K-Subspace clustering and its application in sparse component analysis

Zhaoshui He[1,2] and Andrzej Cichocki[1]

1-Laboratory for Advanced Brain Signal Processing RIKEN Brain Science Institute, Wako-shi, Saitama 351-0198, Japan
2- School of Electronic and Information Engineering, South China University of Technology Guangzhou, China, 510640

Abstract. The K-subspace clustering algorithm is established for sparse component analysis and overcome the difficulty that conventional SCA algorithms can not overcome. The conventional SCA algorithm can only perform single dominant SCA, can not perform multiple dominant SCA, but the proposed SCA algorithm based on K-subspace clustering can overcome this difficulty.

1 Introduction

Clustering is one of the most widely used techniques for exploratory data analysis. It can be used for signal compression, blind source separation, feature extraction, regularization in inverse problems, especially in the applications of sparse component analysis and undetermined blind source separation [1]-[5][8]-[11]. Usually the clustering plays a key role in sparse component analysis.

The conventional SCA is based on the single-dominant-component assumption that for each time, there is only one dominant component and others components are insignificant. So the scatter plot of observation signals shows clear line orientations geometrically. But for multiple dominant components SCA, for example, if there are two or more dominant components for each time, there will be no line orientations at the scatter plot of observation signals and the conventional SCA algorithms will fail. However Georgiev, Theis and Cichocki [1] pointed out that as long as the number of nonzero components in source signals is smaller than the number m of observation signals, SCA is solvable and gave the corresponding algorithm. But they did not consider the noise. In practice it is impossible to rigorously satisfy their assumption (the number of nonzero components in source signals is smaller than the number m of observation signals). So here we continue to discuss the "multiple dominant components SCA problem". The K-subspace clustering approach is proposed to perform multiple dominant components SCA method in this paper.

2 Problem statement of K-subspace clustering

Give a $m \times n$ complex valued matrix:

$$\boldsymbol{B} = \begin{pmatrix} b_{11} & \cdots & b_{1n} \\ \vdots & \ddots & \vdots \\ b_{m1} & \cdots & b_{mn} \end{pmatrix} = (\boldsymbol{b}_1, \cdots, \boldsymbol{b}_n) \in C^{m \times n}, \tag{1}$$

where C denotes complex valued number. If $m \le n$, assume that any a $m \times m$ submatrix of $\boldsymbol{B}$ is invertible. Give an integer $k(<m)$ and a data set $\boldsymbol{X} = (\boldsymbol{x}(1), \cdots, \boldsymbol{x}(T)) \in C^{m \times T}, (T \gg m)$. So $\boldsymbol{B}$ has C_n^k different $m \times k$ submatrice and denote them as $\boldsymbol{B}_1, \boldsymbol{B}_2, \cdots, \boldsymbol{B}_{C_n^k}$. The column vectors of each $m \times k$ submatrix of $\boldsymbol{B}_i$ can span a linear space. Without the loss of generality, we suppose that $\boldsymbol{B}_i$ is composed by the column vectors $\boldsymbol{b}_{i1}, \boldsymbol{b}_{i2}, \cdots, \boldsymbol{b}_{ik}$ of matrix $\boldsymbol{B}$, where $\{i1, i2, \cdots, ik\} \subset \{1, 2, \cdots, n\}$. The column vectors $\boldsymbol{b}_{i1}, \boldsymbol{b}_{i2}, \cdots, \boldsymbol{b}_{ik}$ span a linear space $span\{\boldsymbol{B}_i\} = span\{\boldsymbol{b}_{i1}, \boldsymbol{b}_{i2}, \cdots, \boldsymbol{b}_{ik}\}$. So we can obtain C_n^k linear spaces using $m \times k$ submatrice $\boldsymbol{B}_1, \boldsymbol{B}_2, \cdots, \boldsymbol{B}_{C_n^k}$. Indicate $K = C_n^k$. The K-subspace clustering problem can be described as follows: for the given data set $\boldsymbol{X} = (\boldsymbol{x}(1), \cdots, \boldsymbol{x}(T))$, how to estimate a $m \times n$ matrix $\boldsymbol{B}$ such that the distance sum $\sum_{t=1}^{T} \min_{i=1,\cdots,C_n^k} d(\boldsymbol{x}(t), \boldsymbol{B}_i)$ is minimized, where $d(\boldsymbol{x}(t), \boldsymbol{B}_i)$ means the distance from point $\boldsymbol{x}(t)$ to the linear space spaned by the k column vectors of $\boldsymbol{B}_i$.

3 The distance formula from a point to a sub-linear-space

Consider a point $\boldsymbol{p} = (p_1, \cdots, p_m)^T$ (see Figure 1), we attempt to calculate the distance $d(\boldsymbol{p}, \mathcal{L})$ from the point $\boldsymbol{p}$ to linear space $\mathcal{L}$, whose basis matrix is

$$\boldsymbol{L} = \begin{pmatrix} l_{11} & \cdots & l_{1k} \\ \vdots & \ddots & \vdots \\ l_{m1} & \cdots & l_{mk} \end{pmatrix} = (\boldsymbol{l}_1, \cdots, \boldsymbol{l}_k). \tag{2}$$

This problem can convert to search a point $\boldsymbol{l}_*$(in the space $\mathcal{L}$), which is closest to $\boldsymbol{p}$, i.e., we can deal with this problem by solving the following optimization problem:

$$\begin{cases} d^2(\boldsymbol{p}, \mathcal{L}) = \min_{\boldsymbol{l}} g(\boldsymbol{l}) = \min_{\boldsymbol{b}} \|\boldsymbol{p} - \boldsymbol{l}\|^2, \\ \text{subject to}: \boldsymbol{l} \in \mathcal{L}, \end{cases} \tag{3}$$

where $\boldsymbol{l} = (l_1, \cdots, l_m)^T \in C^m$, $\boldsymbol{p} = (p_1, \cdots, p_m)^T \in C^m$. And $\boldsymbol{l}$ can be linearly represented:

$$\boldsymbol{l} = \alpha_1 \boldsymbol{l}_1 + \cdots + \alpha_k \boldsymbol{l}_k = \boldsymbol{L}\boldsymbol{\alpha}. \tag{4}$$

From equation (4), solving problem (3) can be described as

$$\begin{aligned} &\min_{\boldsymbol{\alpha} \in C^k} g(\boldsymbol{\alpha}) = \min_{\boldsymbol{\alpha} \in C^k} \|\boldsymbol{p} - \boldsymbol{L}\boldsymbol{\alpha}\|^2 = \min_{\boldsymbol{\alpha} \in C^k} \langle \boldsymbol{p} - \boldsymbol{L}\boldsymbol{\alpha}, \boldsymbol{p} - \boldsymbol{L}\boldsymbol{\alpha} \rangle \\ &= \min_{\boldsymbol{\alpha} \in C^k} (\boldsymbol{p} - \boldsymbol{L}\boldsymbol{\alpha})^H (\boldsymbol{p} - \boldsymbol{L}\boldsymbol{\alpha}) = \min_{\boldsymbol{\alpha} \in C^k} (\boldsymbol{p}^H - \boldsymbol{\alpha}^H \boldsymbol{L}^H)(\boldsymbol{p} - \boldsymbol{L}\boldsymbol{\alpha}), \end{aligned} \tag{5}$$

where $\langle \bullet, \bullet \rangle$ stands for inner product. For any real valued function $f(z)$ of a complex valued variable z the gradients with respect to the real and imaginary part are obtained by taking derivatives formally with respect to the conjugate quantities z^*, ignoring the nonconjugate occurrences of z [2][6][7], i.e.,

$$\frac{\partial f(z)}{\partial R(z)}+i\frac{\partial f(z)}{\partial I(z)}=2\frac{\partial f(z)}{\partial z^*}. \tag{6}$$

Therefore the derivative of cost function $g(\bullet)$ with respect to $\boldsymbol{\alpha}^*$ is

$$\frac{\partial g(\bullet)}{\partial \boldsymbol{\alpha}^*}=-\boldsymbol{L}^H\boldsymbol{p}+\boldsymbol{L}^H\boldsymbol{L}\boldsymbol{\alpha}. \tag{7}$$

Let $\partial g(\bullet)/\partial \boldsymbol{\alpha}^*=0$, we get $\boldsymbol{\alpha}=\left(\boldsymbol{L}^H\boldsymbol{L}\right)^{-1}\boldsymbol{L}^H\boldsymbol{p}$. (8)

Substitute equation (8) and equation (4) into (3), we have

$$d(\boldsymbol{p},\mathcal{L})=\sqrt{\boldsymbol{p}^H\boldsymbol{p}-\boldsymbol{p}^H\boldsymbol{L}\left(\boldsymbol{L}^H\boldsymbol{L}\right)^{-1}\boldsymbol{L}^H\boldsymbol{p}}. \tag{9}$$

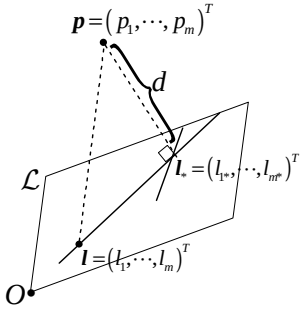

Fig.1 The distance from a point to a line

4 K-subspace clustering algorithm

K-subspace clustering algorithm (here $K=C_n^k$) can be outlined as follows:

(1) Input the observation dataset $\boldsymbol{X}=(\boldsymbol{x}(1),\cdots,\boldsymbol{x}(T))\in C^{m\times T},(T\gg m)$. Initialize the basis matrix $\boldsymbol{B}$ as $\hat{\boldsymbol{B}}\in C^{m\times K}$, normalize $\hat{\boldsymbol{B}}$ by $\hat{\boldsymbol{b}}_j=\hat{\boldsymbol{b}}_j/\left\|\hat{\boldsymbol{b}}_j\right\|, j=1,\cdots,n$. And denote its $K\left(K=C_n^k\right)$ $m\times k$ submatrice as $\hat{\boldsymbol{B}}_1,\hat{\boldsymbol{B}}_2,\cdots,\hat{\boldsymbol{B}}_K$.

(2) **Partition stage**: calculate each distance $d\left(\boldsymbol{x}(t),\hat{\boldsymbol{B}}_i\right),i=1,\cdots,K$ from the observation sample point $\boldsymbol{x}(t)$ to linear space $span\left\{\hat{\boldsymbol{B}}_i\right\}$ using the distance formula (9). The observation sample point $\boldsymbol{x}(t)\in\theta\left(\hat{\boldsymbol{B}}_i\right)$ if and only if $d\left(\boldsymbol{x}(t),\hat{\boldsymbol{B}}_i\right)=\min\left\{d\left(\boldsymbol{x}(t),\hat{\boldsymbol{B}}_j\right),j=1,\cdots,K\right\}$, where $\theta\left(\hat{\boldsymbol{B}}_i\right)$ is a data vector set. By this means, assign the T observation sample points $\boldsymbol{x}(1),\cdots,\boldsymbol{x}(T)$ in observation matrix $\boldsymbol{X}=\left[\boldsymbol{x}(1),\cdots,\boldsymbol{x}(T)\right]$ into K different clusters $\theta\left(\hat{\boldsymbol{B}}_i\right),i=1,\cdots,K$.

(3) Update the cluster centroids matrix stage: update the cluster centroids matrix $\hat{\boldsymbol{B}}$:

3.1) Consider each cluster $\theta\left(\hat{\boldsymbol{B}}_i\right)$, assume it contains $T^{(i)}$ entries $\boldsymbol{x}^{(i)}(1),\cdots,\boldsymbol{x}^{(i)}\left(T^i\right)$, which compose a matrix $\boldsymbol{X}^{(i)}=\left[\boldsymbol{x}^{(i)}(1),\cdots,\boldsymbol{x}^{(i)}\left(T^{(i)}\right)\right]$. For the

symmetrical matrix $\frac{1}{T}\left(\boldsymbol{X}^{(i)}\right)^H \boldsymbol{X}^{(i)} = \boldsymbol{V}^{(i)}\boldsymbol{D}^{(i)}\left(\boldsymbol{V}^{(i)}\right)^H$, apply EVD decomposition. Suppose $d_1^{(i)},\cdots,d_k^{(i)}$ are the largest k eigenvalues and $\boldsymbol{v}_1^{(i)},\cdots,\boldsymbol{v}_k^{(i)}$ are the corresponding eigenvectors, we get $\boldsymbol{E}_i = \left[\boldsymbol{v}_1^{(i)},\cdots,\boldsymbol{v}_k^{(i)}\right], i = 1,\cdots,K$.

3.2) Update the column vector $\hat{\boldsymbol{b}}_j$ of $\hat{\boldsymbol{B}}$. Without losing generality, suppose $\hat{\boldsymbol{b}}_j$ is the column vector of the $m\times k$ submatrice $\hat{\boldsymbol{B}}_{j1},\cdots,\hat{\boldsymbol{B}}_{jC_{n-1}^{k-1}}$ of $\hat{\boldsymbol{B}}$, respectively. We can update the column vector $\hat{\boldsymbol{b}}_j$ by solving the following optimization problem:

$$J\left(\hat{\boldsymbol{b}}_j\right) = \min_{\hat{\boldsymbol{b}}_j} \sum_{r=1}^{C_{n-1}^{k-1}} d^2\left(\hat{\boldsymbol{b}}_j, \boldsymbol{E}_{jr}\right), \text{subject to} \hat{\boldsymbol{b}}_j^H \hat{\boldsymbol{b}}_j = 1. \tag{10}$$

The solution of problem (10) is the eigenvector corresponding to the smallest eigenvalue of matrix $\sum_{r=1}^{C_{n-1}^{k-1}} \boldsymbol{E}_{jr}\left(\boldsymbol{E}_{jr}^H\boldsymbol{E}_{jr}\right)^{-1}\boldsymbol{E}_{jr}^H$. From formula (9), problem (10) can be changed into

$$\max_{\hat{\boldsymbol{b}}_j} \hat{\boldsymbol{b}}_j^H \left(\sum_{r=1}^{C_{n-1}^{k-1}} \boldsymbol{E}_{jr}\left(\boldsymbol{E}_{jr}^H\boldsymbol{E}_{jr}\right)^{-1}\boldsymbol{E}_{jr}^H\right)\hat{\boldsymbol{b}}_j, \text{subject to } \hat{\boldsymbol{b}}_j^H\hat{\boldsymbol{b}}_j = 1. \tag{11}$$

(11) can be changed into the following optimization problem without constraint:

$$\max_{\hat{\boldsymbol{b}}_j} J\left(\hat{\boldsymbol{b}}_j\right) = \max_{\hat{\boldsymbol{b}}_j} \frac{\hat{\boldsymbol{b}}_j^H \left(\sum_{r=1}^{C_{n-1}^{k-1}} \boldsymbol{E}_{jr}\left(\boldsymbol{E}_{jr}^H\boldsymbol{E}_{jr}\right)^{-1}\boldsymbol{E}_{jr}^H\right)\hat{\boldsymbol{b}}_j}{\hat{\boldsymbol{b}}_j^H\hat{\boldsymbol{b}}_j}. \tag{12}$$

From equation (12), let $\frac{\partial J}{\partial \hat{\boldsymbol{b}}_j^*} = 0$ and we have

$$\frac{\partial J}{\partial \hat{\boldsymbol{b}}_j^*}\cdot\hat{\boldsymbol{b}}_j^H\hat{\boldsymbol{b}}_j + J\cdot\frac{\partial\left(\hat{\boldsymbol{b}}_j^H\hat{\boldsymbol{b}}_j\right)}{\partial\hat{\boldsymbol{b}}_j^*} = \left(\sum_{r=1}^{C_{n-1}^{k-1}} \boldsymbol{E}_{jr}\left(\boldsymbol{E}_{jr}^H\boldsymbol{E}_{jr}\right)^{-1}\boldsymbol{E}_{jr}^H\right)\hat{\boldsymbol{b}}_j \Rightarrow$$

$$\frac{\partial J}{\partial \hat{\boldsymbol{b}}_j^*}\cdot\hat{\boldsymbol{b}}_j^H\hat{\boldsymbol{b}}_j = \left[\left(\sum_{r=1}^{C_{n-1}^{k-1}} \boldsymbol{E}_{jr}\left(\boldsymbol{E}_{jr}^H\boldsymbol{E}_{jr}\right)^{-1}\boldsymbol{E}_{jr}^H\right) - J\right]\hat{\boldsymbol{b}}_j = 0. \tag{13}$$

From equation (13), obviously $\hat{\boldsymbol{b}}_j$ is the eigenvector of $\sum_{r=1}^{C_{n-1}^{k-1}} \boldsymbol{E}_{jr}\left(\boldsymbol{E}_{jr}^H\boldsymbol{E}_{jr}\right)^{-1}\boldsymbol{E}_{jr}^H$. In this way $\hat{\boldsymbol{b}}_1,\hat{\boldsymbol{b}}_2,\cdots,\hat{\boldsymbol{b}}_n$ respectively are updated, so $\hat{\boldsymbol{B}} = \left[\hat{\boldsymbol{b}}_1,\hat{\boldsymbol{b}}_2,\cdots,\hat{\boldsymbol{b}}_n\right]$ is updated.

(4) Return to step (2) and repeat until $\hat{\boldsymbol{B}}$ converges.

(5) Output the estimation of basis matrix $\boldsymbol{B}^* = \left[\hat{\boldsymbol{b}}_1,\cdots,\hat{\boldsymbol{b}}_n\right]$.

Remark In analogy to K-means, we call this algorithm "K-subspace clustering".

5 Sparse component analysis

Sparse Component Analysis (SCA) can be expressed as follows:

$$\boldsymbol{X} = \boldsymbol{A}\boldsymbol{S} + \boldsymbol{V}, \tag{14}$$

where $\boldsymbol{X} = \left[\boldsymbol{x}(1),\cdots,\boldsymbol{x}(T)\right] \in C^{m\times T}\ (T >> m)$ is the given data (observation) matrix, $\boldsymbol{A} = \left[\boldsymbol{a}_1,\cdots,\boldsymbol{a}_m\right] \in C^{m\times n}$ is unknown mixing (basis) matrix (not necessary sparse) and $\boldsymbol{S} \in C^{n\times T}$ is also unknown matrix representing sparse sources or hidden components,

T is the number of available sample, m the number of observations and n the number of sources. Our main objective is to find a reasonable basis matrix $\boldsymbol{A}$ such that the coefficients in matrix $\boldsymbol{S}$ are as sparse as possible. Here we mainly discuss the $k(>1)$ dominant components SCA. We use the two-stage methods to solve the SCA problem: estimate at the first step the basis matrix $\boldsymbol{A}$ using K-subspace clustering approach and in the next step to estimate the coefficient matrix $\boldsymbol{S}$ using the following Minimum Distance Decomposition (MDD):

(1) Denote $K\left(K=C_n^k\right)$ $m\times k$ submatrice of $\boldsymbol{A}$ as $\boldsymbol{A}_1,\boldsymbol{A}_2,\cdots,\boldsymbol{A}_K$.

(2) Calculate each distance $d\left(\boldsymbol{x}(t),\boldsymbol{A}_i\right), i=1,\cdots,K$ from the observation sample point $\boldsymbol{x}(t)$ to linear space $span\{\boldsymbol{A}_i\}$ using the distance formula (9). The observation sample point $\boldsymbol{x}(t)\in\theta(\boldsymbol{A}_i)$ if and only if $d\left(\boldsymbol{x}(t),\boldsymbol{A}_i\right)=\min\left\{d\left(\boldsymbol{x}(t),\boldsymbol{A}_j\right), j=1,\cdots,K\right\}$, where $\theta(\boldsymbol{A}_i)$ is a data vector set. By this means, the T observation sample points $\boldsymbol{x}(1),\cdots,\boldsymbol{x}(T)$ are assigned into K different clusters $\theta(\boldsymbol{A}_i), i=1,\cdots,K$.

(3) Suppose $\boldsymbol{A}_i=[\boldsymbol{a}_{i1},\cdots,\boldsymbol{a}_{ik}]$. If $\boldsymbol{x}(t)\in\theta(\boldsymbol{A}_i)$, the estimation of the column vector $\boldsymbol{s}(t), t=1,\cdots,T$ of coefficient matrix $\boldsymbol{S}$ is as follows:

$$\begin{cases}\begin{pmatrix} s(i1,t)\\ \vdots\\ s(ik,t)\end{pmatrix}=\left(\boldsymbol{A}_i^H\boldsymbol{A}_i\right)^{-1}\boldsymbol{A}_i^H\boldsymbol{x}(t),\\ s(j,t)=0, \text{for } j\neq i1,\cdots,ik.\end{cases} \tag{15}$$

6 Numerical experiments and result analysis

We give a SCA experiment. To check how well the mixing matrix is estimated, we introduce the following the Biased Angles Sum (BAS), the sum of angles between the column vectors (of mixing matrix) and their corresponding estimations:

$$BAS\left(\boldsymbol{A},\hat{\boldsymbol{A}}\right)=\sum_{i=1}^{n} acos\left(\left\langle \boldsymbol{a}_i,\hat{\boldsymbol{a}}_i\right\rangle\right), \tag{16}$$

where $acos(\bullet)$ is the inverse cosine function, $\langle\bullet,\bullet\rangle$ the inner product and $\boldsymbol{A}=[\boldsymbol{a}_1,\cdots,\boldsymbol{a}_n]$.

In addition, the Signal to Interference Ratio (SIR) is employed to measure the accuracy of the estimations of coefficient matrix. It is shown as follows:

$$SIR(s,\hat{s})=10\log\left[\|s\|_2^2/\|s-\hat{s}\|_2^2\right](\text{dB}). \tag{17}$$

Consider that $\hat{s}$ is permitted to scale up to a nonzero constant factor $c(c\neq 0)$ with the corresponding s, we rescale $\hat{s}$ to be the same energy level as s before computing its SIR. Usually, when SIR >18dB, the estimation is acceptable.

The basis matrix is taken by randomly as follows:

$$\boldsymbol{A}=\begin{pmatrix} 0.7930 & -0.7428 & 0.1404 & 0.9021\\ 0.1480 & -0.5901 & 0.7010 & -0.3691\\ -0.5910 & -0.3161 & -0.6992 & -0.2235\end{pmatrix}, \tag{18}$$

The two dominant components sparse source matrix $\boldsymbol{S} \in R^{4\times 10000}$ was generated artificially and added Gaussian noise. After 7 iterations, the algorithm converged. It took less than 2 seconds. The initial value $\boldsymbol{A}^{(0)}$ and estimation $\hat{\boldsymbol{A}}$ of $\boldsymbol{A}$ are respectively as follows:

$$\boldsymbol{A}^{(0)} = \begin{pmatrix} 0.3137 & -0.7592 & 1.4124 & 0.9980 \\ 1.3431 & -0.6031 & -0.5058 & 0.1446 \\ -1.3657 & -0.3231 & -0.4132 & -0.7089 \end{pmatrix} \quad \hat{\boldsymbol{A}} = \begin{pmatrix} -0.1405 & -0.7428 & 0.9021 & 0.7931 \\ -0.7010 & -0.5901 & -0.3691 & 0.1481 \\ 0.6992 & -0.3163 & -0.2236 & -0.5908 \end{pmatrix}$$

$BAS\left(\boldsymbol{A}, \hat{\boldsymbol{A}}\right) = 5.3933\text{e-}004$, and the estimations of 4 components of coefficient matrix are 36.5774dB, 108.0649dB, 56.4985dB and 47.0995. The result is very good.

7 Conclusions

Conventional SCA methods only can deal with single dominant component SCA problem. To overcome the drawback of conventional SCA methods, in this paper the K-subspace clustering algorithm is proposed to perform multiple dominant components SCA. In fact the K-subspace algorithm can be used to sparsely represent the complex valued data. Here we are sorry that we can not give the examples of complex sparse representation for the limited space.

References

[1] P. G. Georgiev, F. Theis, and A. Cichocki, Sparse component analysis and blind source separation of underdetermined mixtures, *IEEE Trans. on Neural Networks*, July 2005, Vol. 16(4), pp. 992-996.

[2] L. Parra, C. Spence. Convolutive blind separation of non-stationary sources. *IEEE Trans. Speech and audio processing*, vol. 8(3), pp. 320-327, May 2000.

[3] M. Aharon, M. Elad, A.M. Bruckstein. K-SVD and its non-negative variant for dictionary design. *Proceedings of the SPIE conference wavelets*, Vol. 5914, July 2005.

[4] Y.Q. Li, S. Amari, A. Cichocki, and D. W. C. Ho: "Underdetermined Blind Source Separation Based on Sparse Representation", *IEEE Trans. on Signal Processing* (in print).

[5] Y. Q. Li, A. Cichocki, and S. Amari, "Blind estimation of channel parameters and source components for EEG signals: A sparse factorization approach, *IEEE Transactions on Neural Networks*, 2006, (accepted for publication)

[6] D. Brandwood, A complex gradient operator and its application in adaptive array theory. *Proc. Inst. Elect. Eng*., vol 130, pp.11-16, Feb. 1983.

[7] K. Janich, *Einfuhrung in Die Funktionentheorie*. Berlin, Germany: Springer-Verlag, 1997, ch. 2.

[8] P. Bofill, M. Zibulevsky. Undetermined blind source separation using sparse representations. *Signal processing*, vol.81, 2353-2362, 2001.

[9] Y.Q. Li, A. Cichocki, S. Amari. Analysis of sparse representation and blind source separation. *Neural computation*, vol.16, 1193-1234, 2004.

[10] I. Takigawa, M. Kudo, J. Toyama, Performance analysis of Minimum l_1-norm solutions for underdetermined source separation, *IEEE Trans. Signal processing*, vol.52(3):582-591, March 2004.

[11] O. Yilmaz, S. Rickard. Blind separation of speech mixtures via time-frequency masking. IEEE Trans. Signal Processing, vol.52(7), pp.1830-1847, 2004.