Underdetermined blind source separation (uBSS), sparse component analysis (SCA)

Ivica Kopriva

Ruđer Bošković Institute

e-mail: ikopriva@irb.hr ikopriva@gmail.com
Web: http://www.lair.irb.hr/ikopriva/

Course outline

- Motivation with illustration of applications (lecture I)
- Mathematical preliminaries with principal component analysis (PCA)? (lecture II)
- Independent component analysis (ICA) for linear static problems: information-theoretic approaches (lecture III)
- ICA for linear static problems: algebraic approaches (lecture IV)
- ICA for linear static problems with noise (lecture V)
- Dependent component analysis (DCA) (lecture VI)

Course outline

- Underdetermined blind source separation (BSS) and sparse component analysis (SCA) (lecture VII/VIII)
- Nonnegative matrix factorization (NMF) for determined and underdetermined BSS problems (lecture VIII/IX)
- BSS from linear convolutive (dynamic) mixtures (lecture X/XI)
- Nonlinear BSS (lecture XI/XII)
- Tensor factorization (TF): BSS of multidimensional sources and feature extraction (lecture XIII/XIV)

Seminar problems

1. Blind separation of two uniformly distributed signals with maximum likelihood (ML) and AMUSE/SOBI independent component analysis (ICA) algorithm. Blind separation of two speech signals with ML and AMUSE/SOBI ICA algorithm. **Theory, MATLAB demonstration and comments of the results.**

2. Blind decomposition/segmentation of multispectral (RGB) image using ICA, dependent component analysis (DCA) and nonnegative matrix factorization (NMF) algorithms. **Theory, MATLAB demonstration and comments of the results.**

3. Blind separation of acoustic (speech) signals from convolutive dynamic mixture. **Theory, MATLAB demonstration and comments of the results.**

Seminar problems

4. Blind separation of images of human faces using ICA and DCA algorithms (innovation transform and ICA, wavelet packets and ICA) **Theory, MATLAB demonstration and comments of the results.**

5. Blind decomposition of multispectral (RGB) image using sparse component analysis (SCA): clustering + L_p norm ($0<p\leq 1$) minimization. **Theory, MATLAB demonstration and comments of the results.**

6. Blind separation of four sinusoidal signals from two static mixtures (a computer generated example) using sparse component analysis (SCA): clustering + L_p norm ($0<p\leq 1$) minimization in frequency (Fourier) domain. **Theory, MATLAB demonstration and comments of the results.**

Seminar problems

7. Blind separation of three acoustic signals from two static mixtures (a computer generated example) using sparse component analysis (SCA): clustering + L_p norm ($0<p\leq 1$) minimization in time-frequency (short-time Fourier) domain. **Theory, MATLAB demonstration and comments of the results.**

8. Blind extraction of five pure components from mass spectra of two static mixtures of chemical compounds using sparse component analysis (SCA): clustering a set of single component points + L_p norm ($0<p\leq 1$) minimization in m/z domain. **Theory, MATLAB demonstration and comments of the results.**

9. Feature extraction from protein (mass) spectra by tensor factorization of disease and control samples in joint bases. Prediction of prostate/ovarian cancer. **Theory, MATLAB demonstration and comments of the results.**

Blind source separation

A theory for multichannel blind signal recovery requiring minimum of *a priori* information.

Problem:

$$\mathbf{X}=\mathbf{AS} \quad \mathbf{X}\in R^{NxT},\ \mathbf{A}\in R^{NxM},\ \mathbf{S}\in R^{MxT}$$

Goal: find **A** and **S** based on **X** only.

Solution $\mathbf{X}=\mathbf{A}\mathbf{T}^{-1}\mathbf{T}\mathbf{S}$ must be characterized with $\mathbf{T}=\mathbf{P}\Lambda$ where **P** is permutation and Λ is diagonal matrix i.e.: $\mathbf{Y}\cong\mathbf{P}\Lambda\mathbf{S}$

A. Cichocki, S. Amari, "Adaptive Blind Signal and Image Processing," John Wiley, 2002.

Independent component analysis

- Number of mixtures *N* must be greater than or equal to *M*.

- source signals $s_i(t)$ must be statistically independent.

$$\mathrm{p}(\mathbf{s}) = \prod_{m=1}^{M} p_m(s_m)$$

- source signals $s_m(t)$, except one, must be non-Gaussian.

$$\{C_n(s_m) \neq 0\}_{m=1}^{M} \qquad \forall n > 2$$

- mixing matrix **A** must be nonsingular.

$$\mathbf{W} \cong \mathbf{A}^{-1}$$

A. Hyvarinen, J. Karhunen, E. Oja, "Independent Component Analysis," John Wiley, 2001.

Blind image separation – an example

s

x

T.-M. Huang, V. Kecman, I. Kopriva, "Kernel Based Algorithms for Mining Huge Data Sets: Supervised, Semi-supervised and Unsupervised Learning," Springer Series: Studies in Computational Intelligence, Vol. 17, XVI, ISBN: 3-540-31681-7, 2006.

Blind image separation – an example

y - PCA

y - ICA (min *I*(**y**))

Underdetermined BSS

- uBSS occurs when number of measurements *N* is less than number of sources *M*. Resulting system of linear equations

$$\mathbf{x}=\mathbf{A}\mathbf{s}$$

is underdetermined. Without constraints on **s** unique solution does not exist even if **A** is known:

$$\mathbf{s}=\mathbf{s}_p+\mathbf{s}_h=\mathbf{A}^{\dagger}\mathbf{x}+\mathbf{V}\mathbf{z} \quad \mathbf{A}\mathbf{V}\mathbf{z}_h=\mathbf{0}$$

where **V** spans null-space of **A** that is *M*-*N* dimensional.

- However, if **s** is sparse enough **A** can be identified and unique solution for **s** can be obtained. This is known as sparse component analysis (SCA).

Underdetermined BSS

1. Y. Li, A. Cichocki, S. Amari, "Analysis of Sparse Representation and Blind Source Separation," Neural Computation **16**, 1193-1234 (2004).
2. Y. Li, S. Amari, A. Cichocki, D.W.C. Ho, S. Xie, "Underdetermined Blind Source Separation Based on Sparse Representation," IEEE Trans. on Signal Processing **54**, 423-437 (2006).
3. P. Georgiev, F. Theis, A. Cichocki, "Sparse Component Analysis and Blind Source Separation of Underdetermined Mixtures," IEEE Trans. on Neural Networks **16**, 992-996 (2005).
4. P. Bofill, M. Zibulevsky, "Underdetermined blind source separation using sparse representations," Signal Processing **81**, 2353-2362 (2001).
5. D. Luengo, I. Santamaria, L. Vielva, "A general solution to blind inverse problems for sparse input signal," Neurocomputing **69**, 198-215 (2005).
6. I. Takigawa, M. Kudo, J. Toyama, "Performance Analysis of Minimum l_1-Norm Solutions for Underdetermined Source Separation," IEEE Tr. on Signal Processing **52**, 582-591 (2004).
7. D. L. Donoho, M. Elad, "Optimally sparse representation in general (non-orthogonal) dictionaries via l_1 minimization," Proc. Nat. Acad. Sci. **100**, 2197-2202 (2003).
8. J. A. Tropp, A.C. Gilbert, "Signal Recovery from Random Measurements via Orthogonal Matching Pursuit," IEEE Transactions on Information Theory **53**, 4655-4666 (2007).
9. Y. Washizava, A. Cichocki, "On-Line *k*-plane clustering learning algorithm for sparse component analysis," in *Proceedings of ICASSP'06*, Toulouse, France, 2006, pp. 681-684.
10. F. M. Naini, G.H. Mohimani, M. Babaie-Zadeh, Ch. Jutten, "Estimating the mixing matrix in Sparse Component Analysis (SCA) based on partial *k*-dimensional subspace clustering," Neurocomputing **71**, 2330-2343 (2008).
11. V. G. Reju, S.N. Koh, I. Y. Soon, "An algorithm for mixing matrix estimation in instantaneous blind source separation," Signal Processing **89**, 1762-1773 (2009).
12. S. G. Kim, C.D. Yoo, "Underdetermined Blind Source Separation Based on Subspace Representation," IEEE Trans. Signal Processing **57**, 2604-2614 (2009).

uBSS – L_p norm minimization: $0< p \leq 1$

SCA-based solution of the uBSS problem is obtained in two stages:

1) estimate basis or mixing matrix **A** using data clustering, ref.[9,10].

2) estimating sources **s** solving underdetermined linear systems of equations **x**=**As**. Provided that **s** is sparse enough, solution is obtained at the minimum of L_p-norm, ref.[1,2,6,7,8,13,14].

- L_1-norm is often used as a replacement for L_0-quasi-norm since it is convex and, thus, provides unique solution. However, it is sensitive to presence of noise i.e. presence of errors in sparse approximation. L_1-norm based solution is not the sparsest one.

13. R. Chartrand, Exact reconstructions of sparse signals via nonconvex minimization, IEEE Signal Process. Let., 14 (2007), 707-710.
14. L. Foucart, Sparsest solution of underdetermined linear systems via l_q minimization for $0<q\leq 1$, Appl. Comp. Harmon. Anal. 26 (2009) 395-407.

uBSS – L_p norm minimization: $0< p \leq 1$

•Unique SCA-based solution of the uBSS problem **x**=**As** is obtained if **s** has (*M*-*N*+*1*)-zero components or if it is *N*-1 sparse.

•Signal is *k*-sparse if it has *k* non-zero components, i.e. $k=||\mathbf{s}||_0$.

•If uBSS problem is not sparse in original domain it is transformed in domain where enough level of sparseness can be achieved: $T(\mathbf{x})=\mathbf{A}T(\mathbf{s})$.

•Time-frequency and time-scale (wavelet) bases are employed for this purpose most often.

15. R. Gribonval and M. Nielsen, "Sparse representations in unions of bases," IEEE Transactions on Information Theory **49**, 3320-3325 (2003).
16. J. .A. Tropp, "Greed is good: Algorithmic results for sparse approximation," IEEE Transactions on Information Theory **50**, 2231-2242 (2004).

uBSS – L_p norm minimization: $0< p \leq 1$

•In addition to sparseness requirement certain degree of incoherence of the basis or mixing matrix **A** is required as well, ref.[7,15,16]. Mutual coherence is defined as the largest absolute and normalized inner product between different columns in **A**, what reads as

$$\mu\{\mathbf{A}\} = \max_{1\leq i,j\leq M \text{ and } i\neq j} \frac{\left|\mathbf{a}_i^T \mathbf{a}_j\right|}{\left\|\mathbf{a}_i\right\|\left\|\mathbf{a}_j\right\|}$$

The mutual coherence provides a measure of the worst-case similarity between the basis vectors. It indicates how much two closely related vectors may confuse any pursuit algorithm (solver of the underdetermined linear system of equations). Perfect recovery condition for **s** relates sparseness requirement on **s** and coherence of **A**:

$$\left\|\mathbf{s}\right\|_0 < \frac{1}{2}\left(1+\frac{1}{\mu\{\mathbf{A}\}}\right)$$

uBSS – L_p norm minimization: $0< p \leq 1$

It was found in ref. [17] that above criterion although true from the worse-case standpoint does not reflect accurately actual behavior of sparse representations and pursuit's algorithms performance. Average measure of coherence is proposed in [17] coined *t-averaged mutual coherence* to better characterize behavior of sparse representations.

$$\mu_t\{\mathbf{A}\} = \frac{\sum_{1\le i,j\le M \text{ and } i\neq j} \left(|g_{ij}| \ge t\right)|g_{ij}|}{\sum_{1\le i,j\le M \text{ and } i\neq j} \left(|g_{ij}| \ge t\right)}$$

where $\mathbf{G=A^TA}$ is Gramm matrix. It applies $\mu_t\{\mathbf{A}\} \ge t$ as well as

$$\lim_{t\to 1} \mu_t\{\mathbf{A}\} = \mu\{\mathbf{A}\}$$

17. M. Elad, "Optimized Projections for Compressed Sensing," IEEE Transactions on Signal Processing **55**, 5695-5702 (2007).

uBSS – L_p norm minimization: $0< p \leq 1$

•In the context of blind source separation scenario properties of the mixing matrix **A** can not be predefined or selected i.e. they are problem dependent and given.Yet, **A** dictates necessary level of sparseness of **s** to, possibly, obtain unique solution of the uBSS problem: **x=As**. To obtain this solution it is necessary:

- to estimate **A** as accurately as possible.
- to find representation (transformation) $T(\mathbf{x})=\mathbf{A}T(\mathbf{s})$ where $T(\mathbf{s})$ is as sparse as possible.
- to construct algorithms for solving underdetermined system of equations $T(\mathbf{x})=\mathbf{A}T(\mathbf{s})$ that are robust with respect to the presence of noise i.e. errors in sparse approximation of $T(\mathbf{s})$: $T(\mathbf{s})$ is approximately k-sparse with k dominant and number of small coefficients. If possible performance of the algorithm should remain robust if k increases.

uBSS – L_p norm minimization: $0< p \leq 1$

- Solving underdetermined system of linear equations **x**=**As** amounts to solving:

$$\hat{\mathbf{s}}(t) = \arg\min_{\mathbf{s}(t)} \|\mathbf{s}(t)\|_0 \quad \text{s.t. } \hat{\mathbf{A}}\mathbf{s}(t) = \mathbf{x}(t) \quad \forall t = 1,\ldots,T$$

or for problems with noise or approximation error:

$$\hat{\mathbf{s}}(t) = \arg\min_{\mathbf{s}(t)} \frac{1}{2}\left\|\hat{\mathbf{A}}\mathbf{s}(t) - \mathbf{x}(t)\right\|_2^2 + \lambda \|\mathbf{s}(t)\|_0 \qquad \forall t = 1,\ldots,T$$

$$\hat{\mathbf{s}}(t) = \arg\min_{\mathbf{s}(t)} \|\mathbf{s}(t)\|_0 \quad \text{s.t. } \left\|\hat{\mathbf{A}}\mathbf{s}(t) - \mathbf{x}(t)\right\|_2^2 \leq \varepsilon \qquad \forall t = 1,\ldots,T$$

Minimization of L_0–norm of **s** is combinatorial problem that is NP-hard. For larger dimension M it becomes computationally infeasible. Moreover, minimization of L_0–norm is very sensitive to noise i.e. presence of small coefficients.

uBSS – L_1 norm minimization

Replacement of L_0-norm by L_1-norm is done quite often. That is known as convex relaxation of the minimum L_0-norm problem. This leads to linear programming, [1,5-7]:

$$\hat{\mathbf{s}}(t) = \arg\min_{\mathbf{s}(t)} \sum_{m=1}^{\hat{M}} s_m(t) \quad \text{s.t. } \hat{\mathbf{A}}\mathbf{s}(t) = \mathbf{x}(t) \quad \forall t = 1,\ldots,T$$

$$\text{s.t. } \mathbf{s}(t) \geq 0$$

L_1-regularized least square problem ref.[8,18,19]:

$$\hat{\mathbf{s}}(t) = \arg\min_{\mathbf{s}(t)} \frac{1}{2}\left\|\hat{\mathbf{A}}\mathbf{s}(t) - \mathbf{x}(t)\right\|_2^2 + \lambda \left\|\mathbf{s}(t)\right\|_1 \qquad \forall t = 1,\ldots,T$$

and L_2-regularized linear problem [19,20]:

$$\hat{\mathbf{s}}(t) = \arg\min_{\mathbf{s}(t)} \left\|\mathbf{s}(t)\right\|_1 \quad \text{s.t. } \left\|\hat{\mathbf{A}}\mathbf{s}(t) - \mathbf{x}(t)\right\|_2^2 \leq \varepsilon \qquad \forall t = 1,\ldots,T$$

18. S..J. Kim, K. Koh, M. Lustig, S. Boyd, D. Gorinevsky, "An Interior-Point Method for Large-Scale -Regularized Least Squares,"IEEE Journal of Selected Topics in Signal Processing **1**, 606-617 (2007), **http://www.stanford.edu/~boyd/l1_ls/.**
19. E.. van den Berg, M.P. Friedlander, "Probing the Pareto Frontier for Basis Pursuit Solutions," SIAM J. Sci. Comput. 31, 890-912 (2008).
20. M.A.T. Figuiredo, R.D. Nowak, S.J. Wright, "Gradient Projection for Sparse Reconstruction: Application to Compressed Sensing and Other Inverse Problems," IEEE Journal on Selected Topics in Signal Processing **1**, 586-597 (2007).

uBSS – L_1 norm minimization

Provided that prior on **s**(t) is Laplacian, maximum likelihood approach to maximization of posterior probability $P(\mathbf{s}|\mathbf{x},\mathbf{A})$ yields minimum L_1-norm as the solution:

$$\hat{\mathbf{s}}(t) = \max_{\hat{\mathbf{A}}\mathbf{s}(t)=\mathbf{x}(t)} P\left(\mathbf{s}(t)\middle|\mathbf{x}(t),\hat{\mathbf{A}}\right)$$

$$= \max_{\hat{\mathbf{A}}\mathbf{s}(t)=\mathbf{x}(t)} P\left(\mathbf{x}(t)\middle|\mathbf{s}(t),\hat{\mathbf{A}}\right) P\left(\mathbf{s}(t)\right)$$

$$\propto \max_{\hat{\mathbf{A}}\mathbf{s}(t)=\mathbf{x}(t)} P\left(\mathbf{s}(t)\right)$$

$$= \max_{\hat{\mathbf{A}}\mathbf{s}(t)=\mathbf{x}(t)} \exp -\left(\left|\mathbf{s}_1(t)\right| + \ldots + \left|\mathbf{s}_M(t)\right|\right)$$

$$= \min_{\hat{\mathbf{A}}\mathbf{s}(t)=\mathbf{x}(t)} \left|\mathbf{s}_1(t)\right| + \ldots + \left|\mathbf{s}_M(t)\right|$$

$$= \min_{\hat{\mathbf{A}}\mathbf{s}(t)=\mathbf{x}(t)} \left\|\mathbf{s}(t)\right\|_1$$

uBSS – L_1 norm minimization

Sequence of MATLAB commands for solution of the problems **x**=**As** using command `linprog`:

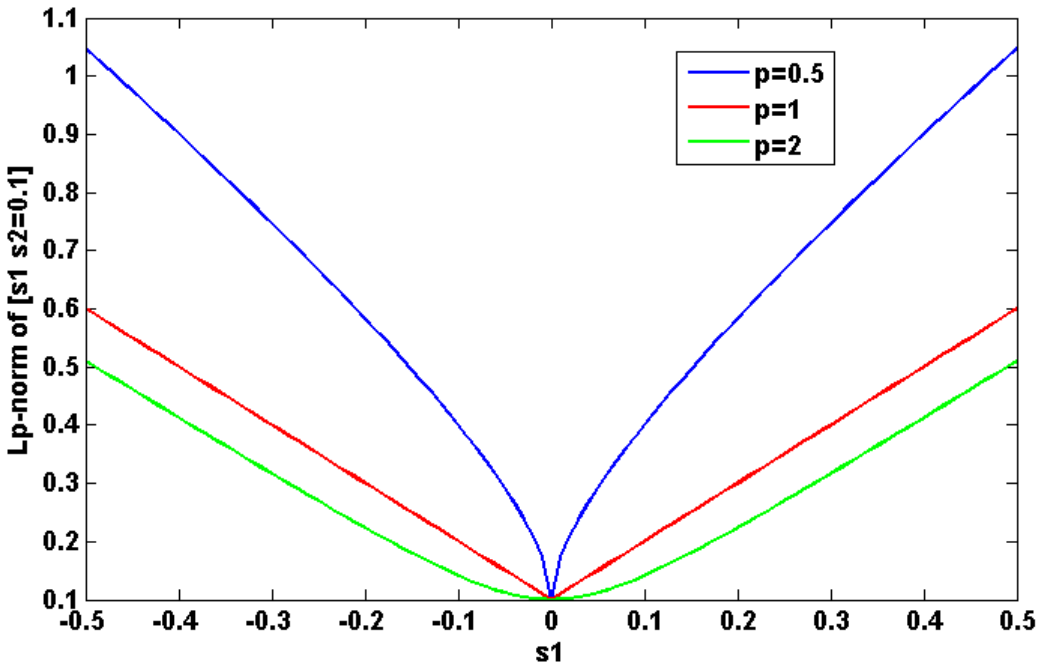
```
% Linear progamming solution
% solves linear program min(x) f'*x s.t. Ax=b, lb<=x<=ub.
f = ones(M,1);
lb = zeros(M,1);
ub = 1000*ones(M,1);

for m=1:T
    x=X(:,m);
    [sh,fval,exitflag,output]=linprog(f,[],[],A,x,lb,ub,[]);
    SH(:,m)=sh;
end
```

• What happens if $P(\mathbf{s})$ is not Lalpacian? For distributions $P(\mathbf{s})$ sparser than Laplacian, minimum L_1-norm approach will not yield the sparsest solution!!!!

uBSS – L_p norm minimization: $0< p \leq 1$

Minimizing L_p-norm, $0<p<1$, of **s** yields better performance when solving underdetermined system **x=As** than when using L_1-norm minimization.

This occurs despite the fact that minimization of L_p-norm, $0<p<1$ is non-convex problem. Yet, in practical setting (when noise or approximation errors are present) its local minimum can be smaller than global minimum of L_1 i.e. min L_p-norm solution is sparser than min L_1-norm solution.



L_p-norm of [s_1 0.1] : $\|\mathbf{s}\|_p = \left(\sum_{m=1}^{M} |S_m|_{22}^{p} \right)^{1/p}$

uBSS – L_p norm minimization: $0 < p \leq 1$

The idea of ref. [21] was to replace L_0-norm by continuous parametric approximation:

$$\|\mathbf{s}\|_0 \approx M - F_\sigma(\mathbf{s})$$

where:

$$F_\sigma(\mathbf{s}) = \sum_m f_\sigma(s_m)$$

and:

$$f_\sigma(s_m) = \exp\left(-\frac{s_m^2}{2\sigma^2}\right)$$

approximates indicator function of a set {0}.

21. H. Mohimani, M. Babaie-Zadeh, C. Jutten, "A fast approach for overcomplete sparse decomposition based on smoothed L_0 norm," IEEE Trans. Signal Process. 57 (2009) 289-301.

uBSS – L_p norm minimization: $0 < p \leq 1$

Smaller parameter σ brings us closer to $L_0(\mathbf{s})$, while larger σ yields smoother approximation that is easier to optimize.

Minimizing approximation of $L_0(\mathbf{s})$ is equivalent to maximize $F_\sigma(\mathbf{s})$. The idea is to maximize $F_\sigma(\mathbf{s})$ for large σ and than use obtained solution as initial value for next maximization of $F_\sigma(\mathbf{s})$ for smaller σ.

After each iteration computed approximation of $\mathbf{s}$ is projected back onto the constraining set $\mathbf{As}=\mathbf{x}$:

$$\mathbf{s} \leftarrow \mathbf{s} - \mathbf{A}^T \left(\mathbf{A}\mathbf{A}^T \right)^{-1} \left(\mathbf{A}\mathbf{s} - \mathbf{x} \right)$$

Matlab code for smooth L_0 algorithm can be downloaded from:

http://ee.sharif.ir/~SLzero/

uBSS – L_p norm minimization: $0 < p \leq 1$

- Initialization:
  1) Let $\hat{\mathbf{s}}_0$ be equal to the minimum ℓ^2 norm solution of $\mathbf{A}\mathbf{s} = \mathbf{x}$, obtained by pseudo-inverse of $\mathbf{A}$.
  2) Choose a suitable decreasing sequence for σ, $[\sigma_1 \ldots \sigma_J]$ (see Remarks 5 and 6 of the text).
- For $j = 1, \ldots, J$:
  1) Let $\sigma = \sigma_j$.
  2) Maximize (approximately) the function F_σ on the feasible set $\mathcal{S} = \{\mathbf{s} \mid \mathbf{A}\mathbf{s} = \mathbf{x}\}$ using L iterations of the steepest ascent algorithm (followed by projection onto the feasible set):
     - Initialization: $\mathbf{s} = \hat{\mathbf{s}}_{j-1}$.
     - For $\ell = 1 \ldots L$ (loop L times):
       a) Let $\boldsymbol{\delta} \triangleq [s_1 \exp(-s_1^2/2\sigma^2), \ldots, s_n \exp(-s_n^2/2\sigma^2)]^T$.
       b) Let $\mathbf{s} \leftarrow \mathbf{s} - \mu\boldsymbol{\delta}$ (where μ is a small positive constant).
       c) Project $\mathbf{s}$ back onto the feasible set $\mathcal{S}$:
       $$\mathbf{s} \leftarrow \mathbf{s} - \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}(\mathbf{A}\mathbf{s} - \mathbf{x}).$$
  3) Set $\hat{\mathbf{s}}_j = \mathbf{s}$.
- Final answer is $\hat{\mathbf{s}} = \hat{\mathbf{s}}_J$.

Fig. 1. Final SL0 algorithm.

uBSS – L_p norm minimization: $0 < p \leq 1$

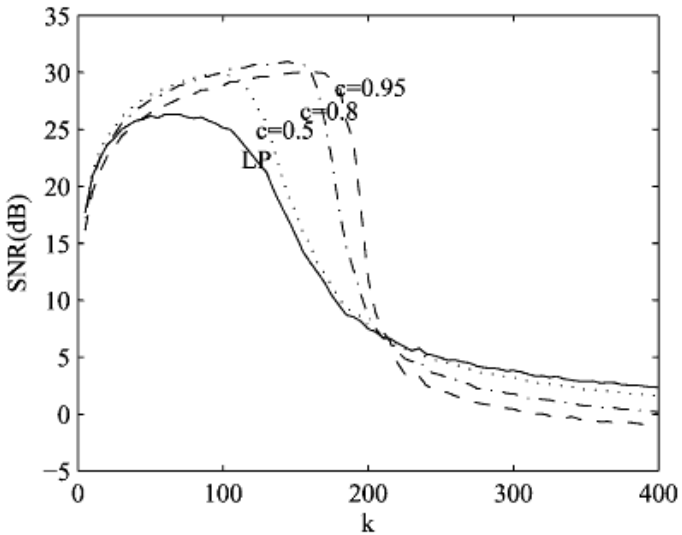


Fig. 6. Averaged SNRs (over 100 runs of the algorithm) versus k, the average number of active sources, for SL0 algorithm with several values of c, and for LP. The parameters are $m = 1000, n = 400, \sigma_1 = 1, \sigma_I = 0.01, \sigma_n = 0.01$.

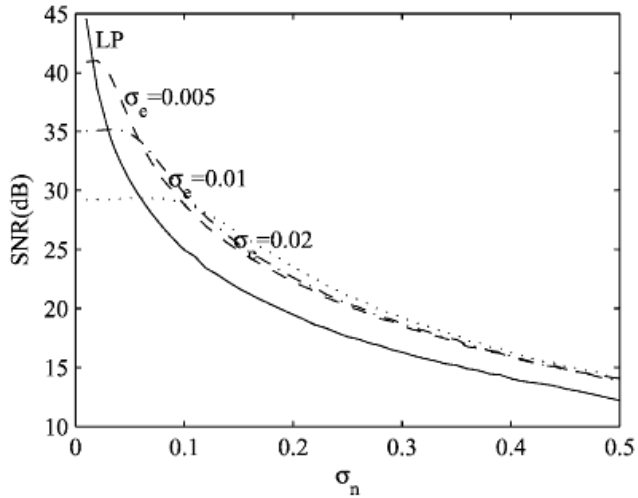


Fig. 7. Averaged SNRs (over 100 runs of the algorithm) versus the noise power σ_n for different values of σ_I, and for LP. The parameters are $m = 100, n = 400, k = 100, \sigma_1 = 1$, and $c = 0.8$.

$$SNR[dB] = 20\log\left(\frac{\mathbf{s}}{\mathbf{s}-\hat{\mathbf{s}}}\right)$$

Iteratively reweighted least square (IRLS) algorithm outline

$$\min \|\mathbf{s}\|_p \ \ s.t. \ \ \mathbf{As} = \mathbf{x} \ \rightarrow \min \sum_{m=1}^{M} w_m s_m^p$$

Initialize: ε=1, $\mathbf{s}^{(0)} = pinv(\mathbf{A})\mathbf{x}$, k=1.

do

repeat

$$w_m = \left(\left(s_m^{(k-1)} \right)^2 + \varepsilon \right)^{p/2-1}$$

$$\mathbf{Q}_k = diag\left\{ 1/w_m \right\}$$

$$\mathbf{s}^{(k)} = \mathbf{Q}_k \mathbf{A}^T \left(\mathbf{A}\mathbf{Q}_k \mathbf{A}^T \right)^{-1} \mathbf{x}$$

$$k = k+1$$

until $\left\| \mathbf{s}^{(k)} - \mathbf{s}^{(k-1)} \right\|_2 < \sqrt{\varepsilon}/100$

ε=ε/10

while ε>10^{-8}

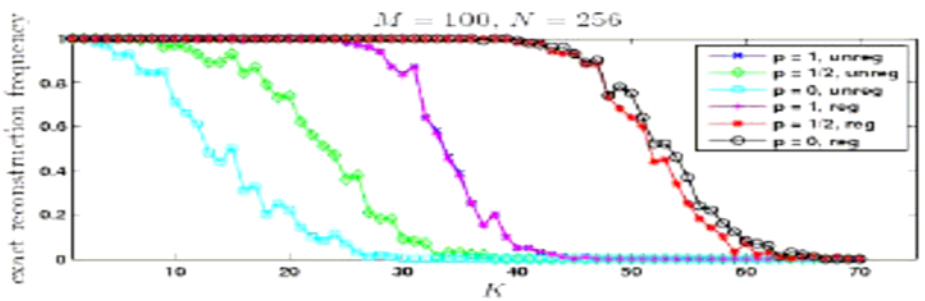


Fig. 1. Plots of recovery frequency as a function of K. Regularized IRLS has a much higher recovery rate than unregularized IRLS, except when $p = 1$ when they are almost identical. Regularized IRLS recovers the greatest range of signals when p is small, while unregularized IRLS performs less well for small p than when $p = 1$.

22. R. Chartrand, Exact reconstructions of sparse signals via nonconvex minimization, IEEE Signal Process. Let., 14 (2007), 707-710.
23. I. Daubechies, R. Devore, M. Fornassier,C. S. Gunturk "Iteratively reweigghted least squares minimization for sparse recovery," Communications on Pure and Applied Mathematics,vol. LXIII (2010) 1-38.

Iterative soft/hard thresholding

L_1-regularized least square problem:

$$\hat{\mathbf{s}}(t) = \arg\min_{\mathbf{s}(t)} \frac{1}{2}\left\|\hat{\mathbf{A}}\mathbf{s}(t) - \mathbf{x}(t)\right\|_2^2 + \lambda \left\|\mathbf{s}(t)\right\|_1 \qquad \forall\, t = 1,\ldots,T$$

can be reformulated within analytic soft thresholding representation [24, 25]:

$$B(\mathbf{s}^{(k)}(t)) = \mathbf{s}^{(k)} + \mathbf{A}^T\left(\mathbf{x}(t) - \mathbf{A}\mathbf{s}^{(k)}(t)\right)$$

$$s_m^{(k+1)}(t) = \begin{cases} B(\mathbf{s}^{(k)}(t))_m - sign(\,B(\mathbf{s}^{(k)}(t))_m)\lambda/2, & \left|B(\mathbf{s}^{(k)}(t))_m\right| > \lambda/2 \\ 0, \quad otherwise \end{cases}$$

where $\lambda=\sigma^2$ provided that error term (noise) has normal distribution. Otherwise some kind of cross-validation (trial and error) needs to be applied.

24. D. L. Donoho, Denoising by soft-thresholding, IEEE Trans. Information Theory, 41 (1995), 613-627.
25. I. Daubechies, M. Defrise, D.M. Christine, An iterative thresholding algorithm for linear inverse problems with a sparsity constraint, Comm. Pure and Appl. Math., LVII (2004) 1413-1457.

Iterative soft/hard thresholding

L_0-regularized least square problem:

$$\hat{\mathbf{s}}(t) = \arg\min_{\mathbf{s}(t)} \frac{1}{2}\left\|\hat{\mathbf{A}}\mathbf{s}(t) - \mathbf{x}(t)\right\|_2^2 + \lambda\left\|\mathbf{s}(t)\right\|_0 \qquad \forall t = 1,\ldots,T$$

can be reformulated within analytic hard thresholding representation [26]:

$$s_m^{(k+1)}(t) = \begin{cases} s_m^{(k)}(t) - sign(s_m^{(k)}(t))\lambda/2, & \left|s_m^{(k)}(t)\right| > \lambda/2 \\ 0, \quad otherwise \end{cases}$$

where $\lambda=\sigma^2$ provided that error term (noise) has normal distribution. Otherwise some kind of cross-validation (trial and error) needs to be applied.

26. R. Chartrand, V. Staneva, Restricted isometry properties and nonconvex compressive sensing, Inverse Problems, 24 (2008) 1-14.

Iterative soft/hard thresholding

Very recently it has been proven in [27] $L_{1/2}$-regularizer is the most sparse and robust among L_p regularizers when $1/2 \le p < 1$, and when $0 < p < 1/2$, the L_p regularizers have similar properties as the $L_{1/2}$ regularizer. In [27] it is shown that solution of:

$$\hat{\mathbf{s}}(t) = \arg\min_{\mathbf{s}(t)} \frac{1}{2}\left\|\hat{\mathbf{A}}\mathbf{s}(t) - \mathbf{x}(t)\right\|_2^2 + \lambda \sum_{m=1}^{M} \left|s_m(t)\right|^{1/2} \qquad \forall\, t = 1,\ldots,T$$

can be converted to a series of convex weighted L_1 regularized problems:

$$\mathbf{s}^{(k+1)}(t) = \arg\min_{\mathbf{s}(t)} \frac{1}{2}\left\|\hat{\mathbf{A}}\mathbf{s}(t) - \mathbf{x}(t)\right\|_2^2 + \lambda \sum_{m=1}^{M} \frac{1}{\sqrt{\left|s_m^{(k)}(t)\right|}} \left|s_m(t)\right| \qquad \forall\, t = 1,\ldots,T$$

27. X. ZongBen, Z. Hai, W. Yao, C. XiangYu, L. Yong, $L_{1\text{-}2}$ regularization, Science China, series F, 53 (2010) 1159-1169.

Iterative soft/hard thresholding

It has been further derived in [28] a fast solver for $L_{1/2}$-regularized problems based on thresholding representation theory. It is proven in [28] that an analytically expressive thresholding representation exists among all L_p-regularizes $0<p<1$ only for $p=1/2$.

$$\mathbf{u}^{(k)}(t) = B_\mu(\mathbf{s}^{(k)}(t)) = \mathbf{s}^{(k)}(t) + \mu\mathbf{A}^T\left(\mathbf{x}(t) - \mathbf{A}\mathbf{s}^{(k)}(t)\right) \quad \forall t = 1,\ldots,T \qquad 0 < \mu \le \|\mathbf{A}\|_2^{-1}$$

$$\mathbf{s}^{(k+1)}(t) = \mathbf{H}_{\lambda\mu,1/2}\left(\mathbf{u}^{(k)}(t)\right)$$

$$\mathbf{H}_{\lambda\mu,1/2}\left(\mathbf{u}^{(k)}(t)\right) = \left[h_{\lambda\mu,1/2}(u_1^{(k)}(t)) \ldots h_{\lambda\mu,1/2}(u_M^{(k)}(t))\right]^T$$

$$h_{\lambda\mu,1/2}(u_m^{(k)}(t)) = \begin{cases} \frac{2}{3}u_m^{(k)}(t)\left(1+\cos\left[\frac{2\pi}{3} - \frac{2\varphi_\lambda(u_m^{(k)}(t))}{3}\right]\right), & u_m^{(k)}(t) > \frac{\sqrt[3]{54}}{4}(\lambda\mu)^{2/3} \\ 0, & otherwise \end{cases}$$

$$\varphi_\lambda(u_m^{(k)}(t)) = \arccos\left(\frac{\lambda}{8}\left(\frac{|u_m^{(k)}(t)|}{3}\right)^{-3/2}\right) \qquad \lambda^{*(k)} = \frac{\sqrt{96}}{9\mu_0}\left|\left[B_{\mu_0}\left(\mathbf{s}^{(k)}(t)\right)\right]_{\mathrm{supp}(\mathbf{s})+\mathbf{1}}\right|^{3/2} \quad for\ some\ fixed\ \mu_0 > 0.$$

28. Z. Xu, X. Chang, F. Xu, H. Zhang, L_{1-2} Regularization: A Thresholding Representation Theory and Fast Solver, *accepted* for IEEE Tr. Neural Networks.

Estimation of the mixing matrix: single component points

Accuracy of the estimation of the mixing matrix **A** can be improved significantly when it is estimated on a set of single component points i.e. points where only one component/source is active, ref. [11,12].

At such "*t*" points of single source activity the following relation holds:

$$\mathbf{x}_t = \mathbf{a}_j s_{jt}$$

where *j* denotes the source index that is active at point "*t*", i.e. at these points the mixing vector $\mathbf{a}_j$ is collinear with data vector $\mathbf{x}_t$. It is assumed that data vector and source components are complex. If not, Hilbert transform-based analytical expansion can be used to obtain complex representation, ref. [29].

29. I.Kopriva, I. Jerić, "Blind separation of analytes in nuclear magnetic resonance spectroscopy and mass spectrometry: sparseness-based robust multicomponent analysis," Analytical Chemistry **82**: 1911-1920 (2010).

Estimation of the mixing matrix: single component points

If single source points can not be find in original domain a linear transform such as wavelet transform, Fourier transform or Short-time Fourier transform can be used to obtain sparser representation:

$$T(\mathbf{x})_t = \boldsymbol{a}_j T(\mathrm{s}_j)_t$$

Since the mixing vector is real, the real and imaginary part of data vector $\mathbf{x}_t$ must point in the same direction when real and imaginary part of s_{jt} have the same sign. Otherwise, they must point into opposite directions. Thus, such points can be identified using:

$$\left| \frac{R\{\mathbf{x}_t\}^{\mathrm{T}} I\{\mathbf{x}_t\}}{\|R\{\mathbf{x}_t\}\| \|I\{\mathbf{x}_t\}\|} \right| \geq \cos(\Delta\theta)$$

where $R\{\mathbf{x}_t\}$ and $I\{\mathbf{x}_t\}$ denote real and imaginary part of $\mathbf{x}_t$, and $\Delta\theta$ denotes angular displacement from a direction of 0 or π radians.

Estimation of the mixing matrix: clustering

Assuming unit L_2-norm of $\boldsymbol{a}_m$ and N=2 we can parameterize column vectors in a plane by one angle

$$\mathbf{a}_m = [\cos(\varphi_m) \;\; \sin(\varphi_m)]^{\mathrm{T}}$$

Assuming that **s** is *1*-sparse in representation domain estimation of **A** and M is obtained by means of data clustering algorithm, [10]:

We remove all data points close to the origin for which applies: $\left\{\left|\mathbf{x}(t)\right|_2 \le \varepsilon\right\}_{t=1}^{T}$ where ε represents some predefined threshold.

Normalize to unit L_2-norm remaining data points $\mathbf{x}(t)$, i.e., $\left\{\mathbf{x}(t) \to \mathbf{x}(t) \big/ \left|\mathbf{x}(t)\right|_2\right\}_{t=1}^{\bar{T}}$

Estimation of the mixing matrix: clustering

Calculate function *f*(**a**):

$$f(\mathbf{a}) = \sum_{t=1}^{\bar{T}} \exp\left(-\frac{d^2(\mathbf{x}(t), \mathbf{a})}{2\sigma^2} \right)$$

where $d(\mathbf{x}(t), \mathbf{a}) = \sqrt{1 - (\mathbf{x}(t) \cdot \mathbf{a})^2}$ and $(\mathbf{x}(t) \cdot \mathbf{a})$ denotes inner product. Parameter σ is called dispersion. If set to sufficiently small value, in our experiments this turned out to be $\sigma \approx 0.05$, the value of the function *f*(**a**) will approximately equal the number of data points close to **a**. Thus by varying mixing angle φ we effectively cluster data.

• Number of peaks of the function *f*(**a**) corresponds with the estimated number of materials *M*. Locations of the peaks correspond with the estimates of the mixing angles $\{(\hat{\varphi}_m)\}_{m=1}^{\hat{M}}$, i.e., mixing vectors $\{\hat{\mathbf{a}}_m\}_{m=1}^{\hat{M}}$.

Estimation of the mixing matrix: clustering

• hierarchical clustering by MATLAB function `clusterdata`. It is assumed that number of clusters (sources) is given (known). The method is deterministic and memory demanding.

•*k*-means clustering by MATLAB function `kmeans`. It is assumed that a number of clusters *M* (corresponds with number of sources) is given. For each dana point $\mathbf{x}(t)$ to assign to cluster *m* we need to assign a binary indicator variable $r_{tm}=1$ and $r_{tj}=0$ for $j\neq m$. That is known as 1-of-*M* coding scheme. We also defined a prototype cluster centers: μ_m, *m=1,…,M.* The objective function known as a *distortion measure* is defined:

$$J=\sum_{t=1}^{\bar{T}}\sum_{m=1}^{M} r_{tm}\left\|\mathbf{x}(t)-\boldsymbol{\mu}_m\right\|_2^2 \quad r_{tm}=\begin{cases}1 & if \quad m=\arg\min_j \left\|\mathbf{x}(t)-\boldsymbol{\mu}_j\right\| \\ 0 & otherwise\end{cases}$$

For fixed r_{tm} we can solve for μ_m from:

$$\frac{\partial J}{\partial \boldsymbol{\mu}_m}=0 \quad \rightarrow \quad \boldsymbol{\mu}_m=\sum_{t=1}^{\bar{T}} r_{tm}\mathbf{x}(t) \Big/ \sum_{t=1}^{\bar{T}} r_{tm}$$

Since, *k*-means clustering is a first order method it is sensitive on initial choice of μ_m, *m=1,…,M.*

Estimation of the mixing matrix: clustering

• Mean shift clustering [30-32]. Let $\left\{\mathbf{x}_0(t) \in \mathbb{R}^N\right\}_{t=1}^{\bar{T}_0}$ be a dataset to be clustered. A kernel based density estimate at some point $\mathbf{x}_0(t)$ is:

$$p_{\mathbf{x}_0,\sigma}\left(\mathbf{x}_0(t)\right) = \frac{1}{\bar{T}_0}\sum_{j=1}^{\bar{T}_0} G_\sigma\left(\mathbf{x}_0(t) - \mathbf{x}_0(j)\right) \qquad \textit{where} \quad G_\sigma\left(\mathbf{x}_0(t) - \mathbf{x}_0(j)\right) = \exp\left(-\frac{\left\|\mathbf{x}_0(t) - \mathbf{x}_0(j)\right\|_2^2}{\sigma^2}\right)$$

The initial dataset $\mathbf{X}_0$ is transformed into series of steps. Let the Reny's cross entropy between the current dataset $\mathbf{X}$ and initial dataset $\mathbf{X}_0$ be defined as:

$$H\left(\mathbf{X}, \mathbf{X}_0\right) = -\log\int\left(p^2\left(\mathbf{X}, \mathbf{X}_0\right)\mathbf{dX}\right) \cong -\log\frac{1}{\bar{T}\bar{T}_0}\sum_{i=1}^{\bar{T}}\sum_{j=1}^{\bar{T}_0} G_\sigma\left(\mathbf{x}(i) - \mathbf{x}_0(j)\right)$$

30. S.Rao, A. Medeiros Martins, J. C. Principe, "Mean shift: An information theoretic perspective," Pattern Recognition Letters **30**: 222-230 (2009).
31. D. Comaniciu, P. Meer, "Mean Shift: A Robust Approach Toward Feature Space Analysis," IEEE Trans. Patt. Anal. Machine Intell. **24**: 603-619 (2002).
32. Y. Cheng, "Mean Shift, Mode Seeking, and Clusterin," IEEE Trans. Patt. Anal. Machine Intell. **17**: 790-799 (1995).

Estimation of the mixing matrix: clustering

The purpose of transformation series is to move $\mathbf{X}_0$ into $\mathbf{X}$ such that all the samples converge toward modes of $p_{\mathbf{X}0}$. Hence original dataset $\mathbf{X}_0$ will be smoothed while cross entropy between $\mathbf{X}_0$ and $\mathbf{X}$ being minimal ($\mathbf{X}$ is determined completely by series of transformations and $\mathbf{X}_0$. Hence the current movement is obtained by maximizing argument of $H(\mathbf{X},\mathbf{X}_0)$:

$$J(\mathbf{X}) = \max_{\mathbf{x}} \frac{1}{\bar{T}\bar{T}_0} \sum_{i=1}^{\bar{T}} \sum_{j=1}^{\bar{T}_0} G_\sigma\left(\mathbf{x}(i) - \mathbf{x}_0(j)\right)$$

$$\frac{\partial}{\partial \mathbf{x}(t)} J(\mathbf{X}) = \frac{1}{\bar{T}\bar{T}_0} \sum_{j=1}^{\bar{T}_0} G_\sigma\left(\mathbf{x}(t) - \mathbf{x}_0(j)\right)\left(\frac{\mathbf{x}_0(j) - \mathbf{x}(t)}{\sigma^2}\right) = \frac{1}{\bar{T}} \nabla p_{\mathbf{x}_0,\sigma}\left(\mathbf{x}(t)\right) = 0$$

Hence, maximization of $J(\mathbf{X})$ is in the direction of the gradient of density $p_{\mathbf{X}0}$ i.e. the current sample $\mathbf{x}(t)$ is moved toward mode of $p_{\mathbf{X}0}$. The algorithm is expected to has sel-fstopping capability at the modes (the gradient is zero) and it will move samples where associated probability is low (the gradient is maximal) fast toward the modes of $p_{\mathbf{X}0}$. Hence, the *mean shift* algorithm that follows is gradient ascent algorithm with automatically adjustable step size, whereas the gradinet is actually never computed explicitly.

Estimation of the mixing matrix: clustering

It follows from $\partial J(\mathbf{X})/\partial \mathbf{x}(t)=0$:

$$\mathbf{x}(t)^{(k+1)} = m(\mathbf{x}(t)^{(k)}) = \frac{\sum_{j=1}^{\bar{T}} G_\sigma\left(\mathbf{x}(t)^{(k)} - \mathbf{x}_0(j)\right)\mathbf{x}_0(j)}{\sum_{j=1}^{\bar{T}} G_\sigma\left(\mathbf{x}(t)^{(k)} - \mathbf{x}_0(j)\right)}$$

that is the sample mean at $\mathbf{x}(t)$. The term $m(\mathbf{x}(t)^{(k)}) - \mathbf{x}(t)^{(k)}$ is called the *mean shift*. It follows:

$$\mathbf{x}(t)^{(k+1)} - \mathbf{x}(t)^{(k)} = m(\mathbf{x}(t)^{(k)}) - \mathbf{x}(t)^{(k)} = \sigma^2 \nabla_{\mathbf{x}(t)} \log p_{\mathbf{X}_0,\sigma}(\mathbf{X})$$

Thus, the samples are moved in the direction of normalized density gradient with increasing density values (modes). Algorithm is stopped when:

$$\frac{1}{\bar{T}} \sum_{i=1}^{\bar{T}} d^{(k)}\left(\mathbf{x}(i)\right) < tol$$

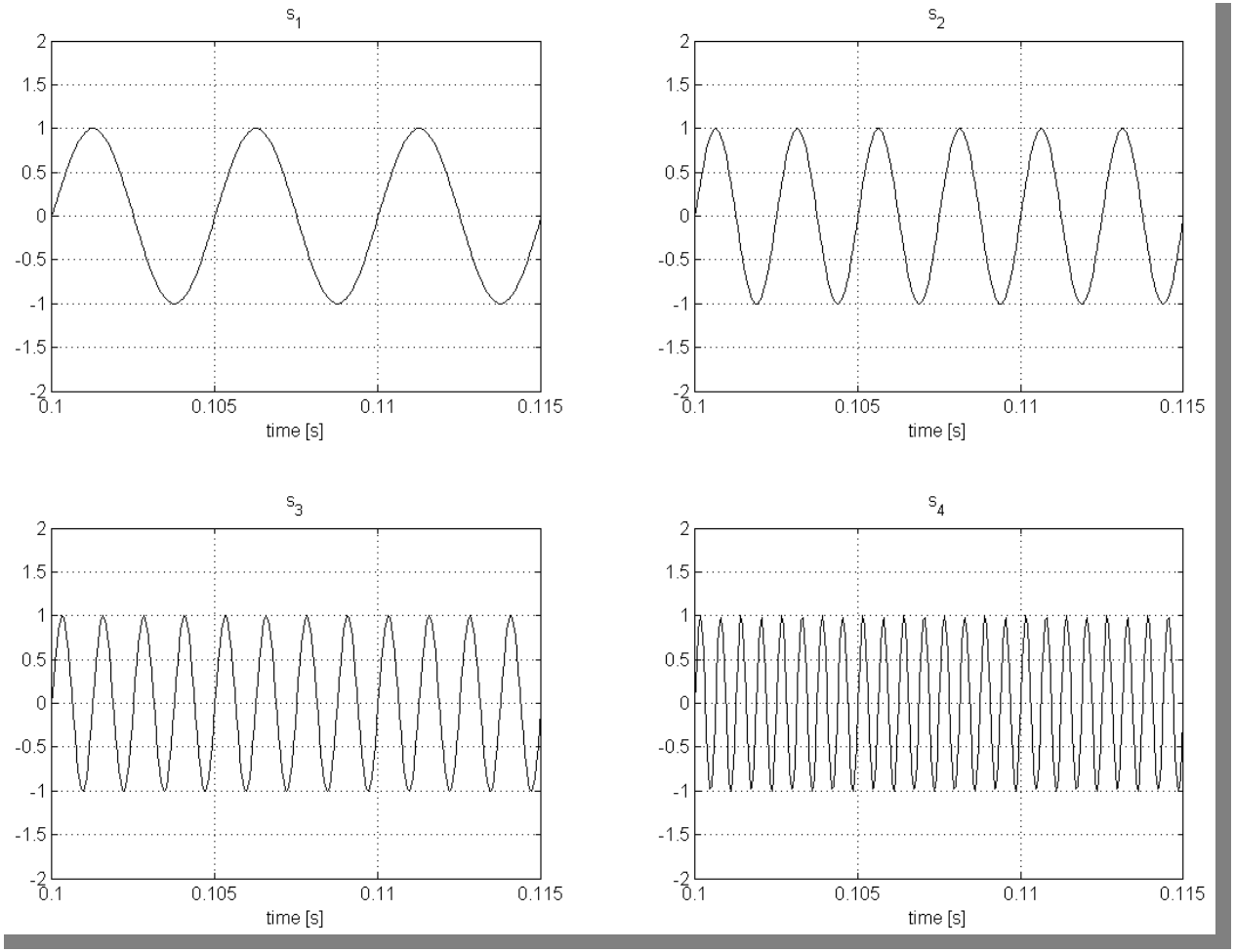
$$d^{(k)}\left(\mathbf{x}(i)\right) = \left\| \mathbf{x}^{(k)}(i) - \mathbf{x}^{(k-1)}(i) \right\|_2$$

where for example $tol=10^{-6}$. The number of modes toward algorithm converged determenis number of clusters (sources). Centers of the clusters represent the mixing vectors.

Blind separation of four sine signals from two mixtures

Four sinusoidal signals with frequencies 200Hz, 400Hz, 800Hz and 1600Hz.

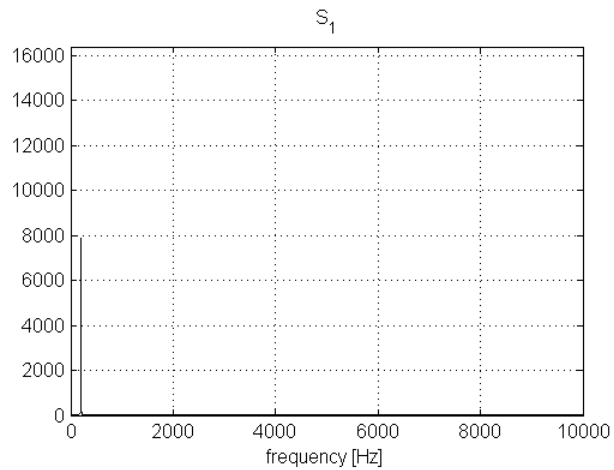
TIME DOMAIN

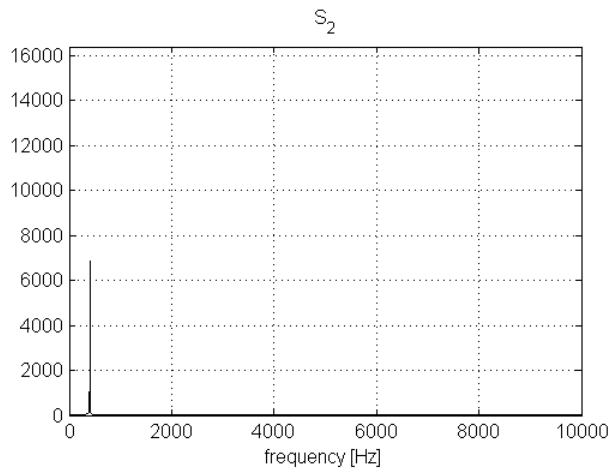


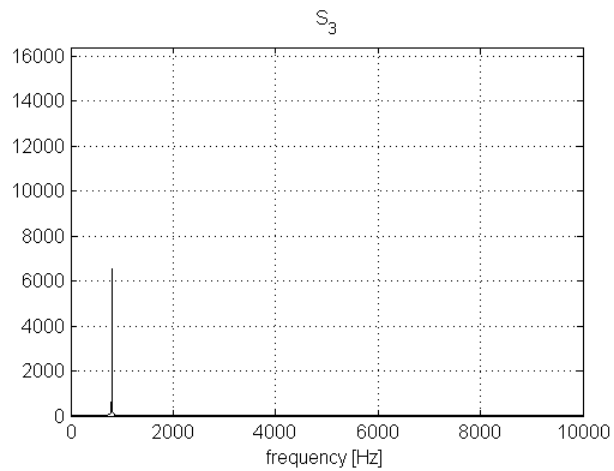
Blind separation of four sine signals from two mixtures

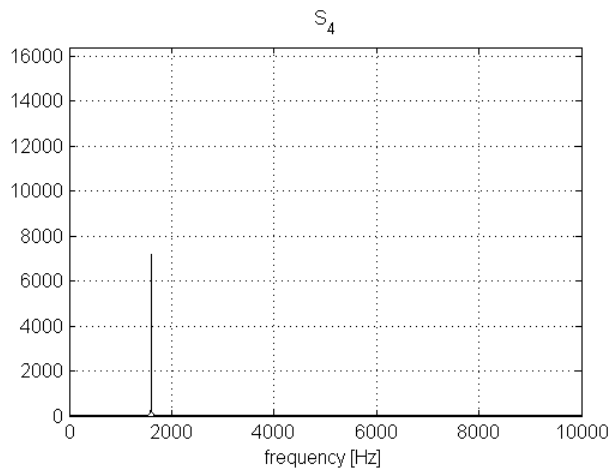
Four sinusoidal signals with frequencies 200Hz, 400Hz, 800Hz and 1600Hz.

FREQUENCY DOMAIN



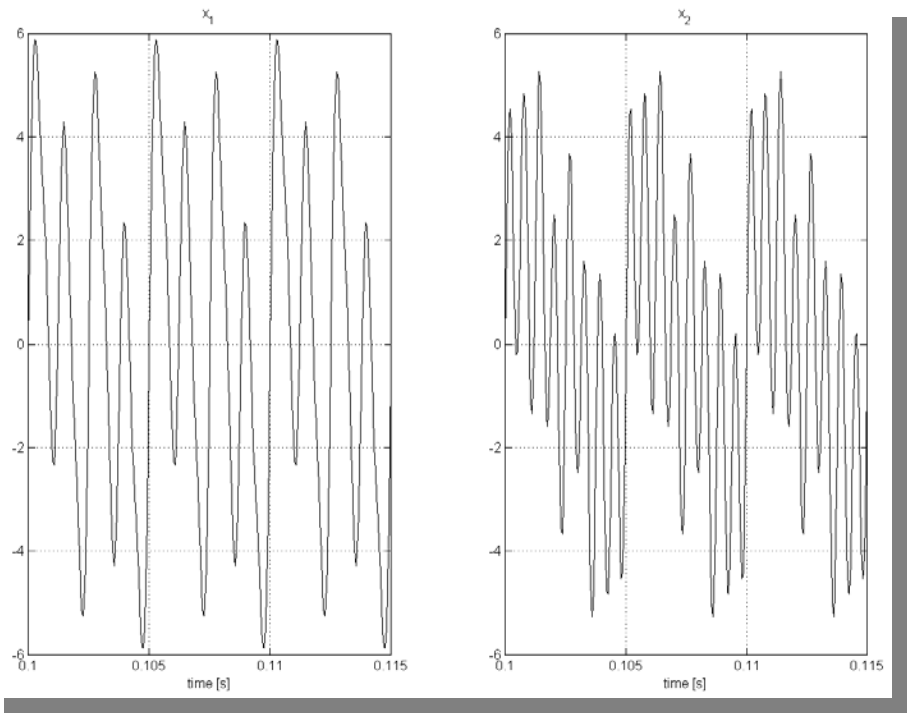


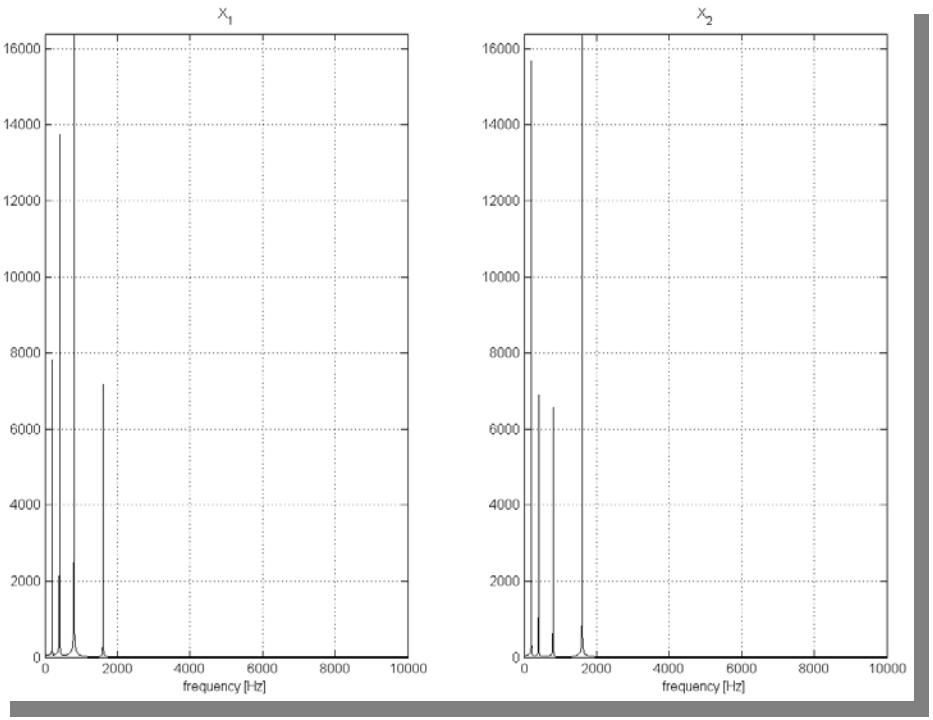




Blind separation of four sine signals from two mixtures

Two mixed signals



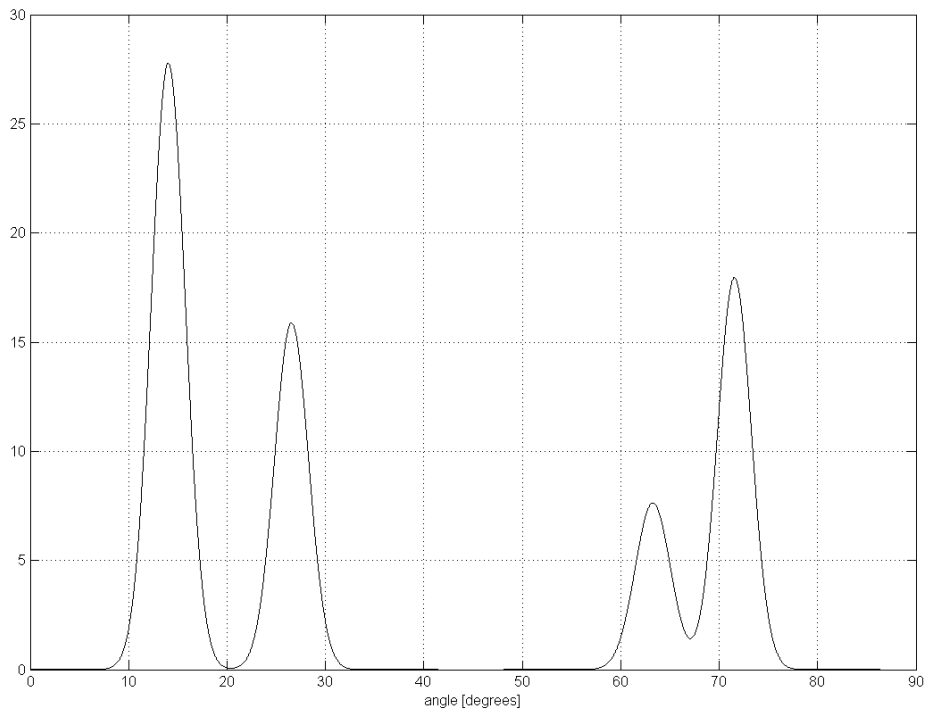


TIME DOMAIN

FREQUENCY DOMAIN

Blind separation of four sine signals from two mixtures

Clustering function



A=[63.44^0 26.57^0 14.04^0 71.57^0]

AH=[14.03^0 26.55^0 63.26^0 71.55^0]

Blind separation of four sine signals from two mixtures

Linear programming based estimation of the sources based on estimated mixing matrix **A**

$$\begin{bmatrix} \mathbf{x}_r(\omega) \\ \mathbf{x}_i(\omega) \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{A} \end{bmatrix} \begin{bmatrix} \mathbf{s}_r(\omega) \\ \mathbf{s}_i(\omega) \end{bmatrix}$$

or:

$$\overline{\mathbf{x}}(\omega) = \overline{\mathbf{A}}\,\overline{\mathbf{s}}(\omega)$$

$\mathbf{s}_r(\omega)$ and $\mathbf{s}_i(\omega)$ are not necessarily nonnegative. Thus, constraint $\overline{\mathbf{s}}(\omega) \geq \mathbf{0}$ required by linear program is not satisfied. In such a case it is customary to introduce dummy variables: $\mathbf{u},\mathbf{v} \geq \mathbf{0}$, such that $\overline{\mathbf{s}}(\omega) = \mathbf{u} - \mathbf{v}$.

Blind separation of four sine signals from two mixtures

Introducing:

$$\mathbf{z}(\omega)=\begin{pmatrix}\mathbf{u}\\ \mathbf{v}\end{pmatrix} \qquad \tilde{\mathbf{A}}=\begin{bmatrix}\overline{\mathbf{A}} & -\overline{\mathbf{A}}\end{bmatrix}$$

yields:

$$\hat{\mathbf{z}}(\omega)=\arg\min_{\mathbf{z}(\omega)}\sum_{m=1}^{4M} z_m(\omega) \quad \text{s.t.} \quad \tilde{\mathbf{A}}\mathbf{z}(\omega)=\overline{\mathbf{x}}$$

$$\mathbf{z}(\omega)\geq\mathbf{0}$$

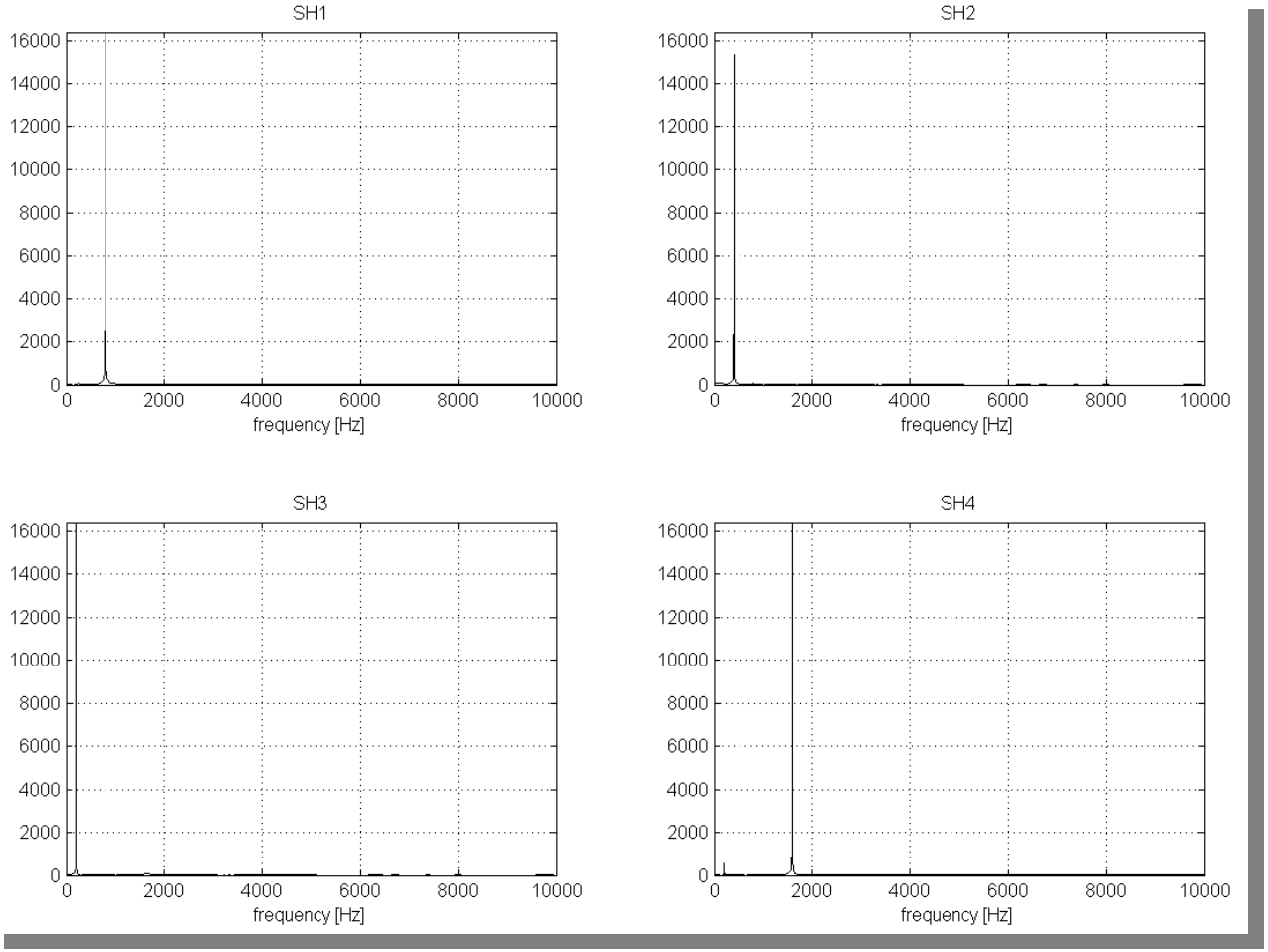
We obtain $\overline{\mathbf{s}}(\omega)$ from $\hat{\mathbf{z}}(\omega)$ as:

$$\overline{\mathbf{s}}(\omega)=\hat{\mathbf{u}}-\hat{\mathbf{v}}$$

and s(t) as:

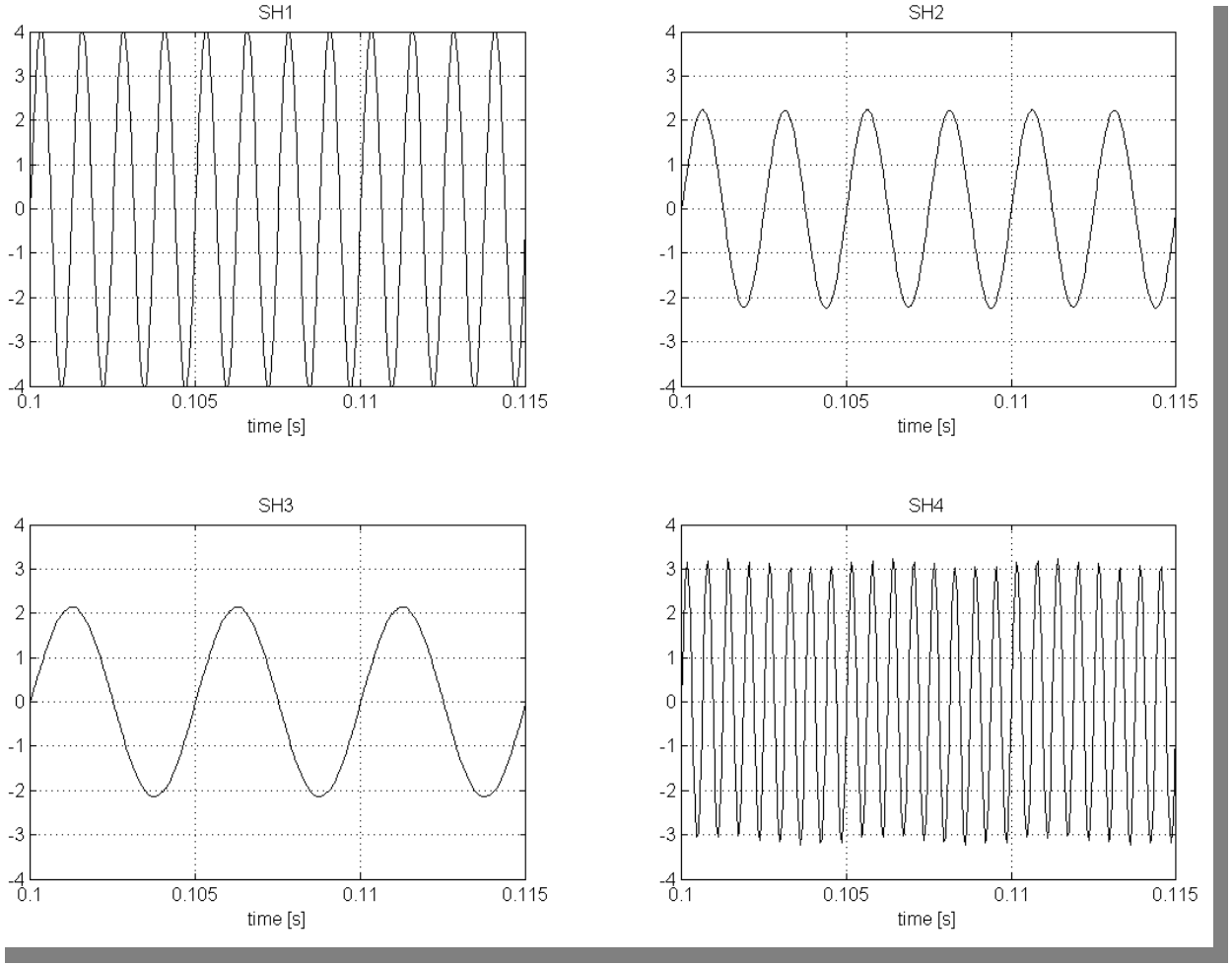
$$s_m(t)=IDFT\left[\overline{s}_m(\omega)\right]$$

Blind separation of four sine signals from two mixtures



Magnitudes of the estimated sources in FREQUENCY DOMAIN

Blind separation of four sine signals from two mixtures

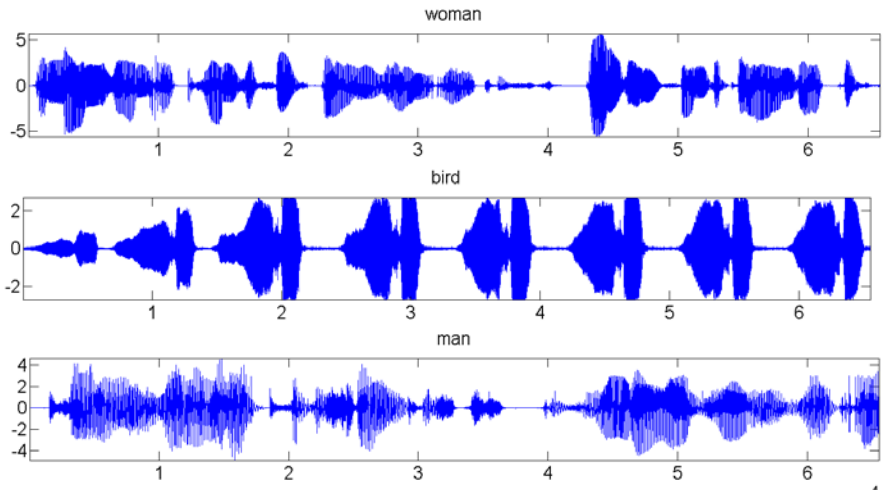


Estimated sources in TIME DOMAIN

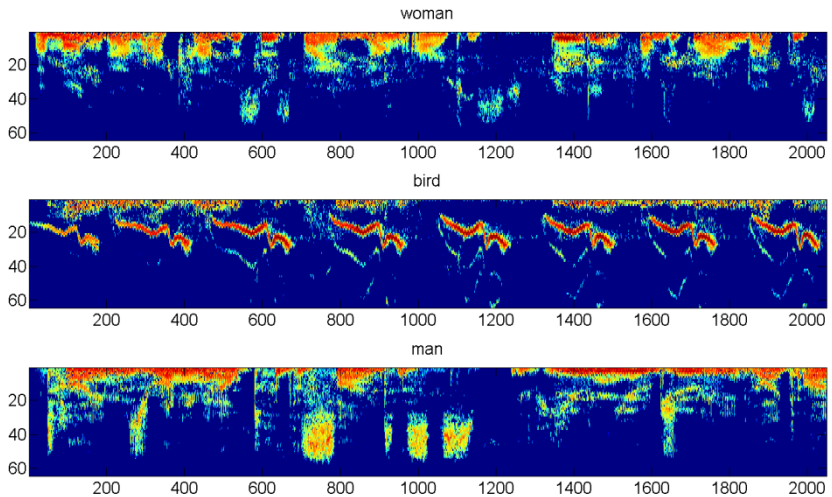
Blind separation of three sounds from two mixtures

Blind separation of three sounds from two mixtures

Three source signals are female and male voice and bird’s sound:



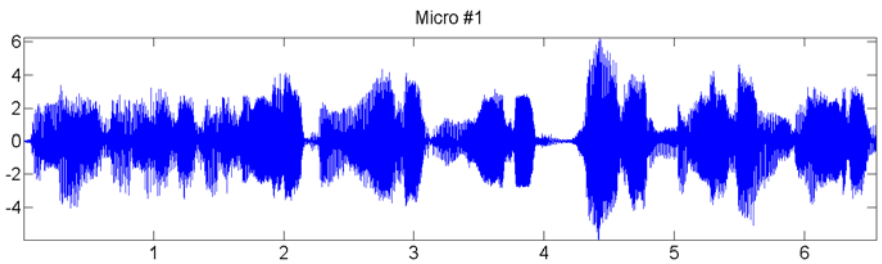
Time domain waveforms

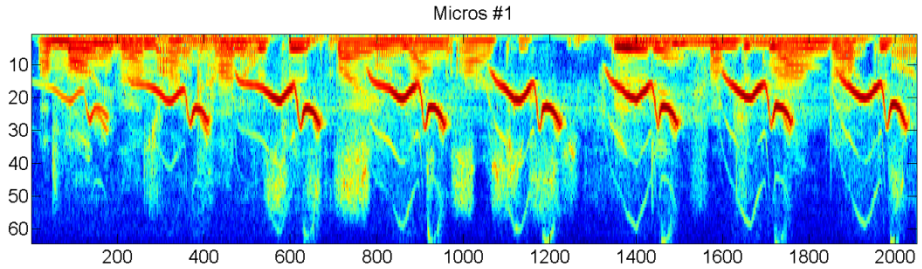


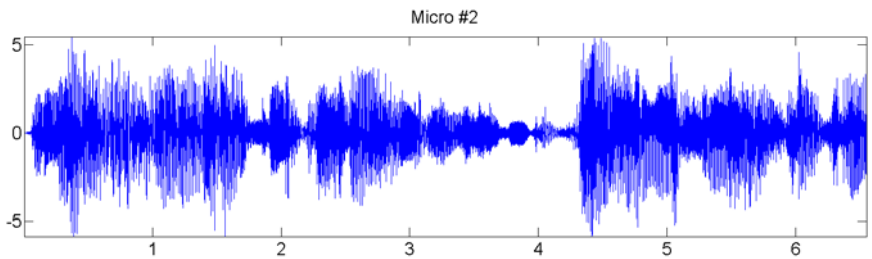
Time-frequency representations

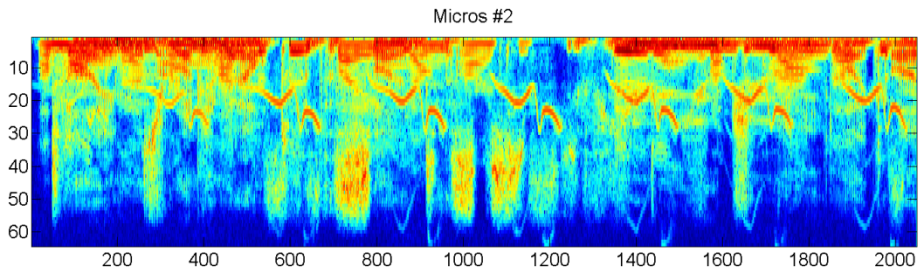
Blind separation of three sounds from two mixtures

Two mixtures of sounds:







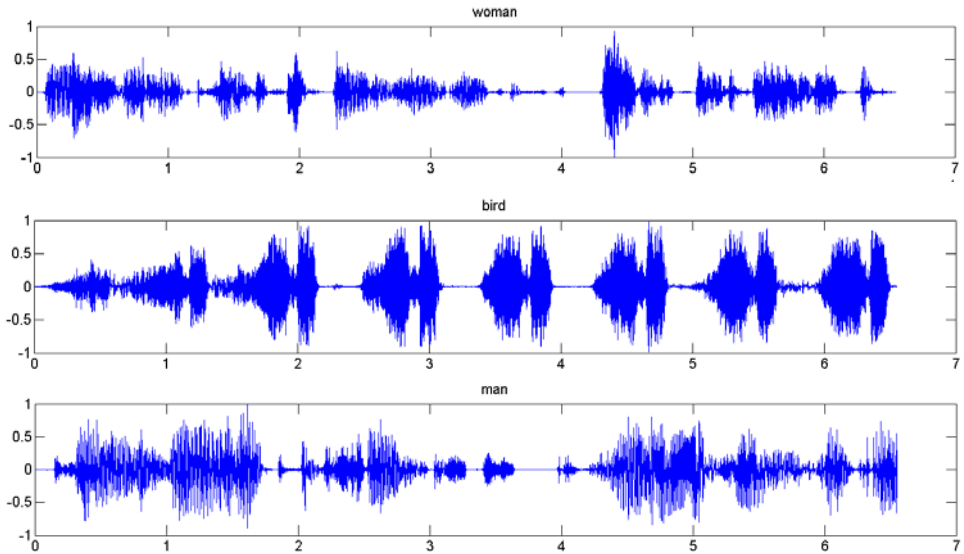


Time domain waveforms

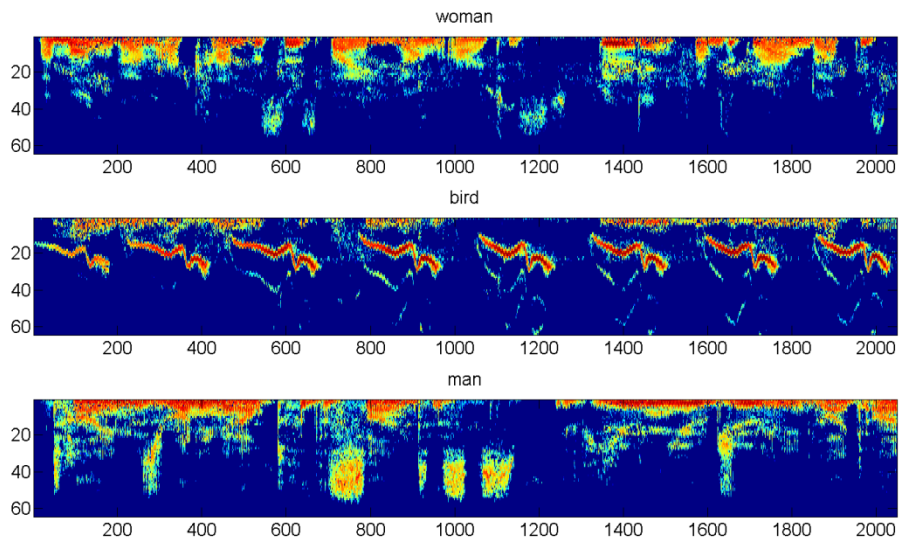
Time-frequency representations

Blind separation of three sounds from two mixtures

Three extracted sounds combining clustering on a set of single source points and linear programming in time-frequency domain:



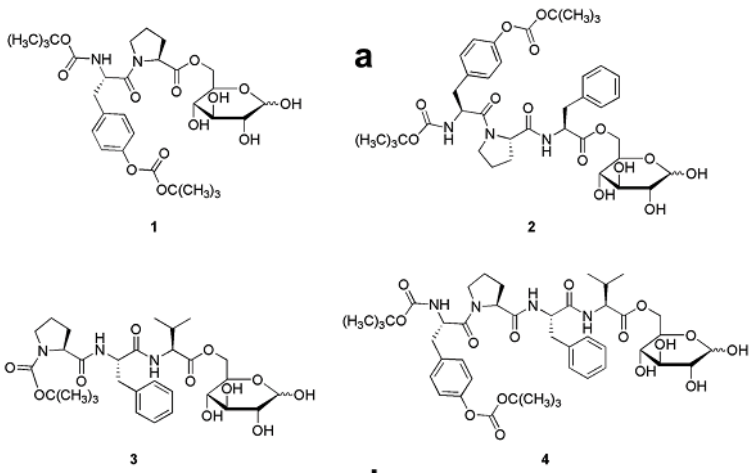
Time domain waveforms



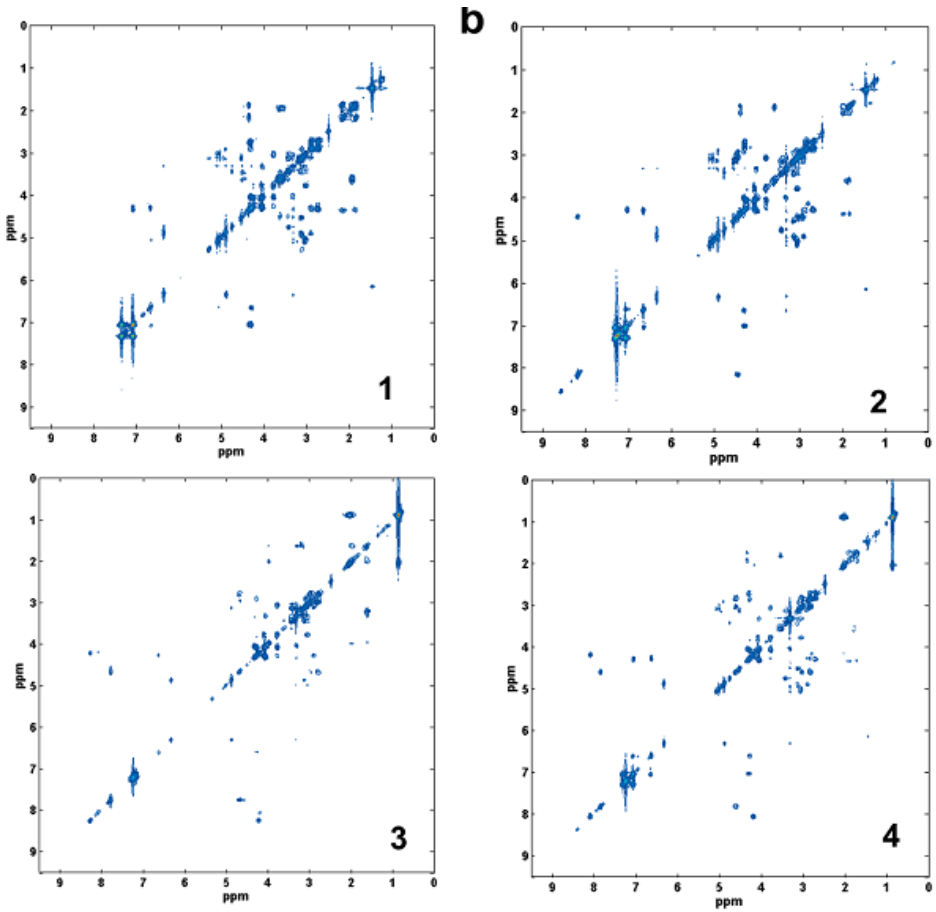
Time-frequency representations

Blind extraction of analytes (pure components) from mixtures of chemical compounds in NMR spectroscopy and mass spectrometry

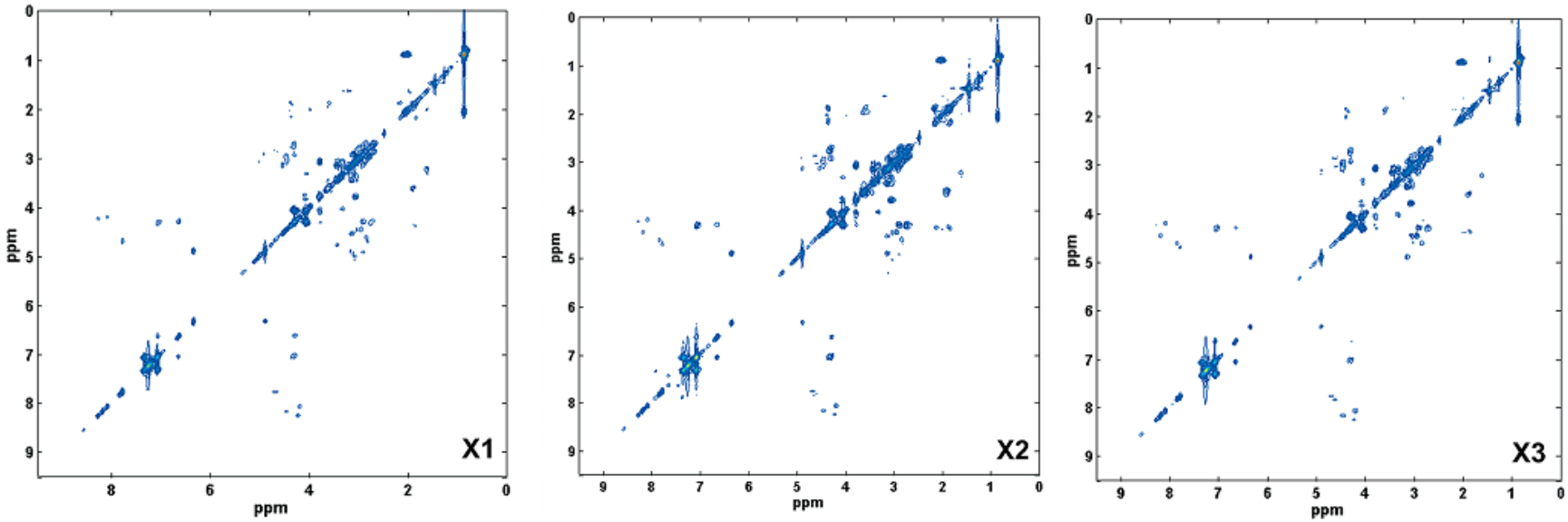
I. Kopriva, I. Jerić (2010). Blind separation of analytes in nuclear magnetic resonance spectroscopy and mass spectrometry: sparseness-based robust multicomponent analysis, **Analytical Chemistry 82**:1911-1920 (IF: 5.71).
I. Kopriva, I. Jerić, V. Smrečki (2009). Extraction of multiple pure component ^{1}H and ^{13}C NMR spectra from two mixtures: novel solution obtained by sparse component analysis-based blind decomposition, **Analytica Chimica Acta**, vol. 653, pp. 143-153 (IF: 3.14).
I. Kopriva, I. Jerić (2009). Multi-component Analysis: Blind Extraction of Pure Components Mass Spectra using Sparse Component Analysis, **Journal of Mass Spectrometry**, vol. 44, issue 9, pp. 1378-1388 (IF: 2.94).
I. Kopriva, I. Jerić, A. Cichocki (2009). Blind Decomposition of Infrared Spectra Using Flexible Component Analysis," **Chemometrics and Intelligent Laboratory Systems** 97 (2009) 170-178 (IF: 1.94)**.**



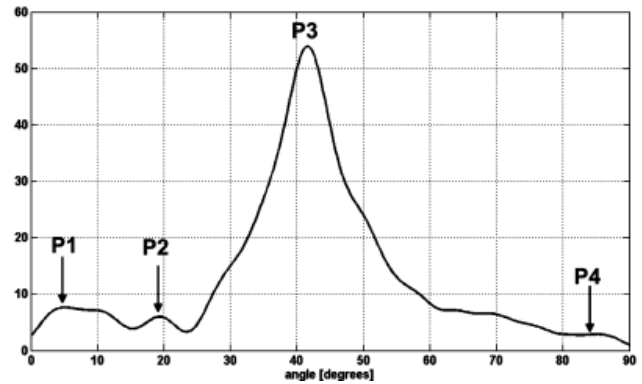
Structure of four analytes (glycopeptides)

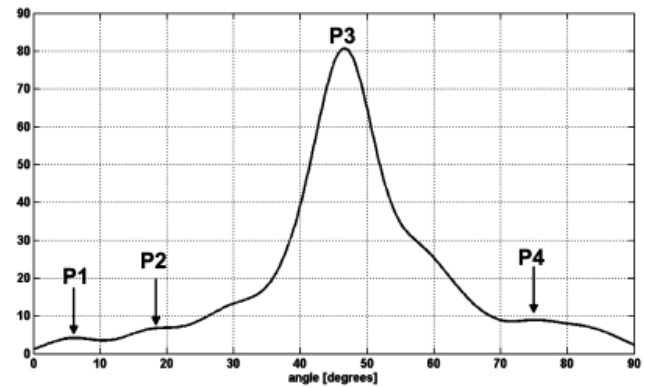


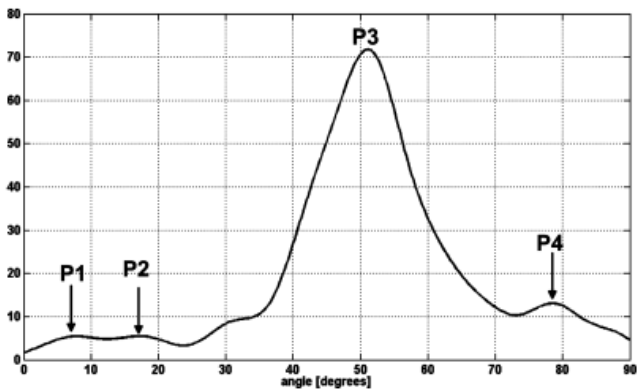
COSY NMR spectra of four analytes



COSY NMR spectra of three mixtures







Clustering functions calculated on a set of 203 SAPs in 2D wavelet domain in 2D subspaces: X_1X_2 , X_1X_3 and X_2X_3.

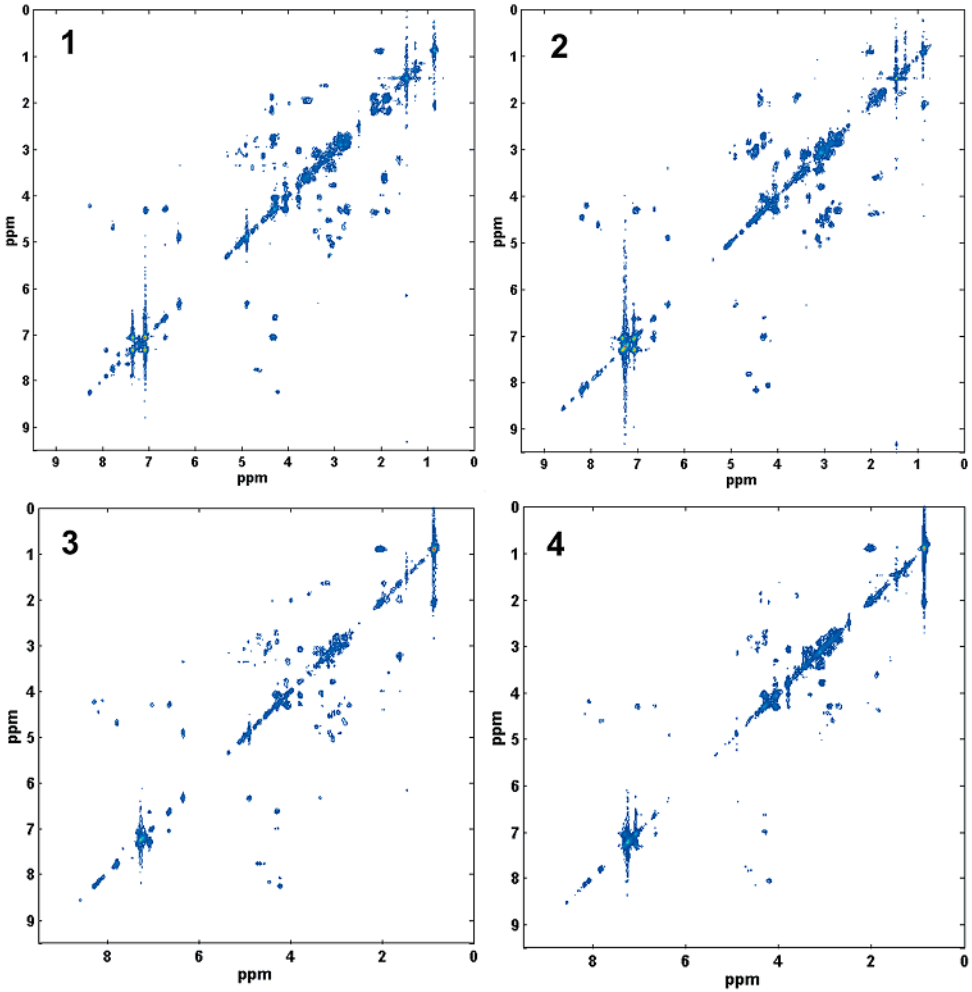
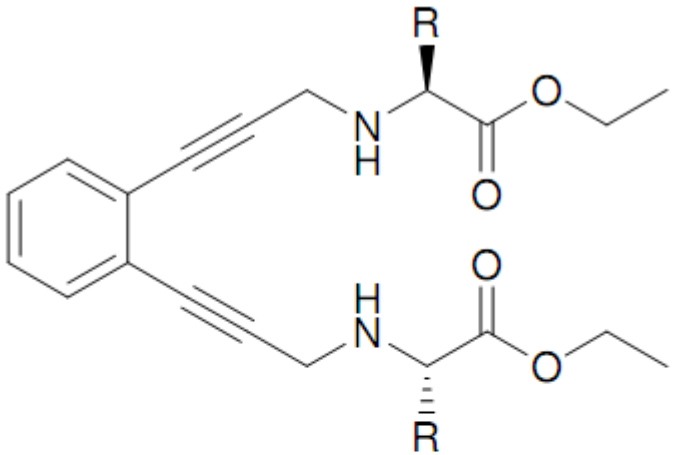


Table 1. Normalized Correlation Coefficients for (a) Pure Analytes 1−4; (b) Analytes 1−4 Estimated on 203 SAPs Detected in Symmlet 8 Wavelet Domain; (c) Analytes 1−4 Estimated on 23 SAPs Detected in Fourier Domain; (d) Analytes 1−4 Estimated by Means of JADE ICA Algorithm from Four Mixtures[a]

entry		An_1	An_2	An_3	An_4
a	An_1	**1**	0.5509	0.1394	0.3730
	An_2	0.5509	**1**	0.3051	0.5120
	An_3	0.1394	0.3051	**1**	0.7965
	An_4	0.3730	0.5120	0.7965	**1**
b	$\hat{An}_1$	**0.8931**	0.4753	0.2638	0.4132
	$\hat{An}_2$	0.5634	**0.8579**	0.2795	0.5366
	$\hat{An}_3$	0.1945	0.5048	**0.8990**	0.7953
	$\hat{An}_4$	0.4386	0.6124	0.8060	**0.8381**
c	$\hat{An}_1$	**0.8924**	0.6009	0.2754	0.4602
	$\hat{An}_2$	0.5482	**0.8469**	0.3107	0.5695
	$\hat{An}_3$	0.0931	0.4101	**0.8432**	0.7249
	$\hat{An}_4$	0.3108	0.3411	**0.8236**	0.7331
d	$\hat{An}_1$	0.7189	0.7090	0.6805	**0.7939**
	$\hat{An}_2$	0.6873	0.7571	0.6524	**0.7790**
	$\hat{An}_3$	0.6606	0.7325	0.7142	**0.8177**
	$\hat{An}_4$	0.6322	0.7232	0.7474	**0.8342**

[a] A significant degree of correlation between spectra of true analytes caused failure of the ICA-based extraction of analytes, part d. An_1-An_4 pure analytes **1−4**; $\hat{An}_1-\hat{An}_4$ estimated analytes **1−4**.

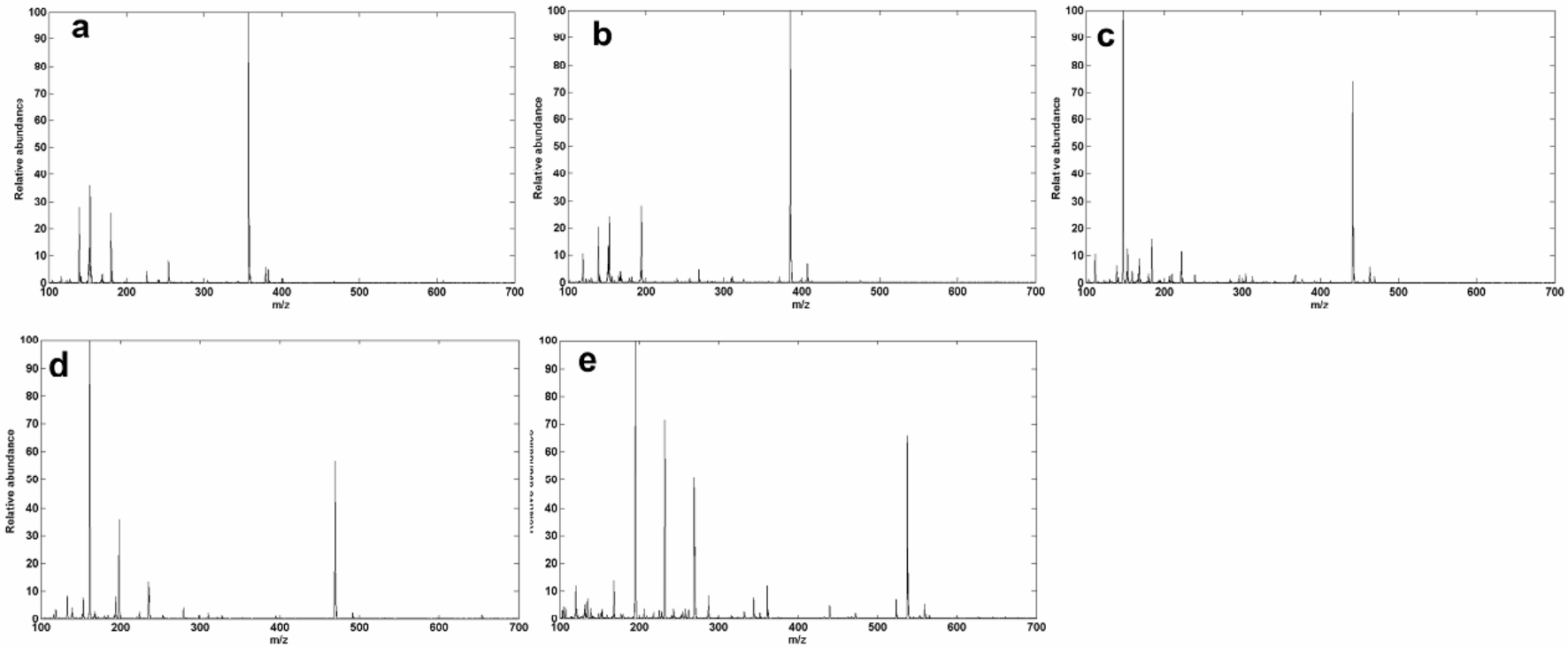
Estimated COSY NMR spectra of analytes in 2D Fourier domain



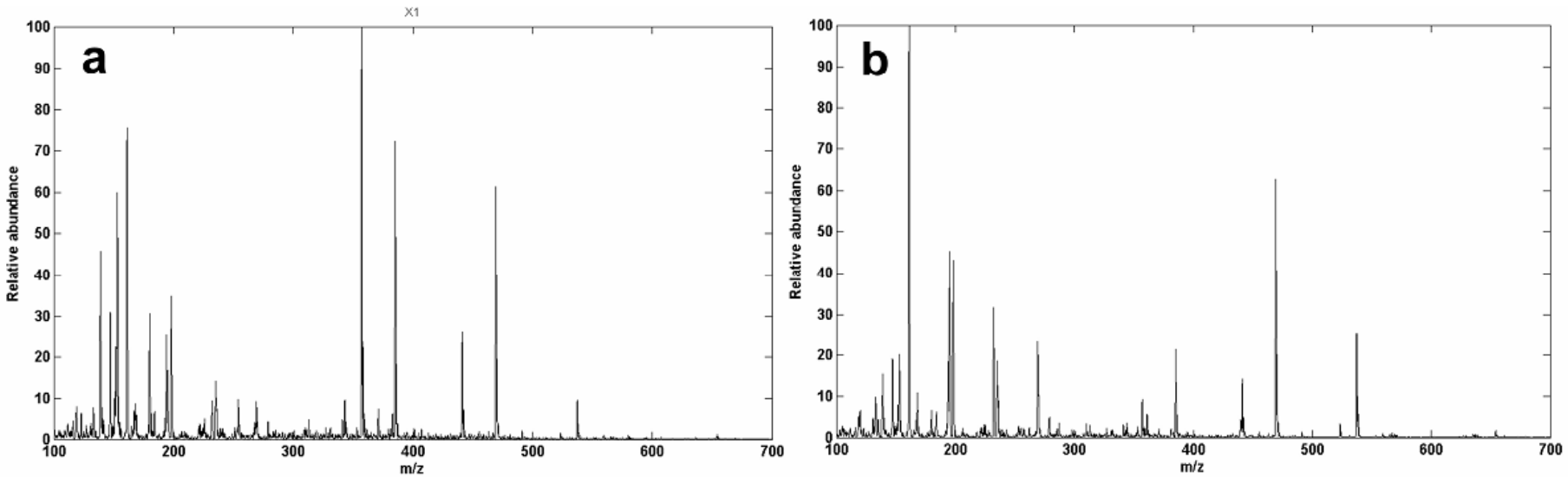
5 R=H
6 R=CH_3
7 R=$CH(CH_3)_2$
8 R=$CH_2CH(CH_3)_3$
9 R=$CH_2C_6H_5$

Figure S-1.

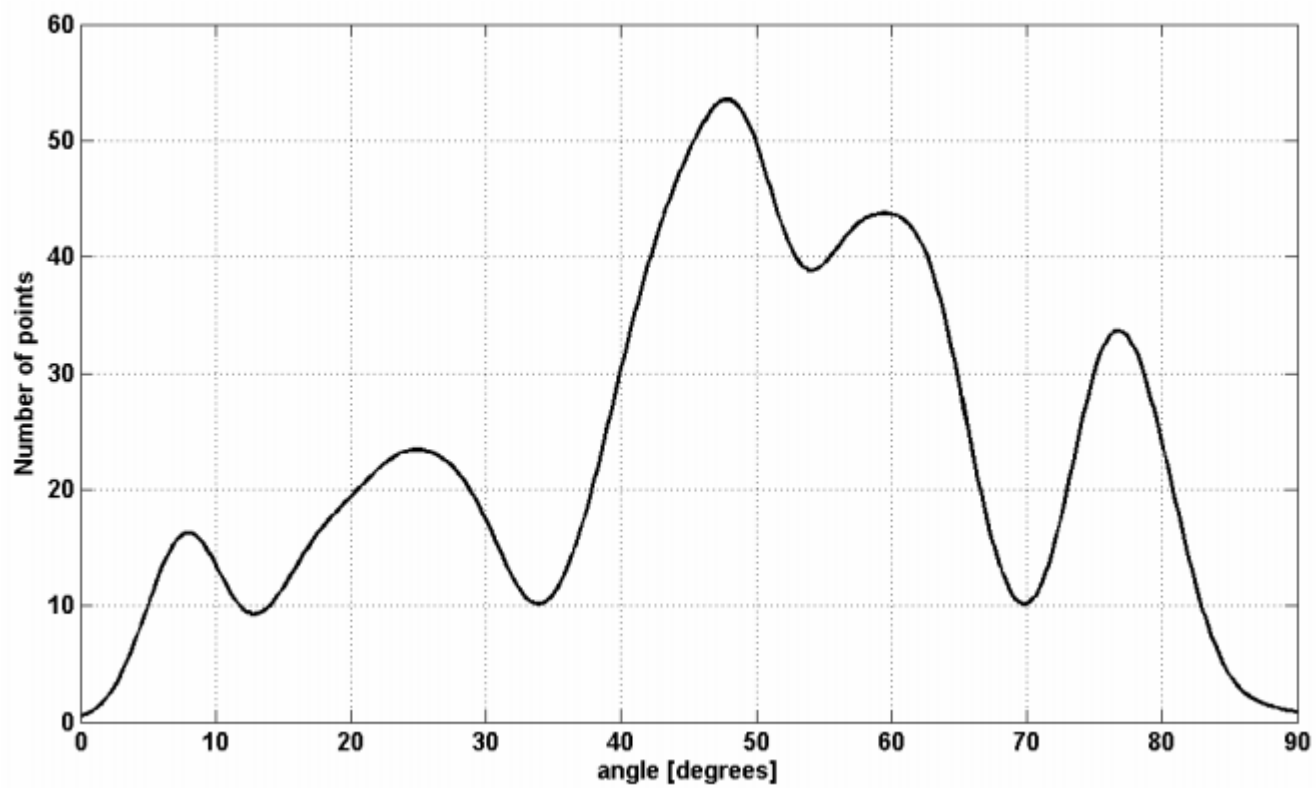
Chemical structure of five pure components.



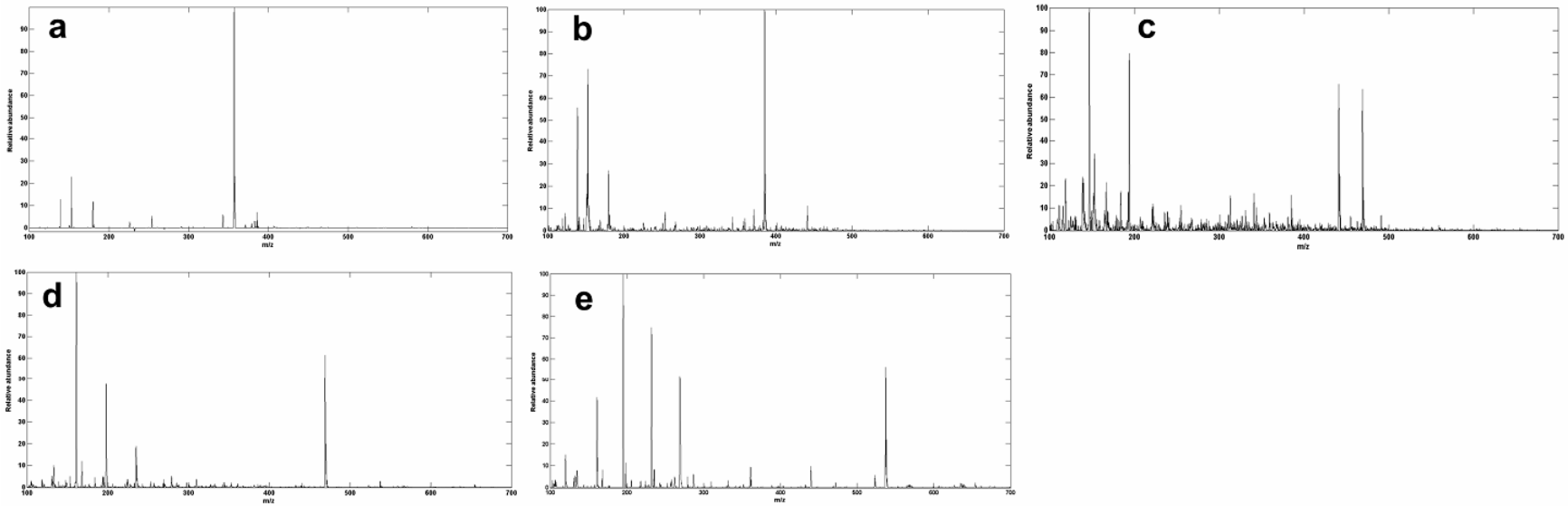
Mass spectra of five pure components.



Mass spectra of two mixtures



Dana clustering function in the mixing anagle domain. Five peaks indicate presence of five components in the mixtures spectra.



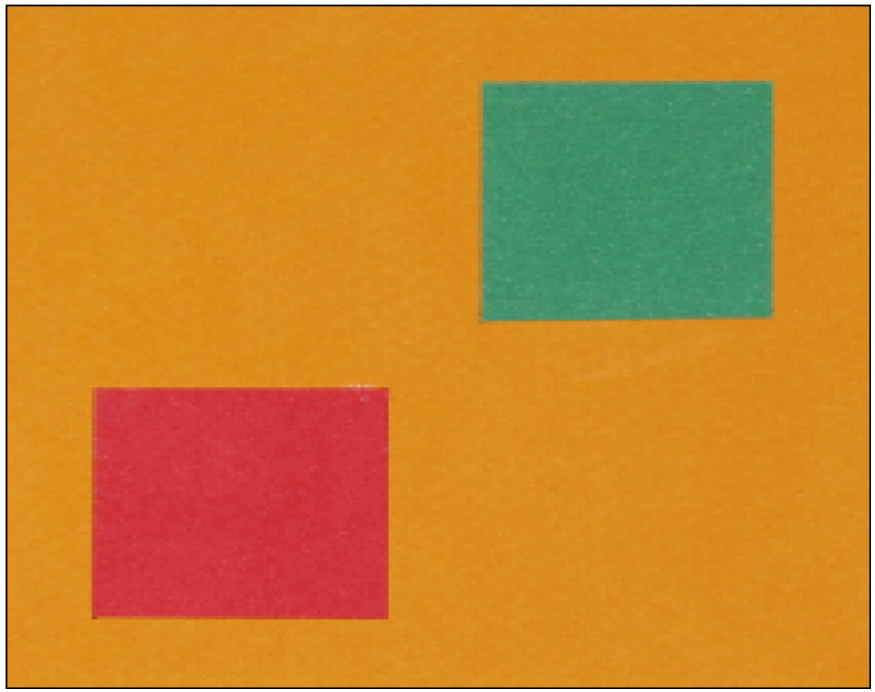
Estimated mass spectra of five pure components.

Table S-1. Normalized correlation coefficients for (a) pure analytes **5-9**; (b) analytes **5-9** estimated on 290 SAPs detected by using analytical representation (3) and *clusterdata* algorithm.*

entry		An_5	An_6	An_7	An_8	An_9
a	An_5	**1**	0.1268	0.0456	0.0266	0.0075
	An_6	0.1268	**1**	0.0321	0.0332	0.0379
	An_7	0.0456	0.0321	**1**	0.0134	0.0030
	An_8	0.0265	0.0332	0.0134	**1**	0.0029
	An_9	0.0075	0.0379	0.0030	0.0029	**1**
b	$\hat{An}_5$	**0.9038**	0.0305	0.0044	0.0002	0.0120
	$\hat{An}_6$	0.3162	**0.8294**	0.1198	0.0325	0.0043
	$\hat{An}_7$	0.0959	0.2334	**0.7275**	0.2009	0.0038
	$\hat{An}_8$	0.0043	0.0038	0.0124	**0.9736**	0.0293
	$\hat{An}_9$	0.0121	0.0161	0.0073	0.2097	**0.9437**

*An_5-An_9 pure analytes **5-9**; $\hat{An}_5$- $\hat{An}_9$ estimated analytes **5-9**.

Unsupervised segmentation of RGB image with three materials: NMF with sparseness constrains, DCA, ICA.



Original RGB image

I. Kopriva and A . Cichocki, "Sparse component analysis-based non-probabilistic blind decomposition of low-dimensional multi-spectral images," *Journal of Chemometrics*, vol. 23, Issue 11, pp. 590-597 (2009).

Unsupervised segmentation of multispectral images

Evidently degree of overlap between materials in spatial domain is very small i.e. $s_m(t)*s_n(t)\approx\delta_{nm.}$. Hence RGB image decomposition problem can be solved either with clustering and L_1-norm minimization or with HALS NMF algorithm with sparseness constraints.

For the L_1-norm minimization estimate of the mixing (spectral reflectance matrix) **A** and number of materials M is necessary. For HALS NMF only estimate of M is necessary. Both tasks can be accomplished by data clustering algorithm presented in ref.[10].

Because materials in principle do not overlap in spatial domain it applies $||\mathbf{s}(t)||_0\approx 1$

Unsupervised segmentation of multispectral images

Assuming unit L_2-norm of $\boldsymbol{a}_m$ we can parameterize column vectors in 3D space by means of azimuth and elevation angles

$$\mathbf{a}_m = [\cos(\varphi_m)\sin(\theta_m) \;\; \sin(\varphi_m)\sin(\theta_m) \;\; \cos(\theta_m)]^{\mathrm{T}}$$

Due to nonnegativity constraints both angles are confined in $[0,\pi/2]$. Now estimation of **A** and M is obtained by means of data clustering algorithm:

- We remove all data points close to the origin for which applies: $\left\{\left|\mathbf{x}(t)\right|_2 \le \varepsilon\right\}_{t=1}^{T}$ where ε represents some predefined threshold.

- Normalize to unit L_2-norm remaining data points $\mathbf{x}(t)$, i.e., $\left\{\mathbf{x}(t) \to \mathbf{x}(t)/\left|\mathbf{x}(t)\right|_2\right\}_{t=1}^{\bar{T}}$

Unsupervised segmentation of multispectral images

• Calculate function *f*(**a**):

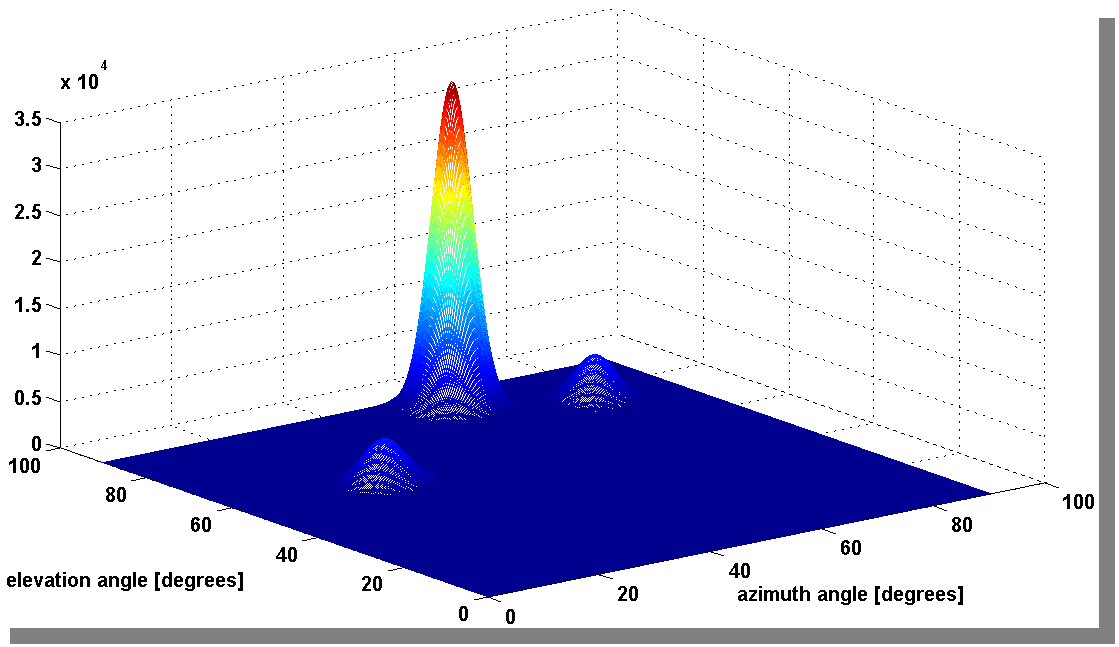
$$f(\mathbf{a}) = \sum_{t=1}^{\bar{T}} \exp\left(-\frac{d^2(\mathbf{x}(t), \mathbf{a})}{2\sigma^2} \right)$$

where $d(\mathbf{x}(t), \mathbf{a}) = \sqrt{1 - (\mathbf{x}(t) \cdot \mathbf{a})^2}$ and $(\mathbf{x}(t) \cdot \mathbf{a})$ denotes inner product. Parameter σ is called dispersion. If set to sufficiently small value, in our experiments this turned out to be $\sigma \approx 0.05$, the value of the function *f*(**a**) will approximately equal the number of data points close to **a**. Thus by varying mixing angles $0 \leq \varphi, \theta \leq \pi/2$ we effectively cluster data.

• Number of peaks of the function *f*(**a**) corresponds with the estimated number of materials *M*. Locations of the peaks correspond with the estimates of the mixing angles $\left\{ \left(\hat{\varphi}_m, \hat{\theta}_m \right) \right\}_{m=1}^{\hat{M}}$, i.e., mixing vectors $\{ \hat{\mathbf{a}}_m \}_{m=1}^{\hat{M}}$.

Unsupervised segmentation of RGB image with three materials: NMF with sparseness constrains, DCA, ICA.

Clustering algorithm is used to estimate number of materials M.



Thee peaks suggest existence of three materials in the RGB image **i.e. *M*=3**.

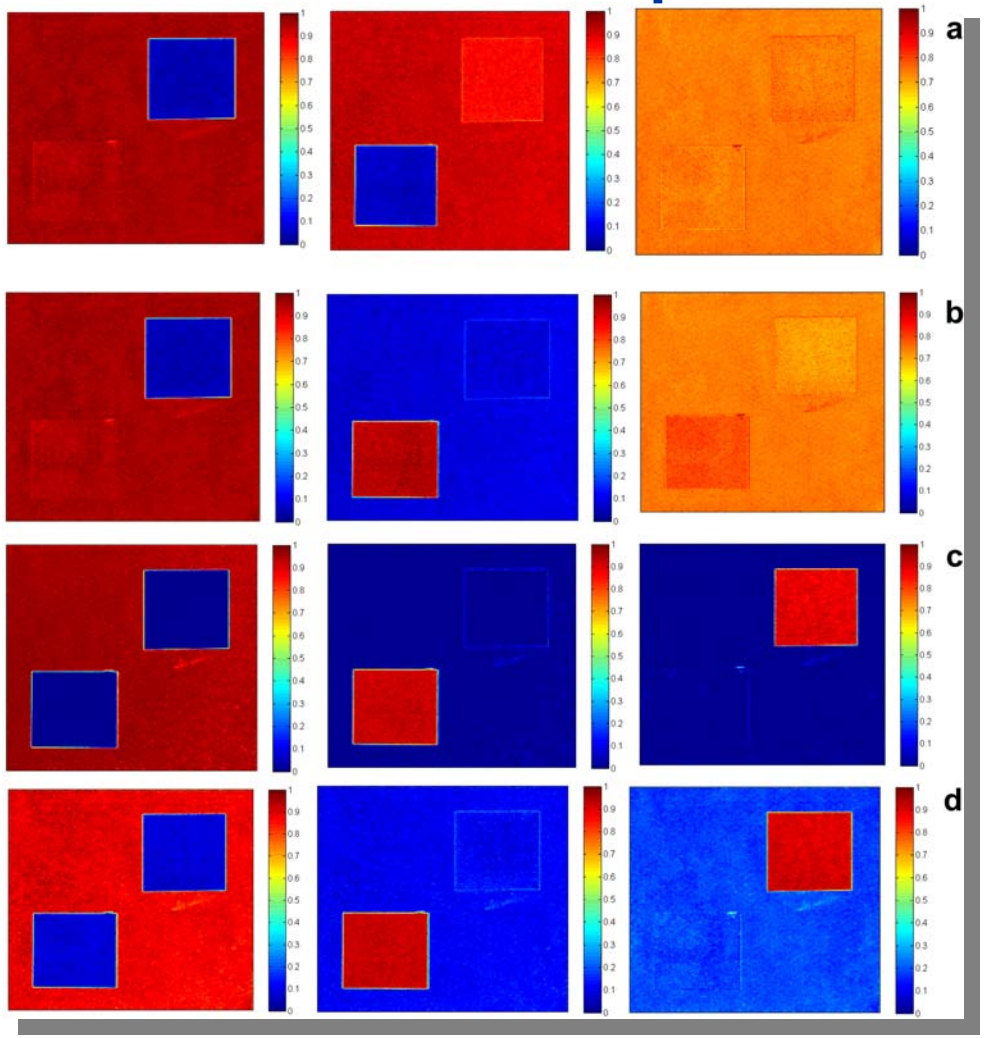
Unsupervised segmentation of RGB image with three materials: NMF with sparseness constrains, DCA, ICA.

Spatial maps of the materials were extracted by NMF with 25 layers, linear programming, ICA and DCA methods.

Extracted spatial maps were rescaled to the interval [0,1] where 0 means full absence of the material and 1 means full presence of the material.

This enables visualization of the quality of decomposition process. Zero probability (absence of the material) is visualized with dark blue color and probability one (full presence of the material) is visualized with dark red color.

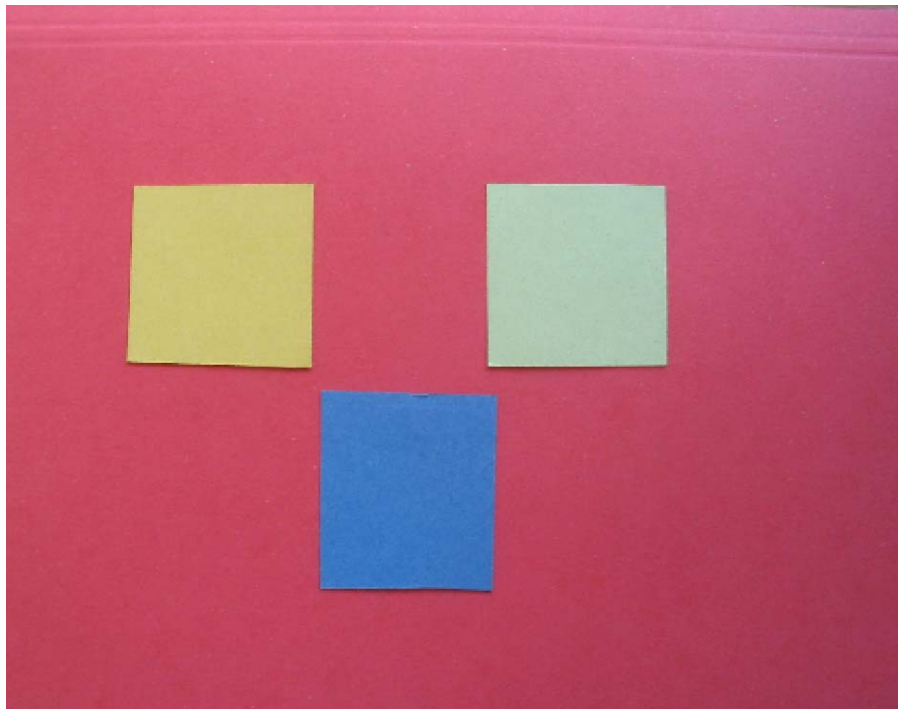
Unsupervised segmentation of RGB image with three materials: NMF with sparseness constrains, DCA, ICA.



a) DCA
b) ICA
c) NMF
d) Linear programming

Unsupervised segmentation of multispectral images

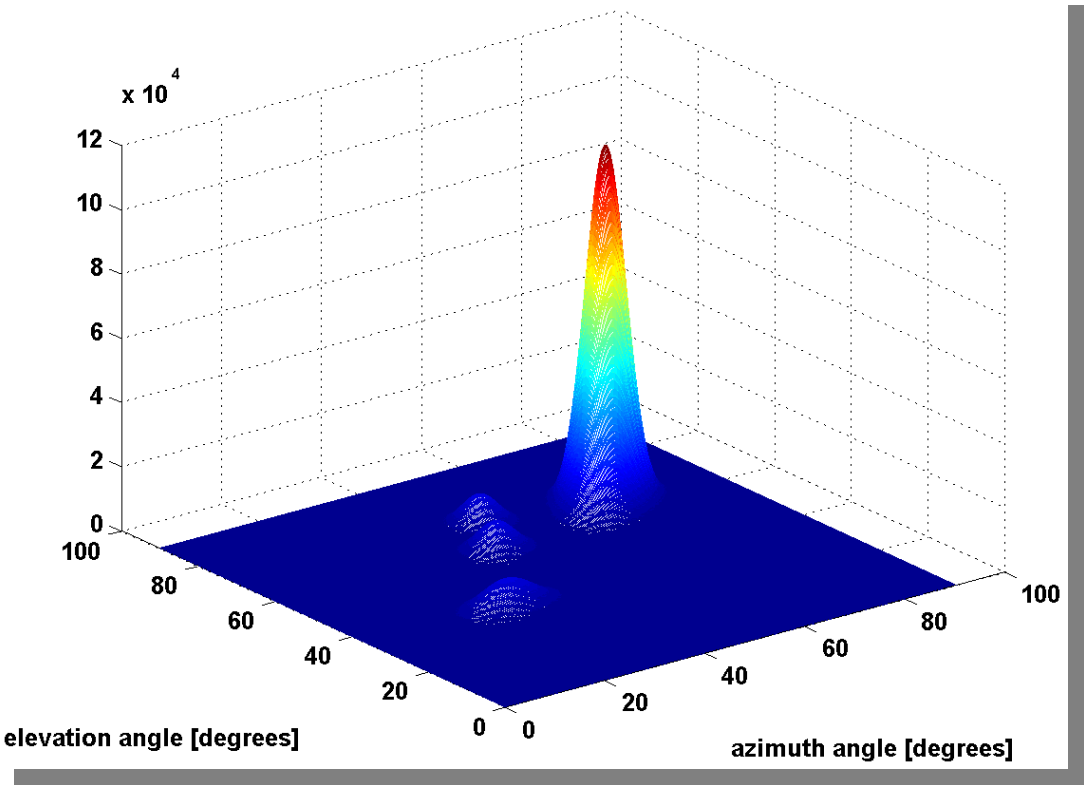
Consider blind decomposition of the RGB image (N=3) composed of four materials (M=4):



I. Kopriva and A . Cichocki, "Sparse component analysis-based non-probabilistic blind decomposition of low-dimensional multi-spectral images," *Journal of Chemometrics*, vol. 23, Issue 11, pp. 590-597 (2009).

Unsupervised segmentation of multispectral images

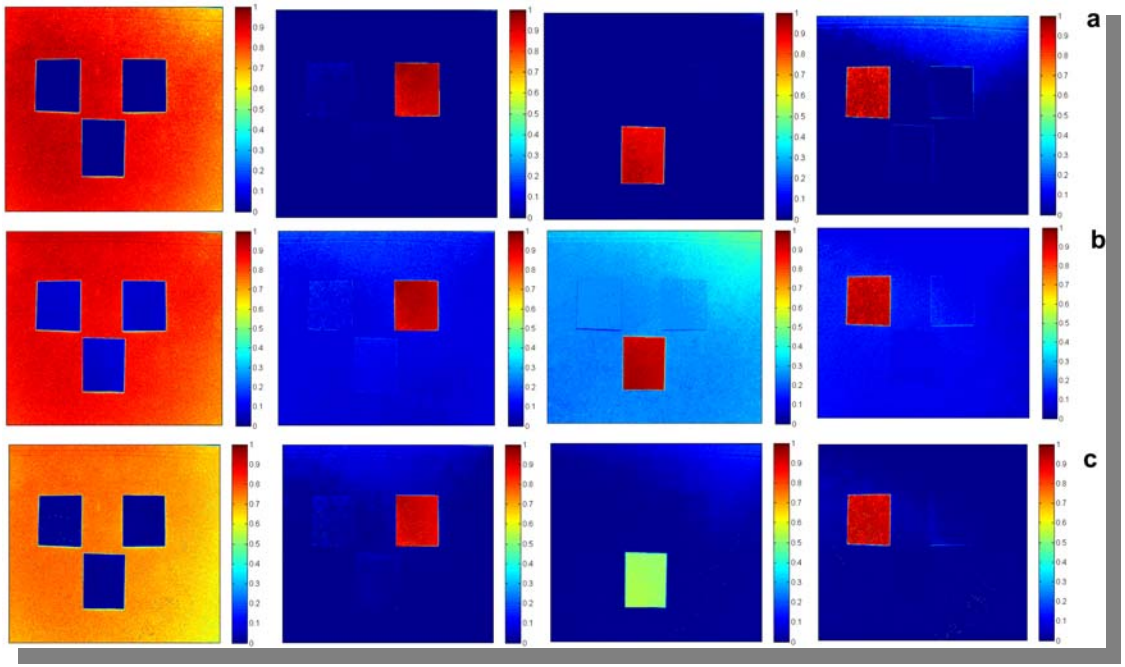
For shown experimental RGB image clustering function is obtained as:



Four peaks suggest existence of four materials in the RGB image **i.e. *M*=4**.

Unsupervised segmentation of multispectral images

Spatial maps of the materials extracted by HALS NMF with 25 layers, linear programming and interior point method [18] are obtained as:



a) 25 layers HALS NMF; b) Interior point method, [74,90]; c) Linear programming.

18. S. J. Kim, K. Koh, M. Lustig, S. Boyd, D. Gorinevsky, "An Interior-Point Method for Large-Scale L_1 -Regularized Least Squares,"IEEE Journal of Selected Topics in Signal Processing **1**, 606-617 (2007). http://www.stanford.edu/~boyd/l1_ls/.

Unsupervised segmentation of multispectral images

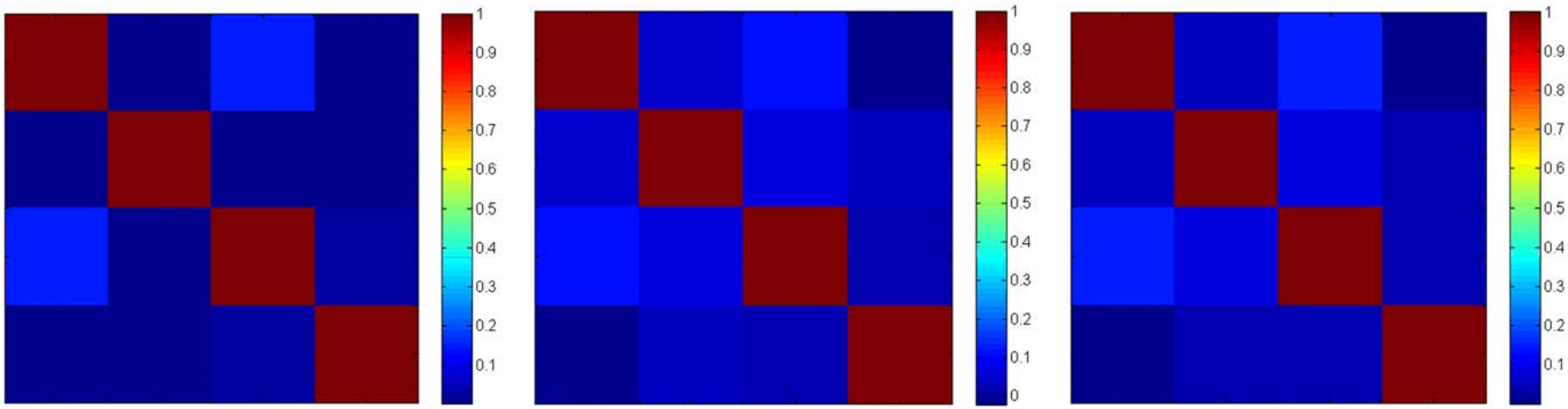
Because materials in the experimental RGB image are orthogonal (they do not overlap in spatial domain) we can evaluate performance of the employed blind image decomposition methods via the correlation matrix defined as $\mathbf{G=SS}^{\mathrm{T}}$. For perfect estimation the correlation matrix will be diagonal and performance is visualized as deviation from diagonal matrix. To quantify decomposition quality numerically we compute the correlation index in dB scale as

$$CR = -10\log_{10}\sum_{\substack{i,j=1\\ j\neq i}}^{M} g_{ij}^{2}$$

where before calculating correlation matrix **G** rows of **S** are normalized to unit L_2-norm.

Unsupervised segmentation of multispectral images

Correlation matrices



From left to right: 25 layers HALS NMF; Interior point method, [18]; c) Linear programming.

CR performance measure in dB

	Multilayer HALS NMF	**Interior-point method**	**Linear program**
CR [dB]	13.67	9.97	7.77
CPU time [s]*	3097	7751	3265

*MATLAB environment on 2.4 GHz Intel Core 2 Quad Processor Q6600 desktop computer with 4GB RAM.