Semi-supervised" Speech Enhancement

In this experiment, we implement the "Semi-supervised" speech enhancement model described in Section 2.2. For this experiment, we vary the following parameters:

- number of noise components $\in \{32, 64, 128\}$
- number of speech components $\in \{32, 64, 128\}$
- $\gamma \in \{0.01, 0.1, 1\}$
- $\lambda_n \in \{0.01, 0.1, 1\}$

- $\lambda_s \in \{1\}$[1]

# Noise Components	# Speech Components	γ	λ_n	SDR Improvement
128	32	0.01	1	1.02 (0.15)
64	32	1	1	1.01 (0.09)
128	32	1	1	1.01 (0.07)
64	32	0.01	1	1.01 (0.09)
32	32	1	1	1.00 (0.08)
32	128	0.01	0.01	0.78 (0.07)
32	128	0.1	0.01	0.77 (0.12)
64	128	1	1	0.77 (0.13)
32	128	1	0.01	0.76 (0.06)
128	128	0.1	0.1	0.74 (0.11)

Table 4: Improvement in SDR (SDR(Enhanced Audio) - SDR(Noisy Audio)). Due to the high number of parameter configurations, we report the top 5 and bottom 5 parameter configurations.

From Table 4, we see that the "Semi-supervised" methods does produce an enhanced signal. On average, the top models have a 1dB improvement in SDR relative to the original noisy signal. Perceptually, the background noise is noticeably softer, but far from completely removed. Although using different data, top published models seem to achieve improvements between 5dB - 10dB.

Qualitatively, we see that the number of speech components has a big impact on model performance. Using too many features significantly deteriorates SDR improvement. Intuitively, this makes sense since adding more components might start to include atoms that also model noise. Additionally, we see that a high level of regularization on the noise (λ_n) helps performance.

4 Conclusion

In this report we explore unsupervised and semi-supervised methods for speech enhancement. Although, performance is not high compared to supervised methods, these models do not rely on clean speech data (which might be difficult to obtain). Through our experiments, we found the following:

- Matrix factorization methods have enough power to adequately reconstruct speech and noise signals. Using as few as 20 dictionary components, we can reconstruct signals to reasonable fidelity
- Purely unsupervised speech enhancement using clustering of dictionary atoms is difficult.
- "Semi-supervised" speech enhancement provides a reasonable compromise between supervised and unsupervised methods. Although enhancement is not as good as supervised methods, we can enhance noisy signals when we only have previous knowledge about noise sources.

Implementation

Spectral features and inverse STFT were computed using Librosa McFee et al. [2015]. NMF, Sparse PCA, and Dictionary Learning were computed using scikit learn Pedregosa et al. [2011]. Non-negative Sparse Coding was implement using the multiplicative update rule derived in Eggert and Körner [2004].

References

Jacob Benesty and Shoji Makino. *Speech enhancement*. Springer Science & Business Media, 2005.

[1]Initial runs over multiple values of λ_s found that $\lambda_s = 1$ produced the best results. To save time, we excluded $\lambda_s = 0.1$ and $\lambda_s = 0.01$.

Julian Eggert and Edgar Körner. Sparse coding and nmf. In *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, volume 4, pages 2529–2533. IEEE, 2004.

C. Fvotte, N. Bertin, and J.-L. Durrieu. Nonnegative matrix factorization with the itakura-saito divergence. with application to music analysis. *Neural Computation*, 21(3):793–830, Mar. 2009.

John S Garofolo, Lori F Lamel, William M Fisher, Jonathan G Fiscus, and David S Pallett. Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1. *NASA STI/Recon Technical Report N*, 93, 1993.

Daniel W Griffin and Jae S Lim. Signal estimation from modified short-time fourier transform. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 32(2):236–243, 1984.

Patrik O Hoyer. Non-negative matrix factorization with sparseness constraints. *The Journal of Machine Learning Research*, 5:1457–1469, 2004.

Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by nonnegative matrix factorization. *Nature*, 401:788–791, 1999.

Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th Python in Science Conference*, 2015.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Mikkel N Schmidt, Jan Larsen, and Fu-Tien Hsiao. Wind noise reduction using non-negative sparse coding. In *Machine Learning for Signal Processing, 2007 IEEE Workshop on*, pages 431–436. IEEE, 2007.

Paris Smaragdis, Cedric Fevotte, Gautham J Mysore, Nasser Mohammadiha, and Matthias Hoffman. Static and dynamic source separation using nonnegative factorizations: A unified view. *Signal Processing Magazine, IEEE*, 31(3):66–75, 2014.

Emmanuel Vincent, Rémi Gribonval, and Cédric Févotte. Performance measurement in blind audio source separation. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14(4):1462–1469, 2006.

Kevin Wilson, Bhiksha Raj, Paris Smaragdis, Ajay Divakaran, Kevin W. Wilson, Bhiksha Raj, Paris Smaragdis, and Ajay Divakaran. Speech denoising using nonnegative matrix factorization with priors. In *in IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2008.

Hui Zou, Trevor Hastie, and Robert Tibshirani. Sparse principal component analysis. *Journal of computational and graphical statistics*, 15(2):265–286, 2006.