Factor Analysis From Quadratic Sampling

Sundeep Prabhakar Chepuri, *Member, IEEE*

***Abstract*—Factor analysis decomposition, i.e., decomposition of a covariance matrix as a sum of a low-rank positive semidefinite matrix and a diagonal matrix is an important problem in a variety of areas, such as signal processing, machine learning, system identification, and statistical inference. In this letter, the focus is on computing the factor analysis decomposition from a set of quadratic (or symmetric rank-one) measurements of a covariance matrix. Commonly used minimum trace factor analysis heuristic can be adapted to solve this problem when all the measurements are available. However, the resulting convex program is not suitable for processing large-scale or streaming data. Therefore, this letter presents a low-complexity iterative algorithm, which recovers the unknowns through a series of rank-one updates. The iterative algorithm performs better than the convex program when only a finite number of data snapshots are available.**

***Index Terms*—Covariance sketching, factor analysis, Kaczmarz method, quadratic measurements, stochastic gradient descent.**

I. Introduction

Estimation of a structured covariance matrix plays a central role in many areas, such as signal processing, machine learning, system identification, and statistical inference. It is gaining a significant amount of attention, particularly in scenarios where the number of measurements is much smaller than the ambient dimension of the data.

In this letter, the focus is on recovering a covariance matrix $\boldsymbol{\Sigma} \in \mathbb{C}^{N \times N}$ from M *quadratic measurements* of the form

$$b_i = \boldsymbol{a}_i^H \boldsymbol{\Sigma} \boldsymbol{a}_i, \quad i = 1, 2, \ldots, M. \tag{1}$$

Here, the measurements are obtained using a set of length-N *sketching vectors* $\{\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_M\}$ with $M < N^2$. The superscript $(\cdot)^H$ denotes the complex conjugate transpose. The quadratic measurement scheme, which is sometimes also referred to as *covariance sketching*, has been recently advocated for different structured covariance estimation problems (including phase retrieval, sparse and/or low-rank covariance estimation) [1]–[5] in order to limit the costs of storing, processing, or communicating the data.

The aim of this letter is to estimate $\boldsymbol{\Sigma}$ under the assumption that it admits a *factor analysis decomposition*

$$\boldsymbol{\Sigma} = \boldsymbol{L} + \boldsymbol{D} = \boldsymbol{F}\boldsymbol{F}^H + \boldsymbol{D}. \tag{2}$$

That is, $\boldsymbol{\Sigma}$ can be decomposed into a sum of a low-rank positive semidefinite matrix $\boldsymbol{L} \in \mathbb{C}^{N \times N}$ of known rank R and a nonnegative diagonal matrix $\boldsymbol{D} \in \mathbb{C}^{N \times N}$. Since $\boldsymbol{L}$ is a low-rank positive semidefinite matrix, it can be represented with a matrix $\boldsymbol{F} \in \mathbb{C}^{N \times R}$ as $\boldsymbol{L} = \boldsymbol{F}\boldsymbol{F}^H$. Recovering $\boldsymbol{\Sigma}$ now amounts to estimating the unknown matrices $\boldsymbol{F}$ and $\boldsymbol{D}$. Therefore, the main question of interest is, *can we estimate* $\boldsymbol{F}$ *and* $\boldsymbol{D}$ *from the quadratic measurements* $\{b_1, b_2, \ldots, b_M\}$?

In practice, we will not sketch the true covariance matrix $\boldsymbol{\Sigma}$, but sketch a stream of length-N data vectors $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_S\}$ or the sample covariance matrix

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{S} \sum_{j=1}^{S} \boldsymbol{x}_j \boldsymbol{x}_j^H \tag{3}$$

computed from S data snapshots. Forming the sample covariance matrix $\hat{\boldsymbol{\Sigma}}$ costs about order $N^2 S$ flops. Forming, storing, or transmitting $\hat{\boldsymbol{\Sigma}}$ might become prohibitive depending on the scale of the problem. In contrast, using the quadratic subsampling scheme (1), we need to transmit and/or store only a set of MS samples $\{|\boldsymbol{a}_i^H \boldsymbol{x}_j|^2, 1 \leq i \leq M, 1 \leq j \leq S\}$.

A. Identifiability and Uniqueness

For identifiability, generally more equations are needed than the number of unknowns. Here, the number of equations available is M, and the number of unknowns is cNR *real* parameters of $\boldsymbol{F}$ ($c = 1$ for real matrices $\boldsymbol{F}$ and $c = 2$ for complex matrices $\boldsymbol{F}$) and N parameters of $\boldsymbol{D}$. In total, the number of unknowns is $N(cR + 1)$, which leads to the condition $M \geq N(cR + 1)$. Also, it gives an upper bound $R \leq \min\{c^{-1}(M/N - 1), N\}$ on the factor rank (the decomposition (2) is useless for $R \geq N$).

It should be noted that $\boldsymbol{F}$ can be identified only up to an arbitrary unitary factor $\boldsymbol{V}$ at the right as $\boldsymbol{F}\boldsymbol{V}$ is also a solution.

B. Context

Factor analysis is useful to describe a latent variable model [6]–[10], in which the observed data $\boldsymbol{x} \in \mathbb{C}^N$ may be explained with a small number of unobserved latent variables (or the so-called factors) $\boldsymbol{\theta} \in \mathbb{C}^R$ and factor loadings collected in an $N \times R$ matrix $\boldsymbol{F}$ as

$$\boldsymbol{x} = \boldsymbol{F}\boldsymbol{\theta} + \boldsymbol{\nu}$$

with a probabilistic prior on $\boldsymbol{\theta}$ that $\mathbb{E}\{\boldsymbol{\theta}\boldsymbol{\theta}^H\} = \boldsymbol{I}$. Here, $\boldsymbol{\nu} \in \mathbb{C}^N$ denotes the noise vector with a diagonal covariance matrix

Manuscript received July 3, 2017; revised September 5, 2017; accepted September 25, 2017. Date of publication October 16, 2017; date of current version November 22, 2017. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ashish Pandharipande.

The author is with the Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Delft 2628CD The Netherlands (e-mail: s.p.chepuri@tudelft.nl).

Color versions of one or more of the figures in this letter are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LSP.2017.2763154

$\mathbb{E}\{\boldsymbol{\nu}\boldsymbol{\nu}^H\} = \boldsymbol{D}$. For such models, the covariance matrix of $\boldsymbol{x}$, i.e., $\boldsymbol{\Sigma} = \mathbb{E}\{\boldsymbol{x}\boldsymbol{x}^H\}$, admits a factor analysis decomposition. When $\boldsymbol{D}$ is not a scaled identity matrix (i.e., if the noise is not white), the commonly used eigenvalue decomposition is not useful anymore as it does not reveal the subspace of $\boldsymbol{F}$.

The developed framework is useful for applications like sensor array calibration and interference cancellation in radio astronomy [9], [11] where large arrays are predominant (e.g., the square kilometer array [12]), topology inference [13], [14] of large-scale networks, and image restoration in cryo-electron microscopy [15], [16], to list a few.

C. Related Works and Contributions

Conventional factor analysis decomposition algorithms solve for the unknown matrices $\boldsymbol{F}$ and $\boldsymbol{D}$, assuming $\boldsymbol{\Sigma}$ is completely known, using iterative techniques, such as expectation-maximization algorithm [17], [18], semidefinite program [19], [20], or alternating least squares (involving eigenvalue decomposition in each iteration) [9]. These algorithms cannot be used to compute the factor analysis decomposition when only an affine slice of $\boldsymbol{\Sigma}$ is available.

The contributions of this letter are as follows. To begin with, in Section II, we develop a convex program for estimating $\boldsymbol{F}$ and $\boldsymbol{D}$ from the quadratic measurements. Although the resulting semidefinite programming problem can be solved using off-the-shelf solvers, it might be computationally intensive, and also, it is not suitable for processing streaming data. Therefore, a new low-complexity iterative algorithm is proposed in Section III for computing and updating the factor analysis decomposition as more data becomes known. The developed algorithms are also useful when $\boldsymbol{D}$ is a scaled identity matrix.

II. Convex Program

Letting $\mathcal{A} : \mathbb{C}^{N\times N} \to \mathbb{R}^M$ denote a linear transformation that maps Hermitian matrices into real-valued vectors, the quadratic measurements (1) collected in $\boldsymbol{b}$ as $\boldsymbol{b} = [b_1, \ldots, b_M]^T$ can be compactly expressed as

$$\boldsymbol{b} = \mathcal{A}(\boldsymbol{\Sigma}) = \mathcal{A}(\boldsymbol{F}\boldsymbol{F}^H + \boldsymbol{D}).$$

Given $\boldsymbol{b}$, the factor analysis decomposition may be computed by solving the following rank minimization problem:

$$\begin{array}{ll} \underset{\boldsymbol{L},\boldsymbol{D}}{\text{minimize}} & \text{rank}(\boldsymbol{L}) \\ \text{subject to} & \mathcal{A}(\boldsymbol{L}+\boldsymbol{D}) = \boldsymbol{b} \\ & \boldsymbol{L} \succeq \boldsymbol{0},\ \boldsymbol{D} \succeq \boldsymbol{0} \text{ diagonal} \end{array} \tag{4}$$

with variables $\boldsymbol{L} \in \mathbb{C}^{N\times N}$ and $\boldsymbol{D} \in \mathbb{R}^{N\times N}$. Rank minimization problems are in general NP hard, and a common relaxation technique [21], [22] is to replace the rank function in the objective with a trace operator. After this standard convex relaxation, (4) simplifies to

$$\begin{array}{ll} \underset{\boldsymbol{L},\boldsymbol{D}}{\text{minimize}} & \text{trace}(\boldsymbol{L}) \\ \text{subject to} & \mathcal{A}(\boldsymbol{L}+\boldsymbol{D}) = \boldsymbol{b} \\ & \boldsymbol{L} \succeq \boldsymbol{0},\ \boldsymbol{D} \succeq \boldsymbol{0} \text{ diagonal.} \end{array} \tag{5}$$

The semidefinite program (5) is very similar to the well-known minimum trace factor analysis problem [19], [20] (see also references therein), however, with the main difference being that [19] and [20] assume that the covariance matrix $\boldsymbol{\Sigma}$ is completely known. In contrast, in (5), we only have access to an affine slice of the covariance matrix $\boldsymbol{\Sigma}$.

When only a finite number of snapshots are available, the measurements are not consistent with the model (1). Therefore, to allow for noise in the measurements, it might be more appropriate to replace the equality constraint in (5) with $\|\mathcal{A}(\boldsymbol{L}+\boldsymbol{D}) - \boldsymbol{b}\|_2 \le \varepsilon$, where ε is an appropriately chosen parameter.

In general, there is no guarantee that $\hat{\boldsymbol{L}}$ obtained from (5) will be of rank R. Nonetheless, we may compute an estimate of $\boldsymbol{F}$ by rank reduction. Let $\hat{\boldsymbol{L}} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^H$ with

$$\boldsymbol{U} = \begin{array}{cc|c} [\boldsymbol{U}_1 & \boldsymbol{U}_2] & N; \\ R & N-R & \end{array} \quad \boldsymbol{\Lambda} = \begin{array}{c|c} \begin{bmatrix} \boldsymbol{\Lambda}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Lambda}_2 \end{bmatrix} & \begin{array}{c} R \\ N-R \end{array} \\ \begin{array}{cc} R & N-R \end{array} & \end{array}.$$

Then, an estimate of $\boldsymbol{F}$ will be $\hat{\boldsymbol{F}} = \boldsymbol{U}_1\boldsymbol{\Lambda}_1^{1/2}$. Although, the semidefinite program (5) can be solved efficiently, it still needs a factorization to compute the low-dimensional subspace of interest as we lift up the problem of recovering a low-dimensional subspace into that of recovering a low-rank matrix.

III. Iterative Algorithm

In this section, we develop an iterative algorithm for computing the factor analysis decomposition from a stream of quadratic measurements. The developed algorithm is an adaptation of the Kaczmarz method [23], [24], which is a special case of stochastic gradient descent. It consists of a rank-one update in each iteration to *simultaneously* update the low-dimensional subspace as well as the nonnegative diagonal matrix. The formulation is inspired by [2], and the update equations, in fact, reduce to the ones in [2] for the special case with $\boldsymbol{D} = \boldsymbol{0}$.

By defining a diagonal matrix $\boldsymbol{Q} = \boldsymbol{D}^{1/2}$, (2) can be rewritten as $\boldsymbol{\Sigma} = \boldsymbol{F}\boldsymbol{F}^H + \boldsymbol{Q}^2$. The proposed algorithm sweeps through the measurements, and at each iteration we solve a constrained least squares problem. Given the previous estimate, denoted by $(\boldsymbol{F}_k, \boldsymbol{Q}_k)$, the update equations are obtained by solving the following optimization problem

$$\begin{array}{ll} \underset{\boldsymbol{X},\boldsymbol{Y}}{\text{minimize}} & \|\boldsymbol{F}_k - \boldsymbol{X}\|_F^2 + \|\boldsymbol{Q}_k - \boldsymbol{Y}\|_2^2 \\ \text{subject to} & \boldsymbol{a}_{r(k)}^H[\boldsymbol{X}\boldsymbol{X}^H + \boldsymbol{Y}^2]\boldsymbol{a}_{r(k)} = b_{r(k)} \end{array} \tag{6}$$

with variables $\boldsymbol{X} \in \mathbb{C}^{N\times R}$ and $\boldsymbol{Y} \in \mathbb{R}^{N\times N}$ (diagonal). Here, k is the iteration index and $r(k)$ is chosen from the set $\{1, 2, \ldots, M\}$ uniformly at random or in a cyclic manner,

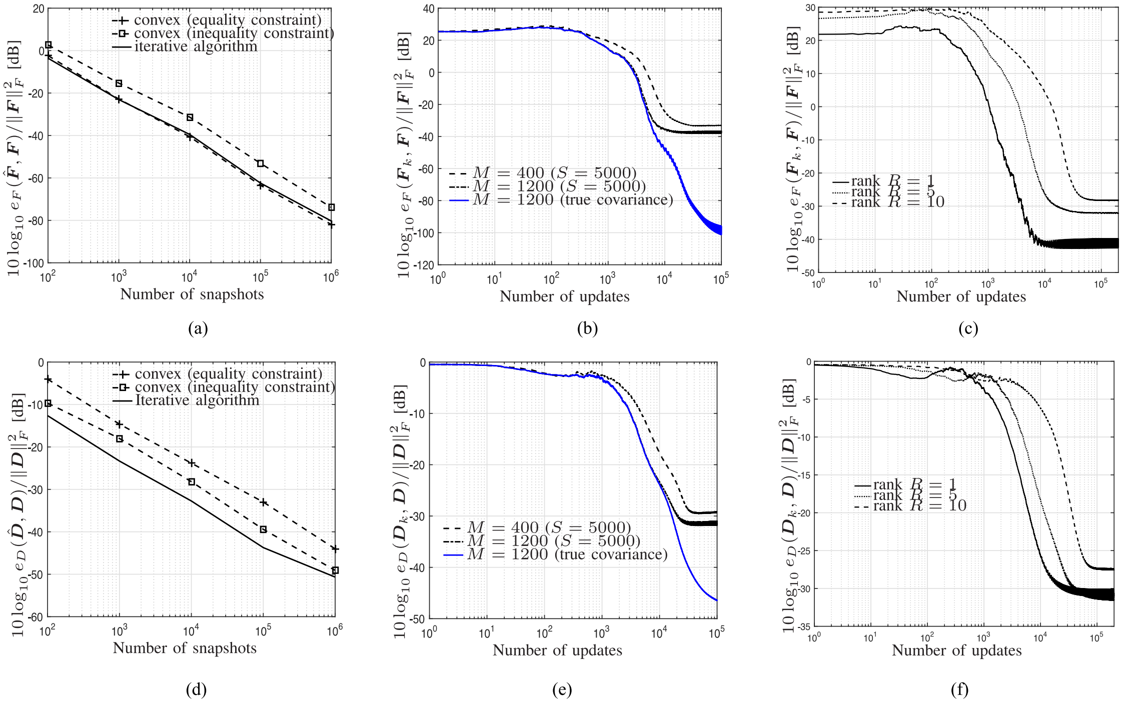

Fig. 1. Plots in the first row and second row are related to the performance of the estimators $\hat{\boldsymbol{F}}$ and $\hat{\boldsymbol{D}}$, respectively. (a) and (d) Performance of the convex program (5), the convex program with inequality constraint, and the iterative algorithm as a function of the number of snapshots. (b) and (e) Convergence of the iterative algorithm for different number of measurements, with and without finite sample effects, as a function of the number of iterations. (c) and (f) Convergence of the iterative algorithm for different values of the factor rank as a function of the number of iterations.

Algorithm 1: Stochastic gradient descent for factor analysis.

```
Require $\{b_i, 1 \le i \le M\}$ and $K$.
Initialize $\boldsymbol{F}_0 \in \mathbb{C}^{N\times R}$ and $\boldsymbol{D}_0 \in \mathbb{R}^{N\times N}$ (diagonal).
  1. for $k = 1$ to $K$
  2.   Set $r(k) = (k-1 \mod M) + 1$.
  3.   Update $\boldsymbol{F}_{k+1}$ and $\boldsymbol{D}_{k+1}$ using (7) and (8), respectively.
  4. end
Return $\hat{\boldsymbol{F}} = \boldsymbol{F}_K$ and $\hat{\boldsymbol{D}} = \boldsymbol{D}_K$.
```

i.e., $r(k) = (k-1 \mod M) + 1$. The solution of (6), denoted by $(\hat{\boldsymbol{X}}, \hat{\boldsymbol{Y}})$, will be the current estimate $(\boldsymbol{F}_{k+1}, \boldsymbol{Q}_{k+1})$.

The optimization problem (6) can be solved in *closed form* using Lagrange multipliers. The derivation is given in Appendix A. The update equations are given by

$$\boldsymbol{F}_{k+1} = \left[\boldsymbol{I} - \alpha_{r(k)}\boldsymbol{P}_{r(k)}\right]\boldsymbol{F}_k \tag{7}$$

$$\begin{aligned}\boldsymbol{Q}_{k+1} &= \text{diag}\left(\left[\boldsymbol{I} - \alpha_{r(k)}\boldsymbol{P}_{r(k)}\right]\boldsymbol{Q}_k\right)\\ \boldsymbol{D}_{k+1} &= \boldsymbol{Q}_{k+1}^2.\end{aligned} \tag{8}$$

Here, diag$(\cdot)$ operator forms a diagonal matrix from the diagonal entries of its argument. The *rank-one projector*, $\boldsymbol{P}_{r(k)}$, onto the measurement space spanned by $\boldsymbol{a}_{r(k)}$, given in (15), may be cached if the measurements are reused. The closed-form expression of the *gain* $\alpha_{r(k)}$ is given in (16).

In summary (see Algorithm 1), the iterative algorithm consists of choosing a random initial point $(\boldsymbol{F}_0, \boldsymbol{D}_0)$, then performing the rank-one update for estimating the low-dimensional subspace and the diagonal matrix successively until convergence. Convergence may be determined by observing when the estimates stop changing. Each update step approximately costs order N^2R flops. There is no tuning parameter involved in this method and it is also well suited for processing streaming data as the kth update step requires only the observation $b_{r(k)}$ to be available.

IV. Numerical Experiments

In this section*, we demonstrate the performance of the developed convex program and the iterative algorithm. For the simulations, we use $N = 50$ and the sketching vectors $\{\boldsymbol{a}_i, 1 \le i \le M\}$ are chosen randomly, and independently, from a zero-mean unit-variance Gaussian distribution. We generate $\boldsymbol{F}$ with independent and identically distributed real Gaussian entries having zero mean and unit variance. The diagonal entries of $\boldsymbol{D}$ are also generated randomly, and independently, from an uniform distribution on $(0, 10)$. The performance of the developed algorithms is shown in terms of the subspace estimation error and the squared error of the estimator $\hat{\boldsymbol{D}}$. The subspace estimation error is defined as

$$\begin{aligned}e_F(\hat{\boldsymbol{F}}, \boldsymbol{F}) &= \underset{\boldsymbol{V}^H\boldsymbol{V}=\boldsymbol{I}}{\text{minimize}}\ \|\hat{\boldsymbol{F}} - \boldsymbol{F}\boldsymbol{V}\|_F^2\\ &= \sum_{i=1}^{R}[\sigma_i^2(\hat{\boldsymbol{F}}) - 2\sigma_i(\boldsymbol{F}^H\hat{\boldsymbol{F}}) + \sigma_i^2(\boldsymbol{F})]\end{aligned}$$

where $\sigma_i(\boldsymbol{A})$ denotes the ith singular value of $\boldsymbol{A}$ with the singular values being sorted in the descending order. The squared

*Software to reproduce results of this letter can be downloaded from http://cas.et.tudelft.nl/∼sundeep/sw/FAquad17.zip error of the estimator $\hat{\boldsymbol{D}}$ is defined as $e_D(\hat{\boldsymbol{D}}, \boldsymbol{D}) = \|\hat{\boldsymbol{D}} - \boldsymbol{D}\|_F^2$. The convex optimization problem (5) is solved using CVX [25]. The initial point $(\boldsymbol{F}_0, \boldsymbol{D}_0)$ for the iterative algorithm is chosen randomly.

To begin with, in Fig. 1(a) and (d), we show the performance of the convex program and the iterative algorithm for $R = 2$ and for different number of snapshots. Here, the sample covariance matrix $\hat{\boldsymbol{\Sigma}}$ formed from $S = \{10^2, 10^3, 10^4, 10^5, 10^6\}$ snapshots is sketched with $M = 1000$ sketching vectors. Thus, the measurements are noisy. For the iterative algorithm, the error is shown for $K = 50000$ iterations. This means that the available measurements are reused 50 times. When the convex program (5) is solved with an equality constraint (by ignoring noise), we can see that the subspace estimates are very similar to that of the iterative algorithm, however, the performance of the estimator $\hat{\boldsymbol{D}}$ from the convex program is worse than that of the iterative algorithm. This is because the convex program fits noise to $\hat{\boldsymbol{D}}$ to obtain a low-rank matrix $\hat{\boldsymbol{L}}$. When the convex program is solved with the inequality constraint $\|\mathcal{A}(\boldsymbol{L} + \boldsymbol{D}) - \boldsymbol{b}\| \leq \varepsilon$, the error in $\hat{\boldsymbol{D}}$ is traded with the rank of $\hat{\boldsymbol{L}}$. This means that the rank of $\hat{\boldsymbol{L}}$ will be sensitive to the choice of ε. Although for the sake of illustration, we use $\varepsilon = \|\mathcal{A}(\boldsymbol{\Sigma}) - \mathcal{A}(\hat{\boldsymbol{\Sigma}})\|_2$, in practice ε should be tuned, e.g., through cross validation. The iterative algorithm, on the other hand, is free from such a tuning parameter and is also less sensitive to finite sample effects.

In Fig. 1(b) and (e), we fix the factor rank $R = 2$ and show the convergence of the iterative algorithm for $M = \{400, 1200\}$ as a function of the number of updates. It can be seen that the estimates improve slowly after $N(R+1)$ updates (for which the number of measurements is more than the number of unknowns) and the error drops quickly when the available measurements are recycled. With $S = 5000$, due to the finite sample noise, the error saturates, whereas in the noiseless scenario with the true covariance matrix, the iterations converge to the true solution.

In Fig. 1(c) and (f), we fix the number of measurements $M = 1000$ and the number of snapshots $S = 5000$, and show the convergence for $R = \{1, 5, 10\}$ as a function of the number of updates. The number of unknowns increase as R increases and as a consequence, for a fixed M, the convergence rate depends on R with the convergence being slower for larger values of R. As before, the error saturates because of the (finite sample) noise.

V. Concluding Remarks

This letter presents a convex program as well as an iterative algorithm for computing the factor analysis (i.e., low-rank plus diagonal) decomposition from a set of quadratic rank-one measurements of a covariance matrix. The developed semidefinite convex program can be solved efficiently when all the measurements are available. To process streaming or large-scale data, Kaczmarz-like stochastic gradient descent algorithm has been developed, wherein each update step involves a simple rank-one projection to simultaneously update the low-rank subspace and the diagonal matrix. The iterative algorithm performs better than the convex program when only a finite number of data snapshots are available.

Appendix A

In this appendix, the closed-form solution to (6) using Lagrange multipliers is derived. The Lagrangian of (6) is given by

$$\mathcal{L}(\boldsymbol{X}, \boldsymbol{Y}, \lambda) = \|\boldsymbol{F}_k - \boldsymbol{X}\|_F^2 + \|\boldsymbol{Q}_k - \boldsymbol{Y}\|_2^2 + \lambda\left(\boldsymbol{a}_{r(k)}^H[\boldsymbol{X}\boldsymbol{X}^H + \boldsymbol{Y}^2]\boldsymbol{a}_{r(k)} - b_{r(k)}\right)$$

where λ is a scalar Lagrange multiplier corresponding to the equality constraint. Setting the derivate of $\mathcal{L}(\boldsymbol{X}, \boldsymbol{Y}, \lambda)$ towards $\boldsymbol{X}$, $\boldsymbol{Y}$, and λ to zero, we obtain the equations

$$\boldsymbol{X}(\lambda) = (\boldsymbol{I} + \lambda \boldsymbol{a}_{r(k)}\boldsymbol{a}_{r(k)}^H)^{-1}\boldsymbol{F}_k; \tag{9}$$

$$\boldsymbol{Y}(\lambda) = (\boldsymbol{I} + \lambda \boldsymbol{a}_{r(k)}\boldsymbol{a}_{r(k)}^H)^{-1}\boldsymbol{Q}_k; \tag{10}$$

$$\boldsymbol{a}_{r(k)}^H[\boldsymbol{X}\boldsymbol{X}^H + \boldsymbol{Y}^2]\boldsymbol{a}_{r(k)} = b_{r(k)}. \tag{11}$$

The optimal λ may be computed by substituting (9) and (10) in (11), and this results in

$$\boldsymbol{a}_{r(k)}^H(\boldsymbol{I} + \lambda \boldsymbol{a}_{r(k)}\boldsymbol{a}_{r(k)}^H)^{-1}\left[\boldsymbol{F}_k\boldsymbol{F}_k^H + \boldsymbol{Q}_k^2\right] \times (\boldsymbol{I} + \lambda \boldsymbol{a}_{r(k)}\boldsymbol{a}_{r(k)}^H)^{-1}\boldsymbol{a}_{r(k)} = b_{r(k)}. \tag{12}$$

We will now use the matrix-inversion lemma [26] to obtain the following identities:

$$(\boldsymbol{I} + \lambda \boldsymbol{a}_{r(k)}\boldsymbol{a}_{r(k)}^H)^{-1}\boldsymbol{a}_{r(k)} = \frac{1}{1 + \lambda\|\boldsymbol{a}_{r(k)}\|_2^2}\boldsymbol{a}_{r(k)}; \tag{13}$$

$$(\boldsymbol{I} + \lambda \boldsymbol{a}_{r(k)}\boldsymbol{a}_{r(k)}^H)^{-1} = \mathbf{I} - \frac{\lambda \boldsymbol{a}_{r(k)}\boldsymbol{a}_{r(k)}^H}{1 + \lambda\|\boldsymbol{a}_{r(k)}\|_2^2}. \tag{14}$$

Using (13) in (12), we can compute the solution for λ explicitly as

$$\lambda = \frac{1}{\|\boldsymbol{a}_{r(k)}\|_2^2}\left[\pm\left(\frac{\boldsymbol{a}_{r(k)}^H(\boldsymbol{F}_k\boldsymbol{F}_k^H + \boldsymbol{Q}_k^2)\boldsymbol{a}_{r(k)}}{b_{r(k)}}\right)^{1/2} - 1\right].$$

The solution corresponding to the larger λ that minimizes the cost function is chosen. Using (14), and substituting the optimal λ in (9) and (10), we obtain the solution to (6) as

$$\hat{\boldsymbol{X}} = \left[\boldsymbol{I} - \alpha_{r(k)}\boldsymbol{P}_{r(k)}\right]\boldsymbol{F}_k;$$

$$\hat{\boldsymbol{Y}} = \operatorname{diag}\left(\left[\boldsymbol{I} - \alpha_{r(k)}\boldsymbol{P}_{r(k)}\right]\boldsymbol{Q}_k\right),$$

where the rank-one projector $\boldsymbol{P}_{r(k)}$ is given by

$$\boldsymbol{P}_{r(k)} = \frac{\boldsymbol{a}_{r(k)}\boldsymbol{a}_{r(k)}^H}{\|\boldsymbol{a}_{r(k)}\|_2^2} \tag{15}$$

and the *gain* $\alpha_{r(k)}$ is given by

$$\alpha_{r(k)} = 1 - \left(\frac{b_{r(k)}}{\|\boldsymbol{F}_k^H\boldsymbol{a}_{r(k)}\|_2^2 + \|\boldsymbol{Q}_k\boldsymbol{a}_{r(k)}\|_2^2}\right)^{1/2}. \tag{16}$$

It is easy to see that $\boldsymbol{P}_{r(k)}$ is a projection onto the measurement space spanned by the $r(k)$th sketching vector $\boldsymbol{a}_{r(k)}$, and the gain $\alpha_{r(k)}$ provides a correction within that space.

REFERENCES

[1] Y. Chen, Y. Chi, and A. J. Goldsmith, "Exact and stable covariance estimation from quadratic sampling via convex programming," *IEEE Trans. Inf. Theory*, vol. 61, no. 7, pp. 4034–4059, Jul. 2015.

[2] Y. Chi and Y. M. Lu, "Kaczmarz method for solving quadratic equations," *IEEE Signal Process. Lett.*, vol. 23, no. 9, pp. 1183–1187, Sep. 2016.

[3] C. D. White, S. Sanghavi, and R. Ward, "The local convexity of solving systems of quadratic equations," *arXiv:1506.07868*, 2015.

[4] E. J. Candes, T. Strohmer, and V. Voroninski, "Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming," *Commun. Pure Appl. Math.*, vol. 66, no. 8, pp. 1241–1274, 2013.

[5] Y. Chen and E. Candes, "Solving random quadratic systems of equations is nearly as easy as solving linear systems," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 739–747.

[6] C. Spearman, "General intelligence, objectively determined and measured," *Amer. J. Psychol.*, vol. 15, pp. 201–293, 1904.

[7] D. Lawley and A. Maxwell, "Factor analysis as a statistical method," *J. Roy. Statist. Soc. Ser. D,* vol. 12, no. 3, pp. 209–229, 1962.

[8] J. Fan, Y. Fan, and J. Lv, "High dimensional covariance matrix estimation using a factor model," *J. Econometrics*, vol. 147, no. 1, pp. 186–197, 2008.

[9] A.-J. van der Veen and S. J. Wijnholds, "Signal processing tools for radio astronomy," in Proc. *Handbook Signal Process. Syst*. Springer, 2013, pp. 421–463.

[10] D. Bertsimas, M. S. Copenhaver, and R. Mazumder, "Certifiably optimal low rank factor analysis," *J. Mach. Learn. Res.*, vol. 18, no. 29, pp. 1–53, 2017.

[11] A. Sardarabadi, "Covariance matching techniques for radio astronomy calibration and imaging," Ph.D. dissertation, TU Delft, Fac. EEMCS, Delft, The Netherlands, Jun. 2016, iSBN 978-94-6186-670-7.

[12] P. E. Dewdney, P. J. Hall, R. T. Schilizzi, and T. J. L. Lazio, "The square kilometre array," *Proc. IEEE*, vol. 97, no. 8, pp. 1482–1496, Aug. 2009.

[13] A. Anandkumar, D. Hsu, A. Javanmard, and S. Kakade, "Learning linear Bayesian networks with latent variables," in *Proc. 30th Int. Conf. Mach. Learn.*, 2013, pp. 249–257.

[14] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, "Learning Laplacian matrix in smooth graph signal representations," *IEEE Trans. Signal Process.*, vol. 64, no. 23, pp. 6160–6173, Dec. 2016.

[15] X.-C. Bai, G. McMullan, and S. H. Scheres, "How cryo-em is revolutionizing structural biology," *Trends Biochem. Sci.*, vol. 40, no. 1, pp. 49–57, 2015.

[16] T. Bhamre, T. Zhang, and A. Singer, "Denoising and covariance estimation of single particle cryo-em images," *J. Struct. Biol.*, vol. 195, no. 1, pp. 72–81, 2016.

[17] A.-K. Seghouane, "An iterative projections algorithm for ML factor analysis," in *Proc. IEEE Workshop Mach. Learn. Signal Process.*, 2008, pp. 333–338.

[18] C. M. Bishop, *Pattern Recognition and Machine Learning*. Berlin, Germany: Springer, 2006.

[19] A. Shapiro, "Rank-reducibility of a symmetric matrix and sampling theory of minimum trace factor analysis," *Psychometrika*, vol. 47, no. 2, pp. 187–199, 1982.

[20] J. Saunderson, V. Chandrasekaran, P. A. Parrilo, and A. S. Willsky, "Diagonal and low-rank matrix decompositions, correlation matrices, and ellipsoid fitting," *SIAM J. Matrix Anal. Appl.*, vol. 33, no. 4, pp. 1395–1416, 2012.

[21] M. Mesbahi and G. P. Papavassilopoulos, "On the rank minimization problem over a positive semidefinite linear matrix inequality," *IEEE Trans. Autom. Control*, vol. 42, no. 2, pp. 239–243, Feb. 1997.

[22] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM Rev.*, vol. 52, no. 3, pp. 471–501, 2010.

[23] S. Kaczmarz, "Angenähert auflösung von systemen linearer gleichungen," *Int. Bulletin Polish Acad. Sci. Lett.*, vol. 35, pp. 355–357, 1937.

[24] T. Strohmer and R. Vershynin, "A randomized Kaczmarz algorithm with exponential convergence," *J. Fourier Anal. Appl.*, vol. 15, no. 2, pp. 262–278, 2009.

[25] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.0 beta," http://cvxr.com/cvx, Sep. 2012.

[26] W. W. Hager, "Updating the inverse of a matrix," *SIAM Rev.*, vol. 31, no. 2, pp. 221–239, 1989.