Non-negative Matrix Factorization: A possible way to learn sound dictionaries

Hiroki Asari

Tony Zador Lab
Watson School of Biological Sciences
Cold Spring Harbor Laboratory

August 22, 2005

Abstract

Auditory system detects sound signals and uses the temporal-frequency information of the sound signals to conduct sound identification, sound localization, and sound source separation. Thanks to the past studies, we know that the hair cells at cochlea show frequency-dependent responses against input sound signals. But, little is known about how the sound information is conducted to and processed within the auditory cortex.

Recently, B.A.Pearlmutter and A.M.Zador proposed an algorithm for monaural source separation under the assumption that the precise head-related transfer function (HRTF) and all the sound dictionary elements are known (ref.[1]). To apply this algorithm to the real sound signals, here I propose that non-negative matrix factorization (NMF) applied to the spectrograms of sound signals would successfully give a set of sound dictionary elements.

When NMF was applied to solo-music signals with an appropriate value for the rank of factorization, it could extract instrument-specific patterns of basis spectrograms, each of which has a peak frequency for different notes. Interestingly, the sound signals converted back from the obtained basis spectrograms sounded more or less like the corresponding instrument, which suggests that the obtained basis spectrograms, or basis sound elements, would be a good candidate for the sound dictionary.

When NMF was applied to sound signals played with several different instruments, the obtained basis sound elements can be categorized into each instrument-specific pattern by hand. In addition, using the categorized elements, sound signals can be reconstructed corresponding to each instrument part of the original music. The fact that source separation can be done using the basis sound elements obtained by NMF suggests that NMF would be a possible way to learn sound dictionaries from sound signals.

1 Introduction

1.1 Monaural source separation

We live in a world with a full variety of sounds from a different combination of sound sources. Thanks to the great feats of the brain, however, we can listen to only those sounds that we pay attention to, ignoring all the other irrelevant sounds. It seems that our brain accomplishes such a complex task with the aid of binaural and monaural sound cues as well as visual cues. One of the most important cues is the filtering effects ("HRTF") by the head, torso, and pinnae, which organisms would learn by experience. With the prior knowledge of HRTF, theoretical work showed that monaural source separation can be done, if in addition we know all the "sound dictionary" elements for all the sound sources (ref.[1]).

In this study, I addressed how we or computer can learn the huge sound dictionary. Because sound signals contain temporal and frequency information, sound signals can be depicted in a two-dimensional plane as spectrograms. Therefore, it would be reasonable to think that we can learn sound dictionaries by learning representative subset of spectrograms for each sound source. Here I propose that an algorithm called non-negative matrix factorization can be used for this purpose.

1.2 Non-negative matrix factorization (NMF)

NMF is an algorithm to factorize a data matrix under the non-negativity constraints (ref.[2], [3]). Because the non-negativity constraints allow only additive combination of bases, NMF gives parts-based representation of the original dataset. In addition, compared to other factorization methods such as principal component analysis (PCA) or independent component analysis (ICA), it is easy to think of the meanings of the obtained bases. Therefore, when NMF is applied to a set of spectrograms of music, each basis elements would be expected to represent different notes for different instruments, which turned out to be true.

2 Methods

All the programming and data analyses were done in Matlab[1].

2.1 Non-negative matrix factorization (NMF)

The original data matrix is given as an $n \times m$ matrix V, each column of which contains the n data values for one of the m spectrograms. Then, the data matrix V is approximated by NMF as

$$V \approx WH = \sum_{a=1}^{r} W_{ia} H_{a\mu}$$

where the rank of factorization, r, is chosen as $nm > nr + rm$ to compress the original data V into WH. The dimension of the factors W and H are $n \times r$ and $r \times m$, respectively. Each column of the matrix W contains one of the basis spectrograms, and the matrix H represents the coefficients for reconstructing the original data matrix V.

The cost function, F, to be maximized by NMF implementation is given by the next equation:

$$F = \sum_{i,\mu} \Big(V_{i\mu} \log (WH)_{i\mu} - (WH)_{i\mu} \Big) \tag{1}$$

proof: Here it is reasonable to think that sound signals are made up of a sparse combination of basis elements at each discrete time point. This sparseness allows to make a model that each data point $V_{i\mu}$ is generated by a Poisson distribution with the mean value $(WH)_{i\mu}$. The likelihood of the Poisson distribution is

$$P_{i\mu}\,(V|WH) = \exp[-(WH)]\ \frac{(WH)^V}{V!},$$

where the indices are omitted to simplify the equation. The logarithm of both sides is taken to transform the equation into

$$\log P_{i\mu} = V \log(WH) - WH - \log(V!).$$

By summing $\log P_{i\mu}$ over i and μ, the likelihood that V is generated by WH is written as

$$\sum_{i,\mu} \log P_{i\mu} = \sum_{i,\mu} \Big(V_{i\mu} \log (WH)_{i\mu} - (WH)_{i\mu} - \log(V_{i\mu}!) \Big).$$

Because the value for $\log(V_{i\mu}!)$ is constant with respect to W and H, we can drop the term $\log(V_{i\mu}!)$, and we will get the cost function F as in equation (1). ***Q.E.D.***

[1]Please contact me at *asari@cshl.edu* if you want to use M-files for this analysis.

To maximize the cost function F, the following update rules are applied (ref.[2]).

$$\begin{aligned} H_{a\mu} &\longleftarrow H_{a\mu} \sum_i H_{a\mu} \frac{V_{i\mu}}{(WH)_{i\mu}} \\ W_{ia} &\longleftarrow W_{ia} \sum_\mu \frac{V_{i\mu}}{(WH)_{i\mu}} H_{a\mu} \\ W_{ia} &\longleftarrow \frac{W_{ia}}{\sum_j W_{ja}} \end{aligned} \tag{2}$$

These update rules find a pair of W and H that gives an approximation $V \approx WH$ by converging to a local maximum of the cost function F. NMF would be applied several times with different non-negative initial values for W and H to find the best approximation for the dataset V. The first two update rules do preserve the non-negativity of W and H, and the third update rule shows the normalization of the column of W ($\sum_i W_{ia} = 1$). The monotonic convergence of the cost function F by these update rules can be proved by using EM algorithm as shown below (ref.[3],[4]).

Expectation-Maximization algorithm Expectation-Maximization (EM) algorithm is a technique to solve maximum likelihood problems with two-step procedures. At the first "expectation" step, a good auxiliary function for the original cost function is formulated, and the auxiliary function is maximized at the second "maximization" step, which is guaranteed to give a non-decreasing value for the original cost function. These two steps are repeated as necessary until the original cost function is converged enough to a local maximum. Therefore, the key for EM algorithm is to find an appropriate auxiliary function.

Expectation step

Let F be a cost function to be maximized. Then, an auxiliary function $G(h, h^t)$ for $F(h)$ is defined as a function that satisfies the following conditions:

$$G(h, h^t) \leq F(h), \qquad G(h, h) = F(h)$$

Maximization step

If G is an auxiliary function for F, then F is non-decreasing with the following update rule:

$$h^{t+1} = \arg\max_h G(h, h^t)$$

proof: $F(h^{t+1}) \geq G(h^{t+1}, h^t) \geq G(h^t, h^t) = F(h^t)$ ***Q.E.D***

In fact, h^{t+1} is not necessarily maximizing G but can be chosen so that G is non-decreasing (Fig.1).

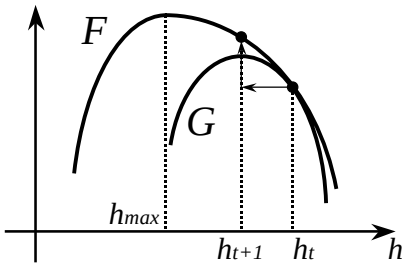

Figure 1: Maximizing the auxiliary function G guarantees to give a non-decreasing value for F.

Proof of the NMF update rules:

The update rules shown in equation (2) can be proved by considering a modified version of EM algorithm. Under the non-negativity constraints $W \geq 0$ and $H \geq 0$, define $G_1(H, H^t)$ and $G_2(H, H^t)$ as

$$\begin{aligned} G_1(H, H^t) &= \sum_{i,\mu} V_{i\mu} \sum_a \frac{W_{ia} H_{a\mu}}{(WH)_{i\mu}} \log(W_{ia} H^t_{a\mu}) - \sum_{i,\mu} (WH^t)_{i\mu} \\ G_2(H, H^t) &= -\sum_{i,\mu} V_{i\mu} \sum_a \frac{W_{ia} H_{a\mu}}{(WH)_{i\mu}} \log \frac{W_{ia} H^t_{a\mu}}{(WH^t)_{i\mu}}. \end{aligned}$$

By fixing W in equation (1), the cost function F can be thought of as a function of H, satisfying the following relation:

$$F(H) = G_1(H, H) + G_2(H, H).$$

Then, $F(H)$ is non-decreasing if $G_1(H, H^t)$ is maximized with respect to H^t, because the inequality $G_2(H^t, H^{t+1}) \geq G_2(H^t, H^t)$ is true[2] for any H^{t+1}. Therefore, by setting $\partial G_1(H, H^t)/\partial H^t = 0$, the update rule for H can be derived as shown in equation (2). The update rule for W can also be derived in a similar method. ***Q.E.D.***

2.2 Principal Component Analysis and Independent Component Analysis

Principal component analysis (PCA) and independent component analysis (ICA) are methods to simplify datasets by replacing a group of the original variables with a single new variable. Both PCA and ICA generate a set of new variables $\{\gamma_i : i = 1, \cdots, n\}$, and the original data x is decomposed into a linear combination of the new variables:

$$x = \sum_i a_i \gamma_i.$$

But, PCA chooses the basis set $\{\gamma_i : i = 1, \cdots, n\}$ as orthogonal, whereas ICA chooses it not as orthogonal but as statistically independent.

[2]Note: $\sum_i p_i \log q_i \geq \sum_i q_i \log q_i, \quad \sum_i p_i = \sum_i q_i = 1$

Principal component analysis PCA was done by using a built-in matlab code `svd` (singular value decomposition, SVD). SVD decomposes a data matrix $\boldsymbol{X}$ into $\boldsymbol{X} = \boldsymbol{USV}^T$ with a diagonal matrix $\boldsymbol{S}$ and unitary matrices $\boldsymbol{U}$ and $\boldsymbol{V}$. Then, $\boldsymbol{US}$ corresponds to principal components given by PCA.

Independent component analysis Independent Component Analysis (ICA) was done by an algorithm called "FastICA" (ref.[5]). Independent components $\boldsymbol{Y}$ are derived by rotating the principal components $\boldsymbol{U}$ by an orthogonal matrix $\boldsymbol{B}$, that is, $\boldsymbol{Y} = \boldsymbol{UB}$. In "FastICA" algorithm, the fixed-point algorithm is used to find $\boldsymbol{B}$ which is based on 4th-cumulant for the criterion of the independency.

2.3 Preparation of various datasets for NMF analyses

Facial images Three hundred and fifty two facial images were downloaded from F.A.C.E.S directory at the internal web site of Cold Spring Harbor Laboratory[3]. All the images were trimmed and shrunk into 20×20 pixel images.

***Kanji* chinese characters** A dataset of *kanji* chinese characters was prepared by rasterizing "MS gothic" fonts (30 points). The dataset contains images (30×30 pixels) of 1006 characters[4] that students in japan would learn by the age of 12.

Images comprised of non-overlapping arbitrary bases Twenty five non-overlapping arbitrary bases were generated by the two-dimensional normal distribution $N[(\mu_x, \mu_y), \sigma]$ with mean (μ_x, μ_y) and standard deviation σ in 25×25 pixel images.

$$\begin{aligned} (\mu_x, \mu_y) &= (5i - 2, 5j - 2), \qquad (i, j = 1, 2, 3, 4, 5) \\ \sigma &= 2.5 \end{aligned}$$

Each basis image was then normalized to sum to unity. Note that these bases were not completely non-overlapping, but can reasonably be regarded as non-overlapping (Fig.2). Then, 1000 images were generated for NMF analyses by randomly-weighted additive combination of the bases.

Images comprised of overlapping arbitrary bases To make overlapping arbitrary bases, two random points were chosen for each basis image within a two-dimensional plane (x, y) $(0 \leq x, y \leq 25)$. A line segment was generated by connecting these two points, and the distance, d, was calculated between the line segment and a point of interest on the plane. Then, the value for each point was determined by $\exp[-d/2]$, and normalized to sum to unity. These steps were iterated to generate 25 overlapping arbitrary bases (Fig.3).

Datasets were generated by an additive combination of the bases with either random coefficients or random sparse coefficients to test the effects of sparseness on the results.

[3] ~~URL: http://intranet.cshl.edu/frame.html?link=faces~~
The web site is not open to public. Instead, face image data are available at the following web sites (Aug 22, 2005).
CBCL database: http://cbcl.mit.edu/cbcl/software-datasets/FaceData2.html
ORL database: http://www.uk.research.att.com/facedatabase.html

[4] National curriculum standards for elementary school issued by Ministry of Education, Culture, Sports, Science, and Technology (written in Japanese): http://www.mext.go.jp/b_menu/shuppan/sonota/990301b/990301d.htm
The list is available for example at AOZORA-BUNKO: http://www.aozora.gr.jp/kanji_table/kyouiku_list.zip

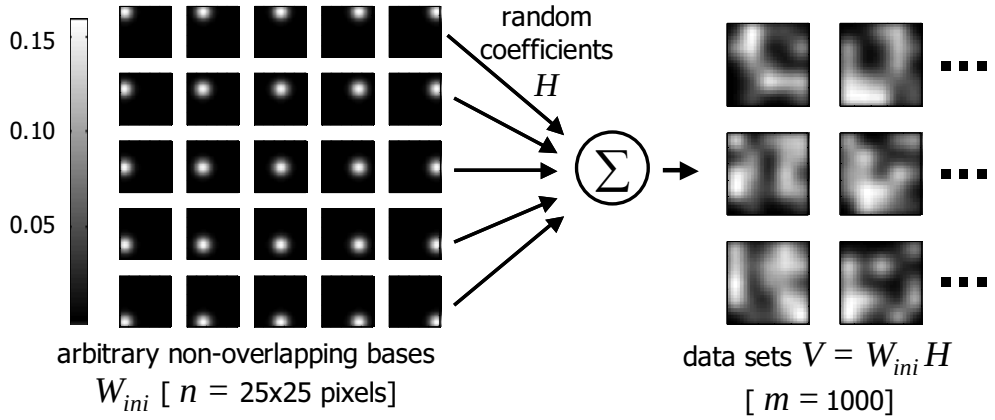

Figure 2: An example of non-overlapping arbitrary bases W_{ini} and some of the generated data set V.

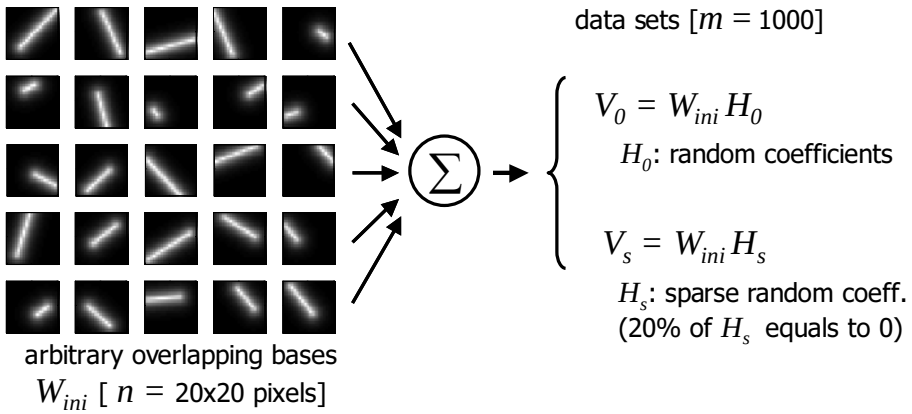

Figure 3: An example of overlapping arbitrary bases W_{ini} and a schematic to generate datasets V_0 and V_s

2.4 NMF analyses on sound signals

Music data were obtained from music CDs as WAVE format sound files with sampling frequency of $11.025 \sim 22.05$ kHz. The sound signals were then fragmented into short overlapped pieces (300 msec), and each of them was converted into logarithmic-scale spectrogram ($\Delta f = 1/12$ octave, $\Delta t = 3$ msec). Each spectrogram had about 6000 data points, and about 1000 spectrograms were used for the NMF analyses with the rank of factorization $r \sim 100$. The obtained basis spectrograms were characterized and categorized by the main peak frequency (first formant) or the pattern for the peaks. In addition, the obtained basis spectrograms were converted back into sound signals (see below for the algorithm) to confirm if the categorization of the spectrograms was reasonable or not. Then, according to the categorization, the sound signals were reconstructed to test whether or not the reconstructed sounds corresponded to each instrument part of the original music.

Algorithm to convert spectrograms back into sound signals First of all, the peak frequencies for each discrete time of a target spectrogram were picked up. Second, the amplitudes for each peak frequency were estimated by the interpolation of the discrete ones to make sound signals with an arbitrary sampling frequency. Then, sine-waves, whose phase was zero at the time point zero, were generated for each peak frequency with the estimated amplitude. By combining all the sine-waves with different frequencies, the sound signals with the arbitrary sampling frequency can be obtained from a spectrogram.

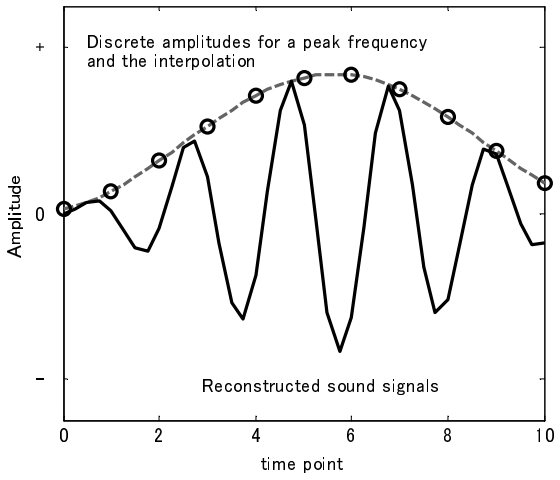

Figure 4: Conversion from spectrogram back into sound signal.
Open circles show example values for a specific peak frequency in a spectrogram at each discrete time. The interpolation is shown in dashed line, and the reconstructed sound signals for this peak frequency is shown in a solid line.

3 Results

In the first section, NMF was applied to various datasets such as facial images to find out characteristics of NMF. The results of the factorization by NMF were also compared with those of other methods such as PCA to validate the use of NMF. In the second section, NMF was applied to datasets of spectrograms obtained from music data. See suplementary data for sound signal data files (WAVE format).

3.1 Characterization of NMF

Example 1: Facial images To reproduce the result of D.D.Lee and H.S.Seung (ref.[2]), I firstly applied NMF to $m = 352$ facial images, each consisting of $n = 20 \times 20$ pixels, with the rank of factorization $r = 36$. Although the number of the original image data $V[m \times n]$ was about one-seventh of that used by D.D.Lee and H.S.Seung ($V[m = 2429, n = 19 \times 19]$), the consistent results were obtained, that is, the basis images W given by NMF ($V \approx WH$) showed parts-based representation as shown in figure 5.

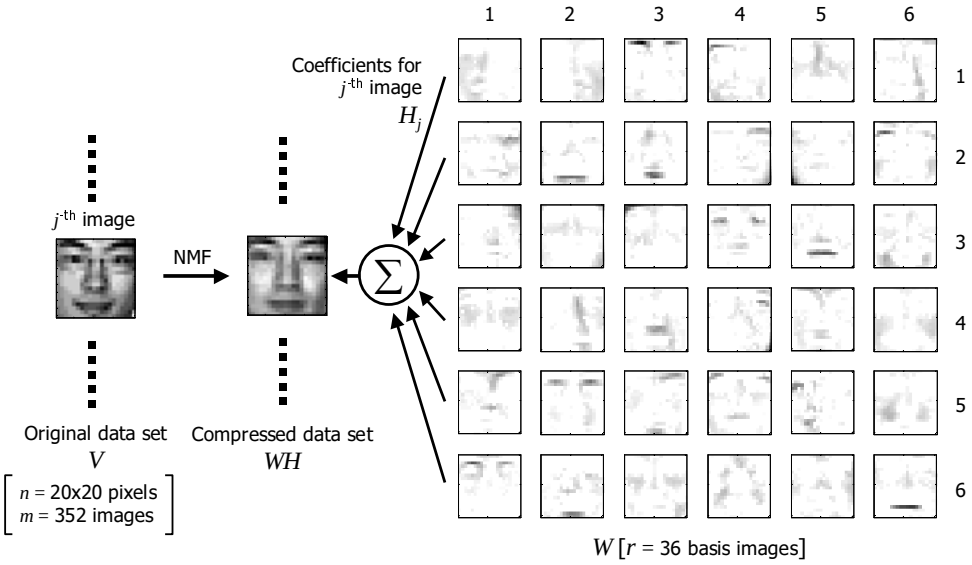

Figure 5: NMF finds an approximate factorization of the form $V \approx WH$.
Each of the obtained basis images W shows distinct parts of faces. For example, W at the position (column, row) $= (4, 3), (2, 5)$ show "eyes," and those at the position $(3, 2)$ and $(5, 3)$ show "mouth."

Example 2: *Kanji* Chinese characters As another example, I applied NMF to a dataset of Chinese characters. Because some of the Chinese characters consist of a combination of "left" and "right" parts or a combination of "top" and "bottom" parts, I expected that NMF would extract these parts that are used frequently in the original dataset. The result of the factorization with the rank $r = 25$ showed that some basis elements represented parts used frequently in the dataset (Fig.6). On the other hand, PCA and ICA failed to find any "meaningful" basis elements but learned wholistic representation (Fig.6, data not shown).

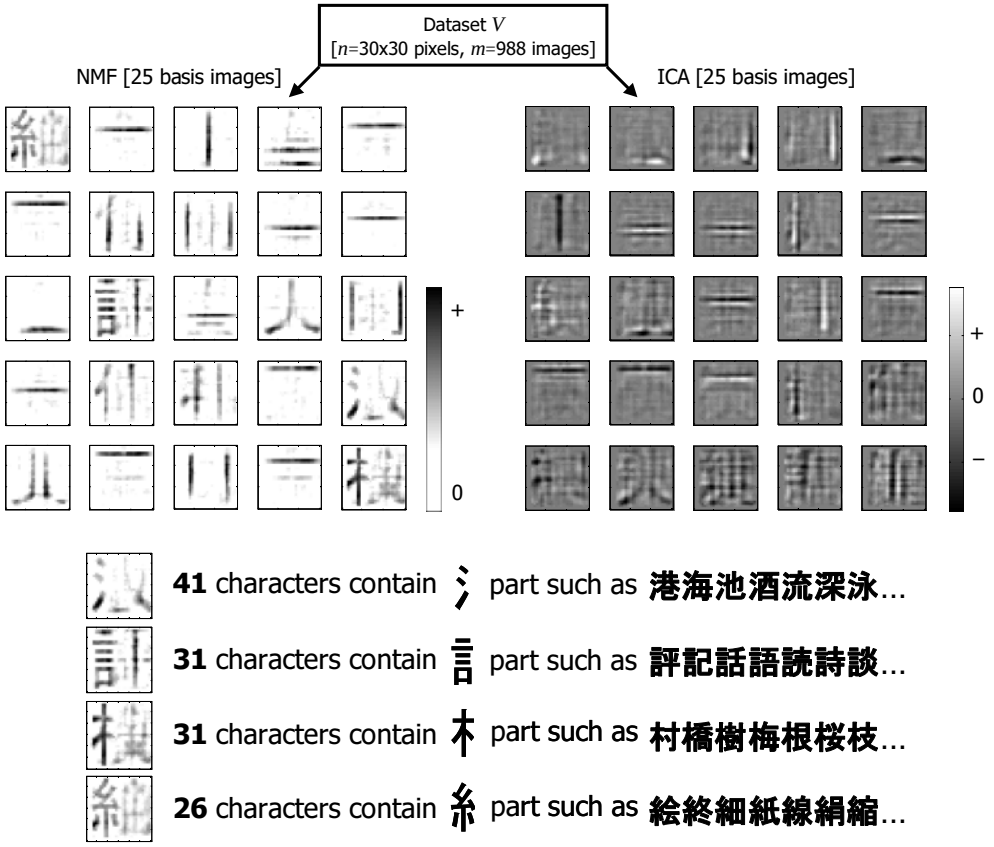

Figure 6: NMF found "meaningful" parts while ICA failed.

NMF and ICA were applied to a dataset of $m = 1006$ *kanji* Chinese characters ($n = 30 \times 30$ pixels). Some of the basis images obtained by NMF showed "meaningful" parts of *kanji* Chinese characters as shown in the bottom panel. In contrast, ICA learns wholistic representation.

Example 3: Dataset comprised of non-overlapping arbitrary bases To confirm if NMF can really extract *the* basis images for the original dataset, I prepared a set of 1000 images V, each of which is generated by randomly-weighted (H) summation of arbitrary non-overlapping bases W_{ini} ($V = W_{ini}H$, see *Methods* section and Fig.2 for details). Then, NMF was applied to see if the dataset V is separated back into the non-overlapping bases W_{ini}. When the rank of factorization was chosen correctly, that is, chosen to be the exact number of the real basis images W_{ini} ($r = 25$ in the case of Fig.2), the bases W obtained by NMF had one-to-one correspondence to the original ones W_{ini} (Fig.7). When the rank was chosen by far smaller or larger than the correct value, however, the obtained basis images showed quite different patterns from W_{ini} (data not shown). In contrast, when the rank was chosen a little smaller or larger than the correct value, some images were screwed up and showed redundancy but the others remained to have good one-to-one correspondence to W_{ini} (data not shown). These results showed that it is important to choose an appropriate value for the rank of factorization r, although I have not come up with a good criterion or theory to do so (see *discussion*).

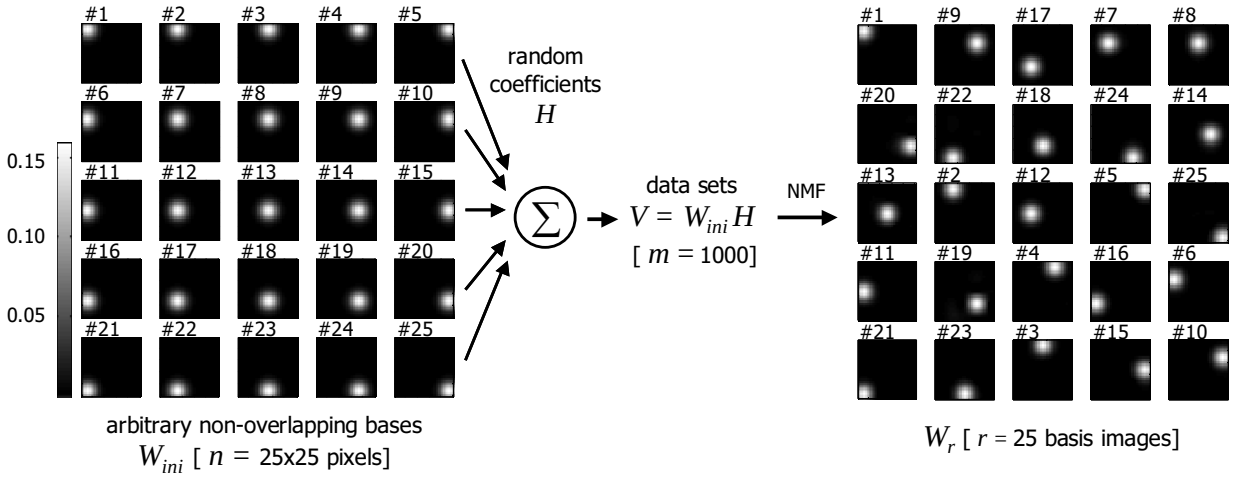

Figure 7: The original non-overlapping bases (W_{ini}) and those obtained by NMF (W_r). Note that there is one-to-one correspondence between W_{ini} and W_r. The numbering was done by hand.

Example 4: Dataset comprised of overlapping arbitrary bases I then generated datasets by an additive combination of the arbitrary overlapping bases W_{ini} with either random coefficients ($V_0 = W_{ini}H_0$) or random sparse coefficients ($V_s = W_{ini}H_s$, see *Methods* section and Fig.3 for details). The basis images obtained by NMF with the correct value for the rank of factorization ($r = 25$) showed that NMF worked better when each image of the dataset is a sparse additive combination of the original bases W_{ini} (Fig.8). In contrast, PCA and ICA failed to break the dataset V_s back into the original bases W_{ini}, although some of the basis images obtained by ICA had one-to-one correspondence to W_{ini} (Fig.8). These results suggest the validity of using NMF for sound analyses because sound signals are supposed to consist of a sparse combination of sound dictionaries (ref.[1]).

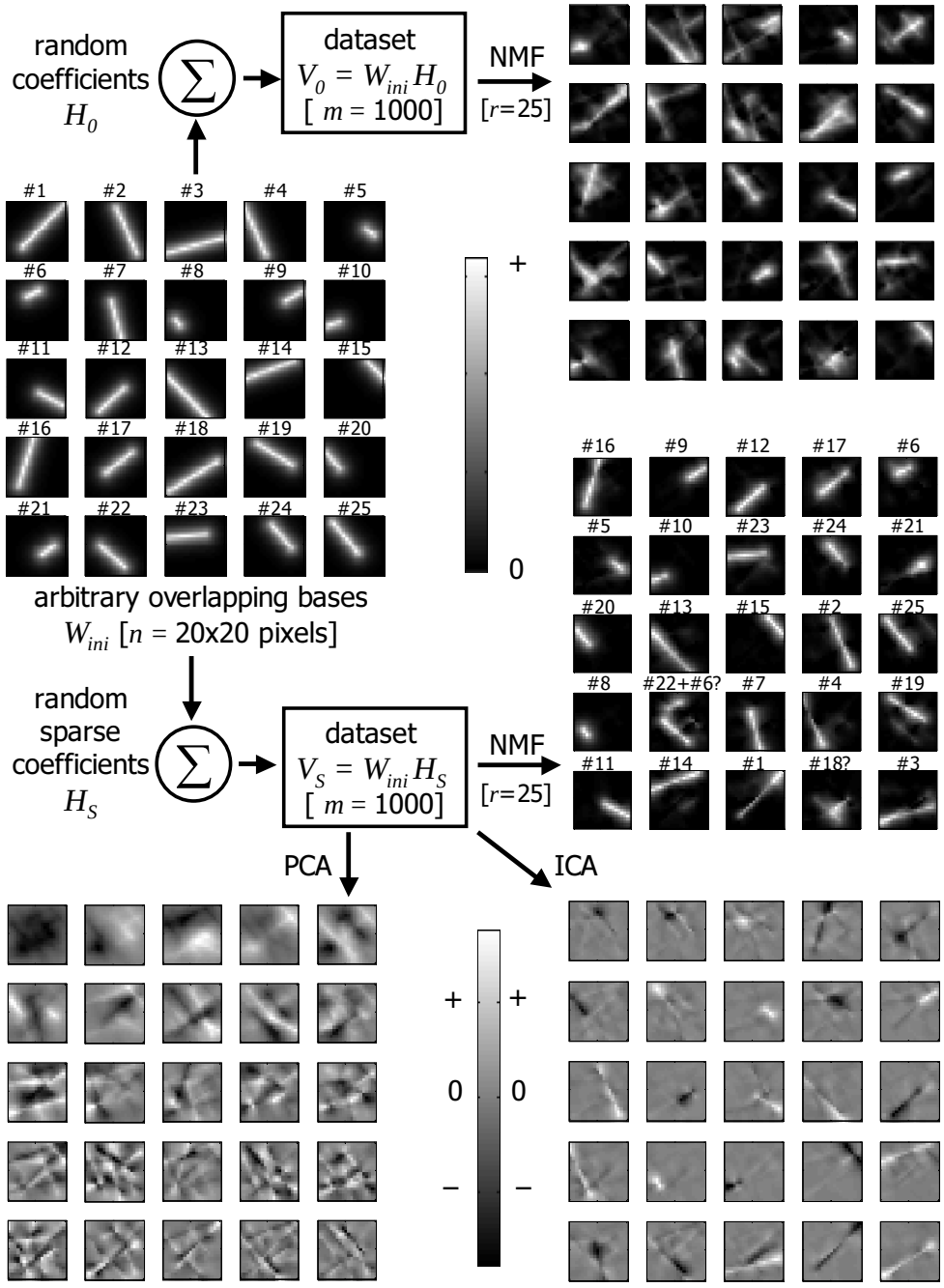

Figure 8: The original overlapping bases W_{ini} and those obtained by NMF, PCA, and ICA
The basis images obtained by NMF from V_s showed good one-to-one correspondence to W_{ini} (except for #18), while the other methods failed to show good fidelity.

3.2 NMF analyses on sound signals

Solo music Several sets of spectrograms of solo music (cello, violin, flute, piano, etc) were used for NMF analyses. As shown in figure 9, when NMF was applied to cello solo music, the obtained basis images showed characteristic patterns for cello, each of which represented different peak frequencies or "notes." Moreover, when they were converted back into sound signals, they did sound like cello (see *Supplementary data*). This is true for other instruments such as violin and flute, although NMF did not work very well to extract the characteristic patterns or "sounds" for piano solo music (data not shown). Interestingly, the basis spectrograms can be categorized into onset-, continuous-, and ending-patterns for each instruments. In addition, because these basis spectrograms form a harmonics of each instrument (see Fig.9 for example), and because they did sound like a harmonics when the basis spectrograms were converted back into sound signals (see *Supplementary data*), it can be said that NMF could separate sound signals into sound dictionary elements of each note of each instrument.

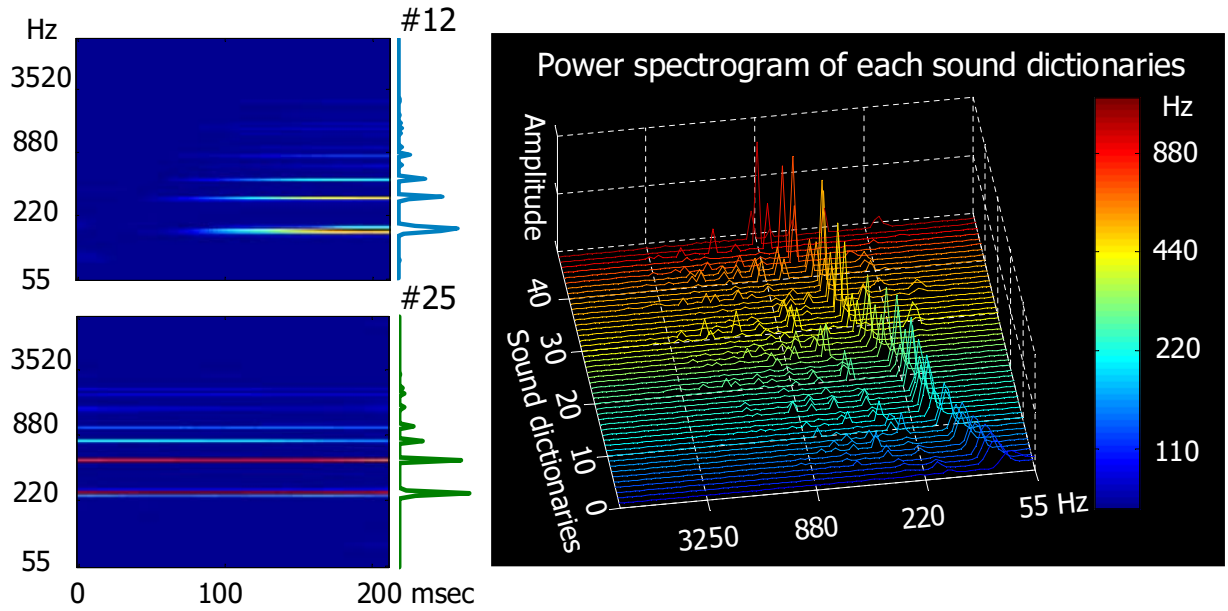

Figure 9: Application of NMF to cello solo music

The original music is "J.S.Bach, cello suite No.1 Prelude ($\sim 2'27''$, 22.05 kHz)." The dataset consists of $m = 983$ log-scaled spectrograms ($n = 6532$: $\Delta f = 1/12$ octave, $\Delta t = 3$ msec) with the time range of 210 msec. The left two paneles show examples of basis images obtained by NMF ($r = 49$), both of which clearly show the characteristic pattern for string instrument, i.e., lots of small resonance peaks in addition to the main peak frequency. The basis image #12 has an onset-pattern and #25 has a continuous-pattern. The right panel shows the power spectrogram of sound dictionaries over the whole time range of the spectrograms. Each of them has different peak frequency or "note," and they form a harmonics of the cello sounds.

Music played with several different instruments I then prepared datasets of music palyed with several different instruments to try source separation based on the basis elements obtained by NMF. When NMF was applied to a quartet (flute, violin, viola, and violoncello), the obtained basis spectrograms ($r = 64$) can be categorized into at least two groups by hand. Spectrograms in one category showed a single peak frequency, or "flute" patterns, while those in the other category showed a strong peak frequency with several resonance frequencies, or "string-instrument" patterns (Fig.10). Based on this categorization, when only "flute" pattern spectrograms were used to reconstruct sound signals, I got sound signals of only the flute part of the original music (see *Supplementary data*). On the other hand, when the other "string-instrument" pattern spectrograms were used for the reconstruction of the sound signals, I got sound signals without the "flute" part (Fig.10, see *Supplementary data*). This showed that source separation can be done based on the sound dictionaries NMF gave, and this is true for other cases such as sourse separation between trumpet and pipe organ[5], or between flute and harp[6] (data not shown).

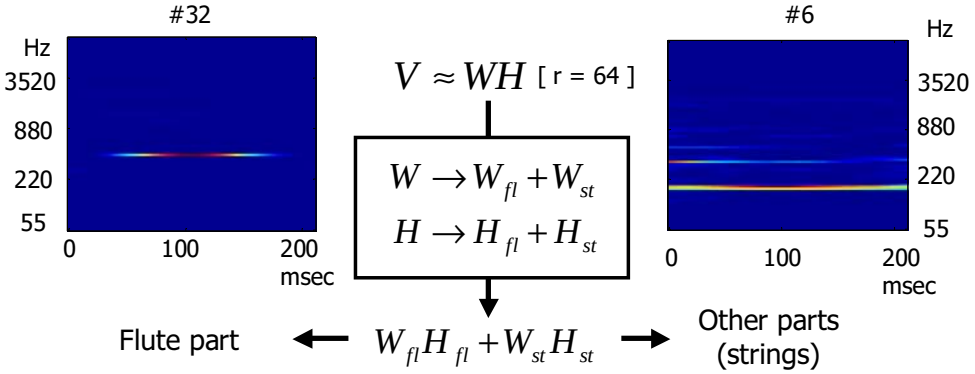

Figure 10: Sourse separation based on the basis elements obtained by NMF
The original music is "W.A.Mozart, Flute quartet in D, K.285 Rondeau ($\sim 1'39''$, 22.05 kHz)." The dataset consists of $m = 659$ log-scaled spectrograms ($n = 6615$: $\Delta f = 1/12$ octave, $\Delta t = 3$ msec) with the time range of 210 msec. The basis spectrograms W obtained by NMF can be categorized into two (W_{fl} and W_{st}) by hand based on the pattern of peak frequencies. The spectrogram #32 is one of the "flute" pattern basis spectrograms W_{fl}, and the spectrogram #6 is one of the "string-snstrument" pattern basis spectrograms W_{st}. Source separation can be done based on the categorization ($V_{fl} = W_{fl}H_{fl}$ versus $V_{st} = W_{st}H_{st}$).

[5] J.S.Bach, Cantata "Jesu, joy of Man's Desiring" BWV147 ($\sim 3'45''$, 16 kHz)
$\Longrightarrow$ Log-scaled spectrogram data sets $V[n = 6090 \times m = 1427]$, the rank of factorization $r = 64$
[6] W.A.Mozart, Concerto for flute and harp in C, K.299:3 Rondeau. Allegro ($\sim 8'49''$, 16 kHz)
$\Longrightarrow$ Log-scaled spectrogram data sets $V[n = 6090 \times m = 3527]$, the rank of ractorization $r = 144$

4 Discussion

I have described some characteristics of NMF compared to PCA and ICA. I show how NMF would break down a set of spectrograms into basis elements, and also show that the basis elements of sound signals can be used for source separation of the original music pieces.

This work would contribute to solving monaural source separation problem theoretically. The algorithm for this problem proposed by B.A.Pearlmutter and A.M.Zador requires two kinds of prior knowledge, HRTF and sound dictionaries (ref.[1]). HRTF itself can be numerically obtained by careful measurements[7], and biologically speaking, organisms would learn their own HRTF by experience with the aid of visual cues. Therefore, the assumption that we know HRTF is quite reasonable. On the other hand, although we can tell piano or flute sounds as "piano" or "flute" for example, it has not been addressed how we can acquire this knowledge of the sound dictionaries. I here used NMF to learn representative subsets of spectrograms for each instrument, and showed it is possible to do source separation based on the bases obtained by NMF. Because this result suggests that NMF would be a possible way to learn sound dictionaries, it would be very interesting to use these basis sound elements as one of the prior knowledge for the monaural source separation algorithm.

In this study, I used music pieces as sound signals because I could easily expect that the elements of music would be notes. As I expected, the basis elements obtained by NMF showed different notes for each instrument, and they formed a harmonics of the instruments on the whole. Now that we know NMF works on music, I think it is very interesting to apply NMF to natural or vocal sounds to see what basis sound elements of nature or languages are. In addition, to test the biological meanings of this algorithm, it would be intriguing to record neural responses against the basis elements of sound signals. In the visual system, it is well-known that rod and cones detect light with specific wavelength while neural responses at the visual cortex depend on the orientation of lines and edges, and their movements in specific directions (ref.[6]). But the "receptive fields" of auditory system has not been addressed very well. Considering that the sound dictionaries are obtained by a learning process called NMF, and considering that our responses to familiar sounds are usually better than those to unfamiliar ones, I think it is possible that sound dictionaries obtained by NMF form "receptive fields" at some level of the auditory system.

4.1 The rank of factorization for NMF

A big problem for NMF is that the rank of factorization r is an arbitrary number, which is a classic problem on a "modeling" in general. Although the rank of factorization would be critical for the results as I showed here, there seems to be no good criterion how to choose an appropriate value for r. To compress a data matrix $V[n \times m]$ into $W[m \times r] \cdot H[r \times n]$, r should satisfy the inequality $nm > mr + rn$. This is the only constraint for r, but unfortunately, this inequality would not be very helpful to determine an appropriate value for r.

In the case of music, because I expected that the sound dictionary would be notes of each instrument, I could speculate the value for r, which worked quite well. But in other cases such as natural sounds, it would be required to do cross-validation, using AIC (Akaike's information criterion) for an error criterion for example.

[7]For example, see the next web site
URL: http://interface.cipic.ucdavis.edu/CIL_html/CIL_HRTF_database.htm

Acknowledgements

I thank Anthony Zador, Barak Pearlmutter, and all the members in Zador laboratory for helpful discussion. I thank Christian Mathens for allowing me to use his Matlab code for calculating logarithmic-scale spectrograms. This work is supported by Farish-Gerry Fellowship for Watson School of Biological Sciences.

References

[1] B.A.Pearlmutter and A.M.Zador. Monaural source separation using spectral cues. *Submitted*.

[2] D.D.Lee and H.S.Seung. Learning the parts of objects by non-negative matrix factorization. Nature, 401:788–791, 1999.

[3] D.D.Lee and H.S.Seung. Algorithms for non-nagetive matrix factorization. Adv. Neural Info. Proc. Syst., 13:556–562, 2001.

[4] T.Hastie, R.Tibshirani, and J.Friedman. The Elements of Statistical Learning. Springer, 2001.

[5] A.Hyvarinen and E.Oja. A fast fixed-point algorithm for independent component analysis. Neural Comp., 9:1483–1492, 1997.

[6] T.N.Wiesel and D.H.Hubel. Spatial and chromatic interactions in the lateral geniculate body of the rhesus monkey. J. Neurophysiol., 29:1115–1156, 1966.

Supplementary data: list of WAVE files

This is a list of some example WAVE files for the NMF analyses.

Solo music (cello) J.S.Bach, cello suite No.1 Prelude ($\sim 2'27''$, 22.05 kHz)

- `cello-ori.wav` : A part of original sound signals
- `cello-sp.wav` : Sound signals for the dataset V (before NMF)
- `cello-nmf.wav` : Sound signals for the compressed data WH (after NMF)
- `cello12.wav` : Sound signals for the basis spectrogram #12
- `cello25.wav` : Sound signals for the basis spectrogram #25
- `cello-dic.wav` : Sound signals for all the basis spectrograms

Quartet W.A.Mozart, Flute quartet in D, K.285 Rondeau ($\sim 1'39''$, 22.05 kHz)

- `285-ori.wav` : A part of original sound signals
- `285-sp.wav` : Sound signals for the dataset V (before NMF)
- `285-nmf.wav` : Sound signals for the compressed data WH (after NMF)
- `285-6.wav` : Sound signals for the basis spectrogram #6
- `285-32.wav` : Sound signals for the basis spectrogram #32
- `285-dic.wav` : Sound signals for all the basis spectrograms
- `285-fl.wav` : Sound signals for only "flute" part $V_{fl} = W_{fl}H_{fl}$
- `285-st.wav` : Sound signals for the other string parts $V_{st} = W_{st}H_{st}$