Strict and relaxed rank constrained ML estimator of structured covariance matrices

Bosung Kang[1], Vishal Monga[1] and Muralidhar Rangaswamy[2]
[1]Department of Electrical Engineering, The Pennsylvania State University
[2]Air Force Research Lab, WPAFB, Dayton, OH.

November 2, 2012

1 Introduction

We proposed a rank constrained maximum likelihood estimator of structured covariance matrices which exploits the rank information and structural property of the clutter matrix [1]. Therefore we have two constraints for the optimization problem we solved, $rank(\mathbf{R}_c) = r$ and $\mathbf{R} = \mathbf{R}_c + \sigma^2\mathbf{I}$. In terms of the rank constraint, we used relaxation to the rank so that the closed form solution for the estimator can be derived by solving a convex optimization problem using KKT conditions [2]. Strictly speaking, in other words, we used the maximum rank constraint instead of just rank constraint. We will show merits of relaxed rank constrained estimator to the strict rank constraint and there is no difference between two estimators in view of performance.

2 Two optimization problems

In the case of known noise level, the initial optimization problem for covariance estimator is

$$\begin{cases} \min\limits_{\mathbf{R}} & tr\{\mathbf{R}^{-1}\mathbf{S}\} + \log(|\mathbf{R}|) \\ s.t. & \mathbf{R} = \sigma^2\mathbf{I} + \mathbf{R}_c \\ & rank(\mathbf{R}_c) = r \\ & \mathbf{R}_c \succeq \mathbf{0} \\ & \mathbf{R} \succeq \sigma^2\mathbf{I} \end{cases} \tag{1}$$

By introducing transformation variables and analyzing the eigenstructure of $\mathbf{R}$, we have a final convex optimization problem:

$$\begin{cases} \min\limits_{\boldsymbol{\lambda}} & \mathbf{d}^T\boldsymbol{\lambda} - \mathbf{1}^T\log\boldsymbol{\lambda} \\ s.t. & \mathbf{U}\boldsymbol{\lambda} \preceq \mathbf{0} \\ & -\boldsymbol{\lambda} \preceq -\boldsymbol{\varepsilon} \\ & \boldsymbol{\lambda} \preceq \mathbf{1} \\ & \mathbf{E}\boldsymbol{\lambda} = \mathbf{h} \end{cases} \tag{2}$$

A closed form solution for (2) can in fact be derived using KKT conditions [2] in constrained optimization. The optimal solution $\boldsymbol{\lambda}^\star$ is

$$\lambda_i^\star = \begin{cases} \min(1, \dfrac{1}{d_i}) & \text{for } i = 1, 2, \dots, r \\ 1 & \text{for } i = r+1, r+2, \dots, N \end{cases} \tag{3}$$

The constraints of (2) imply the rank information of the covariance matrix. That is,

$$\lambda_1 \le \lambda_2 \le \cdots \le \lambda_r \le \lambda_{r+1} = \lambda_{r+2} = \cdots = \lambda_N = 1 \tag{4}$$

, which means $rank(\mathbf{R}_c) = r$. However, the constraint $\lambda_r \leq \lambda_{r+1}$ can cause the rank to be less than r when the equality holds, i.e., $\lambda_r = \lambda_{r+1} = 1$. In other words, we use relaxation about the rank constraint, $rank(\mathbf{R}_c) \leq r$.
Now we see a strict rank constrained problem. To make the problem use a strict rank constraint, we have to replace the inequality with a strict inequality. Then the constraints are

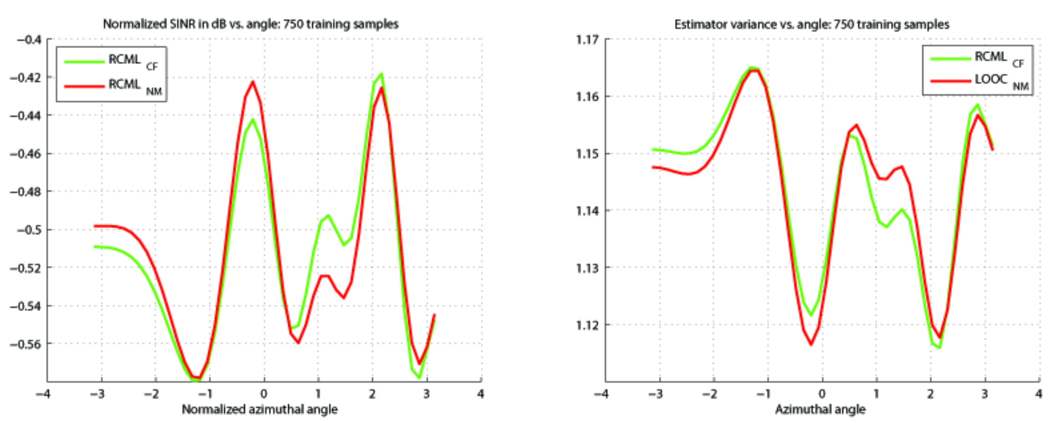
$$\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_r < \lambda_{r+1} = \lambda_{r+2} = \cdots = \lambda_N = 1 \tag{5}$$

The only difference between strict and relaxed rank constraints is a strict inequality between λ_r and λ_{r+1}. Since the cost function still remains a convex function and the feasible sets of λ_i's are still convex, the problem is obviously a convex optimization problem. Though we cannot get a closed form solution because the feasible set is open - it is highly likely that the solutions are very close. This is because the problem is very well conditioned (see Boyd: Convex Optimization Chapter 5 "Perturbation and Sensitivity Analysis") and replacing a relaxed inequality by a strict one is a very mild perturbation on the problem which will cause little or no change to the solution.

The problem is simple enough and we employ Barrier methods from the MATLAB implementation toolbox to obtain the numerical solution of the problem which employs the strict rank constraint. The experimental results in the next Section indeed establish the performance of two estimators is very close to each other and hardly discriminable.

3 Experimental result

We compare the performance of two estimators. One is the closed form solution (3) which comes from a originally proposed rank constrained estimation and the other is a numerical solution from estimation using a strict rank constraint. To solve the problem numerically, we use CVX, a package for specifying and solving convex programs [3, 4]. Figure 1 shows the performance of two estimators in terms of normailized SINR and estimator variance. We denote the closed form solution and the numerical solution of the rank constrained ML estimation by $\mathrm{RCML}_{\mathrm{CF}}$ and $\mathrm{RCML}_{\mathrm{NM}}$, respectively. We can see the gap of performance of two estimators is less than 0.03 in SINR and 0.01 in estimator variance at most. This difference is much less than that among other compared estimators such as SMI, FML, LOOC, and so on, and therefore we can conclude that these two estimators have pretty much the same performance.



(a) (b)

Figure 1: Performance of two estimators for $K = 750$: (a) Normalized SINR vs. normalized azimuthal angle and (b) Estimator variance vs. normalized azimuthal angle

4 Summary

We proposed the rank constrained ML estimator of structured covariance matrix. Though we use the relaxed rank constraint, we showed it performs as well as the estimator using the strict rank constraint through experimental results. In addition, we can also derive the closed form solution by introducing relaxation to the rank constraint, which makes it atrractive from an implementation viewpoint.

References

[1] B. Kang, V. Monga, and M. Rangaswamy, "Rank Constrained ML Estimation of Structured Covariance Matrices," *Accepted to IEEE Trans. Aerosp. Electron. Syst. with minor revisions*, 2012.

[2] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2nd edition, 2004.

[3] Inc. CVX Research, "CVX: Matlab software for disciplined convex programming, version 2.0 beta," `http://cvxr.com/cvx`, September 2012.

[4] M. Grant and S. Boyd, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control*, V. Blondel, S. Boyd, and H. Kimura, Eds., Lecture Notes in Control and Information Sciences, pp. 95–110. Springer-Verlag Limited, 2008, `http://stanford.edu/~boyd/graph_dcp.html`.