SPARSE AND LOW RANK DECOMPOSITION USING l_0 PENALTY

M.O. Ulfarsson[†], *V. Solo*[‡] *and G. Marjanovic*[‡]

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

ABSTRACT

High dimensional data is often modeled as a linear combination of a sparse component, a low-rank component, and noise. An example is a video sequence of a busy scene where the background is the low-rank part and the foreground, e.g. moving pedestrians, is the sparse part. Sparse and low rank (SLR) matrix decomposition is a recent method that estimates those components. In this paper we develop an l_0 based SLR method and an associated tuning parameter selection method based on the extended Bayesian information criterion (EBIC) method. In simulations the new algorithm is compared with state of the art algorithms from the literature.

Index Terms— Sparse and Low Rank Matrix Decomposition, Cyclic Descent, Extended BIC, l_0 penalty.

1. INTRODUCTION

Due to ever increasing data acquisition capability in important fields such as genomics, brain imaging, and remote sensing the need for efficient algorithms for processing such data is increasing. Usually such data is assumed to have some known structure such as sparsity, positivity or smoothness.

A typical characteristic of high dimensional data is that it lies, approximately, on a low dimensional linear subspace. Many classical algorithms attempt to exploit that property, e.g., principal component analysis [1], independent component analysis [2], and sparse component analysis [3].

In the presence of large sparse noise, i.e., outliers, the classical methods often break down. Motivated by this [4, 5] introduced a noiseless sparse low rank decomposition (SLR) method that decomposes the noiseless observed matrix into a sum of a sparse matrix and a low rank matrix. Furthermore they provided conditions for the exact recovery of those matrices.

A limitation of these papers is that they focus on the noiseless case since in real applications there is always some additive noise. The paper [6] provided a remedy and extended the method to handle the noisy case. Other papers that treat the noisy case include [7] that also handles the case of missing data, [8] which developed a randomized estimation algorithm, and [9] that presents an SLR algorithm assuming structured sparsity in the sparse matrix.

SLR has been found useful for a great variety of applications: [10] applies SLR for the separation of MRI images into background and dynamic components; [11] develops a SLR decomposition for alignment of images; [12] provides a review of SLR methods and its applications.

In this paper we focus on the noisy SLR model and present a novel method based on optimizing an l_0 penalized cost function where the l_0 penalty is both used for enforcing sparsity and low rank. This is different from [6, 7] which focused on the l_1 penalty to encourage sparsity of the sparse component and a nuclear norm to enforce low rank. We additionally developed an extended Bayesian information criterion (EBIC) method [13] for the important problem of selecting the tuning parameters that control the sparseness and the rank. The tuning parameter problem was not treated in [6, 7].

The paper is organized as follows. In section 2 we introduce the SLR model, its associated estimation algorithm, and the tuning parameter selection criterion. Section 3 presents a simulation study and compares the new algorithm to competing methods. Finally, in section 4, conclusions are presented.

1.1. Notation

The Frobenius norm of a matrix $\boldsymbol{X}$ is denoted by $\|\boldsymbol{X}\|_F^2 = \sum_{t=1}^{T}\sum_{v=1}^{M} x_{tv}$; $\|\boldsymbol{X}\|_1 = \sum_{t=1}^{T}\sum_{v=1}^{M} |x_{tv}|$ is the l_1 norm of a matrix $\boldsymbol{X}$; $\|\boldsymbol{X}\|_0 = \sum_{t=1}^{T}\sum_{v=1}^{M} I(x_{tv} \neq 0)$, where $I(\cdot)$ is the indicator function, is the l_0 penalty of a matrix $\boldsymbol{X}$; $\mathcal{H}_h(\cdot)$ is the hard thresholding operator and operates elementwise on its input, i.e. the i,jth element of $\mathcal{H}_h(\boldsymbol{Y})$ is $\mathcal{H}_h(\boldsymbol{Y})_{ij} = y_{tv} I(|y_{tv}| \geq h)$; similarly $\mathcal{S}_h(\boldsymbol{Y})_{ij} = \max(|y_{ij}| - h, 0)\text{sign}(y_{ij})$ is the soft thresholding operator.

2. SPARSE AND LOW RANK MODEL

The sparse and low rank model is given by

$$\boldsymbol{Y} = \boldsymbol{L} + \boldsymbol{X} + \boldsymbol{\epsilon} \tag{1}$$

where $\boldsymbol{Y}$ is a $T \times M$ observed matrix, $\boldsymbol{L}$ is a low rank matrix of rank r where $r < \min(T, M)$, $\boldsymbol{X}$ is a sparse matrix, and

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

$\boldsymbol{\epsilon}$ is additive noise where the elements are independent and identically distributed zero-mean Gaussian random variables with noise variance σ^2. This model was treated in [6] which developed the robust Principal Component Pursuit (RPCP) method that is based on the following criterion:

$$J(\boldsymbol{L},\boldsymbol{X}) = \frac{1}{2\mu}\|\boldsymbol{Y}-\boldsymbol{L}-\boldsymbol{X}\|_F^2 + \lambda\|\boldsymbol{X}\|_1 + \|\boldsymbol{L}\|_*. \qquad (2)$$

The nuclear penalty $\|\cdot\|_*$ is simply an l_1 penalty on the singular values of $\boldsymbol{L}$ and encourages $\boldsymbol{L}$ to be of low rank. In this paper we enforce low rank in a different way and use the property that a low rank matrix $\boldsymbol{L}$ can be written as a product of two lower dimensional matrices $\boldsymbol{A}_{M\times r}$ and $\boldsymbol{S}_{T\times r}$ that are of full rank $r < \min(T,M)$, i.e., $\boldsymbol{L} = \boldsymbol{S}\boldsymbol{A}^T$ [14]. Based on this idea we base our estimation on the following l_0 penalized least squares criterion:

$$\begin{array}{ll} \min_{\boldsymbol{A},\boldsymbol{S},\boldsymbol{X}} & \frac{1}{2}\|\boldsymbol{Y}-\boldsymbol{S}\boldsymbol{A}^T-\boldsymbol{X}\|_F^2 + \frac{h^2}{2}\|\boldsymbol{X}\|_0 \\ \text{s.t.} & \boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{I}_r \end{array} \qquad (3)$$

where the constraint is added to ensure identifiability. We call this problem SLR$_0$. Below in the simulation section we will also be concerned with the sparse low rank problem with l_1 penalty:

$$\begin{array}{ll} \min_{\boldsymbol{A},\boldsymbol{S},\boldsymbol{X}} & \frac{1}{2}\|\boldsymbol{Y}-\boldsymbol{S}\boldsymbol{A}^T-\boldsymbol{X}\|_F^2 + h\|\boldsymbol{X}\|_1 \\ \text{s.t.} & \boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{I}_r. \end{array} \qquad (4)$$

We call the l_1 penalized problem SLR$_1$. Notice that the difference between the form of the weight of the penalties in (3) and (4) is to ensure that h has the same units as σ in both cases.

2.1. Cyclic Descent

The optimization algorithm we use to solve (3) is a cyclic descent (CD) algorithm [15]. The algorithm is based on the following three simple steps that are iterated until convergence (subscript k is the iteration index):

X-step : Given $\boldsymbol{S}_k$ and $\boldsymbol{A}_k$ solve (3) w.r.t. $\boldsymbol{X}$ yielding

$$\begin{aligned} \boldsymbol{X}_{k+1} &= \text{argmin}_{\boldsymbol{X}} J(\boldsymbol{A}_k,\boldsymbol{S}_k,\boldsymbol{X}) \\ &= \mathcal{H}_h(\boldsymbol{Y}-\boldsymbol{S}_k\boldsymbol{A}_k^T) \end{aligned}$$

where $\mathcal{H}_h(\cdot)$ is the hard thresholding operator.

S-step : Given $\boldsymbol{X}_{k+1}$, $\boldsymbol{A}_k$ solve (3) w.r.t. $\boldsymbol{S}$ yielding

$$\begin{aligned} \boldsymbol{S}_{k+1} &= \text{argmin}_{\boldsymbol{S}} J(\boldsymbol{A}_k,\boldsymbol{S},\boldsymbol{X}_{k+1}) \\ &= (\boldsymbol{Y}-\boldsymbol{X}_{k+1})\boldsymbol{A}_k \end{aligned}$$

A-step : Given $\boldsymbol{X}_{k+1}$, $\boldsymbol{S}_{k+1}$ solve (3) w.r.t. $\boldsymbol{A}$ yielding

$$\begin{aligned} \boldsymbol{A}_{k+1} &= \text{argmin}_{\boldsymbol{A}} J(\boldsymbol{A},\boldsymbol{S}_{k+1},\boldsymbol{X}_{k+1}) \\ &= \boldsymbol{P}_r\boldsymbol{Q}_r^T \end{aligned}$$

where $\boldsymbol{P}\boldsymbol{D}\boldsymbol{Q}^T = (\boldsymbol{Y}-\boldsymbol{X}_{k+1})^T\boldsymbol{S}_{k+1}$ is a singular value decomposition (SVD) and $\boldsymbol{P}_r, \boldsymbol{Q}_r$ denote matrices consisting of the first r columns of $\boldsymbol{P}$ and $\boldsymbol{Q}$ respectively.

We call the algorithm CD-SLR$_0$ and summarize it as follows:

```
CD-SLR$_0$
  Input: Data matrix $\boldsymbol{Y}$,$r$ and $h$
  Initialization: $\boldsymbol{A}_0 = \mathbf{0}$ and $\boldsymbol{S}_0 = \mathbf{0}$
  while (Not converged) do
      $\boldsymbol{X}_{k+1} = \mathcal{H}_h(\boldsymbol{Y}-\boldsymbol{S}_k\boldsymbol{A}_k^T)$
      $\boldsymbol{S}_{k+1} = (\boldsymbol{Y}-\boldsymbol{X}_{k+1})\boldsymbol{A}_k$
      $\boldsymbol{P}\boldsymbol{D}\boldsymbol{Q}^T = (\boldsymbol{Y}-\boldsymbol{X}_{k+1})^T\boldsymbol{S}_{k+1}$    (SVD)
      $\boldsymbol{A}_{k+1} = \boldsymbol{P}_r\boldsymbol{Q}_r^T$
  Output: $\hat{\boldsymbol{X}}$,$\hat{\boldsymbol{S}}$ and $\hat{\boldsymbol{A}}$.
```

Remark 1. By construction the CD-SLR$_0$ method ensures monotonicity of the cost iterates, i.e.,

$$\begin{aligned} J(\boldsymbol{A}_k,\boldsymbol{S}_k,\boldsymbol{X}_k) &\geq J(\boldsymbol{A}_k,\boldsymbol{S}_k,\boldsymbol{X}_{k+1}) \\ &\geq J(\boldsymbol{A}_k,\boldsymbol{S}_{k+1},\boldsymbol{X}_{k+1}) \\ &\geq J(\boldsymbol{A}_{k+1},\boldsymbol{S}_{k+1},\boldsymbol{X}_{k+1}) \geq 0. \end{aligned}$$

Remark 2. The CD-SLR$_1$ method (4) is implemented by exchanging the soft thresholding operator for the hard thresholding operator in the $\boldsymbol{X}$ step.

Remark 3. Notice that the $\boldsymbol{S}$-step and the $\boldsymbol{A}$-step can be solved together using the fact that the solution of

$$\hat{\boldsymbol{A}},\hat{\boldsymbol{S}} = \text{argmin}_{\boldsymbol{A},\boldsymbol{S}}\|\boldsymbol{R}-\boldsymbol{S}\boldsymbol{A}^T\|_F^2$$

is $\hat{\boldsymbol{A}} = \boldsymbol{Q}_r$ and $\hat{\boldsymbol{S}} = \boldsymbol{P}_r\boldsymbol{D}_r$ where $\boldsymbol{Q}_r$ consists of the first r left eigenvectors of $\boldsymbol{R}$, $\boldsymbol{P}_r$ consists of the first r right eigenvectors of $\boldsymbol{R}$, and $\boldsymbol{D}_r$ is a diagonal matrix of the first r singular values of $\boldsymbol{R}$. However this requires a costly SVD of a $T\times M$ matrix while our algorithm only requires an SVD of a smaller dimensional $M\times r$ matrix.

2.2. Tuning parameter selection

There are two tuning parameters in the CD-SLR$_0$ model that need to be selected, i.e., the rank r and the sparseness tuning parameter h. Traditional methods for tuning parameter selection include AIC [16], BIC [17], and cross-validation [18]. They were all originally designed for selecting one discrete parameter. However they can be easily modified for selecting multiple continuous and discrete parameters. Based on our experiments with CD-SLR$_0$ these methods select models with too many false positives. Consequently we choose the extended BIC (EBIC) [13] which is known to tightly control the false positive rate. We use EBIC similar to [19]:

$$\begin{aligned}\mathrm{EBIC}_{r,h} &= M\log(\hat{\sigma}^2) + \frac{1}{T}\frac{\|\boldsymbol{Y} - \hat{\boldsymbol{S}}_{r,h}\hat{\boldsymbol{A}}_{r,h}^{T} - \hat{\boldsymbol{X}}_{r,h}\|_F^2}{\hat{\sigma}^2} \\ &+ \frac{(\log(T) + 4\alpha\log(M))d_e(r,h)}{T}\end{aligned}$$

where $\alpha \in [0,1]$ and the effective dimensionality $d_e(r,h)$ of the model is given by

$$d_e(r,h) = Tr + Mr - r^2 + \|\hat{\boldsymbol{X}}_{r,h}\|_0.$$

Note that subscripts have been added on the estimates to emphasize their dependencies on the tuning parameter r, h. In the examples below we use $\alpha = 0.5$ and

$$\hat{\sigma}^2 = \frac{1}{TM}\|\boldsymbol{Y} - \hat{\boldsymbol{S}}\hat{\boldsymbol{A}}^{T} - \hat{\boldsymbol{X}}\|_F^2.$$

2.3. Performance evaluation

To evaluate the performance of our algorithm in the simulations we use the true positive rate (TPR), false positive rate (FPR), and the normalized mean-squared error (nMSE). Before defining FPR and TPR we need few preliminary definitions. Define the null set $\Gamma_0 = \{(t,v) : x_{t,v} = 0\}$, the active set $\Gamma_a = \{(t,v) : x_{t,v} \neq 0\}$, $\hat{\Gamma}_0 = \{(t,v) : \hat{x}_{t,v} = 0\}$ and $\hat{\Gamma}_a = \{(t,v) : \hat{x}_{t,v} \neq 0\}$. True positive (TP) is defined as $\mathrm{TP} = |\Gamma_a \cap \hat{\Gamma}_a|$, false negative (FN) as $\mathrm{FN} = |\Gamma_a \cap \hat{\Gamma}_0|$, false positive (FP) as $\mathrm{FP} = |\Gamma_0 \cap \hat{\Gamma}_a|$ and true negative (TN) as $\mathrm{TN} = |\Gamma_0 \cap \hat{\Gamma}_0|$, here $|\cdot|$ denotes the cardinality of the set. Now TPR and FPR are defined as

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{5}$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}. \tag{6}$$

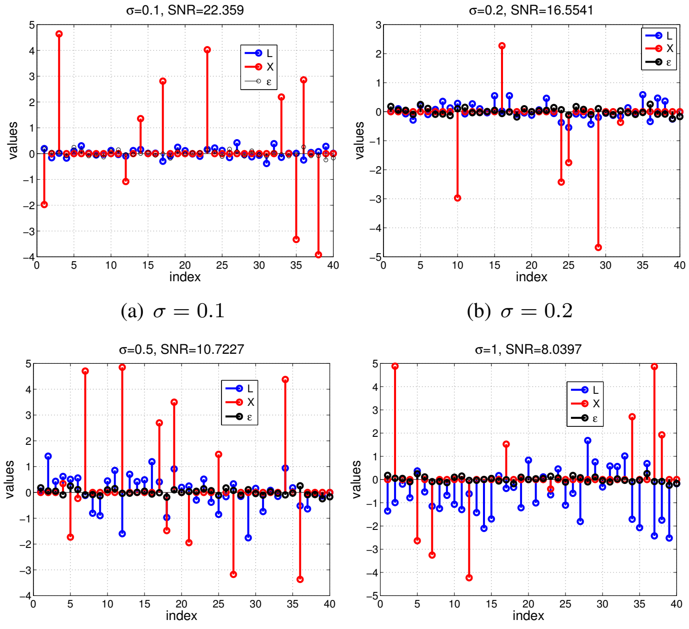
The nMSE is defined as

$$\mathrm{nMSE} = \frac{\|\boldsymbol{L} - \hat{\boldsymbol{S}}\hat{\boldsymbol{A}}^{T} + \boldsymbol{X} - \hat{\boldsymbol{X}}\|_F^2}{\|\boldsymbol{L} + \boldsymbol{X}\|_F^2} \tag{7}$$

where $\boldsymbol{L} + \boldsymbol{X}$ is the true signal and $\hat{\boldsymbol{S}}\hat{\boldsymbol{A}}^{T} + \hat{\boldsymbol{X}}$ is the estimated signal.

3. SIMULATION STUDY

In this section we compare the performance of CD-SLR$_0$, CD-SLR$_1$ and RPCP (2) from [6] using an implementation from [20] which uses the accelerated proximal gradient method. We simulate data according to (1) where $T = M = 200$. The elements in the matrices $\boldsymbol{A}$ and $\boldsymbol{S}$ where drawn from a Gaussian distribution $N(0, 10\frac{\sigma}{\sqrt{T}})$. The matrix $\boldsymbol{X}$ is constructed such that 20 % of its elements are non-zero (active), the non-zero elements are uniformly distributed between -5 and 5. Fig. 1 shows the first 40 elements in the first row of the matrices $\boldsymbol{L}$, $\boldsymbol{X}$, and $\boldsymbol{\epsilon}$ for four different values of the noise variance σ^2.



(a) $\sigma = 0.1$ (b) $\sigma = 0.2$

(c) $\sigma = 0.5$ (d) $\sigma = 1$

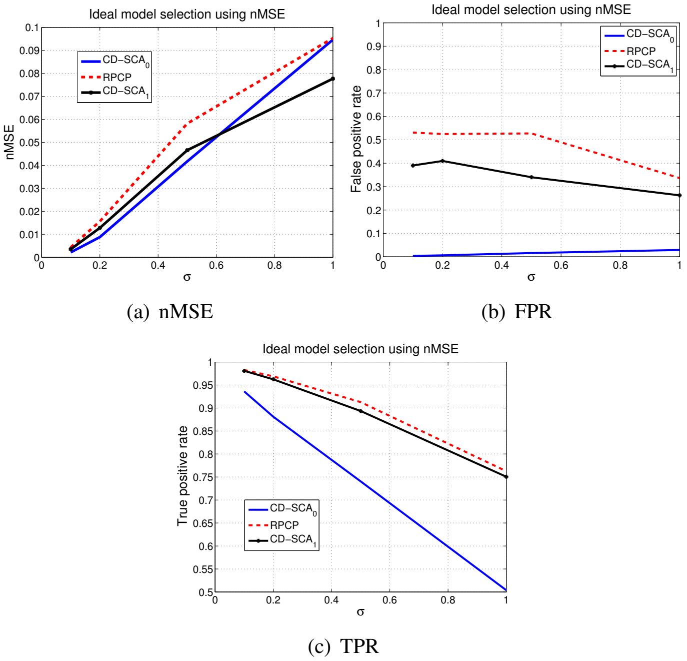
Fig. 1. First 40 elements in the first row of the matrices $\boldsymbol{L}$, $\boldsymbol{X}$, and $\boldsymbol{\epsilon}$ for four different values of the noise variance σ^2.

First we explored the variation of the nMSE (7), TPR (5), and FPR (6) with the standard deviation of the noise σ. We generated a grid of values of the tuning parameters r and h for CD-SLR$_0$ and CD-SLR$_1$ and μ and λ for the RPCP method and for each value of σ selected the tuning parameters based on the lowest value of the nMSE. We note that selecting the tuning parameters based on nMSE is unrealistic in practice since it depends on a knowledge of the true signal. However, this demonstrates the limits of performance in terms of nMSE.

Fig. 2 depicts the averaged (over 10 simulations) nMSE (7), TPR (5), and FPR (6) for the three different algorithms where the tuning parameters are selected based on the best nMSE. The CD-SLR$_0$ and CD-SLR$_1$ methods both perform better than RPCP in terms of nMSE, CD-SLR$_0$ performs better than CD-SLR$_1$ for all values of σ expect $\sigma = 1$. Not unexpectedly CD-SLR$_0$ has the greatest sparsity, i.e. it has lower TPR and FPR than the other methods.

Fig. 3 shows a box plot showing the rank selection for each of the algorithm. The CD-SLR$_0$ and CD-SLR$_1$ methods select the true rank all the time while on the other hand RPCP slightly overestimates the rank at higher noise variance levels.

Fig. 4 show the nMSE, TPR, FPR for each of the method in the more practical settings when the tuning parameters are selected based on the EBIC. Here CD-SLR$_0$ clearly outperforms the other methods in terms of the nMSE.



(a) nMSE

(b) FPR

(c) TPR

Fig. 2. Averaged (over 10 simulations) nMSE (7), TPR (5), and FPR (6) for CD-SLR$_1$, CD-SLR$_0$ and RPCP where the tuning parameters are selected based on the best nMSE

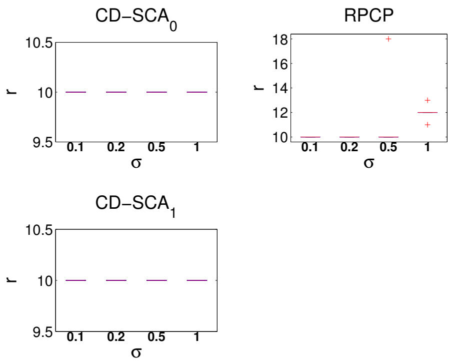


Fig. 3. Box plots showing the rank selection for CD-SLR$_0$, CD-SLR$_1$ and RPCP where the tuning parameters are selected based on the best nMSE.

Fig. 5 shows the boxplot for the rank selection using the EBIC method. EBIC selects the true rank for the CD-SLR$_0$ for all values of the noise variance, the EBIC method slightly underestimates the rank for the CD-SLR$_1$ method. The EBIC method clearly fails at selecting the rank for RPCP.

4. CONCLUSIONS

In this paper we have developed a new algorithm using the l_0 penalty for the SLR problem. The algorithm uses a cyclic descent method for estimation and the extended BIC for the tuning parameter selection of the sparsity parameter and the rank. In simulations the performance of the new method was evaluated under various settings and shown to outperform both the RPCP method and a CD-SLR$_1$ method.

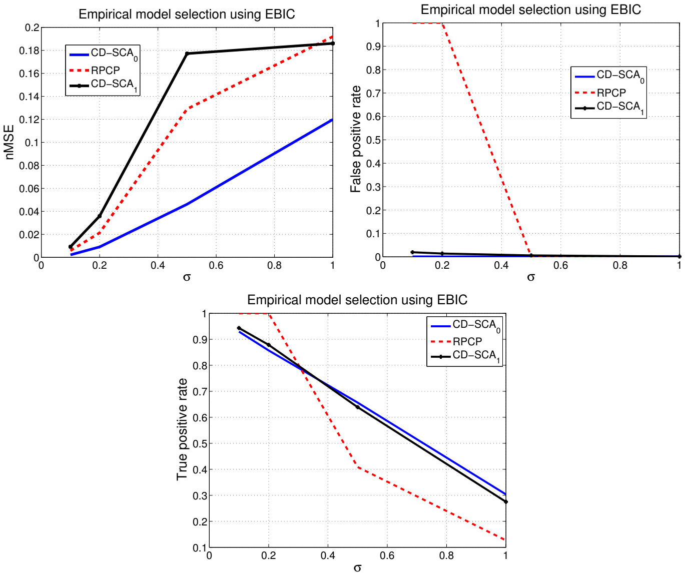


Fig. 4. Averaged (over 10 simulations) nMSE (7), TPR (5), and FPR (6) for CD-SLR$_1$, CD-SLR$_0$ and RPCP where the tuning parameters are selected based on the best EBIC.

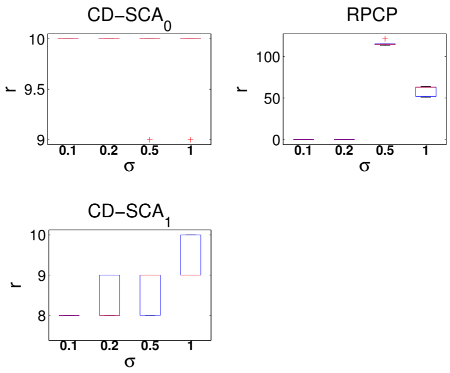


Fig. 5. Box plots showing the rank selection for CD-SLR$_0$, CD-SLR$_1$ and RPCP where the tuning parameters are selected based on the best EBIC.

5. REFERENCES

[1] I.T. Jolliffe, *Principal Component Analysis*, Springer, New York, NY, second edition, 2002.

[2] A. Hyvarien, J. Karhunen, and E Oja, *Independent Component Analysis*, John Wiley and Sons, New York, 2001.

[3] B.A. Olshausen and D.J. Field, "Natural image statistics

and efficient coding," *Computation in Neural systems*, vol. 7, pp. 333–339, 1996.

[4] E. Candes, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis," *Journal of the ACM*, vol. 58, 2011.

[5] V. Chandrasekaran, S. Sanghavi, P.A. Parrilo, and A.S. Willsky, "Sparse and low-rank matrix decomposition," in *27th Annual Allerton Conference on Communication, Control and Computing*, Monticello, Il, 2009, pp. 2962–967.

[6] Z. Zhou, X. Li, J. Wright, E. Candes, and Y. Ma, "Stable principal component analysis," in *IEEE International Symposium on Information Theory*, Austin, Texas, 2010, pp. 1518–1522.

[7] M. Tao and X. Yuan, "Recovering low-rank and sparse components of matrices from incomplete and noisy observations," *SIAM J. Optim.*, vol. 21, no. 1, pp. 57–81, 2009.

[8] T. Zhou and D. Tao, "Godec: Randomized low-rank and sparse matrix decomposition in noisy case," in *Proc. International Conference on Machine Learning*, Bellevue, WA, 2011.

[9] G. Mateos and G. Giannakis, "Robust PCA as bilinear decomposition with outlier-sparsity regularization," *IEEE Trans. Signal Proc.*, vol. 60, no. 10, pp. 5176–5190, 2012.

[10] R. Otazo, E. Candes, and D. Sodickson, "Low-rank and sparse matrix decomposition for accelerated dynamic MRI with separation of background and dynamic components," *Magnetic Resonance in Medicine*, accepted for publication, 2014.

[11] Y. Peng, A. Ganesh, J. Wright, and Y. Ma, "RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images," in *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, 2010.

[12] K. Slavakis, G. Giannakis, and G. Mateos, "Modeling and optimization for big data analytics," *IEEE Signal Processing Magazine*, vol. 31, no. 5, 2014.

[13] J. Chen and Z. Chen, "Extended Bayesian information criteria for model selection with large model spaces," *Biometrika*, vol. 95, no. 3, pp. 759–771, 2008.

[14] G. Reinsel and R. Velu, *Multivariate Reduced Rank Regression*, Springer, New York, first edition, 1998.

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

[16] H. Akaike, "A new look at the statistical model identification," vol. 19, no. 6, pp. 716–723, 1974.

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

[18] M. Stone, "Cross-validory choice and assesment of statistical predictions," *J. Roy. Statist. Soc*, vol. 39, pp. 44–47, 1974.

[19] G. Marjanovic and V. Solo, "On l_q optimization and sparse inverse covariance selection," *IEEE Trans. Signal Proc.*, vol. 62, no. 7, pp. 1644–1654, 2014.

[20] Zhouchen Lin, Arvind Ganesh, John Wright, Leqin Wu, Minming Chen, and Yi Ma, "Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix," in *In Intl. Workshop on Comp. Adv. in Multi-Sensor Adapt. Processing, Aruba, Dutch Antilles*, 2009.