SPARSE COMPONENT ANALYSIS VIA DYADIC CYCLIC DESCENT

M.O. Ulfarsson[†] *and V. Solo*[‡]

[†]University of Iceland, Dept. Electrical Eng., Reykjavik, ICELAND
[‡]University of New South Wales, School of Electrical Eng., Sydney, AUSTRALIA

ABSTRACT

Sparse component analysis (SCA) is a widely used method for solving the blind source separation problem. We develop a new cyclic descent algorithm for SCA based on a dyadic expansion. To select the associated tuning parameter a method based on the Bayesian information criterion is developed. In simulations the new algorithm is compared with state of the art algorithms from the literature.

Index Terms— Sparse Component Analysis, Sparsity, Cyclic Descent.

1. INTRODUCTION

Blind source separation (BSS) refers to the problem of estimating the source signals and the mixing matrix of an unknown linear system whose output is observed. A classical example is the so-called cocktail party problem where several microphones are used to record mixtures of conversations at a gathering and the problem is to isolate individual conversations (speech signals) from this mixture. Sparse component analysis (SCA) is a relatively recent method for solving the BSS problem.

SCA has been successfully applied to solve the BSS problem in fields such as image processing [5], speech processing [6], sensor array processing [7]. SCA has been discussed from a theoretical and algorithmical viewpoint in [8, 9].

1.1. Related work

Sparse components analysis (SCA) is originally due to [5] who solved a version of it with steepest descent algorithms. The properties of SCA depend on the sparsity penalty. SCA with an l_0 penalty is denoted as SCA_0. SCA with an l_1 penalty is denoted as SCA_1. [10] develops an approximate maximum likelihood approach to SCA_1; [11] develops an approximate coordinate descent method for SCA_1 and is discussed further below; [6, 12] expand the signal in a basis and sparsity is applied to the coefficients. This actually converts the problem to a reduced rank regression problem [13, 14] and is outside the scope of this paper. [11] solves the SCA_0 problem with an approximate cyclic descent method. Finally [15] has developed an unusual multi-stage iteration called k-SVD which is also discussed further below.

There are two cases of interest. The overdetermined case where the number of variables in the model is greater than number of sources, and the underdetermined case where the number of sources is greater than the number of variables. [15] focuses on the underdetermined case, while [11] focuses on the overdetermined case.

1.2. Paper Contribution

This paper develops a new SCA_1 algorithm using a version of cyclic descent (CD) (aka co-ordinate descent [16]) which we call dyadic CD. We call the new algorithm SCA_1-DCD. Two tuning parameters need to be specified, the number of sparse components, and the penalty parameter. A Bayesian information criterion (BIC) is developed for selecting them.

The paper is organized as follows. In section 2 we introduce the SCA_1 model. In section 3 we derive the new estimation algorithm for SCA. In Section 4 the BIC criterion for tuning parameter selection is presented. Section 5 presents simulations and compares the new algorithm to competing methods. Finally, in section 6, conclusions are presented.

1.3. Notation

Matrices are presented by bold face capital letters, e.g. $\boldsymbol{S}$. The t-th row vector of $\boldsymbol{S}$ is denoted $\boldsymbol{s}_t^T$, the j-th column vector of $\boldsymbol{S}$ is denoted $\boldsymbol{s}_{(j)}$, and the j,t-th element of $\boldsymbol{S}$ is denoted s_{tj}. The Frobenius norm is denoted as $\|\boldsymbol{S}\|_F^2 = \sum_{tj} s_{tj}^2$. The matrix $\boldsymbol{A}_{-j}$ is equal to $\boldsymbol{A}$ with its j-th column removed.

2. THE SCA_1 PROBLEM

The SCA model is given by

$$\boldsymbol{y}_t = \boldsymbol{A}\boldsymbol{s}_t + \boldsymbol{n}_t, \quad t = 1, ..., T \tag{1}$$

where $\boldsymbol{y}_t$ is a $M \times 1$ vector of observed data $\boldsymbol{A}$ is an $M \times r$ mixing matrix, $\boldsymbol{s}_t$ is a $r \times 1$ source vector, and $\boldsymbol{n}_t \sim N(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_M)$ is a noise vector. The signals and mixing

This work was partly supported by the Research Fund of the University of Iceland and the Icelandic Research Fund (130635-051)

matrix are estimated by minimizing the following penalized least squares criterion

$$\begin{aligned} J(\boldsymbol{A},\boldsymbol{S}) &= \frac{1}{2}\|\boldsymbol{Y}-\boldsymbol{S}\boldsymbol{A}^T\|_F^2 + h\sum_{t,j}|s_{tj}| \\ &= \frac{1}{2}\sum_{t=1}^{T}\|\boldsymbol{y}_t-\boldsymbol{A}\boldsymbol{s}_t\|^2 + h\sum_{t,j}|s_{tj}| \quad (2) \end{aligned}$$

where $\boldsymbol{Y}=[\boldsymbol{y}_t^T]$, and $\boldsymbol{S}=[\boldsymbol{s}_t^T]=[\boldsymbol{s}_{(j)}]=[s_{tj}]$. There is a permutation and sign indeterminacy. This can be seen by noticing that for a $r\times r$ permutation matrix $\boldsymbol{P}$, $\tilde{\boldsymbol{S}}=\boldsymbol{S}\boldsymbol{P}^T$ and $\tilde{\boldsymbol{A}}=\boldsymbol{A}\boldsymbol{P}^T$ we have $\tilde{\boldsymbol{S}}\tilde{\boldsymbol{A}}^T=\boldsymbol{S}\boldsymbol{A}^T$ and $\sum_{tj}|\tilde{s}_{tj}|=\sum_{tj}|s_{tj}|$. The sign indeterminacy can be demonstrated similarly.

3. DYADIC CYCLIC DESCENT

The estimates are given by

$$\begin{aligned} &\hat{\boldsymbol{A}},\hat{\boldsymbol{S}} = \text{argmin} J(\boldsymbol{A},\boldsymbol{S}) \\ &\text{s.t. } \|\boldsymbol{a}_{(j)}\|^2=1, j=1,...,r. \end{aligned} \quad (3)$$

The unit norm constraint on the columns of $\boldsymbol{A}$ serves the purpose of ensuring that we do not get estimates where $\boldsymbol{A}\to\infty$ and $\boldsymbol{S}\to 0$ while $\boldsymbol{S}\boldsymbol{A}^T$ is fixed.

There is no closed form solution to this optimization problem and so we develop a cyclic descent (CD) procedure. A natural approach would be a two stage approach:

A-step: given $\boldsymbol{S}$ update $\boldsymbol{A}$;

S-step: given $\boldsymbol{A}$ update $\boldsymbol{S}$.

However we have found a different approach to be much faster. We recall the dyadic expansion

$$\boldsymbol{S}\boldsymbol{A}^T=\sum_{j=1}^{r}\boldsymbol{s}_{(j)}\boldsymbol{a}_{(j)}^T$$

This leads to a r-step CD as follows. For $j=1,...,r$ given $\boldsymbol{S}_{-j},\boldsymbol{A}_{-j}^T$ update

$$\begin{aligned} \boldsymbol{a}_{(j)}^{(k+1)} &= \arg.\min_a J_j(\boldsymbol{a},\boldsymbol{s}^{(k)}) \\ \boldsymbol{s}_{(j)}^{(k+1)} &= \arg.\min_s J_j(\boldsymbol{a}_{(j)}^{(k+1)},\boldsymbol{s}) \end{aligned}$$

where

$$\begin{aligned} J_j(\boldsymbol{a},\boldsymbol{s}) &= \frac{1}{2}\|\boldsymbol{R}_j-\boldsymbol{s}\boldsymbol{a}^T\|^2+h\|\boldsymbol{s}\|_1 \\ \boldsymbol{R}_j &= \boldsymbol{Y}-\boldsymbol{S}_{-j}\boldsymbol{A}_{-j}^T. \end{aligned}$$

3.1. The mixing vector $\boldsymbol{a}_{(j)}$-step

The $\boldsymbol{a}_{(j)}$-step consists of minimizing

$$\begin{aligned} \boldsymbol{a}_{(j)}^{(k+1)} &= \arg.\min_a \tfrac{1}{2}\|\boldsymbol{R}_j-\boldsymbol{s}_{(j)}^{(k)}\boldsymbol{a}^T\|^2 \\ &\text{s.t. } \|\boldsymbol{a}_{(j)}\|^2=1, j=1,...,r. \end{aligned} \quad (4)$$

Simple application of the Lagrange multiplier theory yields the solution

$$\boldsymbol{a}_{(j)}^{(k+1)}=\frac{\boldsymbol{R}_j^T\boldsymbol{s}_{(j)}^{(k)}}{\|\boldsymbol{R}_j^T\boldsymbol{s}_{(j)}^{(k)}\|}.$$

3.2. The source vector $\boldsymbol{s}_{(j)}$-step

The $\boldsymbol{s}_{(j)}$ step is equivalent to minimizing

$$\boldsymbol{s}_{(j)}^{(k+1)}=\arg.\min_s\frac{1}{2}\|\boldsymbol{R}_j-\boldsymbol{s}\boldsymbol{a}_{(j)}^{(k+1)^T}\|^2+h\sum_{tj}|s_{jt}| \quad (5)$$

This problem is a simple version of the LASSO [2] optimization problem and has the soft-thresholding solution

$$s_{tj}^{(k+1)}=\max(|b_{tj}|-h,0)\text{sgn}(b_{tj}),\quad t=1,...,T. \quad (6)$$

where $\boldsymbol{B}=[b_{tj}]=\boldsymbol{R}_j\boldsymbol{A}$.

Here we discuss the precise relation between our algorithm and those of [11, 15]. sPCA-rSVD [11] also makes use of the dyadic expansion. The first step of [11] is the same as our first step. But in sPCA-rSVD [11] further terms are fitted sequentially so that a full CD is not implemented. This means that the procedure does not converge and that it exhibits inferior performance. This is illustrated in the simulations below.

Algorithm [15] takes the traditional two stage CD approach but with a twist in the A update. Here a dyadic approach is used but in a very different way to ours. $\boldsymbol{A}$ is updated one column at a time. Each update involves a rank 1 singular value decomposition (SVD) but is preceded by a sparsity projection. This means that k-SVD is not a true CD algorithm which is also demonstrated below.

4. TUNING PARAMETER SELECTION

To select the number of components r and the penalty parameter h we use the BIC criterion [17]

$$\text{BIC}_{r,h}=M\log(\frac{\|\boldsymbol{Y}-\hat{\boldsymbol{S}}\hat{\boldsymbol{A}}\|_F^2}{TM})+(n_s+Mr_s-r_s^2)\frac{\log T}{T} \quad (7)$$

where n_s is the number of nonzero parameters in $\hat{\boldsymbol{S}}$ and r_s is the rank of $\boldsymbol{S}$. We select the tuning parameters that minimize the $\text{BIC}_{r,h}$ surface.

5. EXAMPLES

In this section we evaluate the performance of SCA$_1$-DCD vs the k-SVD algorithm [15] and sPCA-rSVD [11]. We use two performance metrics: the normalized MSE (nMSE) which is given by

$$\text{nMSE} = \frac{\|\boldsymbol{S}\boldsymbol{A}^T - \hat{\boldsymbol{S}}\hat{\boldsymbol{A}}^T\|_F^2}{\|\boldsymbol{S}\boldsymbol{A}^T\|_F^2}$$

and the average angle distance (AD) between the columns of the mixing matrix $\boldsymbol{A}$ which is given by

$$\text{AD}(\boldsymbol{A}, \hat{\boldsymbol{A}}) = \frac{1}{r}\sum_{j=1}^{r} \arccos(\boldsymbol{a}_{(j)}^T \hat{\boldsymbol{a}}_{(j)}).$$

Since there is permutation and sign indeterminacy in $\boldsymbol{A}$, the columns of $\boldsymbol{A}$ and $\hat{\boldsymbol{A}}$ were matched before computing AD. Since the development in [11] was focused on the overdetermined case $M > r$ and the underdetermined $M < r$ case in [15] we present two examples focusing on those two cases.

5.1. Example 1 (Over-Determined Case)

The data is simulated according to (1) where the mixing matrix $\boldsymbol{A}$ is selected as an $M \times r$ matrix where $M = 100$ and each element is drawn from a Gaussian distribution with zero mean unit variance. After creation $\boldsymbol{A}$ is scaled so that $\|\boldsymbol{A}\|_F = 1$. The $T \times r$ source matrix $\boldsymbol{S}$ is created by constructing $\boldsymbol{S} = \tilde{\boldsymbol{S}}\boldsymbol{D}$ where $\boldsymbol{D} = \text{diag}(d_1, d_2, ..., d_r)$ and $\tilde{\boldsymbol{S}}$ is a vector of zeros and ones where we set f_S as the fraction of active (nonzero) elements. The noise variance σ^2 is selected according to pre-specified signal to noise variance (SNR) where

$$\text{SNR} = 10\log_{10}\left(\frac{\|\boldsymbol{S}\boldsymbol{A}^T\|_F^2}{TM\sigma^2}\right).$$

In the simulation examples below we examine the performance of SCA$_1$-DCD with respect to k-SVD [15] and sPCA-rSVD [11]. The performance w.r.t. SNR, sparsity and rank is investigated.

5.1.1. Performance w.r.t. SNR

Here we fix the number of components to $r = 2$, $\boldsymbol{D} = \text{diag}(\sqrt{400}, \sqrt{300})$ and the fraction of active elements to $f_S = 0.2$. The performance w.r.t. SNR is evaluated where SNR $= (0.73, -6.66, -9.22, -10.56)$. For each SNR level we generated $\boldsymbol{A}$ once and then $\boldsymbol{Y}$ according to model (1) 100 times. For SCA$_1$-DCD and sPCA-rSVD we use BIC to select the penalty parameter h. Fig 1 shows an example of BIC for SCA$_1$-DCD. Fig. 2 show median AD and median nMSE w.r.t. SNR. SCA$_1$-DCD performs the best both in terms of AD and nMSE and sPCA-rSVD the second best.

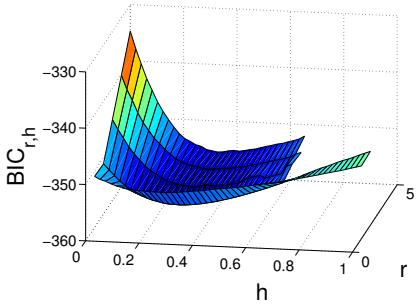
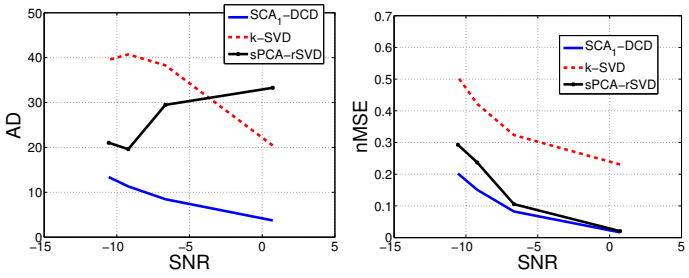


Fig. 1. Example 1 performance w.r.t. SNR. An example of the BIC surface for selecting the penalty parameter h. The minimum is at $r = 2$, $h = 0.29$.

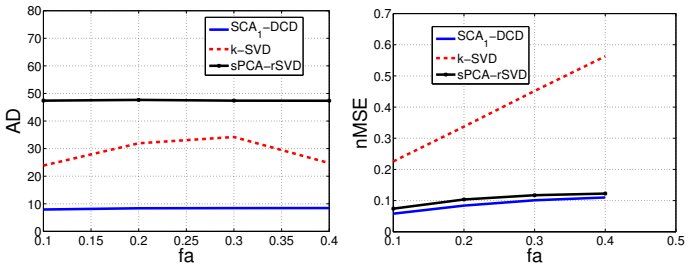


(a) median AD (b) median nMSE

Fig. 2. Example 1 performance w.r.t. SNR, (a) median AD (degrees) vs SNR. (b) median nMSE vs SNR.

5.1.2. Performance w.r.t. sparsity

The number of components and $\boldsymbol{D}$ are selected as before but SNR $= -6.66$. The sparsity is varied such that $f_S = (0.1, 0.2, 0.3, 0.4)$. Fig. 3 show AD and nMSE w.r.t sparsity. For each sparsity level we generated $\boldsymbol{A}$ once and then $\boldsymbol{Y}$ according to model (1) 100 times. Again SCA$_1$-DCD outperforms the other methods by large margin. We note that the performance of SCA$_1$-DCD and sPCA-rSVD do not seem to depend much on the sparsity.

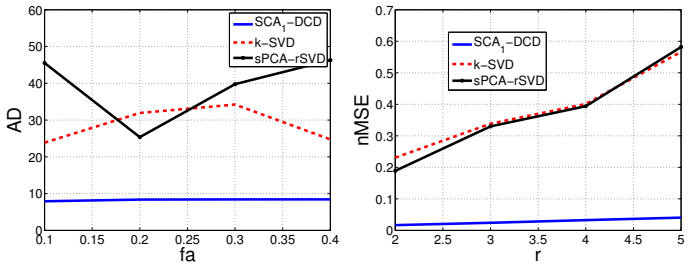


(a) median AD (b) median nMSE

Fig. 3. Example 1 performance w.r.t. sparsity, (a) median AD (degrees) vs sparsity. (b) median nMSE vs sparsity.

5.1.3. Performance w.r.t. rank

The signal to noise ratio is set at SNR $= 0.73$ and the fraction of active elements $f_S = 0.2$. The performance w.r.t. rank is evaluated where $r = 2, 3, 4, 5$ and $d_i = \sqrt{100(r-i+1)+200}$. For each rank level we generated $\boldsymbol{A}$ once and then $\boldsymbol{Y}$ according to model (1) 100 times. Fig 4 show AD and nMSE w.r.t. rank. Yet again SCA_1-DCD



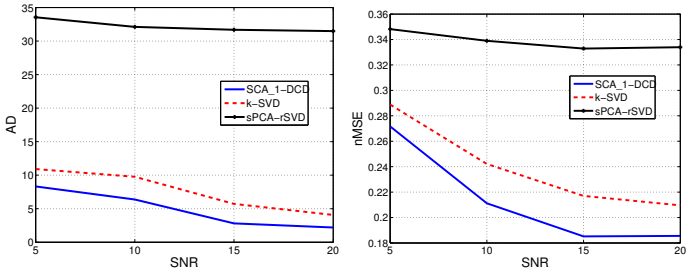
(a) median AD (b) median nMSE

Fig. 4. Example 1 performance w.r.t. rank, (a) median AD (degrees) vs rank. (b) median nMSE vs rank.

outperforms the other methods. The performance of all the methods diminishes with increasing rank.

5.2. Example 2 (Under-Determined Case)

The data is simulated according to (1) where the mixing matrix $\boldsymbol{A}$ is selected as an $M \times r$ matrix where $M = 20$ and $r = 30$. Each element of $\boldsymbol{A}$ is drawn from a Gaussian distribution with zero mean unit variance. The source matrix $\boldsymbol{S}$ is $T \times r$ where $T = 1500$. In each row of $\boldsymbol{S}$ there are $1 \leq T_0 \leq 3$ nonzero elements which are drawn from a $N(0,1)$ distribution. Fig. (5) shows the performance of the methods with respect to SNR. For each SNR value the simulation is performed 100 times. Here we see that SCA_1-DCD outperforms k-SVD. sPCA-rSVD fails relative to the other methods in this case.

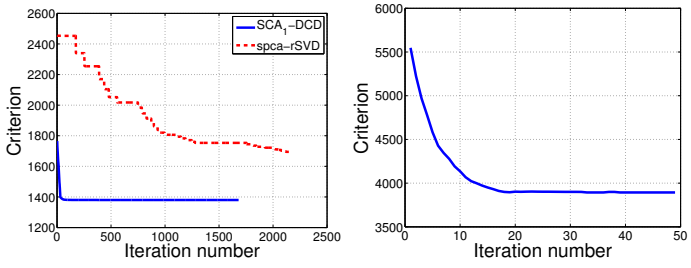


(a) median AD (b) median nMSE

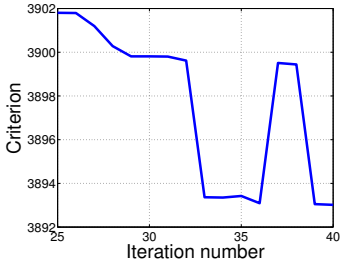
Fig. 5. Example 2 performance w.r.t. SNR, (a) median AD (degrees) vs SNR. (b) median nMSE vs SNR.

5.2.1. Convergence Speed

Fig. 6 shows the optimization criterion for k-SVD SCA_1-DCD and sPCA-rSVD vs iterations. Fig. 6 (a) shows SCA_1-DCD vs sPCA-rSVD. SCA_1-DCD clearly has the faster convergence. It can also be seen that sPCA-rSVD does not convergence. Fig. 6 (b) shows the k-SVD criterion vs iteration. In fact the criterion for k-SVD increases slightly at various places (Fig. 6 (c)) while the SCA_1 criterion is nonincreasing w.r.t iteration (since it is a CD method). We note that the computation time per iteration is smallest for sPCA-rSVD, slightly more for SCA_1-DCD but much more for kSVD. This is reflected in the computation times for generating the plots in Fig. 5. The computation time for kSVD was 29.34 seconds, 1.03 seconds for sPCA-rSVD, and 2.0794 seconds for SCA_1-DCD.



(a) SCD_1-DCD and sPCA-rSVD (b) k-SVD



(c) k-SVD closeup

Fig. 6. Criterion vs iterations for (a) SCD_1-DCD and sPCA-rSVD, (b) k-SVD and (c) k-SVD (closeup).

6. CONCLUSIONS

In this paper we have developed a new algorithm for the l_1 penalized sparse component analysis problem. The algorithm uses a new form of cyclic descent which we call dyadic cyclic descent. We also developed an automatic method for selecting the two tuning parameters involved: the penalty parameter and the number of sparse components. In simulations the performance of SCA_1-DCD was evaluated under various settings and shown to outperform the k-SVD method and the sPCA-rSVD method.

7. REFERENCES

[1] S. Alliney and S. Ruzinsky, "An algorithm for the minimization of mixed l_1 and l_2 norms with application to Bayesian estimation," *IEEE Trans. Signal Proc.*, vol. 42, no. 3, pp. 618–627, 1994.

[2] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Royal Stat. Soc., Series B*, vol. 58, no. 1, pp. 267–288, 1996.

[3] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM review*, vol. 43, no. 1, pp. 129–159, 2001.

[4] J. Hogbom, "Aperture synthesis with a non-regular distribution of interferometer baselines," *Astronomy and Astrophysics Supplement*, vol. 15, pp. 417–426, 1974.

[5] B. Olshausen and D. Field, "Natural image statistics and efficient coding," *Computation in Neural systems*, vol. 7, pp. 333–339, 1996.

[6] M. Zibulevsky and B. Pearlmutter, "Blind source separation by sparse decomposition," *Neural Comput.*, vol. 13, no. 4, 2001.

[7] D. Model and M. Zibulevsky, "Signal reconstruction in sensor arrays using sparse representation," *Signal Processing*, vol. 86, pp. 624–638, 2006.

[8] H. Mohimani, M. Babaie-Zadeh, and C. Jutten, "A fast approach for overcomplete sparse decomposition," *IEEE Trans. Signal Proc.*, vol. 57, no. 1, pp. 289–301, 2009.

[9] R. Gribonval and K. Schnass, "Dictionary identification - sparse matrix factorization via l_1-minimization," *IEEE Trans. Info. Theory*, vol. 56, no. 7, 2010.

[10] A. Hyvarinen, "Independent component analysis in the presence of gaussian noise by maximizing joint likelihood," *Neurocomputing*, vol. 22, pp. 49–67, 1998.

[11] H. Shen and J. Huang, "Sparse principal component analysis via regularized low rank matrix approximation," *J. Multivariate Anal.*, vol. 99, pp. 1015–1034, 2008.

[12] J.Bobin, J. Starck, J. Fadili, and Y.Moudden, "Sparsity and morphological diversity in blind source separation," *IEEE Tran. Image Proc.*, vol. 16, no. 11, 2007.

[13] G. Reinsel and R. Velu, *Multivariate Reduced Rank Regression*, 1st ed. New York: Springer, 1998.

[14] M. Ulfarsson and V. Solo, "Sparse variable reduced rank regression via stiefel optimization," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'11)*, Prague, Czech Republic, 2011.

[15] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Proc.*, vol. 54, no. 11, 2006.

[16] D. G. Luenberger, *Introduction to Linear and Nonlinear Programming*. New York, NY: Addison-Wesley, 1973.

[17] G. Schwartz, "Estimating the dimension of a model," *Ann. Stat.*, vol. 6, no. 2, pp. 461–464, 1978.