

Chapter 2

Dependent Component Analysis Exploiting Nonnegativity and/or Time-Domain Sparsity

Abstract It is well-known that many real-world signals are nonnegative [1–8], i.e., their sample values are either zero or greater than zero, such as images. Obviously, nonnegativity is different from the statistical information of sources. Depending on the kinds of dependent sources, the nonnegativity of the source signals could be exploited to carry out dependent component analysis (DCA), i.e., separate these unknown dependent sources from their observed mixtures. If the sources also have certain level of sparsity in time domain, then the nonnegativity and time-domain sparsity of the source signals can be jointly employed to achieve DCA. In this chapter, three classes of dependent component analysis methods are introduced and analyzed, which are the nonnegative sparse representation (NSR) based methods, the convex geometry analysis (CGA) based methods, and the nonnegative matrix factorization (NMF) based methods. These methods either exploit the nonnegativity of the sources or both the nonnegativity and time-domain sparsity of the sources.

Keywords Nonnegative matrix factorization · Sparse representation · Convex geometry analysis

2.1 Nonnegative Sparse Representation Based Methods

Nonnegativity and sparsity constraints appear in various signal decomposition problems. For instance, in image processing, nonnegative sparse decomposition is related to the extraction of relevant parts from the images whose variables and parameters correspond to pixels [4]; in machine learning, sparseness is closely related to feature selection in learning algorithms, while nonnegativity relates to probability distributions [1]; in environmental science, scientists investigate a relative proportion of different pollutants in water or air, where proportion coefficients are nonnegative and the distributions of pollutants are often sparse [1]. Thus, it is a natural choice of applying NSR to DCA. We start from investigating the sparsity measures for nonnegative signals.

2.1.1 Sparsity Measures for Nonnegative Signals

A number of functions have been designed to measure the sparsity of signals. Some of them are suitable for the measurement of sparsity of a single signal and the others are for measuring the sparsity of a group of signals. For a single nonnegative signal \mathbf{x} , $x_i \geq 0$, $\forall i$ with n samples, one often utilizes the classic L_0 -norm like [9], L_p ($0 < p < 1$)-norm like [10], and L_1 -norm [11] based measures. The L_0 -norm like based measure is expressed as

$$S_{\mathbf{x}} = \#\{i | x_i \neq 0\} \quad (2.1)$$

where $\#\{i\}$ denotes the number of i . Although this measure is traditional in many mathematical settings, it is not suitable for many practical scenarios. One obvious reason is that its robustness against noise is poor. Furthermore, its derivative is zero containing no information. Thus, to find the sparsest solution, one has to employ the exhaustive search approach. This is inconvenient and costly, especially when solving large scale problems. In practice, the L_p ($0 < p < 1$)-norm like or L_1 -norm based measures are often used to approximate it.

The L_p ($0 < p < 1$)-norm like based measure is as follows:

$$S_{\mathbf{x}} = \left(\sum_{i=1}^n x_i^p \right)^{\frac{1}{p}}. \quad (2.2)$$

This measure is a good approximation of the L_0 -norm like counterpart, based on which, less observations are required to separate the sources. The L_p -norm like based measure is also widely used for signal reconstruction in the area of compressed sensing which aims to recover the original high dimensional signal from its low dimensional measurements [10].

The L_1 -norm based measure is defined as

$$S_{\mathbf{x}} = \sum_{i=1}^n x_i. \quad (2.3)$$

In some settings, the L_1 solution can be used to find the support of the L_0 solution. Besides, the L_1 solution can be found efficiently via linear programming (LP). As a result, it is widely used to replace the L_0 based complex problems. Figure 2.1 gives a simple comparison of L_0 , L_p ($p = 0.5$), L_1 and L_2 function curves.

The above measures are very intuitive and widely used in different research areas, such as blind source separation, compressed sensing, pattern recognition, machine learning and so on. However, they are not scaled, and the corresponding quantities do not contain meaningful information. Hence, it is not convenient to use them to compare the sparsity of different signals. Concerning this problem, Hoyer develops the L_1 -norm and L_2 -norm based measure which is scaled from zero to one [12], and

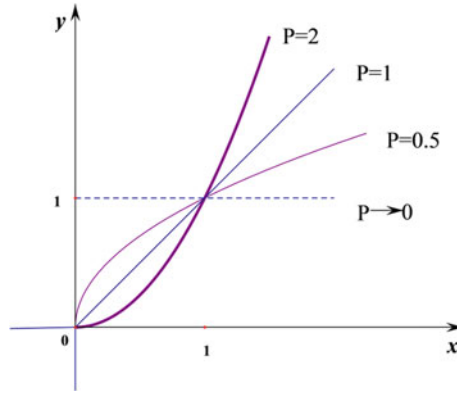


Fig. 2.1 $y = ||x||_p^p$ with different p

Yang et al. propose the following higher-order statistics based measure [4], whose value is also normalized to be in $[0,1]$:

$$S_{\mathbf{x}} = \frac{f_{\max} - (k_4 - \sigma_1 k_1^2 k_2 + \sigma_2 k_1 k_3)}{f_{\max} - f_{\min}} \quad (2.4)$$

where $\sigma_1 > 0$ and $\sigma_2 = (2\sigma_1 - 4)/3$ are two bounded constants, and $f_{\min} = (1 - \sigma_1 + \sigma_2)k_1^4$, $f_{\max} = ((1/n^3) - (\sigma_1/n) + (\sigma_2/n^2))k_1^4$, and $k_i = \|\mathbf{x}\|_i^i$, $i = 1, 2, 3, 4$. Here, n is the number of samples. In the case of $\sigma_1 = 2$, it is easy to derive that $\sigma_2 = 0$, $f_{\min} = -k_1^4$ and $f_{\max} = (1/n^3 - 2/n)k_1^4$. Then it results from (2.4) that

$$S_{\mathbf{x}} = \frac{(\frac{1}{n^3} - \frac{2}{n})k_1^4 - k_4 + 2k_1^2 k_2}{(\frac{1}{n^3} - \frac{2}{n} + 1)k_1^4}. \quad (2.5)$$

For the purpose of visual comparison, we use the statistics based sparsity measure in [4] to compute the $S_{\mathbf{x}}$ values of three signals with different sparsity and the result is shown in Fig. 2.2. It can be seen that the sparsity measure in [4] matches well with the real sparsity of these signals.

As for the sparsity measure of a group of m -dimensional nonnegative signals $\mathbf{X} = [\mathbf{x}_1^T, \dots, \mathbf{x}_r^T]^T$, a simple scheme is to first vectorize them and then use some existing sparsity measure. For example, if we use Hoyer's approach in [12], the sparsity measure for \mathbf{X} can be described as

$$S_{\mathbf{X}} = \frac{\sqrt{mr} - \left(\sum_{i=1}^m \sum_{j=1}^r x_{ij} \right) / \left(\sqrt{\sum_{i=1}^m \sum_{j=1}^r x_{ij}^2} \right)}{\sqrt{mr} - 1}. \quad (2.6)$$

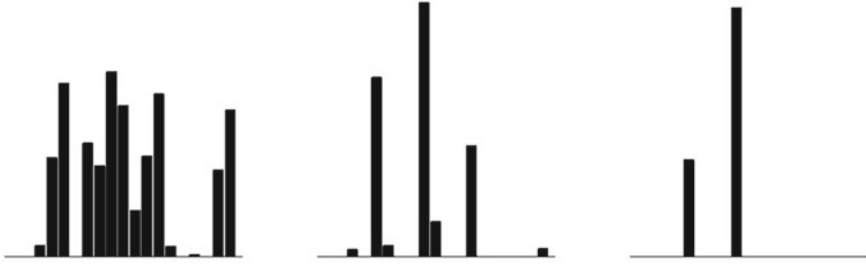


Fig. 2.2 Illustration of various degrees of sparseness. According to (2.5), the $S_{\mathbf{X}}$ values corresponding to the three signals (from left to right) are 0.1, 0.5, 0.9, respectively



Fig. 2.3 Illustration of different degrees of sparseness. According to (2.7), the $S_{\mathbf{X}}$ values corresponding to the three 2-D signals (from left to right) are 0.1, 0.5, 0.9, respectively

The larger the index $S_{\mathbf{X}}$, the sparser the matrix \mathbf{X} . In the case that each row of \mathbf{X} satisfies sum-to-one, one can also use the following determinant based measure [13]:

$$S_{\mathbf{X}} = \det(\mathbf{X}\mathbf{X}^T). \quad (2.7)$$

Figure 2.3 gives an illustration of three 2-D signals with different levels of sparsity measured by (2.7).

There also exist some other useful sparsity measures, such as the L_0^ε measure [14], the $\tanh_{a,b}$ measure [15], the log measure [16], the kurtosis k_4 measure [17], the Gaussian entropy diversity measure H_G , the Shannon entropy diversity measure H_S [18], the pq -mean measure [19], and the following Gini-curve based measure which is originally used to measure the inequality of wealth in economics [16]:

$$S_{\mathbf{x}} = 1 - 2 \sum_{i=1}^n \frac{x_i}{\|\mathbf{x}\|_1} \left(\frac{n-i+\frac{1}{2}}{n} \right) \quad (2.8)$$

Table 2.1 Commonly used sparsity measures

	Measure function
L_0	$-\#\{i x_i \neq 0\}$
L_0^ε	$-\#\{i x_i > \varepsilon\}$
$-L_1$	$-\sum_{i=1}^n x_i$
$-L_p$	$-(\sum_{i=1}^n x_i^p)^{\frac{1}{p}}, 0 < p < 1$
$-\frac{L_2}{L_1}$	$-\frac{\sqrt{\sum_{i=1}^n x_i^2}}{\sum_{i=1}^n x_i}$
$-\tanh_{a,b}$	$-\sum_{i=1}^n \tanh((ax_i)^b)$
$-\log$	$-\sum_{i=1}^n \log(1 + x_i^2)$
κ_4	$\frac{\sum_{i=1}^n x_i^4}{(\sum_{i=1}^n x_i^2)^2}$
H_G	$-\sum_{i=1}^n \log x_i^2$
H_S	$-\sum_{i=1}^n \tilde{x}_i \log \tilde{x}_i^2, \tilde{x}_i = \frac{x_i^2}{\ \mathbf{x}\ _2^2}$
Hoyer	$(\sqrt{x} - \frac{\sum_{i=1}^n x_i}{\sqrt{\sum_{i=1}^n x_i^2}})(\sqrt{n} - 1)^{-1}$
pq -mean	$-(\frac{1}{n} \sum_{i=1}^n x_i^p)^{\frac{1}{p}} (\frac{1}{n} \sum_{i=1}^n x_i^q)^{-\frac{1}{q}}, p < q$
Gini	$1 - 2 \sum_{i=1}^n \frac{x_{(i)}}{\ \mathbf{x}\ _1} (\frac{n-i+\frac{1}{2}}{n}), x_{(1)} \leq x_{(2)} \leq \dots \leq x_{(n)}$

where the elements of \mathbf{x} are with ascending order, i.e., $x_1 \leq x_2 \leq \dots \leq x_n$. Table 2.1 shows a list of commonly used sparsity measures, where the functions are modified such that larger measures correspond to sparser signals and some of them are also shown in [20].

2.1.2 Estimation of Mixing Matrix and Source Signals

Consider the mixing system model $\mathbf{Y} = \mathbf{A}\mathbf{X}$ shown in (1.6) and assume that the source signals \mathbf{X} are nonnegative and sparse. Based on this mixing system model, some NSR methods have been proposed for different scenarios, including the quadratic programming (QP) based method for the determined mixing system [13] and the

clustering based method for the underdetermined mixing case [21]. We will discuss the determined and underdetermined scenarios separately.

2.1.2.1 Determined Mixing System

In the determined mixing system, the number of the observations are equal or greater than that of the sources. Similar to the independence based method for BSS, one can implement DCA by finding a separation matrix \mathbf{B} such that the product \mathbf{BA} is a permutation matrix neglecting the inherent scaling issue. Since the sources are nonnegative and sparse, one can utilize the following optimization model, which exploits the source sparsity based on the determinant measure, to obtain the separation matrix [13]:

$$\text{Maximize : } \det(\mathbf{B}\mathbf{Y}\mathbf{Y}^T\mathbf{B}^T) \quad (2.9)$$

$$\text{s.t. } \begin{cases} \sum_{j=1}^m b_{ij}y_{jt} \geq 0, \forall i, t \\ \sum_{j=1}^m b_{ij} = 1, \forall i \end{cases}$$

where \mathbf{Y} is normalized to be row-sum-to-one in prior. The objective function reflects the sparsity of the estimated sources. Regarding the constraints, the first one denotes the nonnegativity and the second one is used to scale each estimated source to be sum-to-one.

In order to solve the model (2.9), the iterative sparseness maximization based on quadratic programming (ISM-QP) algorithm is developed in [13], where the separation matrix is optimized row-by-row iteratively. For the i th row $\bar{\mathbf{b}}_i$ of \mathbf{B} , the following optimization model is further derived:

$$\text{Maximize : } \bar{\mathbf{b}}_i\mathbf{C}\bar{\mathbf{b}}_i^T \quad (2.10)$$

$$\text{s.t. } \begin{cases} \sum_{j=1}^m b_{ij}y_{jt} \geq 0, \forall i, t \\ \sum_{j=1}^m b_{ij} = 1 \end{cases}$$

where \mathbf{C} is a matrix independent of $\bar{\mathbf{b}}_i$. Specifically,

$$\mathbf{C} = \mathbf{C}_1 + \mathbf{C}_2 + \mathbf{C}_3 \quad (2.11)$$

with

$$\begin{cases} \mathbf{C}_1 = \tilde{\mathbf{X}} \sum_{j=1}^{i-1} (-1)^{i+j} \tilde{\mathbf{b}}_j^T \left[\sum_{t=1}^{i-1} (-1)^{t+i-1} \det \left(\tilde{\mathbf{Y}}_{ij,t(i-1)} \right) \bar{\mathbf{b}}_t \right. \\ \quad \left. + \sum_{t=i}^{n-1} (-1)^{t+i-1} \det \left(\tilde{\mathbf{Y}}_{ij,t(i-1)} \right) \bar{\mathbf{b}}_{t+1} \right] \tilde{\mathbf{X}} \\ \mathbf{C}_2 = (-1)^{i+i} \det \left(\tilde{\mathbf{Y}}_{ii} \right) \tilde{\mathbf{X}} \\ \mathbf{C}_3 = \tilde{\mathbf{X}} \sum_{j=i+1}^n (-1)^{i+j} \tilde{\mathbf{b}}_j^T \left[\sum_{t=1}^{i-1} (-1)^{t+i} \det \left(\tilde{\mathbf{Y}}_{ij,ti} \right) \bar{\mathbf{b}}_t \right. \\ \quad \left. + \sum_{t=i}^{n-1} (-1)^{t+i} \det \left(\tilde{\mathbf{Y}}_{ij,ti} \right) \bar{\mathbf{b}}_{t+1} \right] \tilde{\mathbf{X}} \end{cases}$$

where $\tilde{\mathbf{X}} = \mathbf{Y}\mathbf{Y}^T$, $\tilde{\mathbf{Y}} = \mathbf{B}\mathbf{Y}\mathbf{Y}^T\mathbf{B}^T$, and $\tilde{\mathbf{Y}}_{ij}$ denotes a $(r-1) \times (r-1)$ submatrix of $\tilde{\mathbf{Y}}$ with the i th row and the j th column removed. By analyzing the inequalities in the constraints, the model (2.10) can be rewritten as

$$\text{Maximize : } \bar{\mathbf{b}}_i \mathbf{C} \bar{\mathbf{b}}_i^T \quad (2.12)$$

$$\text{s.t. } \begin{cases} \sum_{j=1}^m b_{ij} v_{jl} \geq 0, \forall l \in \{1, 2, \dots, L\} \\ \sum_{j=1}^m b_{ij} = 1 \end{cases}$$

where $\forall i, \mathbf{v}_i = \sum_{j=1}^m v_{ji}$ denotes the i th extreme point of the convex hull spanned by the observations.

The ISM-QP algorithm for solving DCA with r sources from the observation \mathbf{Y} is summarized in Table 2.2 below. Some highly correlated face images are used to test the effectiveness of this algorithm and Fig. 2.4 shows the separation results. We can see that the ISM-QP algorithm achieves almost perfect recoveries.

Table 2.2 ISM-QP algorithm [13]

Step 1 (Preprocessing)	Normalize each row of \mathbf{Y} to be sum-to-one and find the extreme points $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_L$ of the convex hull spanned by \mathbf{Y}
Step 2 (Initialization)	Let $i = 1$ and set an initial matrix with row-sum-to-one for \mathbf{B}
Step 3 (Iteration)	(i) Obtain the optimal solution $\bar{\mathbf{b}}_i^*$ of (2.12) (ii) Set $i = i + 1$ until a given stop criterion is satisfied (iii) If $i > r$, reset $i = \text{mod}(i, r)$, and if $i = 0$, set $i = r$
Step 4 (Estimation)	The source matrix is estimated by $\mathbf{B}^* \mathbf{Y}$, where $\mathbf{B}^* = [(\bar{\mathbf{b}}_1^*)^T, \dots, (\bar{\mathbf{b}}_r^*)^T]^T$

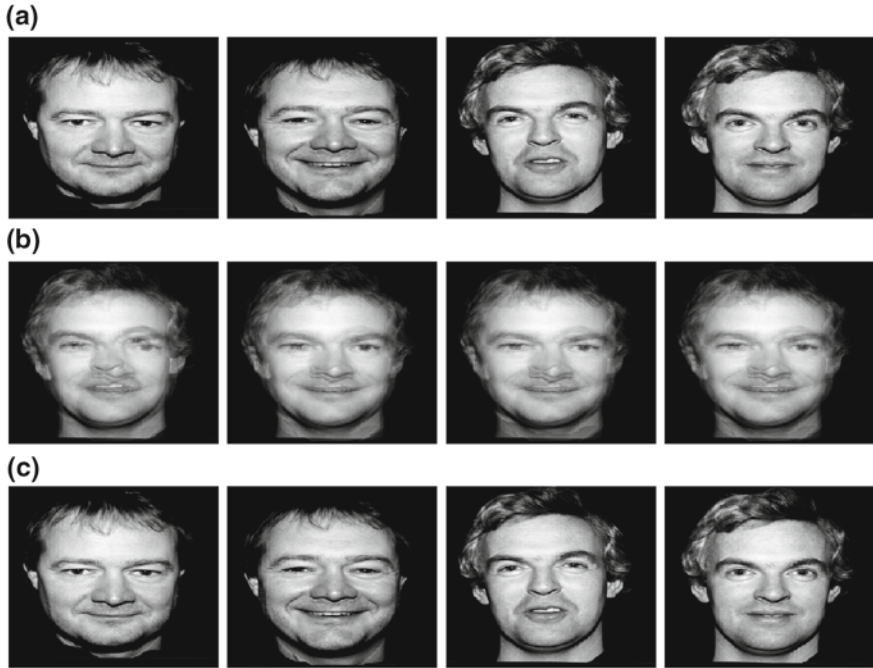


Fig. 2.4 Results of the ISM-QP algorithm in separating correlated face images. **a** Four correlated source images; **b** Four mixtures using random mixing matrix; **c** Four recoveries using ISM-QP

2.1.2.2 Underdetermined Mixing System

In the underdetermined mixing system scenario, the number of sources is greater than that of the observations. Since the mixing matrix is not invertible, it is impossible to implement DCA by searching a separation matrix. A feasible way of solving this challenging problem is to employ a two-step scheme: first estimate the mixing matrix \mathbf{A} and then recover the sources \mathbf{X} . Regarding the estimation of the mixing matrix, a popular approach is the clustering based one, which assumes that the sources are sparse [21]. After the mixing matrix is obtained, the estimation of the sources falls into the sparse reconstruction problem. In order to directly utilize the existing sparse representation algorithms, one can recover the source matrix column by column. In this case, the widely used L_1 -norm based method is a good option [11]. Furthermore, under some conditions, the subspace based scheme gives a more efficient way to conduct source recovery [21]. Table 2.3 shows the detailed structure of the subspace based algorithm.

Table 2.3 Estimating the source matrix \mathbf{X} [21]

Step 1 (Identification)	Calculate the set of k -codimensional subspaces \mathbb{H} produced by taking the linear hull of every subset of the columns of \mathbf{A} with $m - 1$ elements
Step 2 (Iteration)	For $i = 1, \dots, n$ (ii) identify the space $H \in \mathbb{H}$ containing \mathbf{y}_i , and project \mathbf{y}_i onto H to $\tilde{\mathbf{y}}_i$; (ii) if H is produced by the linear hull of column vectors $\mathbf{a}_{i_1}, \dots, \mathbf{a}_{i_{m-1}}$, then find coefficients λ_{ij} such that $\tilde{\mathbf{y}}_i = \sum_{j=1}^{m-1} \lambda_{ij} \mathbf{a}_{ij}$
Step 3 (Estimation)	The estimation of \mathbf{x}_i , $\forall i$ contains λ_{ij} in the place i_j for $j = 1, \dots, m - 1$, and its remaining components are zero

2.1.3 Uniqueness Conditions

Regarding the mixing model $\mathbf{Y} = \mathbf{A}\mathbf{X}$ with $\mathbf{A} \in \mathbb{R}^{m \times r}$ and $\mathbf{X} \in \mathbb{R}^{r \times n}$, the uniqueness conditions of NSR are related to both the mixing matrix \mathbf{A} and the source matrix \mathbf{X} . In the case that \mathbf{A} is determined, i.e., $m \geq r$, we have the following theorem:

Theorem 2.1 *If the mixing matrix \mathbf{A} is full column rank, the source matrix \mathbf{X} is nonnegative, and there exists an $r \times r$ submatrix $\hat{\mathbf{X}}$ satisfying $\det(\hat{\mathbf{X}}\hat{\mathbf{X}}^T) = 1$, where $\hat{\mathbf{X}}$ is normalized to be row-sum-to-one, then it holds that [13]*

$$\mathbf{B}^* \mathbf{A} = \mathbf{P} \quad (2.13)$$

where \mathbf{B}^* is the optimal solution of (2.9) and \mathbf{P} is a permutation matrix.

When the mixing system is underdetermined, i.e., $m < r$, the uniqueness analysis becomes much more complex. If \mathbf{A} is unknown, the following theorem gives the sufficient conditions:

Theorem 2.2 *Assume that $m \leq r \leq n$, any $m \times m$ square submatrix of $\mathbf{A} \in \mathbb{R}^{m \times r}$ is nonsingular, $\mathbf{X} \in \mathbb{R}^{r \times n}$ is sufficiently rich and its each column has at most $m - 1$ nonzero elements, then the matrix $\mathbf{Y} \in \mathbb{R}^{m \times n}$ can be represented uniquely in the form $\mathbf{Y} = \mathbf{A}\mathbf{X}$ if the following conditions are satisfied [21]:*

- (i) *the columns of \mathbf{Y} lie in the union \mathbb{H} of C_r^{m-1} different hyperplanes, each column lies in only one such hyperplane, each hyperplane contains at least m columns of \mathbf{Y} such that each $m - 1$ of them are linearly independent;*
- (ii) *for each $i \in 1, \dots, r$, there exist $p = C_{r-1}^{m-2}$ different hyperplanes $\{H_{i,j}\}_{j=1}^p$ in \mathbb{H} such that their intersection $L_i = \cap_{j=1}^p \{H_{i,j}\}$ is 1-D subspace;*
- (iii) *any m different L_i span the whole \mathbb{R}^m .*

Sometimes, the mixing matrix is known or can be calculated by some methods. In this case, the uniqueness of NSR is related to the uniqueness of nonnegative solution

in an underdetermined system. We have the following theorem:

Theorem 2.3 *Given that \mathbf{y} and $\mathbf{A} \in \mathbb{R}^{m \times r}$ (where $m < r$) for the system $\mathbf{y} = \mathbf{A}\mathbf{x}$ (where $\mathbf{x} \geq \mathbf{0}$) with finite solutions. Let $\hat{\mathbf{x}}$ be a solution to this problem, it is the unique solution if $\hat{\mathbf{x}}$ satisfies [22]*

$$\|\hat{\mathbf{x}}\|_0 < \frac{1}{2t_{\mathbf{A}}} \quad (2.14)$$

where $\|\hat{\mathbf{x}}\|_0$ denotes the number of the non-zero element of $\hat{\mathbf{x}}$, $t_{\mathbf{A}} = \rho(\mathbf{A})/(1 + \rho(\mathbf{A}))$ and $\rho(\mathbf{A})$ denotes the one-sided coherence which is defined as

$$\rho(\mathbf{A}) = \max_{i,j:j \neq i} \frac{|\mathbf{a}_i^T \mathbf{a}_j|}{\|\mathbf{a}_i\|_2^2}. \quad (2.15)$$

From the viewpoint of blind source separation, the above theorem shows the condition of uniquely recovering one column of \mathbf{X} . Also, it can be easily extended to the following corollary:

Corollary 2.1 *Given that $\mathbf{Y} \in \mathbb{R}^{m \times n}$ and $\mathbf{A} \in \mathbb{R}^{m \times r}$ with $m < r$. If $\forall i \in \{1, \dots, n\}$, the solution $\hat{\mathbf{x}}_i$ of $\mathbf{y}_i = \mathbf{A}\mathbf{x}_i$ satisfies*

$$\|\hat{\mathbf{x}}_i\|_0 < \frac{1}{2t_{\mathbf{A}}} \quad (2.16)$$

then $\hat{\mathbf{X}} = [\hat{\mathbf{x}}_1, \dots, \hat{\mathbf{x}}_n]$ is the unique solution of $\mathbf{Y} = \mathbf{A}\mathbf{X}$.

There are more uniqueness results related to the NSR problem. They range from the analysis of the nonnegative solutions to the underdetermined linear equations, including the restricted isometry property related conditions [23], the k-neighborly features [24], etc.

2.2 Convex Geometry Analysis Based Methods

In the context of nonnegative sources, there might be some geometric structures in the observations and the sources. For example, the biomedical image and human portraits are often with *local dominance* feature, under which the source signals correspond to the extreme points of some observation-constructed convex polyhedral set [5]; the hyper-spectral image abundance (or source) matrix has column-sum-to-one feature, such that it corresponds to the minimum volume simplex among those enclose of the observed data [6]; and the dynamic positron emission tomography images and the mass spectra for metabolomics are often the minimum aperture simplicial convex cones which contain their respective mixtures [7]. The use of these geometric features, instead of the statistical features of the sources, can facilitate the blind separation of mutually correlated sources.

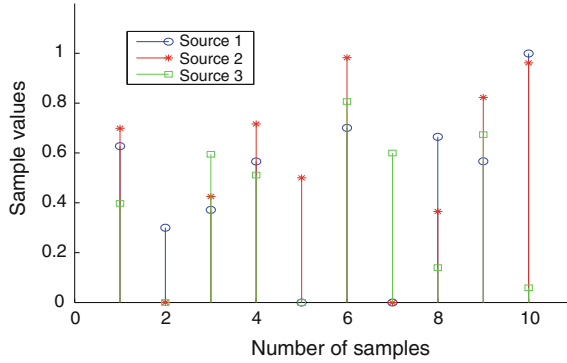


Fig. 2.5 Illustration of three sources with local dominance feature, where the dominant indices are 2, 5, 7 respectively

2.2.1 Geometric Features

Local dominance, also called pure source sample in [25], is an important geometric feature existing in some sources. It means that for each source there is at least one time instant at which the source dominates. A mathematical definition of local dominance is as follows:

Definition 2.1 (*Local dominance*): A group of sources $\bar{\mathbf{x}}_1^T, \dots, \bar{\mathbf{x}}_r^T$ have local dominance feature if for each $i \in \{1, \dots, r\}$, there exists an index l_i such that $\bar{\mathbf{x}}_i(l_i) > 0$ and $\bar{\mathbf{x}}_j(l_i) = 0, \forall j \neq i$.

Figure 2.5 gives an illustration of three sources with local dominance feature, where the dominant indices for the three sources are 2, 5, 7 respectively.

The local dominance feature may be completely satisfied or serve as a good approximation when the source signals are sparse (or contain many zeros). For example, in brain magnetic resonance imaging (MRI), the nonoverlapping region of the spatial distribution of fast perfusion and slow perfusion source images can be larger than 95 % [27]. In the hyperspectral unmixing problem, the abundances of the ground covers are often quite sparse, and thus the source images (corresponding to the abundances) tend to satisfy local dominance [26]. It may also be appropriate to consider this feature when the source signals exhibit high contrast, which could exist in sources such as face images and natural images. The local dominance feature is widely applied to solving the BSS problem [5, 28].

Another geometric feature is the minimum cone feature related to the mixing matrix [25, 29, 30]. From the mixing model $\mathbf{Y} = \mathbf{A}\mathbf{X}$ with $\mathbf{X} \geq \mathbf{0}$, we can see that each column of \mathbf{Y} is the nonnegative linear combination of the columns of \mathbf{A} . This implies that the cones which enclose \mathbf{Y} are related to \mathbf{A} . Specifically, it is found that the vertices of the simplicial cone and convex hull (defined below) with minimum volume correspond to the columns of \mathbf{A} under some conditions.

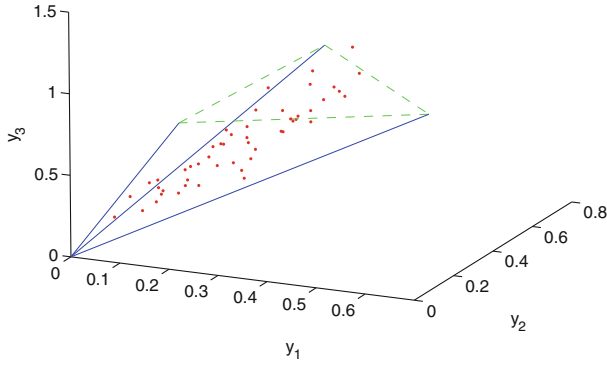


Fig. 2.6 Scatter plot of mixed data included in $\mathbf{Span}^+(\mathbf{A})$

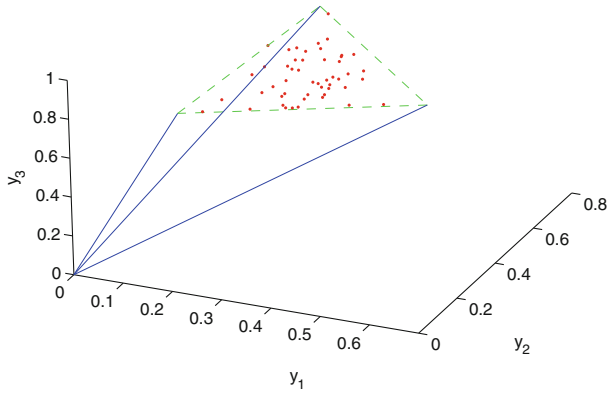


Fig. 2.7 Scatter plot of mixed data included in $\mathbf{Conv}^+(\mathbf{A})$

Definition 2.2 (*Simplicial cone*): The simplicial cone generated by the columns of \mathbf{A} , denoted by $\mathbf{Span}^+(\mathbf{A})$, is defined as [7]:

$$\mathbf{Span}^+(\mathbf{A}) = \{\mathbf{y} | \mathbf{y} = \mathbf{A}\mathbf{x} \text{ with } \mathbf{x} \in \mathbb{R}_+^r\}. \quad (2.17)$$

Definition 2.3 (*Convex hull*): The convex hull generated by the columns of \mathbf{A} , denoted by $\mathbf{Conv}^+(\mathbf{A})$, is defined as [6]:

$$\mathbf{Conv}^+(\mathbf{A}) = \{\mathbf{y} | \mathbf{y} = \mathbf{A}\mathbf{x}, \sum_{i=1}^r \mathbf{x}_i = 1, \mathbf{x}_i \in \mathbb{R}_+^r\}. \quad (2.18)$$

Figures 2.6 and 2.7 illustrate respectively $\mathbf{Span}^+(\mathbf{A})$ and $\mathbf{Conv}^+(\mathbf{A})$ in the case $r = 3$, where \mathbf{A} is randomly generated as

$$\mathbf{A} = \begin{bmatrix} 0.6493 & 0.1765 & 0.2609 \\ 0.2088 & 0.1159 & 0.6469 \\ 0.9641 & 0.8015 & 0.9105 \end{bmatrix}.$$

2.2.2 Estimation of Source Signals

There are many CGA based methods for DCA, which explicitly exploit the local dominance feature of the sources, including the convex analysis of mixtures of non-negative sources using linear programming (CAMNS-LP) [5], the project pursuit (PP) [28], the vertex component analysis (VCA) [26] and the modified VCA [31]. Here, we introduce the first two algorithms for reference. Regarding the CAMNS-LP method, it combines the convex analysis and optimization techniques. In this method, one first finds, through the convex analysis, the true source signals which serve as the extreme points of some observation-constructed polyhedral set. Then, an extreme-point finding algorithm is developed, by taking advantage of the powerful tool of linear programming, for source recovery. Actually, for the given mixture \mathbf{Y} , it first calculates the 2-tuple (\mathbf{C}, \mathbf{d}) as follows:

$$\begin{cases} \mathbf{d} = \frac{1}{m} \sum_{i=1}^m \bar{\mathbf{y}}_i^T \\ \mathbf{C} = [\mathbf{q}_1(\mathbf{U}\mathbf{U}^T), \mathbf{q}_2(\mathbf{U}\mathbf{U}^T), \dots, \mathbf{q}_{r-1}(\mathbf{U}\mathbf{U}^T)] \end{cases} \quad (2.19)$$

where $\mathbf{U} = [\bar{\mathbf{y}}_1^T - \mathbf{d}, \dots, \bar{\mathbf{y}}_m^T - \mathbf{d}] \in \mathbb{R}^{n \times m}$, the notation $\mathbf{q}_i(\mathbf{A})$ denotes the eigenvector associated with the i th principal eigenvalue of the mixing matrix \mathbf{A} , and $\bar{\mathbf{y}}_i$ is the i th row of the mixture matrix \mathbf{Y} . This 2-tuple constructs a meaningful affine hull. Built upon this 2-tuple, a series of LP problems are constructed for separating the sources. Then, the CAMNS-LP method recovers the sources iteratively, where only solvable LP problems need to be processed in each iteration. Table 2.4 shows the structure of the CAMNS-LP algorithm.

As for the PP method, it first maps the observation matrix into a superplane such that one of the rows of the mapped observation matrix has equal elements with value 1. This ensures that the unaccessible source matrix is normalized to be column-sum-to-one. Then, based on the property of the normalized source matrix, it estimates one column of the mixing matrix \mathbf{A} by searching an optimal projection vector for the mapped observation matrix. After that, it estimates another column by searching another optimal vector in the subspace orthogonal to the already estimated columns. All columns of the mixing matrix \mathbf{A} can be obtained by repeating this process. The PP method works under the same conditions as those of the CAMNS-LP method but it has much less computational complexity as it only needs to solve one LP problem [28]. The PP algorithm is summarized in Table 2.5.

Table 2.4 CAMNS-LP algorithm [5]

Step 1 (Preprocessing)	Calculate the 2-tuple (\mathbf{C}, \mathbf{d}) of the given \mathbf{Y} by (2.19)
Step 2 (Initialization)	Set $l = 0$ and $\mathbf{B} = \mathbf{I}_n$, where \mathbf{I}_n is the $n \times n$ identity matrix
Step 3 (Iteration)	<p>While $l \leq r$,</p> <p>(i) let $\mathbf{h} = \mathbf{B}\mathbf{w}$, where $\mathbf{w} \sim N(0, 1)$ is a randomly generated vector;</p> <p>(ii) solve the LPs</p> $p^* = \min_{\alpha: \mathbf{C}\alpha + \mathbf{d} \geq \mathbf{0}} \mathbf{h}^T (\mathbf{C}\alpha + \mathbf{d})$ $q^* = \max_{\alpha: \mathbf{C}\alpha + \mathbf{d} \geq \mathbf{0}} \mathbf{h}^T (\mathbf{C}\alpha + \mathbf{d})$ <p>and obtain their optimal solutions, denoted by α_1^* and α_2^*, respectively;</p> <p>(iii) if $l = 0$, let</p> $\widehat{\mathbf{S}} = [\mathbf{C}\alpha_1^* + \mathbf{d}, \mathbf{C}\alpha_2^* + \mathbf{d}]$ <p>else, update $\widehat{\mathbf{S}}$ by</p> $\widehat{\mathbf{S}} := \begin{cases} [\widehat{\mathbf{S}}, \mathbf{C}\alpha_1^* + \mathbf{d}], & \text{if } p^* \neq 0; \\ [\widehat{\mathbf{S}}, \mathbf{C}\alpha_2^* + \mathbf{d}], & \text{if } q^* \neq 0; \end{cases}$ <p>(iv) update l to be the number of the columns of $\widehat{\mathbf{S}}$, and apply QR decomposition to $\widehat{\mathbf{S}}$ as</p> $\widehat{\mathbf{S}} = \mathbf{Q}_l \mathbf{R}_l$ <p>where $\mathbf{Q}_l \in \mathbb{R}^{n \times l}$ and $\mathbf{R}_l \in \mathbb{R}^{l \times l}$;</p> <p>(v) update \mathbf{B} by</p> $\mathbf{B} := \mathbf{I}_n - \mathbf{Q}_l \mathbf{Q}_l^T$
Step 4 (Estimating \mathbf{X})	Finally, the source matrix \mathbf{X} is estimated by $\widehat{\mathbf{X}} = \widehat{\mathbf{S}}^T$

To give a visual comparison of the performance of the PP and CAMNS-LP algorithms, we use them to test four correlated fingerprint images,¹ where the correlation coefficient matrix \mathbf{C} is:

$$\mathbf{C} = \begin{bmatrix} 1.0000 & 0.7908 & 0.5939 & 0.6965 \\ 0.7908 & 1.0000 & 0.6548 & 0.7712 \\ 0.5939 & 0.6548 & 1.0000 & 0.7767 \\ 0.6965 & 0.7712 & 0.7767 & 1.0000 \end{bmatrix}.$$

The mixing matrix \mathbf{A} is generated randomly as:

$$\mathbf{A} = \begin{bmatrix} 0.7814 & 0.4464 & 0.3072 & 0.3298 \\ 0.4157 & 0.5367 & 0.2705 & 0.3822 \\ 0.4703 & 0.7291 & 0.6629 & 0.4115 \\ 0.4970 & 0.3533 & 0.5180 & 0.9035 \end{bmatrix}.$$

Figure 2.8 shows the source images, the mixtures, and the recoveries by using the PP algorithm and the CAMNS-LP algorithm, respectively. It can be seen that both of

¹ See <http://biometrics.cse.msu.edu/fvc04db/index.html>.

Table 2.5 PP algorithm [28]

Step 1 (Preprocessing)	<p>(i) Obtain \mathbf{u} satisfying $\mathbf{u}^T \mathbf{y}_i > 0, \forall i$ and suppose $u_q \neq 0$</p> <p>(ii) Compute \mathbf{D} by $\mathbf{D} = \text{diag}(\mathbf{1}^T \oslash (\mathbf{u}^T \mathbf{Y}))$ where $\mathbf{1}$ is a all-1 column vector and \oslash denotes component-wise division</p> <p>(iii) Let $\tilde{\mathbf{I}}_m$ be the $m \times m$ identity matrix with the qth row replaced by \mathbf{u}^T. Map \mathbf{Y} into $\tilde{\mathbf{Y}}$ by $\tilde{\mathbf{Y}} = \tilde{\mathbf{I}}_m \mathbf{Y} \mathbf{D}$</p>
Step 2 (Estimating \mathbf{a}_1)	<p>(i) Set $\mathbf{v} = \mathbf{0}$ and generate randomly a full-rank square matrix \mathbf{B}</p> <p>(ii) Update \mathbf{v} using the scheme (related to \mathbf{B}) in [28] and estimate $\hat{\mathbf{a}}_1$ by $\hat{\mathbf{a}}_1 = \mathbf{y}_j$ where $j = \begin{cases} \arg \max(\mathbf{v}^T \tilde{\mathbf{Y}}), & \text{if } \max(\mathbf{b}_1^T \tilde{\mathbf{Y}}) > 0 \\ \arg \min(\mathbf{v}^T \tilde{\mathbf{Y}}), & \text{else} \end{cases}$</p>
Step 3 (Estimating $\mathbf{a}_2, \dots, \mathbf{a}_r$)	<p>For $k = 1, 2, \dots, r - 1$,</p> <p>(i) update $\hat{\mathbf{A}}_k$ and $\hat{\mathbf{A}}_k^\perp$ by $\begin{cases} \hat{\mathbf{A}}_k = [\hat{\mathbf{a}}_1, \dots, \hat{\mathbf{a}}_k] \\ \hat{\mathbf{A}}_k^\perp = (\mathbf{I}_r - \hat{\mathbf{A}}_k (\hat{\mathbf{A}}_k^T \hat{\mathbf{A}}_k)^{-1} \hat{\mathbf{A}}_k^T) \mathbf{H} \end{cases}$ where $\mathbf{H} \in \mathbb{R}^{r \times (r-k)}$ is a matrix of full column rank;</p> <p>(ii) update \mathbf{B} by $\begin{cases} \mathbf{B}(1:r, 1:r-k) = \hat{\mathbf{A}}_k^\perp \\ \mathbf{B}(1:r, r-k+1:r) = \hat{\mathbf{A}}_k \end{cases}$</p> <p>(iii) estimate $\hat{\mathbf{a}}_{k+1}$ using the method shown in Step 2.</p>
Step 4 (Estimating \mathbf{X})	<p>Let $\hat{\mathbf{A}}_r = [\hat{\mathbf{a}}_1, \hat{\mathbf{a}}_2, \dots, \hat{\mathbf{a}}_r]$, then the source matrix is estimated by $\hat{\mathbf{X}} = \hat{\mathbf{A}}_r^{-1} \mathbf{Y}$</p>

them achieve satisfactory separating results. The corresponding CPU running times for these two algorithms are 2.9172 and 40.0455 s, respectively, indicating that the PP algorithm is much faster than the CAMNS-LP method.

Also, there are some methods which do not require the local dominance condition, such as the minimum volume simplex (MVS) [6] and the simplicial cone shrinking algorithm (SCSA) [7]. The MVS algorithm assumes that the sources are column-sum-to-one, i.e., the full additivity. In the noiseless case, one can relax the full additivity assumption by normalizing each column of the data matrix to a unit sum. However, in the noisy case, enforcing this normalization may amplify noise and thus yield a bad estimation of the sources, especially if the number of sources is overestimated. Different from the MVS algorithm, SCSA estimates the mixing matrix and the sources by finding the minimum aperture simplicial cone (MASC) containing the scatter plot of the mixed data. It needs neither the local dominance condition nor the full additivity assumption, applicable to a wider range of applications. Generally,

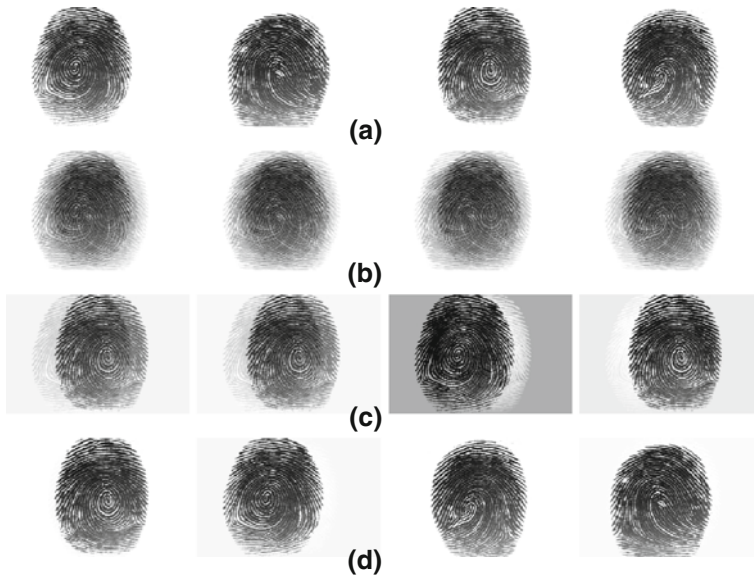


Fig. 2.8 Results of separating correlated fingerprint images by the PP and CAMNS-LP algorithms. **a** Four correlated source images; **b** Four mixtures; **c** Four recoveries using PP; **d** Four recoveries using CAMNS-LP

SCSA first finds a proper initial simplicial cone by using the VCA algorithm in [26], then decreases the aperture of the current simplicial cone iteratively. A summary of SCSA is shown in Table 2.6.

2.2.3 Source Identifiability Analysis

Regarding the source identifiability issue in relation to the mixing model $\mathbf{Y} = \mathbf{A}\mathbf{X}$, there are several conclusions shown in the following theorems.

Theorem 2.4 *Assuming that the sources are nonnegative with local dominance feature and the mixing matrix is full column rank with row-sum-to-one, the extreme points of the following polyhedral set correspond to the r true source vectors [5]:*

$$\{\mathbf{y} \in \mathbb{R}^n | \mathbf{y} = \mathbf{C}\boldsymbol{\alpha} + \mathbf{d} \geq \mathbf{0}, \boldsymbol{\alpha} \in \mathbb{R}^{r-1}\} \quad (2.20)$$

where (\mathbf{C}, \mathbf{d}) is obtained from \mathbf{Y} by (2.19).

This theorem shows how the local dominance feature affects the identification of the sources. If the mixing system satisfies the mentioned conditions, the sources can be recovered by searching the extreme points related to (2.20).

Table 2.6 A summary of SCSA [7]

Step 1 (Initialization)	Set $\mathbf{W} = \mathbf{I}_m$ or find it by using the VCA method and set $\mathbf{D} = \mathbf{W}^{-1}\mathbf{Y}$
Step 2 (Iteration)	<p>(i) $\forall i \in \{1, \dots, r\}$, compute \mathbf{V}_i by</p> $\mathbf{V}_i = \begin{bmatrix} 1 & 0 & \dots & 0 & v_{1i} & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 & v_{2i} & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & v_{(i-1)i} & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & v_{(i+1)i} & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & v_{mi} & 0 & \dots & 1 \end{bmatrix}$ <p>and update \mathbf{W}, \mathbf{D} by</p> $\begin{cases} \mathbf{W} = \mathbf{W}\mathbf{V}_1\mathbf{V}_2 \dots \mathbf{V}_r \\ \mathbf{D} = [\mathbf{R}_r]^{-1}[\mathbf{R}_{r-1}]^{-1} \dots [\mathbf{R}_1]^{-1}\mathbf{D} \end{cases}$ <p>(ii) Compute \mathbf{Q} by</p> $\mathbf{Q}_{(p+1)} = \mathbf{Q}_p - \mu \left[-\frac{(\mathbf{W}^{-1})^T \mathbf{T}^{null} \mathbf{T}^T}{\sigma} + 4\gamma \mathbf{Q}_p (\mathbf{Q}_p^T \mathbf{Q}_p - \mathbf{I}_r) \right]$ <p>where μ is a learning rate parameter, $\mathbf{T} = \mathbf{W}^{-1}\mathbf{Q}_p\mathbf{Y}$, $\mathbf{T}^{null} = \exp(-\mathbf{T}/\sigma)$, $\sigma > 0$, and $\gamma \geq 0$. And let</p> $\begin{cases} \mathbf{W} = \mathbf{Q}\mathbf{W} \\ \mathbf{D} = \mathbf{W}^{-1}\mathbf{Q}^{-1}\mathbf{Y} \end{cases}$ <p>(iii) If $\mathbf{Q} = \mathbf{I}_r$, stop the iteration</p>
Step 3 (Estimation)	\mathbf{A} and \mathbf{X} are estimated by using MATLAB functions $\max(\mathbf{W}, 0)$ and $\max(\mathbf{D}, 0)$, respectively

More recent results about source identification are given in [7] as follows:

Theorem 2.5 $\text{Span}^+(\mathbf{A})$ is the unique nonnegative MASC containing the scatter plot of the mixed data if and only if $\text{Span}^+(\mathbf{I}_r)$ is the unique nonnegative MASC containing the scatter plot of the sources, where \mathbf{I}_r denotes the $r \times r$ identity matrix.

Theorem 2.6 (Necessary condition) If $\text{Span}^+(\mathbf{I}_r)$ is the unique nonnegative MASC containing the scatter plot of the sources, then there is at least one point of the cloud of sources on each facet of $\text{Span}^+(\mathbf{I}_r)$, i.e., $\forall 1 \leq i \leq r$, $\exists k_i$ such that $x_i(k_i) = 0$.

Theorem 2.7 (Sufficient condition 1) If the sources are nonnegative and locally dominant, then $\text{Span}^+(\mathbf{I}_r)$ is the unique nonnegative MASC containing the scatter plot of sources.

Theorem 2.8 (Sufficient condition 2) For each facet of $\text{Span}^+(\mathbf{I}_r)$, if at least $r - 1$ points of the scatter plot of the sources belong to underlined facet, and the vectors corresponding to these points are linearly independent, then $\text{Span}^+(\mathbf{I}_r)$ is the unique nonnegative MASC containing the scatter plot of the sources.

2.3 Nonnegative Matrix Factorization Based Methods

Like the sources, sometimes the mixing matrix is also nonnegative. For example, in remote sensing image processing, both the endmember signature matrix and the abundance matrix are nonnegative [4]. In fluorescence spectroscopy analysis, the pure species spectra and their concentrations are also nonnegative [3]. More practical mixing systems with nonnegative sources and nonnegative mixing matrix can be found in [1]. Since NMF aims to decompose a given nonnegative matrix into the product of two nonnegative matrices [8], it matches well with the BSS problem, or DCA when the sources are spatially correlated. NMF is a well developed scheme which is widely used in the areas of signal processing and pattern recognition. Over the last few years, a number of NMF based methods have been proposed to implement DCA, such as NMF-MVC [32], NMF-L1 [33], and NMF-SMC [4]. Prior to discussing these methods, we first introduce some NMF models.

2.3.1 Nonnegative Matrix Factorization Models

Assume that \mathbf{Y} is a given nonnegative matrix, NMF aims to decompose \mathbf{Y} into the product of two nonnegative matrices, denoted by \mathbf{A} and \mathbf{X} , respectively. Mathematically, the standard NMF can be described as [1, 8]

$$\mathbf{Y} \approx \mathbf{AX} \quad (2.21)$$

where $\mathbf{Y} \in \mathbb{R}_+^{m \times n}$, $\mathbf{A} \in \mathbb{R}_+^{m \times r}$, $\mathbf{X} \in \mathbb{R}_+^{r \times n}$. Clearly, under the case of perfect decomposition, i.e., \mathbf{Y} is equal to \mathbf{AX} , NMF model is equivalent to the noiseless BSS mixing model. This motivates researchers to exploit NMF schemes to solve the BSS problem [1].

In order to achieve NMF, several useful cost or measure functions have been proposed for particular applications. Let $\hat{\mathbf{Y}}$ be the decomposition of \mathbf{Y} . We list three major cost functions here.

- The first function is the Euclidean distance based function [8]

$$D(\mathbf{Y} \parallel \hat{\mathbf{Y}}) = \frac{1}{2} \|\mathbf{Y} - \hat{\mathbf{Y}}\|_2^2 = \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^n (y_{ij} - \hat{y}_{ij})^2 \quad (2.22)$$

which measures the error of the given matrix and its decomposition. It is lower bounded by zero and vanishes if and only if $\mathbf{Y} = \hat{\mathbf{Y}}$.

- The second function is the Kullback-Leibler (KL) divergence based function [34]

$$D(\mathbf{Y} \parallel \hat{\mathbf{Y}}) = \sum_{i=1}^m \sum_{j=1}^n (y_{ij} \log \frac{y_{ij}}{\hat{y}_{ij}} - y_{ij} + \hat{y}_{ij}). \quad (2.23)$$

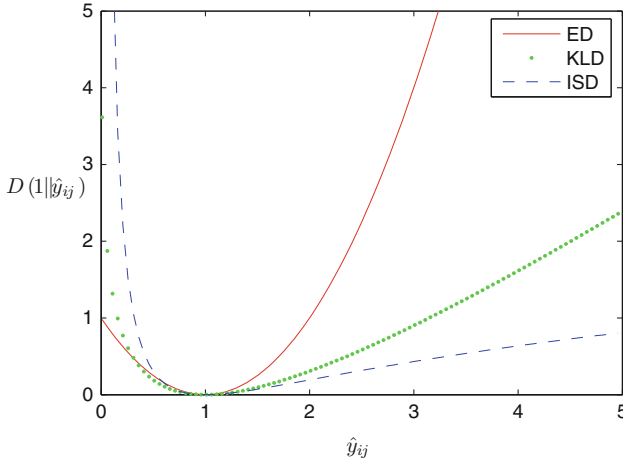


Fig. 2.9 Euclidean distance (ED), KL divergence (KLD) and IS divergence (ISD) versus \hat{y}_{ij} , where $y_{ij} = 1$ [35]

Similar to the Euclidean distance, this function is also lower bounded by zero and vanishes if and only if $\mathbf{Y} = \hat{\mathbf{Y}}$. Since it is not symmetric about \mathbf{Y} and $\hat{\mathbf{Y}}$, it is called divergence.

- The third function is the Itakura-Saito (IS) divergence based function [35]

$$D(\mathbf{Y} \parallel \hat{\mathbf{Y}}) = \sum_{i=1}^m \sum_{j=1}^n \left(\frac{y_{ij}}{\hat{y}_{ij}} - \log \frac{y_{ij}}{\hat{y}_{ij}} - 1 \right). \quad (2.24)$$

Clearly, the IS divergence depends only on the ratio $\frac{y_{ij}}{\hat{y}_{ij}}$. This property is favorable when analyzing most audio signals such as music and speech, where the low frequency components have much higher energy than high frequency components.

Figure 2.9 shows the Euclidean distance, KL divergence and IS divergence under different \hat{y}_{ij} varying from 0 to 5, where $y_{ij} = 1$. We can see that the KL and IS divergences are less sensitive to over-approximation than under-approximation.

There also exist some other divergence based cost functions, such as the α -divergence [36], β -divergence [37], $\alpha\beta$ -divergence [38], and f -divergence [36]. Table 2.7 shows the afore-mentioned distance and divergence based cost functions, which are used for NMF.

Based on the Euclidean distance, the NMF optimization model to (2.21) is

$$\text{Minimize : } D = \frac{1}{2} \|\mathbf{Y} - \mathbf{AX}\|_2^2 \quad (2.25)$$

s.t. $\mathbf{A} \geq \mathbf{0}$ and $\mathbf{X} \geq \mathbf{0}$, where \geq denotes the component-wise inequality. Since NMF does not necessarily generate a desired result, one often needs to add some constraints

Table 2.7 Distance and divergence based cost functions

	Cost function $D(\mathbf{Y} \parallel \hat{\mathbf{Y}})$
Euclidean distance	$\frac{1}{2} \sum_{i=1}^m \sum_{j=1}^n (y_{ij} - \hat{y}_{ij})^2$
KL divergence	$\sum_{i=1}^m \sum_{j=1}^n (y_{ij} \log \frac{y_{ij}}{\hat{y}_{ij}} - y_{ij} + \hat{y}_{ij})$
IS divergence	$\sum_{i=1}^m \sum_{j=1}^n (\frac{y_{ij}}{\hat{y}_{ij}} - \log \frac{y_{ij}}{\hat{y}_{ij}} - 1)$
β -divergence	$\sum_{i=1}^m \sum_{j=1}^n (\frac{y_{ij}^\beta}{\beta(\beta-1)} + \frac{\hat{y}_{ij}^\beta}{\beta} - \frac{y_{ij}\hat{y}_{ij}^{\beta-1}}{\beta-1}), \beta \in \mathbb{R}, \beta \neq 0, 1$
α -divergence	$\frac{1}{\alpha(1-\alpha)} \sum_{i=1}^m \sum_{j=1}^n (\alpha y_{ij} + (1-\alpha)\hat{y}_{ij} - y_{ij}^\alpha \hat{y}_{ij}^{1-\alpha})$
$\alpha\beta$ -divergence	$-\frac{1}{\alpha\beta} \sum_{i=1}^m \sum_{j=1}^n (y_{ij}^\alpha \hat{y}_{ij}^\beta - \frac{\alpha}{\alpha+\beta} y_{ij}^{\alpha+\beta} - \frac{\beta}{\alpha+\beta} \hat{y}_{ij}^{\alpha+\beta}), \alpha, \beta, \alpha+\beta \neq 0$
f-divergence	$\sum_{i=1}^m \sum_{j=1}^n (y_{ij} f(\frac{\hat{y}_{ij}}{y_{ij}}))$

(or regularization/penalty terms) into the model. A general constrained NMF model can be written as [39]

$$\begin{aligned} \text{Minimize : } D_J &= \frac{1}{2} \|\mathbf{Y} - \mathbf{A}\mathbf{X}\|_2^2 + \alpha J(\mathbf{A}) + \beta J(\mathbf{X}) \\ \text{s.t. } \mathbf{A} &\succeq \mathbf{0} \text{ and } \mathbf{X} \succeq \mathbf{0}. \end{aligned} \quad (2.26)$$

Dependent on the practical applications, different constraints could be considered. Some useful constraints are as follows.

- Volume based constraint on \mathbf{A} [29]:

$$J(\mathbf{A}) = \frac{1}{2(r-1)!} \det^2 \left([\mathbf{1} \ \tilde{\mathbf{A}}^T] \right). \quad (2.27)$$

Here, the matrix $\tilde{\mathbf{A}} \in \mathbb{R}^{(r-1) \times r}$ is calculated by

$$\tilde{\mathbf{A}} = \mathbf{U}^T (\mathbf{A} - \mu \mathbf{1}^T) \quad (2.28)$$

where $\mathbf{U} \in \mathbb{R}^{m \times (r-1)}$ is formed by the $r-1$ most significant components of \mathbf{Y} through principal component analysis and the column vector μ contains the means of the rows of \mathbf{Y} .

- Dispersion based constraint on \mathbf{A} [40]:

$$J(\mathbf{A}) = \text{Tr}(\mathbf{A}^T \mathbf{A}) - \frac{1}{m} \text{Tr}(\mathbf{A}^T \mathbf{E} \mathbf{A}) \quad (2.29)$$

where $\text{Tr}(\cdot)$ denotes the trace operator and \mathbf{E} stands for the $m \times m$ matrix whose entries are all one.

- Temporal continuity based constraint on \mathbf{X} [41]:

$$J(\mathbf{X}) = \sum_{j=1}^r \frac{1}{\sigma_j^2} \sum_{t=2}^n (x_{jt} - x_{j(t-1)})^2 \quad (2.30)$$

where $\sigma_j = \sqrt{(1/n) \sum_{t=1}^n x_{jt}^2}$ denotes the standard deviation of the j th component $\bar{\mathbf{x}}_j$ and $\bar{\mathbf{x}}_j$ is the j th row of \mathbf{X} .

- Dependence based constraint on \mathbf{X} [42]:

$$J(\mathbf{X}) = \frac{1}{2} \left[\sum_{i=1}^r \log((\mathbf{X}\mathbf{X}^T)_{ii}) - \log(\det(\mathbf{X}\mathbf{X}^T)) \right]. \quad (2.31)$$

2.3.2 Estimation of Mixing Matrix and Source Signals

To solve the model (2.25), one can utilize the scheme based on the alternatively iterative multiplication updating rule in [34], together with the classic gradient based tool. From (2.25), the partial derivatives of the cost function D with respect to \mathbf{X} and \mathbf{A} are

$$\begin{cases} \frac{\partial D}{\partial \mathbf{X}} = \mathbf{A}^T \mathbf{A} \mathbf{X} - \mathbf{A}^T \mathbf{Y} \\ \frac{\partial D}{\partial \mathbf{A}} = \mathbf{A} \mathbf{X} \mathbf{X}^T - \mathbf{Y} \mathbf{X}^T \end{cases} \quad (2.32)$$

According to the gradient based optimization rule, \mathbf{X} and \mathbf{A} can be updated by

$$\begin{cases} \mathbf{X} := \mathbf{X} - \eta_{\mathbf{X}} \otimes (\mathbf{A}^T \mathbf{A} \mathbf{X} - \mathbf{A}^T \mathbf{Y}) \\ \mathbf{A} := \mathbf{A} - \eta_{\mathbf{A}} \otimes (\mathbf{A} \mathbf{X} \mathbf{X}^T - \mathbf{Y} \mathbf{X}^T) \end{cases} \quad (2.33)$$

where \otimes denotes the component-wise multiplication. To keep the nonnegativity of \mathbf{X} and \mathbf{A} , the learning rates $\eta_{\mathbf{X}}$ and $\eta_{\mathbf{A}}$ are often chosen as

$$\begin{cases} \eta_{\mathbf{X}} = \frac{\mathbf{X}}{\mathbf{A}^T \mathbf{A} \mathbf{X}} \\ \eta_{\mathbf{A}} = \frac{\mathbf{A}}{\mathbf{A} \mathbf{X} \mathbf{X}^T} \end{cases} \quad (2.34)$$

Then, the corresponding iteration formulae for \mathbf{A} and \mathbf{X} are as follows:

$$\begin{cases} \mathbf{X} := \mathbf{X} \otimes \frac{\mathbf{A}^T \mathbf{Y}}{\mathbf{A}^T \mathbf{A} \mathbf{X}} \\ \mathbf{A} := \mathbf{A} \otimes \frac{\mathbf{Y} \mathbf{X}^T}{\mathbf{A} \mathbf{X} \mathbf{X}^T} \end{cases} \quad (2.35)$$

Table 2.8 NMF algorithm [34]

Step 1 (Initialization)	Randomly generate initial \mathbf{A} and \mathbf{X}
Step 2 (Iteration)	While a stop criterion is not satisfied, (i) update \mathbf{X} by $\mathbf{X} := \mathbf{X} \otimes \frac{\mathbf{A}^T \mathbf{Y}}{\mathbf{A}^T \mathbf{A} \mathbf{X}};$ (ii) update \mathbf{A} by $\begin{cases} \mathbf{A} := \mathbf{A} \otimes \frac{\mathbf{Y} \mathbf{X}^T}{\mathbf{A} \mathbf{X} \mathbf{X}^T} \\ \mathbf{A} := \mathbf{A} (\text{diag}(\mathbf{1} \oslash (\sum_{i=1}^m \bar{\mathbf{a}}_i)^T)) \end{cases}$
Step 3 (Estimation)	The final \mathbf{A} and \mathbf{X} are the estimates of the mixing matrix and the sources, respectively

Furthermore, to tackle the inevitable scaling issue, one often normalizes \mathbf{A} by

$$\mathbf{A} := \mathbf{A} (\text{diag}(\mathbf{1} \oslash (\sum_{i=1}^m \bar{\mathbf{a}}_i)^T)) \quad (2.36)$$

where \oslash denotes the component-wise division, $\bar{\mathbf{a}}_i$ is the i th row of \mathbf{A} , and $\text{diag}(\mathbf{x})$ denotes a diagonal matrix whose diagonal entries correspond to the elements of the vector \mathbf{x} . The complete NMF algorithm is shown in Table 2.8.

Regarding the algorithms concerning the constrained NMF model (2.26), they are related to the exact constraints on the sources or the mixing matrix. For unmixing the hyper-spectral data, a source-constrained NMF method is proposed in [4]. Figure 2.10 shows the results of using this method and the traditional Kruse's method² to separate some hyper-spectral images. We can see that both methods obtain meaningful separation results.

In the following, we will further introduce some recently developed algorithms which apply volume constraint on \mathbf{A} , including both batch mode and online mode. According to the analysis in [32], a new volume constraint on \mathbf{A} is $J_{\mathbf{A}} = \det(\mathbf{A}^T \mathbf{A})/2$. Then, (2.26) is simplified as

$$\begin{aligned} \text{Minimize : } D_J &= \frac{1}{2} \|\mathbf{Y} - \mathbf{A} \mathbf{X}\|_2^2 + \frac{\alpha}{2} \det(\mathbf{A}^T \mathbf{A}) \\ \text{s.t. } \mathbf{A} &\succeq \mathbf{0} \text{ and } \mathbf{X} \succeq \mathbf{0}. \end{aligned} \quad (2.37)$$

In this case, the derivatives of D_J with respect to \mathbf{X} and \mathbf{A} are

$$\begin{cases} \frac{\partial D_J}{\partial \mathbf{X}} = \mathbf{A}^T \mathbf{A} \mathbf{X} - \mathbf{A}^T \mathbf{Y} \\ \frac{\partial D_J}{\partial \mathbf{A}} = \mathbf{A} \mathbf{X} \mathbf{X}^T - \mathbf{Y} \mathbf{X}^T + \alpha (\det(\mathbf{A}^T \mathbf{A}) \mathbf{A} (\mathbf{A}^T \mathbf{A})^{-1}) \end{cases}. \quad (2.38)$$

² Available: <http://www.hgimaging.com/PDF/Kruse-JPL2002-AVIRIS-Hyperion.pdf>.

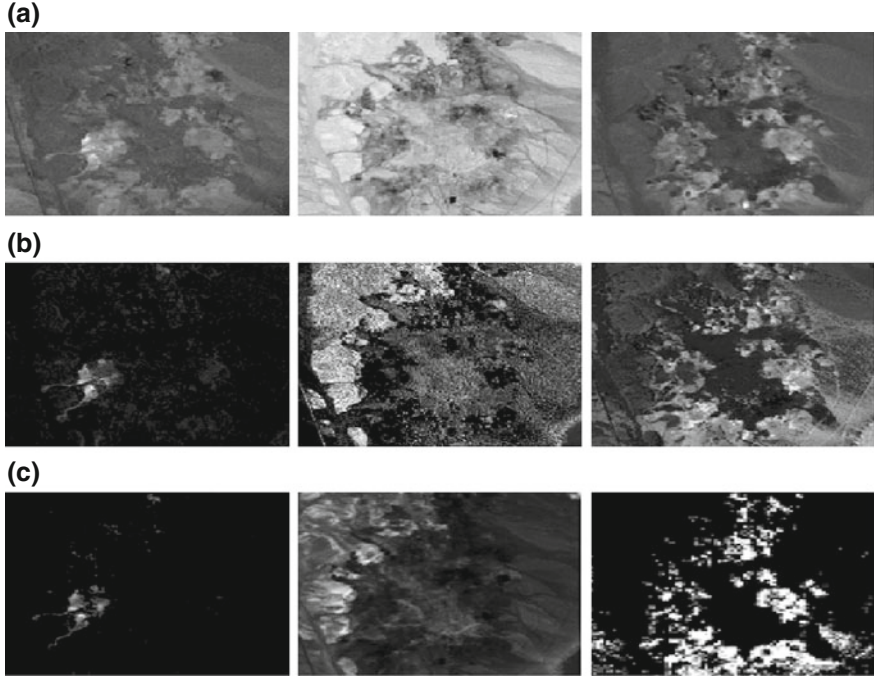


Fig. 2.10 Results of separating real-world hyper-spectral images by the constrained NMF algorithm and Kruse's algorithm. **a** Three source images; **b** Three recoveries using the constrained NMF; **c** Three recoveries using Kruse's method

A traditional gradient based method is utilized to update \mathbf{X} as follows:

$$\begin{aligned}
 \mathbf{X} &:= \mathbf{X} - \eta_{\mathbf{X}} \otimes \frac{\partial D_J}{\partial \mathbf{X}} \\
 &= \mathbf{X} - \eta_{\mathbf{X}} \otimes (\mathbf{A}^T \mathbf{A} \mathbf{X} - \mathbf{A}^T \mathbf{Y}) \\
 &= \mathbf{X} - \eta_{\mathbf{X}} \otimes (\mathbf{A}^T \mathbf{A} \mathbf{X} + \delta_{\mathbf{X}} - \mathbf{A}^T \mathbf{Y} - \delta_{\mathbf{X}}).
 \end{aligned} \tag{2.39}$$

Here, $\delta_{\mathbf{X}}$ denotes a matrix with the same size of \mathbf{X} and its entries all take the small positive value δ . It is used to avoid possible numerical instability. $\eta_{\mathbf{X}}$ denotes the learning rate. By setting $\eta_{\mathbf{X}} = \frac{\mathbf{X}}{\mathbf{A}^T \mathbf{A} \mathbf{X} + \delta_{\mathbf{X}}}$, then \mathbf{X} can be updated by

$$\mathbf{X} := \mathbf{X} \otimes \frac{\mathbf{A}^T \mathbf{Y} + \delta_{\mathbf{X}}}{\mathbf{A}^T \mathbf{A} \mathbf{X} + \delta_{\mathbf{X}}}. \tag{2.40}$$

Now we consider the update of \mathbf{A} . As shown in (2.38), the partial derivative $\partial D_J / \partial \mathbf{A}$ includes the computation of the inverse matrix of $\mathbf{A}^T \mathbf{A}$. This may break the nonnegativity of \mathbf{A} . To conquer this obstacle, the so-called natural gradient (NG)

is employed, which is widely discussed in [43, 44]. In fact, since \mathbf{A} is full column rank, there exists a matrix \mathbf{B} such that $\mathbf{BA} = \mathbf{I}_r$. Consequently, the parameter matrix \mathbf{A} has a special algebraic structure, namely Li group structure, making the variables therein like a curved Riemann manifold. It is known that the natural gradient, instead of the ordinary gradient, is the steepest descent direction in the Riemann manifold [43, 44]. Hence, the natural gradient is utilized to update \mathbf{A} as follows:

$$\begin{aligned}\mathbf{A} &:= \mathbf{A} - \eta_{\mathbf{A}} \otimes \left(\frac{\partial D_J}{\partial \mathbf{A}} \mathbf{A}^T \mathbf{A} \right) \\ &= \mathbf{A} - \eta_{\mathbf{A}} \otimes \left((\mathbf{A}\mathbf{X}\mathbf{X}^T - \mathbf{Y}\mathbf{X}^T + \alpha \det(\mathbf{A}^T \mathbf{A}) \mathbf{A} (\mathbf{A}^T \mathbf{A})^{-1}) \mathbf{A}^T \mathbf{A} \right) \\ &= \mathbf{A} - \eta_{\mathbf{A}} \otimes \left(\mathbf{A}\mathbf{X}\mathbf{X}^T \mathbf{A}^T \mathbf{A} + \alpha \det(\mathbf{A}^T \mathbf{A}) \mathbf{A} + \delta_{\mathbf{A}} - \mathbf{Y}\mathbf{X}^T \mathbf{A}^T \mathbf{A} - \delta_{\mathbf{A}} \right).\end{aligned}\quad (2.41)$$

To ensure the nonnegativity of \mathbf{A} , the learning rate $\eta_{\mathbf{A}}$ is chosen as

$$\eta_{\mathbf{A}} = \frac{\mathbf{A}}{\mathbf{A}\mathbf{X}\mathbf{X}^T \mathbf{A}^T \mathbf{A} + \alpha \det(\mathbf{A}^T \mathbf{A}) \mathbf{A} + \delta_{\mathbf{A}}}. \quad (2.42)$$

Thus, \mathbf{A} can be updated by

$$\mathbf{A} := \mathbf{A} \otimes \frac{\mathbf{Y}\mathbf{X}^T \mathbf{A}^T \mathbf{A} + \delta_{\mathbf{A}}}{\mathbf{A}\mathbf{X}\mathbf{X}^T \mathbf{A}^T \mathbf{A} + \alpha \det(\mathbf{A}^T \mathbf{A}) \mathbf{A} + \delta_{\mathbf{A}}}. \quad (2.43)$$

The NG based minimum volume constrained NMF (NG-MVC-NMF) algorithm [32] is summarized in Table 2.9.

In addition to the batch algorithm for the volume based NMF, the corresponding online learning version has also been developed. Compared with the batch mode which usually suffers from large storage requirement and high computational complexity when the observations are large scale, the online mode or incremental learning scheme is particularly appealing owing to its low computational cost. Table 2.10 shows the incremental NMF with volume constraint (INMF-VC) [30].

Table 2.9 NG-MVC-NMF algorithm [32]

Step 1 (Initialization)	Randomly generate initial \mathbf{A} and \mathbf{X} . Set $\alpha > 0$ and $\delta = 10^{-6}$
Step 2 (Iteration)	<p>While a stop criterion is not satisfied,</p> <p>(i) update \mathbf{X} by</p> $\mathbf{X} := \mathbf{X} \otimes \frac{\mathbf{A}^T \mathbf{Y} + \delta_{\mathbf{X}}}{\mathbf{A}^T \mathbf{A} \mathbf{X} + \delta_{\mathbf{X}}};$ <p>(ii) update \mathbf{A} by</p> $\begin{cases} \mathbf{A} := \mathbf{A} \otimes \frac{\mathbf{Y}\mathbf{X}^T \mathbf{A}^T \mathbf{A} + \delta_{\mathbf{A}}}{\mathbf{A}\mathbf{X}\mathbf{X}^T \mathbf{A}^T \mathbf{A} + \alpha \det(\mathbf{A}^T \mathbf{A}) \mathbf{A} + \delta_{\mathbf{A}}} \\ \mathbf{A} := \mathbf{A}(\text{diag}(\mathbf{1} \oslash (\tilde{\mathbf{a}}))) \end{cases}$
Step 3 (Estimation)	The final \mathbf{A} and \mathbf{X} are the estimates of the mixing matrix and the sources, respectively

Table 2.10 INMF-VC algorithm [30]

Step 1 (Initialization)	<p>Set an initial sample number p for learning process. Then,</p> <p>(i) project the collected $k = p$ samples $\mathbf{y}_1, \dots, \mathbf{y}_k$ to be $[\tilde{\mathbf{y}}_1, \dots, \tilde{\mathbf{y}}_k]$ on the hyperplane $\Pi : \sum_{i=1}^m \tilde{y}_i = 1$, and construct the initial matrix $\tilde{\mathbf{Y}}_k = [\tilde{\mathbf{y}}_1, \dots, \tilde{\mathbf{y}}_k]$;</p> <p>(ii) obtain \mathbf{A}_k and $\tilde{\mathbf{X}}_k$ from $\tilde{\mathbf{Y}}_k$ by using the normal NMF algorithm</p>
Step 2 (Learning)	<p>For the $(k + 1)$th sample \mathbf{y}_{k+1},</p> <p>(i) project it to be $\tilde{\mathbf{y}}_{k+1}$ by</p> $[\tilde{\mathbf{y}}_{k+1}]_i = [\mathbf{y}_{k+1}]_i / \sum_{j=1}^m [\mathbf{y}_{k+1}]_j;$ <p>(ii) let $\mathbf{A}_{k+1} = \mathbf{A}_k$ and update $\tilde{\mathbf{x}}_{k+1}$ by</p> $\tilde{\mathbf{x}}_{k+1} := \tilde{\mathbf{x}}_{k+1} \otimes \frac{\mathbf{A}_{k+1}^T \tilde{\mathbf{h}}_{k+1} + \delta \tilde{\mathbf{x}}_{k+1}}{\mathbf{A}_{k+1}^T \mathbf{A}_{k+1} \tilde{\mathbf{x}}_{k+1} + \delta \tilde{\mathbf{x}}_{k+1}};$ <p>(iii) update \mathbf{A}_{k+1} by</p> $\mathbf{A}_{k+1} := \mathbf{A}_{k+1} \otimes \frac{(\alpha \tilde{\mathbf{Y}}_k \tilde{\mathbf{x}}_k^T + \beta \tilde{\mathbf{y}}_{k+1} \tilde{\mathbf{x}}_{k+1}^T) \mathbf{A}_{k+1}^T \mathbf{A}_{k+1} + \delta \mathbf{A}_{k+1}}{(\alpha \mathbf{A}_{k+1} \tilde{\mathbf{x}}_k^T + \beta \mathbf{A}_{k+1} \tilde{\mathbf{x}}_{k+1} \tilde{\mathbf{x}}_{k+1}^T) \mathbf{A}_{k+1}^T \mathbf{A}_{k+1} + \beta \mu \mathbf{A}_{k+1} + \delta \mathbf{A}_{k+1}}$
Step 3 (Estimation)	<p>Estimate the $(k + 1)$th column of \mathbf{X} by</p> $\mathbf{x}_{k+1} = \tilde{\mathbf{x}}_{k+1} / \sum_{j=1}^m [\mathbf{y}_{k+1}]_j.$ <p>If $k + 1$ is less than the number of the total samples, let $k = k + 1$ and go to Step 2</p>

Table 2.11 NGMCA algorithm [45]

Step 1 (Initialization)	Set a maximum iteration number K and the initial values for \mathbf{A}_0 , \mathbf{X}_0 and λ_1
Step 2 (Iteration)	<p>For $k = 1, 2, \dots, K$</p> <p>(i) normalize the columns of \mathbf{A}_{k-1};</p> <p>(ii) update \mathbf{X}_k by</p> $\mathbf{X}_k = \operatorname{argmin}_{\mathbf{X} \geq 0} \frac{1}{2} \ \mathbf{Y} - \mathbf{A}_{k-1} \mathbf{X}\ _2^2 + \lambda_k \ \mathbf{X}\ _1;$ <p>(iii) update \mathbf{A}_k by</p> $\mathbf{A}_k = \operatorname{argmin}_{\mathbf{A} \geq 0} \frac{1}{2} \ \mathbf{Y} - \mathbf{A} \mathbf{X}_k\ _2^2;$ <p>(iv) Select $\lambda_{k+1} \leq \lambda_k$</p>
Step 3 (Estimation)	\mathbf{A}_K and \mathbf{X}_K are the estimates of the mixing matrix and the source matrix, respectively

Furthermore, the nonnegative generalized morphological component analysis (NGMCA) method [45] is proposed to deal with DCA in the situations where the measurements or the observations are polluted by noise. Different from the traditional NMF algorithms which are usually applicable to determined mixing systems, NGMCA can also be applied to underdetermined mixing systems. Table 2.11 shows the NGMCA algorithm with soft threshold.

2.3.3 Algorithm Analysis

As for the initialization of the decomposition process, there are many useful methods which are verified to be efficient. The singular value decomposition based scheme is one of the best methods for initialization, which has advantages in approximating the data matrix [46]. Other initialization methods include spherical k-means clustering [47], principal component analysis, fuzzy clustering and Gabor wavelets [48], etc.

Regarding the convergence of algorithms, Lin gives a detailed convergence analysis of the multiplicative update algorithms which are widely used in NMF based methods [49], Yang and Yi propose a novel scheme to analyze the convergence of the constrained NMF with application to BSS [50], and Badeau et al. analyze the stability of these algorithms [51].

With regard to the uniqueness of NMF, it is analyzed under different conditions in [52]. In [53], Donoho and Stodden show that the NMF is unique if the involved data satisfies three rules: (a) generative model, (b) separability, and (c) complete factorial sampling. In [54], Laurberg et al. propose the following theorem.

Theorem 2.9 *If $\text{rank}(\mathbf{Y}) = r$, the NMF $\mathbf{Y} = \mathbf{AX}$ is unique if and only if the nonnegative orthant is the only simplicial cone \mathcal{U} with r extreme rays that satisfies*

$$\text{cone}(\mathbf{A}^T) \subseteq \mathcal{U} \subseteq \text{dcone}(\mathbf{X}) \quad (2.44)$$

where $\text{dcone}(\mathbf{X})$ denotes the dual cone of $\text{cone}(\mathbf{X})$.

Furthermore, Schachtner et al. propose a determinant criterion to constrain the solutions of NMF problems and achieve unique and optimal solutions in a general setting [55].

References

1. A. Cichocki, R. Zdunek, A.H. Phan, S. Amari, *Non-Negative Matrix and Tensor Factorization: Applications to Exploratory Multi-Way Data Analysis and Blind Source Separation* (Wiley-Blackwell, Oxford, 2009)
2. P. Sajda, S. Du, T. Brown, R. Stoyanova, D. Shungu, L.P.X. Mao, Nonnegative matrix factorization for rapid recovery of constituent spectra in magnetic resonance chemical shift imaging of the brain. *IEEE Trans. Med. Imaging* **23**(12), 1453–1465 (2004)
3. R.H.C. Gobinet, E. Perrin, Application of non-negative matrix factorization to fluorescence spectroscopy, in: *Proceedings 12th European Signal Processing Conference*, pp. 1095–1098 (2004)
4. Z. Yang, G. Zhou, S. Xie, S. Ding, J. Yang, J. Zhang, Blind spectral unmixing based on sparse nonnegative matrix factorization. *IEEE Trans. Image Process.* **20**(4), 1112–1125 (2011)
5. T.H. Chan, W.K. Ma, C.Y. Chi, Y. Wang, A convex analysis framework for blind separation of non-negative sources. *IEEE Trans. Signal Process.* **56**(10), 5120–5134 (2008)
6. T.H. Chan, C.Y. Chi, Y.M. Huang, W.K. Ma, A convex analysis-based minimum-volume enclosing simplex algorithm for hyperspectral unmixing. *IEEE Trans. Signal Process.* **57**(11), 4418–4432 (2009)

7. W.S.B. Ouedraogo, A. Souloumiac, M. Jaidane, C. Jutten, Non-negative blind source separation algorithm based on minimum aperture simplicial cone. *IEEE Trans. Signal Process.* **62**(2), 376–389 (2014)
8. D.D. Lee, H.S. Seung, Learning of the parts of objects by non-negative matrix factorization. *Nature* **401**(6755), 788–791 (1999)
9. D. Donoho, M. Elad, V. Temlyakov, Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Trans. Inf. Theory* **52**(1), 6–18 (2006)
10. R. Chartrand, Exact reconstruction of sparse signals via nonconvex minimization. *IEEE Signal Process. Lett.* **14**(10), 707–710 (2007)
11. Y. Li, A. Cichocki, S. Amari, Analysis of sparse representation and blind source separation. *Neural Comput.* **16**(6), 1193–1234 (2004)
12. P.O. Hoyer, Non-negative matrix factorization with sparseness constraints. *J. Mach. Learn. Res.* **5**, 1457–1469 (2004)
13. Z. Yang, Y. Xiang, S. Xie, S. Ding, Y. Rong, Nonnegative blind source separation by sparse component analysis based on determinant measure. *IEEE Trans. Neural Netw. Learn. Syst.* **23**(10), 1601–1610 (2012)
14. G. Rath, C. Guillemot, J. Fuchs, Sparse approximations for joint source-channel coding, in *Proc. IEEE 10th Workshop on Multimedia Signal Processing*, 2008, pp. 481–485.
15. J. Karvanen, A. Cichocki, Measuring sparseness of noisy signals, in: *Proceedings Fourth International Symposium on Independent Component Analysis and Blind Signal Separation*, pp. 125–130 (2003)
16. S. Rickard, M. Fallon, The Gini index of speech, in: *38th Proc. Conf. Inf. Sci. Syst.*, Princeton, NJ, (2004)
17. B.A. Olshausen, D.J. Field, Sparse coding of sensory inputs. *Curr. Opin. Neurobiol.* **14**(4), 481–487 (2004)
18. B. Rao, K. Kreutz-Delgado, An affine scaling methodology for best basis selection. *IEEE Trans. Signal Process.* **47**(1), 187–200 (1999)
19. A. Bronstein, M. Bronstein, M. Zibulevsky, Y.Y. Zeevi, Sparse ICA for blind separation of transmitted and reflected images. *Int. J. Imaging Syst. Technol.* **15**(1), 84–91 (2005)
20. N. Hurley, S. Rickard, Comparing measures of sparsity. *IEEE Trans. Inf. Theory* **55**(10), 4723–4741 (2009)
21. P. Georgiev, F. Theis, A. Cichocki, Sparse component analysis and blind source separation of underdetermined mixtures. *IEEE Trans. Neural Netw.* **16**(4), 992–996 (2005)
22. A.M. Bruckstein, M. Elad, M. Zibulevsky, On the uniqueness of nonnegative sparse solutions to underdetermined systems of equations. *IEEE Trans. Inf. Theory* **54**(11), 4813–4820 (2008)
23. E. Candès, T. Tao, Decoding by linear programming. *IEEE Trans. Inf. Theory* **51**(12), 4203–4215 (2005)
24. D. Donoho, J. Tanner, Sparse nonnegative solution of underdetermined linear equations by linear programming. *PNAS* **102**(27), 9446–9451 (2005)
25. F.Y. Wang, C.Y. Chi, T.H. Chan, Y. Wang, Nonnegative least-correlated component analysis for separation of dependent sources by volume maximization. *IEEE Trans. Pattern Anal. Mach. Intell.* **32**(5), 875–888 (2010)
26. J.M.P. Nascimento, J.M.B. Dias, Vertex component analysis: A fast algorithm to unmix hyperspectral data. *IEEE Trans. Geosci. Remote Sens.* **43**(4), 898–910 (2005)
27. Y. Wang, J. Xuan, R. Srikanthana, P.L. Choyke, Modeling and reconstruction of mixed functional and molecular patterns. *Int. J. Biomed. Imaging* **2006**, 1–9 (2006)
28. Z. Yang, Y. Xiang, Y. Rong, S. Xie, Projection-pursuit-based method for blind separation of nonnegative sources. *IEEE Trans. Neural Netw. Learn. Syst.* **24**(1), 47–57 (2013)
29. L. Miao, H. Qi, Endmember extraction from highly mixed data using minimum volume constrained nonnegative matrix factorization. *IEEE Trans. Geosci. Remote Sens.* **45**(3), 765–777 (2007)
30. G. Zhou, Z. Yang, S. Xie, J. Yang, Online blind source separation using incremental nonnegative matrix factorization with volume constraint. *IEEE Trans. Neural Netw.* **22**(4), 550–560 (2011)

31. S. Lopez, P. Horstrand, G.M. Callico, J.F. Lopez, R. Sarmiento, A novel architecture for hyper-spectral endmember extraction by means of the modified vertex component analysis (MVCA) algorithm. *IEEE. J. Sel. Top. Appl. Earth Obs. Remote Sens.* **5**(6), 1837–1848 (2012)
32. G. Zhou, S. Xie, Z. Yang, J. Yang, Z. He, Minimum-volume-constrained nonnegative matrix factorization: enhanced ability of learning parts. *IEEE Trans. Neural Netw.* **22**(10), 1626–1637 (2011)
33. A. Cichocki, R. Zdunek, S.I. Amari, New algorithms for nonnegative matrix factorization in applications to blind source separation, in: *Proceedings 2006 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 5479–5482 (2006)
34. D.D. Lee, H.S. Seung, Algorithms for non-negative matrix factorization. *Adv. Neural Inf. Process. Syst.* **13**, 556–562 (2001)
35. H. Sawada, H. Kameoka, S. Araki, N. Ueda, Multichannel extensions of non-negative matrix factorization with complex-valued data. *IEEE Trans. Audio, Speech, Lang. Process.* **21**(5), 971–982 (2013)
36. A. Cichocki, H. Lee, Y.D. Kim, S. Choi, Non-negative matrix factorization with α -divergence. *Pattern Recognit. Lett.* **29**(9), 1433–1440 (2008)
37. V.Y.F. Tan, C. Févotte, Automatic relevance determination in nonnegative matrix factorization with the β -divergence. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**(7), 1592–1605 (2013)
38. A. Cichocki, S. Cruces, S. Amari, Generalized alpha-beta divergences and their application to robust nonnegative matrix factorization. *Entropy* **13**, 134–170 (2011)
39. V.P. Pauca, J. Piper, R.J. Plemmons, Nonnegative matrix factorization for spectral data analysis. *Linear Algebra Appl.* **416**(1), 29–47 (2006)
40. A. Huck, M. Guillaume, J. Blanc-Talon, Minimum dispersion constrained nonnegative matrix factorization to unmix hyperspectral data. *IEEE Trans. Geosci. Remote Sens.* **48**(6), 2590–2612 (2010)
41. T. Virtanen, Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria. *IEEE Trans. Audio, Speech, Lang. Process.* **15**(3), 1066–1074 (2007)
42. Y. Zhang, Y. Fang, A NMF algorithm for blind separation of uncorrelated signals, in: *Proceedings 2007 IEEE International Conference on Wavelet Analysis and Pattern Recognition*, November 2–4, Beijing, China, pp. 999–1003 (2007)
43. S. Amari, Natural gradient works efficiently in learning. *Neural Comput.* **10**(2), 251–276 (1998)
44. A. Cichocki, S. Amari, *Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications* (Wiley, New York, 2003)
45. J. Rapin, J. Bobin, A. Larue, J.-L. Starck, Sparse and non-negative BSS for noisy data. *IEEE Trans. Signal Process.* **61**(22), 5620–5632 (2013)
46. C. Boutsidis, E. Gallopoulos, SVD based initialization: a head start for nonnegative matrix factorization. *Pattern Recogn.* **41**(4), 1350–1362 (2008)
47. S. Wild, J. Curry, A. Dougherty, Improving non-negative matrix factorizations through structured initialization. *Pattern Recogn.* **37**(11), 2217–2232 (2004)
48. Z. Zheng, J. Yang, Y. Zhu, Initialization enhancer for non-negative matrix factorization. *Eng. Appl. Artif. Intell.* **20**(1), 101–110 (2007)
49. C.-J. Lin, On the convergence of multiplicative update algorithms for nonnegative matrix factorization. *IEEE Trans. Neural Netw.* **18**(6), 1589–1596 (2007)
50. S. Yang, Z. Yi, Convergence analysis of non-negative matrix factorization for BSS algorithm. *Neural Process. Lett.* **31**(1), 45–64 (2010)
51. R. Badeau, N. Bertin, E. Vincent, Stability analysis of multiplicative update algorithms and application to nonnegative matrix factorization. *IEEE Trans. Neural Netw.* **21**(12), 1869–1881 (2010)
52. K. Huang, N.D. Sidiropoulos, A. Swami, Non-negative matrix factorization revisited: uniqueness and algorithm for symmetric decomposition. *IEEE Trans. Signal Process.* **62**(1), 211–224 (2014)
53. D.L. Donoho, V.C. Stodden, When does non-negative matrix factorization give a correct decomposition into parts? *Adv. Neural Inf. Process. Syst.* **16**, 1141–1148 (2003)

54. H. Laurberg, M.G. Christensen, M.D. Plumbley, L.K. Hansen, S.H. Jensen, Theorems on positive data: on the uniqueness of NMF. *Comput. Intell. Neurosci.* **2008**, 1–9 (2008)
55. R. Schachtner, G. Pöppel, E.W. Lang, Towards unique solutions of non-negative matrix factorization problems by a determinant criterion. *Digit. Signal Process.* **21**(4), 528–534 (2011)

Blind Source Separation

Dependent Component Analysis

Xiang, Y.; Peng, D.; Yang, Z.

2015, XII, 94 p. 30 illus., 13 illus. in color., Softcover

ISBN: 978-981-287-226-5