ON REAL AND COMPLEX VALUED ℓ_1-NORM MINIMIZATION FOR OVERCOMPLETE BLIND SOURCE SEPARATION

Stefan Winter,* *Hiroshi Sawada, Shoji Makino*

NTT Communication Science Laboratories,
NTT Corporation,
2-4 Hikaridai, Seika-cho, Soraku-gun,
Kyoto 619-0237, Japan
{wifan,sawada,maki}@cslab.kecl.ntt.co.jp

ABSTRACT

A maximum a-posteriori approach for overcomplete blind source separation based on Laplacian priors usually involves ℓ_1-norm minimization. It requires different approaches for real and complex numbers as they appear for example in the frequency domain. In this paper we compare a combinatorial approach for real numbers with a second order cone programming approach for complex numbers.

Although the combinatorial solution with a proven minimum number of zeros is not theoretically justified for complex numbers, its performance quality is comparable to the performance of the second order cone programming (SOCP) solution. However, it has the advantage that it is faster for complex overcomplete BSS problems with low input/output dimensions.

1. INTRODUCTION

The high quality separation of speech sources is an important prerequisite for further processing such as speech recognition in environments with several active speakers. If the underlying mixing process is unknown it is called blind source separation (BSS). We consider the special case of overcomplete BSS where the number of sources exceeds the number of sensors and take into account convolutive mixtures for reverberant environments.

If the sources are assumed to be statistically independent and Laplacian distributed (which is typical for sparse sources) then a maximum a-posteriori (MAP) based separation approach leads to a two-stage process. In the first step the mixing matrix must be estimated. It is then used in the second step to separate the sources by constrained ℓ_1-norm minimization. Since we work in the frequency domain to account for convolutive mixtures we have to deal with complex numbers.

In this paper we investigate the difference between real and complex valued ℓ_1-norm minimization and its implication for the overcomplete BSS of convolutive mixtures.

In Sec. 2 we first explain the context of overcomplete BSS in which we consider ℓ_1-norm minimization. In Sec. 3.1 and Sec. 3.2 we provide a detailed description of real and complex valued ℓ_1-norm minimization before we consider their differences. The consequences of these differences for practical applications are presented in Sec. 4

*The author is on leave from the Department of Multimedia Communication and Signal Processing, University Erlangen-Nuremberg.

2. OVERCOMPLETE BSS

We will consider a convolutive mixing model with N speech sources $s_i(t)$ $(i = 1 \dots N)$ and M $(M < N)$ sensors that yield linearly mixed signals $x_j(t)$ $(j = 1 \dots M)$. The mixing can be described by

$$x_j(t) = \sum_{i=1}^{N} \sum_{l=1}^{\infty} h_{ji}(l) s_i(t-l) \tag{1}$$

where $h_{ji}(t)$ denotes the impulse response from source i to sensor j.

Instead of solving the problem in the time-domain, we choose a narrowband approach in the frequency domain by applying a short-time discrete Fourier transform (STDFT). Following [1] we can approximate the mixing process in the frequency domain as

$$\mathbf{X}(f,\tau) = \mathbf{H}(f)\mathbf{S}(f,\tau) \ , \tag{2}$$

where $\mathbf{X} \in \mathbb{K}^M$, $\mathbf{H} \in \mathbb{K}^{M\times N}$, $\mathbf{S} \in \mathbb{K}^N$, $\mathbb{K} = \mathbb{C}$ and τ denotes the time frame.

This reduces the problem from convolutive to instantaneous mixtures in each frequency bin f. For simplicity we will omit the frequency and time dependence. Switching to the frequency domain has the additional advantage that the sparseness of the sources is increased [2]. This is very important to ensure good separation performance since the separation is built on the assumption of sparse source signals.

In complete BSS the mixing matrix $\mathbf{H}$ is square and (assuming full rank) invertible. Therefore the BSS problem can be solved by either inverting an estimate of the mixing matrix or directly estimating its inverse and solving (2) for $\mathbf{S}$.

However, this approach does not work for overcomplete BSS where the mixing matrix is not invertible. An optimal solution in a statistical sense can be obtained by maximizing the a-posteriori (MAP) $P(\mathbf{S}|\mathbf{X},\mathbf{H})$ of the sources $\mathbf{S}$ given the mixtures $\mathbf{X}$ and the mixing matrix $\mathbf{H}$.

$$\mathbf{S} = [\ S_1 \dots S_N\ ]^T = \arg\max_{\mathbf{S}} P(\mathbf{S}|\mathbf{X},\mathbf{H}). \tag{3}$$

If we assume source signals whose spectral components have statistically independent phases and amplitudes with uniform and Laplacian distributions, respectively, the cost function results in

$$\min_{\mathbf{S}} \sum_i |S_i|,\ i = 1,\dots,N \qquad \text{s.t.} \qquad \mathbf{HS} = \mathbf{X} \tag{4}$$

for each time instance τ. $|S_i|$ denotes the amplitude of S_i.

0-7803-9154-3/05/$20.00

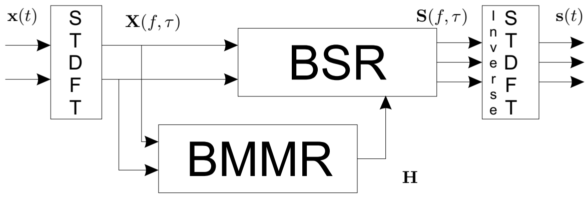

Figure 1: Overall separation system

This can be interpreted as a two-stage process as utilized for example in [1, 2, 3]. In the first step, which we call blind mixing model recovery (BMMR), we estimate the mixing matrix $\mathbf{H}$. Several methods have already been proposed for estimating the mixing matrix in the BMMR step. One example is hierarchical clustering as described in detail in [1]. Once the mixing matrix is known, the second step, which we call blind source recovery (BSR), is basically a constrained ℓ_1-norm minimization to estimate the sources. We assume that we already know the mixing matrix and concentrate in the following section on BSR by ℓ_1-norm minimization.

Before we use the inverse STDFT to obtain time-domain signals, we apply the minimal distortion principle [4] to eliminate scaling ambiguity. The overall separation system is depicted in Fig. 1.

3. CONSTRAINED ℓ_1-NORM MINIMIZATION

Equation (4) cannot be solved analytically for real or complex numbers. After an appropriate transformation numerical solutions can be found in the area of mathematical programming, which deals in general with the optimization of constrained functions [5].

3.1. ℓ_1-norm minimization of real valued problems

If we had to consider only real valued problems ($\mathbb{K} = \mathbb{R}$) we could apply linear programming (LP) [6] which solves problems of the form

$$\min \widehat{\mathbf{c}}^T \widehat{\mathbf{S}}, \qquad \text{s.t.} \quad \widehat{\mathbf{H}}\widehat{\mathbf{S}} = \widehat{\mathbf{X}},\ \widehat{S}_i \geq 0, \quad i = 1, ..., \widehat{N} \tag{5}$$

where $\widehat{\mathbf{c}}, \widehat{\mathbf{S}} \in \mathbb{R}^{\widehat{N}}$, $\widehat{\mathbf{H}} \in \mathbb{R}^{\widehat{M} \times \widehat{N}}$ and $\widehat{\mathbf{X}} \in \mathbb{R}^{\widehat{M}}$. For $\mathbb{K} = \mathbb{R}$ (4) can be transformed into (5) by separating positive and negative values by

$$\widehat{\mathbf{S}} \leftarrow \begin{bmatrix} \mathbf{S}^+ \\ \mathbf{S}^- \end{bmatrix}, \ \widehat{\mathbf{c}} \leftarrow \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \end{bmatrix}, \ \widehat{\mathbf{H}} \leftarrow \begin{bmatrix} \mathbf{H} \\ -\mathbf{H} \end{bmatrix}, \ \widehat{\mathbf{X}} \leftarrow \mathbf{X}. \tag{6}$$

Here $\mathbf{1}$ stands for an unity matrix with appropriate dimensions. $\mathbf{S}^+$ and $\mathbf{S}^-$ are derived from $\mathbf{S}$ by setting all negative values or positive values, respectively, to zero.

Although powerful algorithms for linear programming exist, they are still time consuming. Depending on the dimensions of the problem we can obtain a faster combinatorial algorithm if we use a certain property of the solution. It can be shown that the N-dimensional vector $\mathbf{S}$, that solves (4), contains at least $N - M$ zeros if the columns of $\mathbf{H}$ are normalized [3, 7]. The normalization can be assumed for BSS due to the scaling ambiguity.

The lower limit for the number of zeros can be considered a constraint imposed by the MAP estimation and can easily be fulfilled by setting $N - M$ elements of the solution to zero. Then we only have to determine the remaining M elements. Assuming that we know where to place the zeros the remaining elements are found by multiplying the inverse of the quadratic matrix built by the remaining mixing vectors $\mathbf{h}_i$ with the constraining vector $\mathbf{X}$

$$[\mathbf{h}_{i_1} \ldots \mathbf{h}_{i_M}]^{-1} \mathbf{X}, \qquad i_1, \ldots, i_M \in \{1, \ldots, N\} \tag{7}$$

The correct placement of the zeros can be determined by combinatorially testing all possibilities and accepting the one with the smallest ℓ_1-norm. As a simple example let us consider

$$\mathbf{H} = \begin{bmatrix} 1 & 0.6 & -0.6 \\ 0 & 0.8 & 0.8 \end{bmatrix}, \qquad \mathbf{X} = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}. \tag{8}$$

According to the dimensions of the problem at least one element of the solution $\mathbf{S}$ must be zero. The ℓ_1-norm of the possible solutions are

$$\left| \begin{bmatrix} \begin{bmatrix} 1 & 0.6 \\ 0 & 0.8 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} \\ 0 \end{bmatrix} \right|_1 = 1.25 \tag{9}$$

$$\left| \begin{bmatrix} 0 & \begin{bmatrix} 1 & -0.6 \\ 0 & 0.8 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} \end{bmatrix} \right|_1 = 2 \tag{10}$$

$$\left| \begin{bmatrix} 0 \\ \begin{bmatrix} 0.6 & -0.6 \\ 0.8 & 0.8 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} \end{bmatrix} \right|_1 = 1.\overline{6} \tag{11}$$

where $|\cdot|_p$ denotes the ℓ_p-norm. The chosen solution would be the one corresponding to (9).

This combinatorial method is based on the shortest-path algorithm [3] and the ℓ_0-norm that basically counts the number of non-zero elements. The combinatorial method stands in contrast to the approach in [8] where conditions are given for which the ℓ_0-norm can be calculated by an ℓ_p-norm with $0 < p \leq 1$.

3.2. ℓ_1-norm minimization of complex valued problems

If complex numbers are involved then (6) can no longer be applied because such numbers possess a continuous phase in contrast to a discrete phase of real numbers. Thus we cannot use algorithms that solve linear programming problems for complex valued problems. However, ℓ_1-norm minimization problems (4) with complex numbers ($\mathbb{K} = \mathbb{C}$) can be transformed to second order cone programming (SOCP) problems in the following way.

Equation (4) is equivalent to

$$\min t \in \mathbb{R} \qquad \text{s.t.} \quad \mathbf{X} = \mathbf{H}\mathbf{S}, \text{ and } |\mathbf{S}|_1 \leq t \tag{12}$$

By decomposing $t = \sum_{i=1}^{N} t_i,\ t_i \in \mathbb{R}$, the second constraint $|\mathbf{S}|_1 \leq t$ can be expressed by

$$|\mathbf{S}|_1 = \sum_{i=1}^{N} \left| \begin{bmatrix} \Re(S_i) \\ \Im(S_i) \end{bmatrix} \right|_2 \leq \mathbf{1t} = \mathbf{1}[t_1 \ \cdots \ t_N]^T = t \tag{13}$$

where $\Re(\cdot)$ and $\Im(\cdot)$ denote the real and imaginary parts, respectively. Thus we can rewrite (4) as

$$\min_{\mathbf{t}} \mathbf{1t} \qquad \text{s.t.} \quad \mathbf{X} = \mathbf{H}\mathbf{S}, \ \left| \begin{bmatrix} \Re(S_i) \\ \Im(S_i) \end{bmatrix} \right|_2 \leq t_i, \quad \forall i \tag{14}$$

By defining

$$\widehat{\mathbf{S}} = \begin{bmatrix} t_1 \\ \Re(S_1) \\ \Im(S_1) \\ \vdots \\ t_N \\ \Re(S_N) \\ \Im(S_N) \end{bmatrix} \in \mathbb{R}^{3N}, \qquad \widehat{\mathbf{c}} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 1 \\ 0 \\ 0 \end{bmatrix} \in \mathbb{R}^{3N} \tag{15}$$

$$\widehat{\mathbf{X}} = \left[\begin{array}{c} \Re(\mathbf{X}) \\ \Im(\mathbf{X}) \end{array} \right] \in \mathbb{R}^{2M} \tag{16}$$

$$\widehat{\mathbf{H}} = \left[\begin{array}{cc} \mathbf{0} & \mathbf{0} \\ \Re(\mathbf{h_1}) & \Im(\mathbf{h_1}) \\ -\Im(\mathbf{h_1}) & \Re(\mathbf{h_1}) \\ \vdots & \vdots \\ \mathbf{0} & \mathbf{0} \\ \Re(\mathbf{h_N}) & \Im(\mathbf{h_N}) \\ -\Im(\mathbf{h_N}) & \Re(\mathbf{h_N}) \end{array} \right]^T \in \mathbb{R}^{2M \times 3N} \tag{17}$$

we can write

$$\min_{\widehat{\mathbf{S}}} \widehat{\mathbf{c}}^T \widehat{\mathbf{S}} \quad \text{s.t.} \quad \widehat{\mathbf{X}} = \widehat{\mathbf{H}}\widehat{\mathbf{S}}, \ \left| \left[\begin{array}{c} \Re(S_i) \\ \Im(S_i) \end{array} \right] \right|_2 \le t_i \quad \forall i \tag{18}$$

The second constraint in (18) can be interpreted as a second order cone for each i.

Equation (18) describes an SOCP problem [9], which can be solved numerically for example with `SeDuMi` [10].

3.3. Analysis of real and complex valued ℓ_1-norm minimization

In contrast to the real valued ℓ_1-norm minimization problem where a minimum number of zeros can be guaranteed theoretically in the optimal solution, the number of zeros cannot be predicted with complex valued problems as the following simple example shows. Let

$$\mathbf{H} = \left[\begin{array}{ccc} 1 & 0.6 & \frac{4}{\sqrt{17}} \\ 0 & 0.8 & \frac{0.8+j0.6}{\sqrt{17}} \end{array} \right], \qquad \mathbf{X} = \left[\begin{array}{c} 1 \\ 0.5 \end{array} \right]. \tag{19}$$

Then the ℓ_1-norm of the solution obtained by SOCP is given by

$$|\mathbf{S}_{socp}|_1 = \left| \left[\begin{array}{c} 0.227 + 0.040i \\ 0.511 - 0.091i \\ 0.481 + 0.015i \end{array} \right] \right|_1 = 1.23. \tag{20}$$

It does not contain any zeros as we could expect with real numbers yet solves (4). In comparison, the ℓ_1-norm of the optimal combinatorial solution is given by

$$|\mathbf{S}_{comb}|_1 = \left| \left[\begin{array}{c} 0 \\ \left[\begin{array}{cc} 0.6 & \frac{4}{\sqrt{17}} \\ 0.8 & \frac{0.8+j0.6}{\sqrt{17}} \end{array} \right]^{-1} \left[\begin{array}{c} 1 \\ 0.5 \end{array} \right] \end{array} \right] \right|_1 = 1.24 \tag{21}$$

This observation reveals a very important difference from real valued problems and prevents the theoretical justification of a procedure similar to the combinatorial approach in Sec. 3.1. To better explain this difference between real and complex numbers we take a look at a general solution based on a combinatorial solution and the nullspace $\mathcal{N}(\mathbf{H})$ of $\mathbf{H}$.

Even though the combinatorial solution $\mathbf{S}_{comb}$ does not necessarily minimize the ℓ_1-norm, it fulfills together with the SOCP solution $\mathbf{S}_{socp}$

$$\mathbf{X} = \mathbf{H}\mathbf{S}_{comb} = \mathbf{H}\mathbf{S}_{socp}. \tag{22}$$

By defining the difference $\widehat{\mathbf{S}} = \mathbf{S}_{socp} - \mathbf{S}_{comb}$, (22) becomes

$$\mathbf{H}\mathbf{S}_{comb} = \mathbf{H}\mathbf{S}_{comb} + \underbrace{\mathbf{H}\widehat{\mathbf{S}}}_{=0}. \tag{23}$$

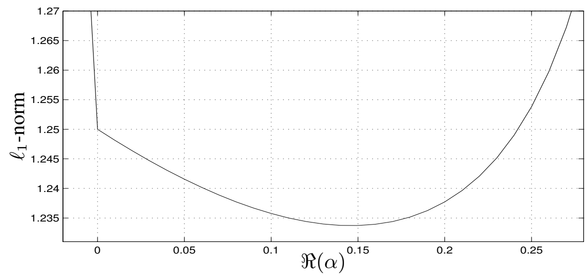

Figure 2: Smooth slope

This means that if we have a combinatorial solution we can limit our search for the minimum ℓ_1-norm solution to the nullspace $\mathcal{N}(\mathbf{H})$, i.e.

$$\min \left| \mathbf{S}_{comb} + \widehat{\mathbf{S}} \right|_1 \tag{24}$$

with

$$\widehat{\mathbf{S}} \in \mathcal{N}(\mathbf{H}) \quad \Leftrightarrow \quad \widehat{\mathbf{S}} = (\mathbf{1} - \mathbf{H}^{-}\mathbf{H})\mathbf{z}, \quad \mathbf{z} \in \mathbb{C}^N \tag{25}$$

where $\mathbf{H}^{-}$ is an arbitrary generalized inverse of $\mathbf{H}$. For $N = 3$ and $M = 2$ we can express the combinatorial solution and the nullspace without loss of generality by

$$\mathbf{S}_{comb} = \left[\begin{array}{c} [\ \mathbf{h}_1 \quad \mathbf{h}_2\ ]^{-1}\mathbf{X} \\ 0 \end{array} \right] \tag{26}$$

$$\mathcal{N}(\mathbf{H}) = \alpha \left[\begin{array}{c} [\ \mathbf{h}_1 \quad \mathbf{h}_2\ ]^{-1}\mathbf{h}_3 \\ 1 \end{array} \right], \qquad \alpha \in \mathbb{C} \tag{27}$$

With (26) and (27), the function to be minimized (24) can be written as

$$\begin{aligned} \left| \mathbf{S}_{comb} + \alpha\widehat{\mathbf{S}} \right|_1 \quad = \quad & |f_{11}(\mathbf{H},\mathbf{X}) + \alpha f_{12}(\mathbf{H},\mathbf{X})| \\ & + |f_{21}(\mathbf{H},\mathbf{X}) + \alpha f_{22}(\mathbf{H},\mathbf{X})| \\ & + |f_{31}(\mathbf{H},\mathbf{X}) + \alpha f_{32}(\mathbf{H},\mathbf{X})|. \end{aligned} \tag{28}$$

Here f_{ij} is a summand that only depends on $\mathbf{H}$ and $\mathbf{X}$, which are constant for any given problem. If only real values are involved then (28) describes a piecewise linear function depending on α whose slope can only change a limited number of times in a discrete manner.

However, once complex numbers are involved, their imaginary part results in an inherent ℓ_2-norm, which leads to smooth slopes as they appear with second order or higher polynomials. This behavior becomes obvious in (18). There the ℓ_1-norm is changed from the sum of absolute values of real numbers to the sum of the ℓ_2-norms of the real and imaginary part. The introduction of the ℓ_2-norm explains the different behavior of complex valued ℓ_1-norm minimization compared with its real counterpart. An example is shown in Fig. 2, where the dependence of the ℓ_1-norm on α is shown (here only the dependence on the real part of α is shown). The combinatorial solution that minimizes the ℓ_1-norm is given there for $\alpha = 0$. However, this is not the solution of (4), which is rather obtained for $\alpha \neq 0$.

Table 1: Experimental conditions

Source directions	50°, 90°, 120°
Sensor distance	40 mm
Source signal length	7.4 seconds
Reverberation time T_R	130 ms
Sampling frequency f_s	8 kHz
Window type	von Hann
Filter length	1024 points
Shifting interval	256 points

Table 2: Detailed performance comparison

Signal combination	Original CS	Original SOCP	Estimated CS	Estimated SOCP
1	0.938	0.913	0.824	0.895
2	0.958	0.948	0.895	0.876
3	0.898	0.955	0.842	0.891
4	0.919	0.912	0.847	0.876
Average	0.928	0.932	0.852	0.884

4. PERFORMANCE COMPARISON

Even though the combinatorial solution (CS) with a minimum number of zeros in Sec. 3.1 cannot be justified theoretically for complex numbers, in practice its performance is comparable to that of the SOCP solution. We tested both approaches with the estimated as well as the original mixing matrix and combinations of three out of four different speech signals. The experimental conditions can be found in Table 1. As a performance measure we used the correlation coefficient

$$\text{cor} = \max_{\tau} \frac{x_{s_i}^T(t) y_i(t+\tau)}{|x_{s_i}(t)|_2 \, |y_i(t)|_2} \tag{29}$$

between the output signal $y_i(t)$ and the corresponding filtered version x_{s_i} of the original speech signal at the microphones. τ determines the shift of the signal vector. The results are shown in Table 2. A subjective evaluation of the separated sources supports this result.

As expected, the SOCP solution yields better results than the combinatorial solution. However, better estimation of the mixing matrix reduces the advantage of the SOCP solution.

While the difference in performance quality might be negligible in practical applications, the computational complexity reveals great differences. The combinatorial solution has a low initial computational complexity but grows exponentially with the input dimension N. On the other hand, the SOCP solution has a high computational complexity even for low input dimensions N, but it grows only according to

$$O\left(\sqrt{N}\log(1/\epsilon)\right). \tag{30}$$

ϵ denotes the precision of the numerical algorithm [10]. Figure 3 illustrates this fact and shows on a logarithmic scale the time required by the two approaches to separate the sources in one frequency bin for different numbers of sources and sensors.

One reason for the big difference in the initial computational complexity can be found in the reusability of previous results. For overcomplete BSS in the frequency domain the minimum ℓ_1-norm solution must be calculated several times with the same mixing matrix. SOCP always starts from scratch and therefore cannot profit from the reuse of earlier results. In contrast, the combinatorial solution is built on the inverses of selected mixing vectors. Once they are calculated they can be reused as long as the mixing matrix does not change.

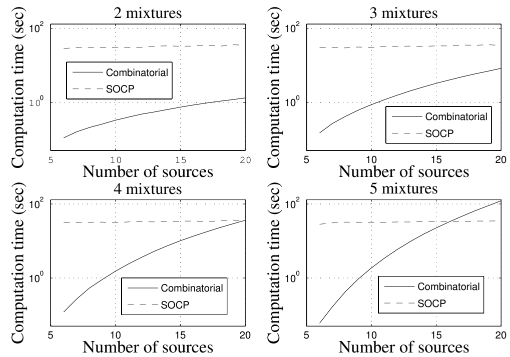

Figure 3: Comparison of computational complexity

5. CONCLUSION

Although the combinatorial solution with at least $N - M$ zeros is not theoretically justified for complex numbers, its performance quality is comparable to that of the SOCP solution. It has the advantage that it is faster for overcomplete BSS problems with low input/output dimensions.

6. REFERENCES

[1] S. Winter, H. Sawada, S. Araki, and S. Makino, "Overcomplete BSS for convolutive mixtures based on hierarchical clustering," in *Proc. ICA 2004*, Sept. 2004, pp. 652–660.

[2] P. Bofill and M. Zibulevsky, "Blind separation of more sources than mixtures using sparsity of their short-time fourier transform," in *Proc. ICA 2000*, June 2000, pp. 87–92.

[3] F. Theis, "Mathematics in independent component analysis," Ph.D. dissertation, University of Regensburg, 2002. [Online]. Available: http://homepages.uni-regensburg.de/˜thf11669/phdthesis.html

[4] K. Matsuoka and S. Nakashima, "Minimal distortion principle for blind source separation," in *Proc. ICA 2001*, Dec. 2001, pp. 722–727.

[5] [Online]. Available: http://www.mathprog.org/

[6] M. S. Lewicki and T. J. Sejnowski, "Learning overcomplete representations," *Neural Computation*, vol. 12, no. 2, pp. 337–365, 2000. [Online]. Available: citeseer.nj.nec.com/lewicki98learning.html

[7] I. Takigawa, M. Kudo, and J. Toyama, "Performance analysis of minimum ℓ_1-norm solutions for underdetermined source separation," *IEEE Trans. Signal Processing*, vol. 52, no. 3, pp. 582–591, Mar 2004.

[8] D. M. Malioutov, M. Cetin, and A. S. Willsky, "Optimal sparse representations in general overcomplete bases," in *Proc. ICASSP 2004*, 2004, pp. 793–796.

[9] A. Pruessner, M. Bussieck, S. Dirkse, and A. Meeraus, "Conic programming in GAMS," INFORMS Annual Meeting, Atlanta, Oct 2003. [Online]. Available: http://www.gams.com/presentations/present˙conic.pdf

[10] J. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization Methods and Software*, vol. 11-12, pp. 625–653, 1999, special issue on Interior Point Methods. [Online]. Available: http://fewcal.kub.nl/sturm/software/sedumi.html