For office use only	Team Control Number	For office use only
T1 ____________	**1911426**	F1 ____________
T2 ____________		F2 ____________
T3 ____________	Problem Chosen	F3 ____________
T4 ____________	**A**	F4 ____________

2019 Mathematical Contest in Modeling(MCM) Summary Sheet

Be a precision audience in the cocktail party

Abstract

Cock Tail party problem (CPP) is a well-known problem in the field of computer speech recognition. Stimuli segregation is easy for human to realize, but hard for computers. Thus for the compound-tone input case, we tend to find out a method that can segregate those input pure-tone signals.

For part one, on the basis of the basilar membrane’s simulation in the literature, first we apply Gammatone filter into our model to shield the noise. Then we consider the nonlinear effect of the auditory system, which shows different power ratios of input and output signal varied with frequency. After that, we use the method of pre-emphasis and framing to deal with the sample discrete signals. Finally we use STFT to treat those signals in frequency domain and get the output signals.

For part two, provided with 3 output signals, we need to refine 5 original signals input into the system. It’s just a simple case of the underdetermined blind source separation question. We set two main steps to solve this problem: determine the mixing matrix A and the original source vector S. For finding the matrix A, we use the k-means cluster method and get it’s column vectors. By confirming the source vector S, we set this to a linear programming problem, searching S to minimize its L1 norm. As the original signals are sparse, the segregated signals have little error compared with the original ones.

In sensitivity analysis, we calculate the underdetermine case by varying the input and output source number, and we obtain that our model is robust and do not rely on those signal number.

Keywords: CPP, Gammatone filter, underdetermined source seperation, k-means cluster method

Contents

1 Introduction

For humans, it is very easy to switch attention between multiple sound sources. It is so easy that we cannot even feel the existence of this process. Unfortunately, machines are hard to achieve the same intelligence as humans. So far, a computer can recognize a single speech signal with high precision, but it cannot handle the case of multiple sound mixing. Figure 1.1 shows the scenario we discussed.

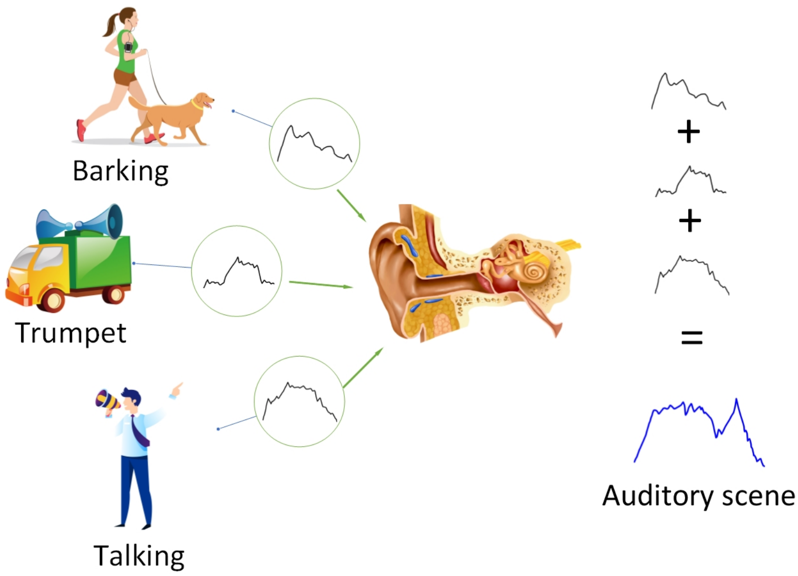

Figure 1.1: Schematic diagram of multiple sound mixing

1.1 Background

The Cocktail Party Problem (CPP), first proposed by *Colin Cherry* in 1953[1]. It is a psychoacoustic phenomenon that refers to the remarkable human ability to selectively attend to and recognize one source of auditory input in a noisy environment, where the hearing interference is produced by competing speech sounds or a variety of noises that are often assumed to be independent of each other. To unveil the mystery and imitate the human performance with a machine, scientists have attempted to view and simplify this complex perceptual task as a learning problem, for which tractable computational solutions are sought. One of the most efficient solutions is Blind Source Separation (BSS).

Blind source separation, also known as blind signal separation, is a method of the separation of a set of source signals from a set of mixed signals without the aid of information about the source signals or the mixing process. The BSS problem is usually divided into two categories: overdetermined blind source separation and underdetermined blind source separation.

1.2 Restatement of the problem

The problem simplified CPP into a system that can input five sound signals and then output three sound signals. Any one of the output signals is a linear combination of five input signals. Without any information except three output signals, we aim to estimate the five signals input.

We reorganize the issue into the following tasks:

- Get the sound we want and shield the rest of the noise;
- Estimate and restore the five signals input;
- Extend the model to be able to determine the number of signals.

1.3 Literature Review

The research on BSS began in 1986[2]. The French scholars *Herault* and *Jutten* proposed a feedback neural network model and a Hebb learning rule-based H-J algorithm to successfully implement two source signals' separation, which is based on a linear mixing model. Subsequently, *E.Sorouchyari* and *P.Comon* et al. systematically proved the convergence and stability of the algorithm, which promoted the development of BSS[3, 4].

In 1991, *L.Tong* pointed out that the reconstructed signal obtained by BSS remains the original waveform characteristics[5].Thus it ensures the information of the source signals is not lost. It guaranteed the solvability of BSS problem.

In 2001, *Bofill* and *Zibulivsky* et al. proposed a method for the underdetermined BSS problem based on "two-stage"[6]. The first stage uses the potential function to recover mixing matrix. The second stage is based on the mixing matrix obtained in the first stage, and separate the source signals by the shortest path decomposition *method*.

2 Assumptions and Notations

2.1 Assumptions and Justifications

- **Every element of the blind signals matrix and source signals is positive.** In the real world, the signals are all positive, and the mixing coefficients of different signals are also positive.
- **The source signals are sparse.** A set of signals is said to be sparse if and only if only one signal is positive at any time. If the input distribution is sparse, the mixing matrix can be estimated either by external optimization or by clustering. Yet, when the signals per se do not satisfy this assumption, sparsity can still be achieved by realizing the separation in a sparser transformed domain.
- **The rank of the blind mixing matrix is exactly the number of rows of the matrix.** Our algorithm takes advantage of the disproportionate nature of each column. Also, if two rows are proportional, then two outputs will be proportional, which is equivalent to the two output signals being the same and can be considered as one signal.

2.2 Notations

Some notations used in this paper are listed in Table 2.1.

Table 2.1: Notations

Notation	Definition	Dimensions/Unit
	Matrix	
A	Blind mixing matrix	$M \times N$
S	Source signals matrix	$N \times T$
X	Output signals matrix	$M \times T$
	Variable	
M	Number of output signals	unitless
N	Number of input signals	unitless
T	Number of simple points	unitless

3 Model establishment

As problem set forth, the system contains five inputs and three outputs. For the number of output is less than input, it is a Underdetermined Blind Source Separation (UBSS) problem. We can solve this kind of problem using clustering method.

3.1 The filter Model

3.1.1 Simulating basilar membrane with Gammatone filter bank

Human can use their basilar membrane to filter the noise and get the sound they want. In the case of pure tones input, when the frequency is high, the maximum amplitude of the sound wave occurs at the bottom of the basilar membrane; On the contray, it appears at the top. As the input sound wave is compound, basilar membrane can sperate those sound components with different frequency and amplitude, which will have the function of spectrum analysis.

Because basilar membrane have such function, we can use a series of filters to simulate the process mentioned above. The Commonly used filter banks include resonant filter bank[7], Round-Exponential filter bank[8] and Gammatone filter bank[9], ect. Because Gammatone filter bank can primely stimulate the physiological properties by using less parameters and has a simple function of unit impulse response, so we apply Gammatone filter bank to our model.

Gammatone filter is defined by its function of unit impulse response in time domain:

$$g_i(t) = At^{n-1}exp(-2\pi b_i t)cos(2\pi f_i + \phi_i)u(t), t \geq 0, 1 \leq i \leq N \tag{3.1}$$

In formular 3.1, A represents the gain of the filter, f_i represents filter's center frequency, ϕ_i means the phase, n is the order of the filter and N is the number of Gammantone filter in the bank. b_i is the decay factor of the filter, it determines the speed of impluse response's attenuation, and it can be defined as follow:

$$bi = 1.019ERB(f_i) \tag{3.2}$$

$ERB(f_i)$ is the equivalent rectangular bandwidth of Gammantone filter, and it has the relationship with filter's center frequency f_i as follow:

$$ERB(f_i) = 24.7 \times (4.37 \times \frac{f_i}{1000} + 1) \tag{3.3}$$

In order to simplify the model, we treat the phase $\phi = 0$, the filter's order $n = 4$. Because all Gammatone filters use the same amplitude A,so parameter A will not affect the final result, we set $A = 1$.

3.1.2 The nonelinearal compression of auditory system

By searching for the literature, we find that auditory system has the characteristic of nonlinearity, which means the output power compression of high frequency signal differs from the low frequency one. This characteristic enhances anti-interference ability of auditory system[10]. Scientists use λ to analyze the nonlinearity of the auditory system, which is defined as below:

$$\lambda = \frac{10\lg(\frac{P_o}{P_r})}{10\lg(\frac{P_i}{P_r})} \tag{3.4}$$

The value of parameter λ varies a lot in different literature, but there are two principles recognized by most scholars:

(1) it shows a strong compressing effect among signals of which frequency above $1KHz$, and λ is almost the same.

(2) if the signal's frequency is below $1KHz$, with its frequency decaying, compressing effect damps a lot.

We use one of functions between λ and signal's freqency f obtainalbe in the literature satisfied those two priciples[11], which can be shown as the graph below:

3.1.3 Refining the signal by FFT method

Basilar membrane's refining and shielding process is similar to choose a group of f_i in a certain range for the Gammatone filter bank. By selecting the center frequency f_i in the range $1KHz - 5KHz$ uniformly, and calculate λ, then we can get the sigals outputted by the auditory system.

In order to make the sigals inputted more precise to the actual and apply to the computers, we need to discretize firstly. Then the discrete signals need to take the procedures of pre-emphasis, framing.

Set $x(n)$ is the signal before pre-emphasis, $y(n)$ is the signal after this process and a is the shifting parameter. So pre-emphasis can be written as formula below:

$$y(n) = x(n) - a \cdot x(n-1) \tag{3.5}$$

For the process of framing, we use hamming window function $w(n)$ to mutiple $y(n)$ to get the signal $s_w(n)$, which are as follow:

$$s_w(n) = y(n) \cdot w(n) \tag{3.6}$$

Set $Y(k)$ and $H_i(k)$ are the discrete spectrum of the signals $y(n)$ and $h_i(n)$ respectively after FFT, and then we can get the spectrum of discrete signals $M_i(k)$ outputted by the filter i:

$$M_i(k) = (Y^2(k)H(k))^{\lambda} \tag{3.7}$$

$$M(k) = \sum_{i=1}^{N} M_i(k) \tag{3.8}$$

N is the number of the filter. Appearently, $M(k)$ is the whole power density spectrum of the output discrete signals. Using the inverse FFT, we can get the discrete output signal $m(n)$ in time domain.

3.2 The UBSS Model

3.2.1 Mathematical expression of UBSS problem

Let $\mathbf{X}^t$ be an M-dimensional column vector corresponding to the output of M sensors at a given discrete time instant t, and let $\mathbf{X}$ be an $M \times T$ matrix corresponding to the sensor data at all times $t = 1, ..., T$. Let $\mathbf{S}$ be the $N \times T$ matrix of underlying source signals and let $\mathbf{A}$ be the $M \times N$ mixing matrix. The problem of blind source separation consist of finding the solution to the following system of equations:

$$\mathbf{X} = \mathbf{AS} \tag{3.9}$$

Because we have no information about the source signals or the mixing process, matrix $\mathbf{A}$ and $\mathbf{S}$ are unknown (We assume matrix $\mathbf{A}$ is of full rank).

Our step is to recover blind mixing matrix $\mathbf{A}$ first and then estimate source signals matrix $\mathbf{S}$.

3.2.2 The recovery of blind mixing matrix

We can decompose Eq. 3.9 into

$$\mathbf{x}^t = \sum_{j=1}^{N} \mathbf{a}^j s_j^t \tag{3.10}$$

We definite $\mathbf{a}^j$ is a column vector of $M \times N$ mixing matrix $\mathbf{A}$. s_j^t is the jth component of the vector $\mathbf{S}$ at time t. $\mathbf{x}^t$ is the output vector at time t. Following from Eq. 3.10, if only one of the sources (say, source i) was different from zero, then all $\mathbf{x}^t$ would be propertional to $\mathbf{a}^i$ and all data points in mixture space would be aligned along the direction of this basis vector. When the sources are sparse, for a given data point t, if one of the sources is significantly larger, the remaining ones are likely to be close to zero. Thus, the density of data in mixture space shows a clear tendency to cluster along the directions of the basis vectors $\mathbf{a}^j$'s.

Estimating the mixing matrix, then, consist of finding the directions of maximum data density. The directions is determined by column vectors $\mathbf{a}^j$ of mixing matrix $\mathbf{A}$. We can determine the directions of maximun data density by using k-means clustering algorithm.

3.2.3 The estimation of source signals

As it is an underdetermined system, Eq. 3.10 are underdetermined equations. Even when mixing matrix $\mathbf{A}$is known, the solution of Eq. 3.10 is not unique. Our approach to sparse UBSS consist of finding the solution that minimizes the l_1 norm.

In the case that blind mixing matrix $\mathbf{A}$ is known, the problem of inferring the source can be formulated independently for each data point x^t, leading to T tractable small problems

$$\min_{\mathbf{s}^t} \frac{1}{2\sigma^2}||\mathbf{A}\mathbf{s}^t - \mathbf{x}^t||^2 + \sum_j \mid s_j^t \mid, \tag{3.11}$$

for $t = 1, ..., T$. Eq. 3.11 is in case with additive Gaussian noise $\mathbf{X} = \mathbf{AS} + \mathbf{V}$, and assuming that $\mathbf{A}$ is uniformly distributed. Variable σ^2 is the variance of the noise $\mathbf{V}$. To simplify the problem, we only consider the case in the absence of noise. Eq. 3.11 can be rewrited as:

$$\min_{\mathbf{s}^t} \sum_j \mid s_j^t \mid \tag{3.12}$$

$$s.t \quad As^t = x^t \qquad t = 1, ..., T \tag{3.13}$$

Eq. 3.12 is equivalent to minimizing the l_1 norm of s^t. It is obviously to find that if the original signals are sparser, the the l_1 norm of s^t will be smaller. We can solve this problem by using linear programming algorithm.

4 Results

4.1 Signals that satisfy the sparsity completely

4.1.1 Data generation

- **Generation of source signals S.** We randomly generate an $N \times T$ matrix. For each column of $\mathbf{S}$, we randomly select one of the rows to be reserved, and set all of the rest to zero.
- **Blind mixing matrix A.** Random generated, and the previous assumptions are satisfied.

4.1.2 Signal separation result

The generated source signals $\mathbf{S}$, output $\mathbf{X}$ are shown in Figure 4.1.

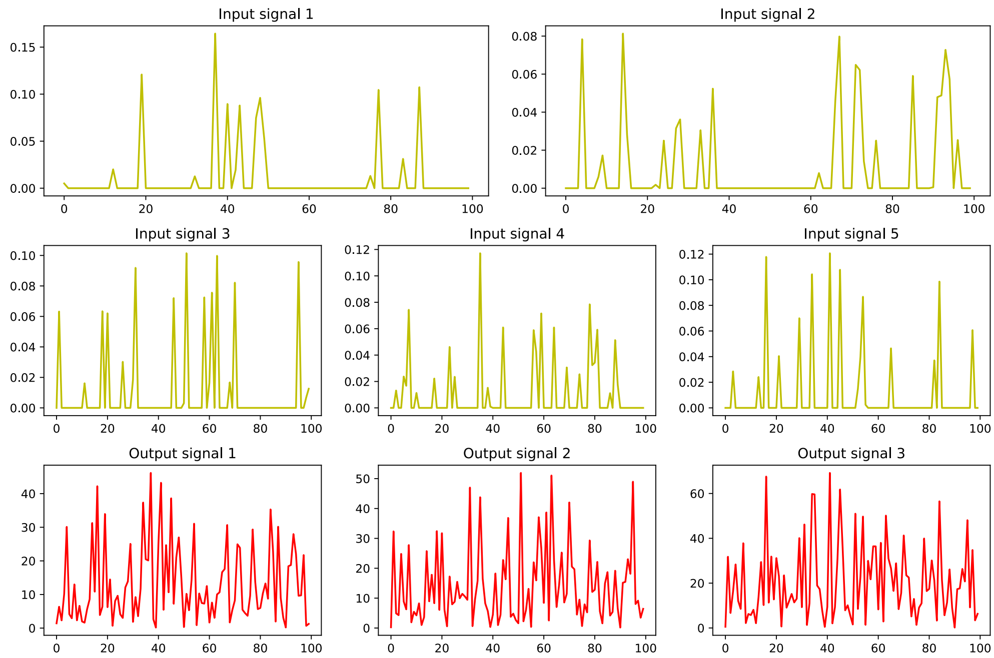

Figure 4.1: Input signals and output signals

In this experiment, the matrix $\mathbf{A}$ we used are shown in equation 4.1.

$$\mathbf{A} = \begin{pmatrix} 11 & 22 & 34 & 45 & 56 \\ 56 & 47 & 28 & 19 & 10 \\ 55 & 64 & 32 & 72 & 21 \end{pmatrix} \tag{4.1}$$

Before we multiple $\mathbf{A}$ with $\mathbf{S}$ to get $\mathbf{X}$, we first normalize each column of $\mathbf{A}$ and sort the columns, then we apply the same sorting indices to $\mathbf{A}$ to make the $\mathbf{A}$ be comparable with our estimated result $\hat{\mathbf{A}}$. The normalized and sorted $\mathbf{A}_{ns}$ is shown in equation 4.2, and th actual $\mathbf{A}$ we used is shown in equation 4.3:

$$\mathbf{A}_{ns} = \begin{pmatrix} 0.64367816 & 0.36170213 & 0.09016393 & 0.16541353 & 0.33088235 \\ 0.11494253 & 0.29787234 & 0.45901639 & 0.35338346 & 0.13970588 \\ 0.24137931 & 0.34042553 & 0.45081967 & 0.48120301 & 0.52941176 \end{pmatrix} \tag{4.2}$$

$$\mathbf{A} = \begin{pmatrix} 56 & 34 & 11 & 22 & 45 \\ 10 & 28 & 56 & 47 & 19 \\ 21 & 32 & 55 & 64 & 72 \end{pmatrix} \tag{4.3}$$

Our result is shown in equation 4.4. Using our model, the estimated $\hat{\mathbf{A}}$ is nearly the same with the original $\mathbf{A}_{ns}$.

$$\hat{\mathbf{A}} = \begin{pmatrix} 0.64367816 & 0.36170213 & 0.09016393 & 0.16541353 & 0.33088235 \\ 0.11494253 & 0.29787234 & 0.45901639 & 0.35338346 & 0.13970588 \\ 0.24137931 & 0.34042553 & 0.45081967 & 0.48120301 & 0.52941176 \end{pmatrix} \tag{4.4}$$

The estimated input signals are shown in Figure 4.2.

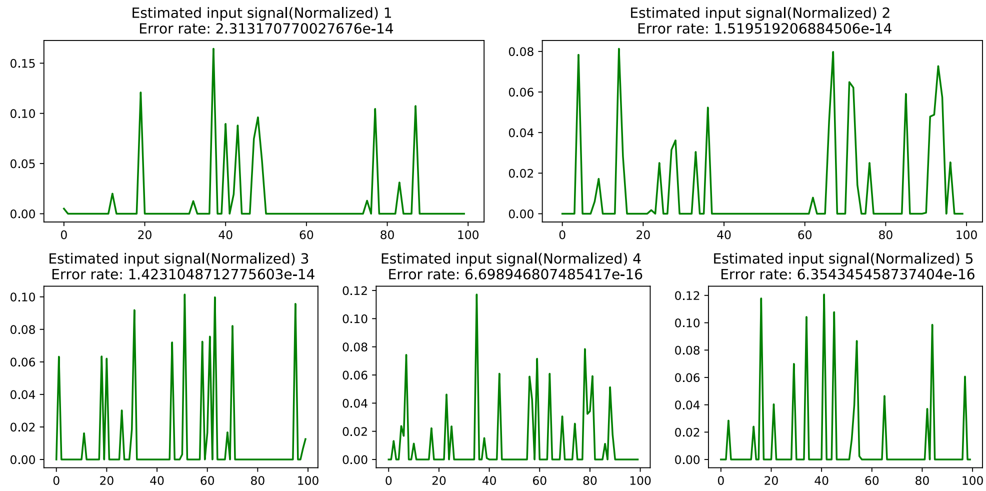

Figure 4.2: Estimated input signals and error rates

The error rates for estimated signals are shown in Table 4.1. The error rate of estimated source signals$\tilde{\mathbf{S}}$ is calculated by equation 4.5 where $||\mathbf{M}||_F$ is the Frobenius norm of matrix $\mathbf{M}$.

$$E = \frac{||\tilde{\mathbf{S}} - \mathbf{S}||_F}{||\mathbf{S}||_F} \tag{4.5}$$

Table 4.1: Error rates

# of signal	1	2	3	4	5
Error rate E	2.3132×10^{-14}	1.5195×10^{-14}	1.4231×10^{-14}	6.6989×10^{-16}	6.3543×10^{-16}

From the above results, we can see that the restored signals are almost the same with the original signals. We can conclude that our model works perfectly on the sparse signals, and the main error is introduced in the second step which derives **S**using the output signals **X**and our estimated $\hat{\mathrm{A}}$.

4.2 Signals that do not satisfy the sparsity completely

4.2.1 Data generation

- **Generation of source signals** S**.** We randomly generate an $N \times T$ matrix.
- **Blind mixing matrix A.** Random generated, and the previous assumptions are satisfied.

For signals that do not satisfy the sparsity completely, input and output signals are shown in Figure 4.3.

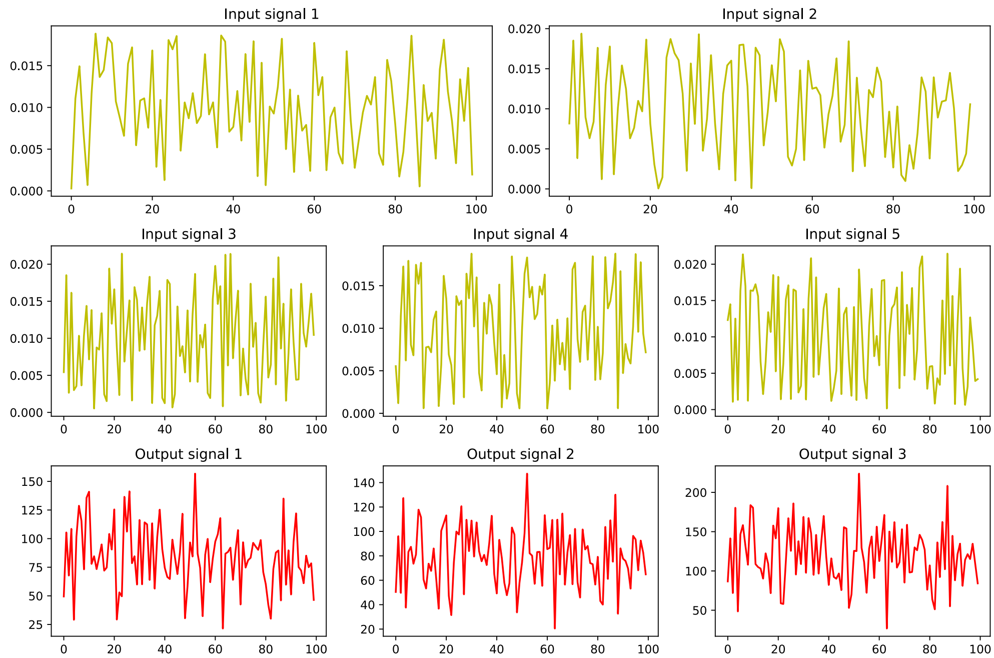

Figure 4.3: Input signals, output signals, and matrix A for non-sparse signals

In this example, we use the same matrix $\mathbf{A}$ in equation 4.3. Our estimated $\hat{\mathbf{A}}$ is shown in 4.6, which differs $\mathbf{A}_{ns}$ a lot. So we use the Normalized Mean Square Error (NMSE) shown in equation 4.6 to measure the error of our estimated $\hat{\mathbf{A}}$.

$$\hat{\mathbf{A}} = \begin{pmatrix} 0.32976212 & 0.37185997 & 0.32307546 & 0.24419688 & 0.29094788 \\ 0.28763198 & 0.22420494 & 0.24989554 & 0.32566172 & 0.27799753 \\ 0.38260589 & 0.4039351 & 0.42702901 & 0.4301414 & 0.43105459 \end{pmatrix} \tag{4.6}$$

$$\text{NMSE=10log}_{10}\left(\frac{||\hat{\mathbf{A}} - \mathbf{A}_{ns}||_F^2}{||\mathbf{A}_{ns}||_F^2}\right) = 20\log_{10}\left(\frac{||\hat{\mathbf{A}} - \mathbf{A}_{ns}||_F}{||\mathbf{A}_{ns}||_F}\right) \tag{4.7}$$

The NMSE result is shown in equation 4.8.

$$\text{NMSE=-8.36988839305828} \tag{4.8}$$

The estimated input signals and error rates are shown in 4.4, and Table 4.2.

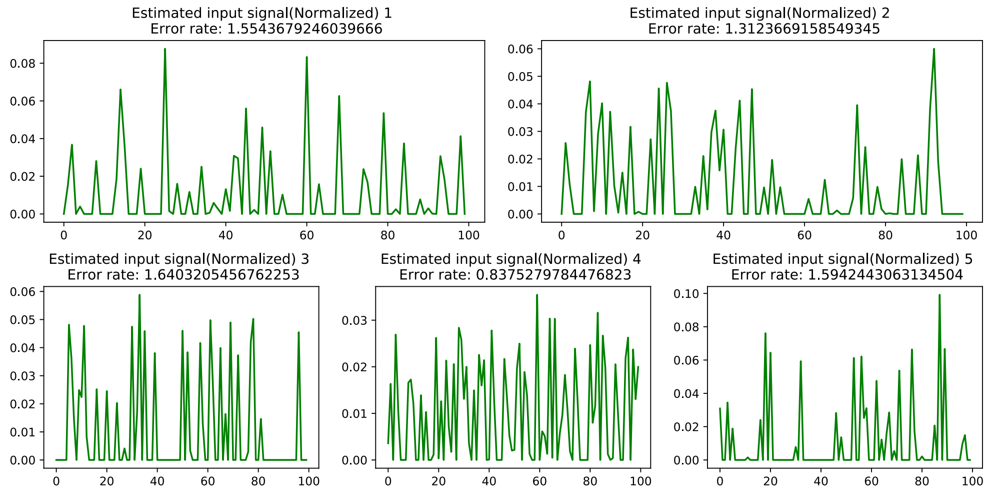

Figure 4.4: Estimated input signals and error rates for non-sparse signals

Table 4.2: Error rates for non-sparse signals

# of signal	1	2	3	4	5
Error rate E	1.5544	1.3124	1.6403	8.3752×10^{-1}	1.5942

Form the results, we can see that our model cannot deal with non-sparsed signals well. The error rates are far larger than those of sparse signals. Both the first step and second step introduce errors, so we need to transform the signals to a sparser domain in order to get better results.

5 Sensitivity Analysis

In our model, the most important parameters are the number of output signals N, the number of input signals M, and the number of simple points T. Once these parameters are determined, we randomly construct the matrix **A**, **S** according to these parameters. Sensitivity Analysis is primarily conducted on these parameters. The blind source separation results obtained using different parameter combinations are shown in Table 5.1.

#	M	N	T	NMSE of estimated $\hat{\mathbf{A}}$	Maximum error rate of estimated $\hat{\mathbf{S}}$
1	5	8	100	-9.47844855412843	1.6462
2	6	8	100	-6.7807464113332605	1.5461
3	6	8	10000	-6.7807464113332605	1.4232
4	8	10	100	-4.37009869738141	1.6465
5	12	15	100	-0.9793842117874039	1.9828

Table 5.1: Blind source separation results

5.1 Sensitivity Analysis for M

From experiment 1 and experiment 2, we can see that the bigger the M, the better our separation result. However, the NMSE might larger because in experiment 2 the **A** has one more row, so the error of **A** might seem to be larger.

5.2 Sensitivity Analysis for N

Form experiment 2, 4, and 5, we can see that the bigger the N, both NMSE and error rate of separation result get larger.

5.3 Sensitivity Analysis for T

From experiment 2 and 3, we can see that our estimated $\hat{\mathbf{A}}$ is the same, but our separation results are better with bigger T.

6 Model expansion

K-means clustering method needs the assumption that signals are sparse in one domain. Hoever, if the signals are not sparse in the domain we choosed, the segregation results appear to be poor. Also, it is not easy to find a domain with all input signals sparse in some cases. Non-negative matrix factorization (NMF) is a method that do not need the assumption of sparse input signals, which will achieve a better result.

NMF method has to dertermine matrix A and original source matrix S, which will achieve for the target written below[12]:

$$\begin{cases} Min: \ D_{F\alpha}(A,S) = \frac{1}{2}||Y - AS|_F^2 + \alpha_A J_A(A) + \alpha_S J_S(S) \\ s.t. \ a_{ij} \geq 0, \ s_{jk} \geq 0, \ \forall i,j,k, \end{cases} \tag{6.1}$$

In 6.1, α_A and α_S are two nonnegative regularization parameters and $J_A(A)$ and $J_S(S)$ are used to enforce a certain application-dependent characteristic of solution. In order to simplify the problem and have a clear comparision to the method of K-means clustering, we set $J_A(A) = 0$ and $J_X(S) = \sum\limits_{jk} |s_{jk}|$.

By applying the standard gradient descent approach to 6.1, we have:

$$\begin{cases} a_{ij} \leftarrow a_{ij} - \eta_{ij} \frac{\partial D_{F\alpha}(A,S)}{\partial a_{ij}} \\ s_{jk} \leftarrow s_{jk} - \eta_{jk} \frac{\partial D_{F\alpha}(A,S)}{\partial x_{jk}} \end{cases} \tag{6.2}$$

6.2 shows the iteration of a_{ij} and s_{jk}, where the parameter η_{ij} and η_{jk} can be determined as[13]:

$$\eta_{ij} = \frac{a_{ij}}{[ASS^T]_{ij}}, \ \eta_{jk} = \frac{s_{jk}}{[AA^TS]_{jk}} \tag{6.3}$$

Take η_{ij} and η_{jk} into formula 6.2, we can get:

$$\begin{cases} a_{ij} \leftarrow a_{ij} \frac{\left[[YS^T]_{ij}\right]_\varepsilon}{[ASS^T]_{ij} + \varepsilon} \\ s_{jk} \leftarrow s_{jk} \frac{\left[[A^TY]_{jk} - \alpha_S\right]_\varepsilon}{[A^TAS]_{jk} + \varepsilon} \end{cases} \tag{6.4}$$

In 6.4, ε avoid the numerater being too small and denominator becoming zero in the iteration process, while $[x]_\varepsilon$ means $max(x, \varepsilon)$. By finding the literature[12], we choose $\varepsilon = 10^{-9}$ and $\alpha_S = 0.25$ in our model.

7 Strengths and Weaknesses

7.1 Strength

- We use a series of filters to simulate the operating mode of basilar membrane.

- Our model can successfully implement five source signals' separation through three outputs of linear mixing.
- By performing domain conversion before separation, our model can deal with the case that source signals are not sparse in the time domain.

7.2 Weakness

- If there is a source signal whose strength is much greater than others', our model cannot estimate the blind mixing matrix and source signals precisely.
- The effect of model estimation depends on the parameters of the mixing matrix.
- The k-means algorithm relies on the selection of random initial points, so there are some differences in the results of each run.

8 Conclusion

By simulating the basilar membrane, we apply Gammatone filter into our model to shield the noise. Our model shows different power ratios of input and output signal varied with frequency, considering the nonlinear effect of the auditory system. Thus, we use the method of pre-emphasis and framing to deal with the sample discrete signals and use STFT to treat those signals in frequency domain and get the output signals.

For UBSS, Our model set two main steps to solve this problem: (1) Use the k-means cluster method and get it's column vectors to recover the mixing matrix A; (2) Set the problem of searching S to a linear programming problem, minimizing its l_1 norm, to estimate the source matrix S. To verify the correctness of our model, we performed a segregation experiment using a random mixing matrix and source signals. Result shows that the restored signals' error rate is very small. Our model works perfectly on the sparse signals. Then we verified that our model cannot deal with the case that source signals don not satisfy the sparsity assumption.

References

[1] Simon Haykin and Zhe Chen. The cocktail party problem. *Neural computation*, 17(9):1875–1902, 2005.

[2] Jeanny Herault and Christian Jutten. Space or time adaptive signal processing by neural network models. In *AIP conference proceedings*, volume 151, pages 206–211. AIP.

[3] Christian Jutten and Jeanny Herault. Blind separation of sources, part i: An adaptive algorithm based on neuromimetic architecture. *Signal processing*, 24(1):1–10, 1991.

[4] Pierre Comon, Christian Jutten, and Jeanny Herault. Blind separation of sources, part ii: Problems statement. *Signal processing*, 24(1):11–20, 1991.

[5] Lang Tong, R-W Liu, Victor C Soon, Y-F and systems. Indeterminacy and identifiability of blind identification. 38(5):499–509, 1991.

[6] Pau Bofill and Michael Zibulevsky. Underdetermined blind source separation using sparse representations. *Signal processing*, 81(11):2353–2362, 2001.

[7] Richard F Lyon and Carver Mead. An analog electronic cochlea. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(7):1119–1134, 1988.

[8] Roy D Patterson. Auditory filters and excitation patterns as representations of frequency resolution. *Frequency selectivity in hearing*, 1986.

[9] PLM Johannesma. The pre-response stimulus ensemble of neurons in the cochlear nucleus. In *Symposium on Hearing Theory, 1972*. IPO, 1972.

[10] LI Zhao-hui and CHI Hui-sheng. Progress in computational modeling of auditory periphery. *ACUSTICA*, 31(5):449–465, 2006.

[11] HU Feng-son and CAO Xiao-yu. Auditory feature extraction based on gammatone filter bank. *Computer Engineering*, 38(21):168–170, 2012.

[12] Andrzej Cichocki, Rafal Zdunek, and Shun-ichi Amari. New algorithms for non-negative matrix factorization in applications to blind source separation. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 5, pages V621–V624. IEEE, 2006.

[13] Paul Sajda, Shuyan Du, and Lucas C Parra. Recovery of constituent spectra using non-negative matrix factorization. In *Wavelets: Applications in Signal and Image Processing X*, volume 5207, pages 321–332. International Society for Optics and Photonics, 2003.

Appendix A

Appendix B