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. 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 component 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 l0 penalty is denoted as SCA0. SCA with an l1 penalty is denoted as SCA1. [10] develops an approximate maximum likelihood approach to SCA1; [11] develops an approximate coordinate descent method for SCA1 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 SCA0 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 the 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 SCA1 algorithm using a version of cyclic descent (CD) (aka co-ordinate descent [16]) which we call dyadic CD. We call the new algorithm SCA1-CD. 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 SCA1 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. S. The t-th row vector of S is denoted st, the j-th column vector of S is denoted sj, and the j, t-th element of S is denoted sjt. The Frobenius norm is denoted as ||S||F = \sum s2 j. The matrix A_{-j} is equal to A with its j-th column removed.

2. THE SCA1 PROBLEM

The SCA model is given by

\[ y_t = A s_t + n_t, \quad t = 1, ..., T \]  

(1)

where \( y_t \) is a \( M \times 1 \) vector of observed data \( A \) is an \( M \times r \) mixing matrix, \( s_t \) is a \( r \times 1 \) source vector, and \( n_t \sim N(0, \sigma^2 I_M) \) is a noise vector. The signals and mixing
matrix are estimated by minimizing the following penalized least squares criterion

\[
J(A, S) = \frac{1}{2} \| Y - SA^T \|^2_F + h \sum_{t, j} |s_{tj}|
\]

where \( Y = [y_t^T] \), and \( S = [s_t^T] = [s_{tj}] = [s_{tj}] \). There is a permutation and sign indeterminacy. This can be seen by noticing that for a \( r \times r \) permutation matrix \( P \), \( S = SP^T \) and \( A = AP^T \) we have \( \hat{SA}^T = SA^T \) and \( \sum_{t, j} |s_{tj}| = \sum_{t, j} |s_{tj}| \). The sign indeterminacy can be demonstrated similarly.

3. DYADIC CYCLIC DESCENT

The estimates are given by

\[
\hat{A}, \hat{S} = \arg \min J(A, S)
\text{s.t.} \| a_{(j)} \|^2 = 1, \; j = 1, \ldots, r.
\]

(3)

The unit norm constraint on the columns of \( A \) serves the purpose of ensuring that we do not get estimates where \( A \to 0 \) and \( S \to 0 \) while \( \hat{SA}^T \) 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 \( S \) update \( A \);

S-step: given \( A \) update \( S \).

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

\[
SA^T = \sum_{j=1}^{r} s_{(j)}a_{(j)}^T
\]

This leads to a \( r \)-step CD as follows. For \( j = 1, \ldots, r \) given \( S_{-j}, A_{-j}^T \) update

\[
a_{(j)}^{(k+1)} = \arg \min_a J_j(a, s_{(j)}^{(k)})
\]

\[
s_{(j)}^{(k+1)} = \arg \min_s J_j(a_{(j)}^{(k+1)}, s)
\]

where

\[
J_j(a, s) = \frac{1}{2} \| R_j - sa^T \|^2 + h \sum_{t, j} |s_{tj}|
\]

\[
R_j = Y - S_{-j}A_{-j}^T
\]

3.1. The mixing vector \( a_{(j)} \)-step

The \( a_{(j)} \)-step consists of minimizing

\[
a_{(j)}^{(k+1)} = \arg \min_a \frac{1}{2} \| R_j - s_{(j)}a^T \|^2
\]

s.t. \( \| a_{(j)} \|^2 = 1, \; j = 1, \ldots, r. \)

(4)

Simple application of the Lagrange multiplier theory yields the solution

\[
a_{(j)}^{(k+1)} = \frac{R_j s_{(j)}^T}{\| R_j s_{(j)} \|}.
\]

3.2. The source vector \( s_{(j)} \)-step

The \( s_{(j)} \) step is equivalent to minimizing

\[
s_{(j)}^{(k+1)} = \arg \min_s \frac{1}{2} \| R_j - sa_{(j)}^{(k+1)} \|^2 + h \sum_{t, j} |s_{tj}|
\]

(5)

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

\[
s_{tj}^{(k+1)} = \max(|b_{tj} | - h, 0) \text{sgn}(b_{tj}), \; t = 1, \ldots, T.
\]

(6)

where \( B = [b_{tj}] = R_j A \).

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. \( A \) 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]

\[
\text{BIC}_{r, h} = M \log \left( \frac{\| Y - \hat{S} \hat{A} \|^2_F}{T^2} \right) + (n_s + Mr_s - r_s^2) \frac{\log T}{T}
\]

(7)

where \( n_s \) is the number of nonzero parameters in \( \hat{S} \) and \( r_s \) is the rank of \( S \). We select the tuning parameters that minimize the BIC_{r, h} surface.
5. Examples

In this section we evaluate the performance of SCA₁-DCD vs the k-SVD algorithm [15] and sPCA-rSVD [11]. We use two performance metrics: the normalized MSE (nMSE) which is given by
\[
nMSE = \frac{\|SA^T - \hat{S}A^T\|_F^2}{\|SA^T\|_F^2},
\]
and the average angle distance (AD) between the columns of the mixing matrix \(A\) which is given by
\[
AD(A, \hat{A}) = \frac{1}{r} \sum_{j=1}^{r} \arccos(\langle a_j^T, \hat{a}_j^T \rangle).
\]

Since there is permutation and sign indeterminacy in \(A\), the columns of \(A\) and \(\hat{A}\) were matched before computing AD.

5.1. Example 1 (Over-Determined Case)

The data is simulated according to (1) where the mixing matrix \(A\) 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 \(A\) is scaled so that \(\|A\|_F = 1\). The \(T \times r\) source matrix \(S\) is created by constructing \(S = SD\) where \(D = \text{diag}(d_1, d_2, ..., d_r)\) and \(S\) is a vector of zeros and ones where we set \(f_S\) as the fraction of active (nonzero) elements. The noise variance \(\sigma^2\) is selected according to pre-specified signal to noise variance (SNR) where
\[
\text{SNR} = 10 \log_{10} \left( \frac{\|SA^T\|_F^2}{TM\sigma^2} \right).
\]

In the simulation examples below we examine the performance of SCA₁-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 \(r = 2\), \(D = \text{diag}(\sqrt{100}, \sqrt{300})\) 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 \(A\) once and then \(Y\) according to model (1) 100 times. For SCA₁-DCD and sPCA-rSVD we use BIC to select the penalty parameter \(h\). Fig 1 shows an example of BIC for SCA₁-DCD. Fig. 2 shows median AD and median nMSE w.r.t. SNR. SCA₁-DCD performs the best both in terms of AD and nMSE and sPCA-rSVD the second best.

5.1.2. Performance w.r.t. sparsity

The number of components and \(D\) 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 shows AD and nMSE w.r.t sparsity. For each sparsity level we generated \(A\) once and then \(Y\) according to model (1) 100 times. Again SCA₁-DCD outperforms the other methods by large margin. We note that the performance of SCA₁-DCD and sPCA-rSVD do not seem to depend much on the 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 \( A \) once and then \( Y \) according to model (1) 100 times. Fig 4 shows AD and nMSE w.r.t. rank. Yet again SCA1-DCD 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 \( A \) is selected as an \( M \times r \) matrix where \( M = 20 \) and \( r = 30 \). Each element of \( A \) is drawn from a Gaussian distribution with zero mean unit variance. The source matrix \( S \) is \( T \times r \) where \( T = 1500 \). In each row of \( S \) 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 SCA1-DCD outperforms k-SVD. sPCA-rSVD fails relative to the other methods in this case.

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 SCA1-DCD was evaluated under various settings and shown to outperform the k-SVD method and the sPCA-rSVD method.
7. REFERENCES


