Quadratic Programming Over Ellipsoids

with Applications to Constrained Linear Regression and Tensor Decomposition

Anh-Huy Phan · Masao Yamagishi · Danilo Mandic · Andrzej Cichocki

Abstract A novel algorithm to solve the quadratic programming problem over ellipsoids is proposed. This is achieved by splitting the problem into two optimisation subproblems, quadratic programming over a sphere and orthogonal projection. Next, an augmented-Lagrangian algorithm is developed for this multiple constraint optimisation. Benefit from the fact that the QP over a single sphere can be solved in a closed form by solving a secular equation [14] and [16], we derive a tighter bound of the minimiser of the secular equation. We also propose to generate a new psd matrix with a low condition number from the matrices in the quadratic constraints. This correction method improves convergence of the proposed augmented-Lagrangian algorithm. Finally, applications of the quadratically constrained QP to bounded linear regression and tensor decompositions are presented.

1 Introduction

Quadratic programming over a single sphere is one of basic optimisation problems and has been extensively studied. For example, the problem was first considered as an eigenvalue problem with linear constraints $\mathbf{N}^T \boldsymbol{x} = \boldsymbol{t}$ by Gander, Golub and Matt [14]. After eliminating the linear constraint, the constrained eigenvalue problem becomes a QP problem

A.-H. Phan and A. Cichocki
Lab for Advanced Brain Signal Processing, Brain Science Institute, RIKEN, Wakoshi, Japan
E-mail: (phan,cia)@brain.riken.jp
A. Cichocki
Skolkovo Institute of Science and Technology (Skoltech), Russia
M.Yamagishi
Tokyo Institute of Technology, Japan
E-mail: myamagi@sp.ce.titech.ac.jp D. Mandic
Imperial College, London, United Kingdom
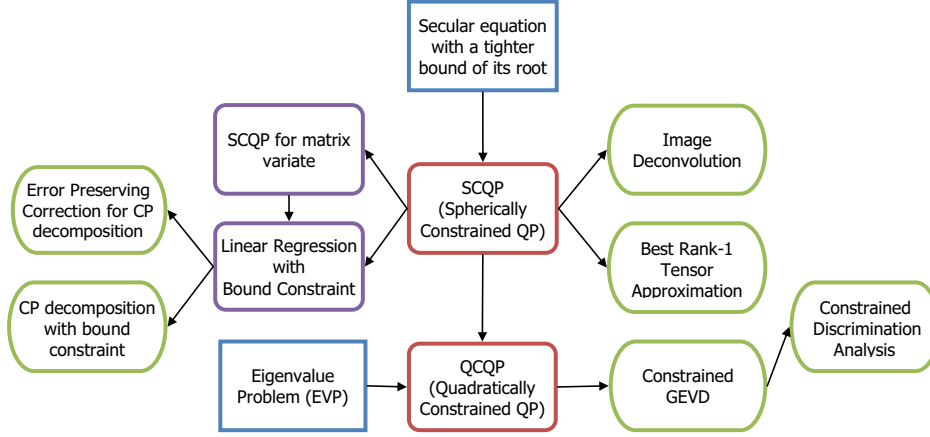E-mail: d.mandic@imperial.ac.uk

over a single sphere. The authors solved a secular equation using an iterative algorithm which starts from an initial point determined based on the eigenvalue of the quadratic term.

A similar study was presented by Hager as a minimization of a quadratic function over a sphere [16]. Hager also considered solving a rational function. Rojas, Santos and Sorensen [30] developed a trust-region algorithm which can be applied to this problem. Some extensions to solving large-scale problems were proposed in [29, 32].

The spherically constrained QP (SCQP) problem was reinvented many times. In [11], Chen and Gao presented a globally optimal solution to the QP with a variable vector constrained inside a ball. They formulated the problem as one-dimensional canonical duality problem and proposed associated numerical algorithms.

For this particular problem, by considering the variable vector in the Stiefel manifold, we can also apply optimization algorithms on a manifold to solve this problem, e.g., using the ManOpt toolbox [7] or the Cran-Nicholson update scheme [33].

A more sophisticated problem is that of minimizing a convex quadratic function over an intersection of ellipsoids $\boldsymbol{x}^T \mathbf{H}_m \boldsymbol{x} = 1$, bearing in mind that quadratic equalities characterize non-convex sets. The non-convex quadratic optimisation problem with quadratic equality constraints is known to be NP-hard [3, 9, 21]. Nevertheless, since gradients and Hessians of the constrained functions can be derived in an analytical form, the problem can be solved using interior-point algorithms for nonlinearly constrained minimization [17]. Alternatively, one can cast a convex quadratic and quadratically constrained optimization problem into a conic optimization problem which can be solved efficiently, e.g, using the Mosek optimisation toolbox [2]. The QP problem with quadratic inequality constraints can also be solved efficiently using the Modified proportioning with gradient projections [12, 13]. Some other common approaches are to

Fig. 1 Flow chart of the Spherically Constrained QP and Quadratically Constrained QP and their novel applications presented in this paper.

convexify the problem using semidefinite relaxation techniques [15], second-order cone programming [18], or mixed SOCP-SDP relaxations [9].

For some particular cases, e.g., a quadratic function with two quadratic constraints, [6] shows that under a suitable assumption, the problem can be solved in polynomial time. Similarly, with some simple convex relaxations the solution can even return the optimal values [22].

In this paper, we develop algorithms for QP with quadratic constraints $x^T \mathbf{H}_m x = 1$, and present novel applications of this optimisation. First, we consider the simple QP over a sphere. In the same spirit as Gander, Golub and Matt [14], Hager [16], we solve the problem by finding a root of a secular equation. Normalisation and conversion methods are introduced to simplify the problem to the one with a smaller number of parameters, when the vector in the linear term comprises zero-entries, or when the matrix in the quadratic term has identical eigenvalues. The conversion is particularly useful for the SCQP for matrix variate in Section 3. We show that the solution to such a constrained QP problem can be deduced from a minimiser of a much smaller similar QP for a vector variate. For the ordinary SCQP, we present new results for finding good bounds of the minimiser. To this end, we perform a slightly different normalisation to that in [14] and [16]. With this new bound, we can even find a good estimate to the global minimiser through solving a truncated problem with a few terms. It is shown that the solution can be found in closed form for some particular cases without resorting iterative algorithms.

In Section 5, we present the linear regression with a bound constraint and formulate it as an equivalent SCQP. This problem has applications in deriving the norm correction method for the CANDECOMP/PARAFAC tensor decomposition (CPD), and for developing the algorithm for the bounded CPD [26].

In Section 6, we will present an algorithm to solve the Quadratic programming over elliptic constraints. The problem with multiple constraints is split into two optimisation sub-problems, one is the quadratic programming over a sphere, and the other being the orthogonal projection. An augmented-Lagrangian algorithm is next developed for this problem. We suggest generating a new psd matrix with a low condition number from the matrices in the quadratic constraints. This correction method is proved to improve convergence of the proposed augmented-Lagrangian algorithm.

We present novel applications of the quadratic programming over a sphere to tensor decompositions, including finding a best rank-1 tensor approximation to symmetric tensors of order-4 [27], and constrained discrimination analysis.

In Section 8, we introduce constrained generalized eigenvalue decomposition, in which eigenvectors impose low-rank structures. The problem is then converted to subproblems related to the ordinary GEVD and the QP over multiple quadratic constraints. Throughout the paper, we provide many examples, including image deconvolution, best rank-1 tensor approximation and image classification, to verify and illustrate our algorithms. In addition, a flow chart in Fig. 1 summarises the studied methods and their applications.

2 Quadratic Programming Over A Single Sphere

Consider a quadratic programming problem with a constraint that the variable vector is on a sphere, i.e., unit-length vector.

Definition 1 (Quadratic Programming over a Single Sphere) Given a positive semi-definite matrix $\mathbf{Q}$ of size $K \times K$ and a vector b of length K, the quadratic programming over a sphere solves the optimisation problem

$$\min \quad \frac{1}{2} x^T \mathbf{Q} x + b^T x, \quad \text{s.t.} \quad x^T x = 1. \quad (1)$$

For the case when b is a zero vector, the problem (1) becomes that of finding the smallest eigenvectors of the matrix

Q. Here, we do not consider this case. In addition, the matrix **Q** only needs to be symmetric so that the positive semidefinite condition on matrix **Q** can be relaxed. We first show that the QP in (1) can be converted to a problem whereby the matrix **Q** is diagonal, and has positive eigenvalues. Then, we simplify the optimization task to that with distinct eigenvalues and non-zero entries b.

2.1 Normalisation, reparameterization and simplication

We shall denote the eigenvalue decomposition of the matrix **Q** in (1) by

$$\mathbf{Q} = \mathbf{U} \, \mathrm{diag}(\boldsymbol{\sigma}) \mathbf{U}^T$$

where $\boldsymbol{\sigma} = [0 \leq \sigma_1 \leq \sigma_2 \leq \cdots \leq \sigma_K]$ comprises the eigenvalues of **Q**, **U** is an orthonormal matrix of size $K \times K$, which consists of eigen-vectors of **Q**.

Since the vector b is non-zero, we can perform the following normalisation and reparameterization

$$\tilde{x} = \mathbf{U}^T x, \quad c = \mathbf{U}^T \frac{b}{\|b\|},$$

$$s = [s_1, \ldots, s_K]^T, \quad s_k = \frac{\sigma_k - \sigma_1}{\|b\|} + 1$$

so that $\tilde{x}$ and c are unit-length vectors, $\tilde{x}^T \tilde{x} = 1$ and $c^T c = 1$, and $s_1 = 1 \leq s_2 \leq \cdots \leq s_K$. Hence, the optimal solution x to the QP problem in (1) can be derived from the following QP

$$\min \quad \frac{1}{2} \tilde{x}^T \, \mathrm{diag}(s) \, \tilde{x} + c^T \tilde{x}, \quad \text{s.t.} \quad \tilde{x}^T \tilde{x} = 1, \tag{2}$$

where $c^T c = 1$ and $s = [s_1 = 1 \leq s_2 \leq \cdots \leq s_K]$.

We next show that the problem (2) can be simplified to the case with distinct eigenvalues, i.e., $s_1 < s_2 < \cdots < s_K$.

We shall denote by J the number of distinct eigenvalues, $\tilde{s} = [\tilde{s}_1 = 1 < \tilde{s}_2 < \cdots < \tilde{s}_J]$, over a set of K eigenvalues, s_k, in (2), and classify $c = [c_1, c_2, \ldots, c_J]$ into J sub-vectors, whereby each c_j consists of entries c_k such that $s_k = \tilde{s}_j$, i.e., $c_j = [c_{k \in \mathcal{I}_j}]$, where $\mathcal{I}_j = \{k : s_k = \tilde{s}_j\}$. In addition, we shall define a vector

$$\tilde{c} = [\|c_1\|, \|c_2\|, \ldots, \|c_J\|].$$

Then, the following relation holds, the proof of which is provided in Appendix A.

Lemma 1 *The minimiser to (2) can be deduced from the minimiser to an SCQP with distinct eigenvalues, that is*

$$\min \quad \frac{1}{2} z^T \, \mathrm{diag}(\tilde{s}) \, z + \tilde{c}^T z \quad \text{subject to} \quad z^T z = 1,$$

as $x_{\mathcal{I}_j} = \dfrac{z_j}{\|c_j\|} c_j$, for a non zero vector c_j, or an arbitrary vector on the ball $\|x_{\mathcal{I}_j}\|^2 = z_j^2$ for a zero c_j, $j = 1, \ldots, J$.

In the sequel, Sections 2.3-2.5 show that for most cases with zero entries $\tilde{c}_j = 0$, e.g., when $j > 1$, the optimal $z_j^\star$ is zero. Hence, $\tilde{x}_{\mathcal{I}_j}$ is also a zero vector.

Next, we consider the case when the entries of the vector c are nonzero. The case with zero entries c_k can be deduced from the former case.

2.2 The case when all c_k are non-zeros

The Lagrangian function of the problem in (2) is given by

$$\mathcal{L}(\tilde{x}, \lambda) = \frac{1}{2} \tilde{x}^T \, \mathrm{diag}(s) \, \tilde{x} + c^T \tilde{x} - \frac{1}{2} \lambda(\tilde{x}^T \tilde{x} - 1).$$

Following the first-order optimality condition, there exists a Lagrange multiplier λ such that

$$\frac{\partial \mathcal{L}(\tilde{x}, \lambda)}{\partial \tilde{x}} = (\mathrm{diag}(s) - \lambda \mathbf{I})\tilde{x} + c = \mathbf{0}. \tag{3}$$

Since c_k are non-zeros, the multiplier λ must not be any s_k, i.e., $\lambda \neq s_k$ for $k = 1, \ldots, K$, thus implying that the minimiser $\tilde{x}^\star$ can be expressed as

$$\tilde{x}^\star = \left[\frac{c_1}{\lambda - s_1}, \ldots, \frac{c_K}{\lambda - s_K} \right]$$

and the Lagrangian function at $\tilde{x}^\star$ is given by

$$\mathcal{L}(\tilde{x}^\star, \lambda) = \frac{1}{2} \sum_{k=1}^{K} \frac{c_k^2 s_k}{(\lambda - s_k)^2} + \sum_{k=1}^{K} \frac{c_k^2}{\lambda - s_k} - \frac{\lambda}{2} \sum_{k=1}^{K} \frac{c_k^2}{(\lambda - s_k)^2} + \frac{\lambda}{2}$$

$$= \frac{\lambda}{2} + \frac{1}{2} \sum_{k=1}^{K} \frac{c_k^2}{\lambda - s_k}.$$

This leads to finding a root λ of the first derivative $f'(\lambda)$, as

$$f'(\lambda) = 1 - \sum_{k=1}^{K} \frac{c_k^2}{(\lambda - s_k)^2}, \tag{4}$$

which minimises the following function

$$\min_{\lambda} \quad f(\lambda) = \lambda + \sum_{k=1}^{K} \frac{c_k^2}{\lambda - s_k}. \tag{5}$$

The *secular equation* in (4) is in a similar form to those derived in [14] and [16], but here, the coefficients c_k are with an additional constraint $c^T c = 1$ and $s_1 = 1$. This constraint will later help to derive a tighter bound for the roots of $f'(\lambda) = 0$.

We will next show that the minimiser $\lambda^\star$ is a minimum root of $f'(\lambda)$. To this end, we first illustrate that $f'(\lambda)$ has a root less than $s_1 = 1$, and prove that this root is the global minimiser to $f(\lambda)$.

Lemma 2 *The first derivative of $f(\lambda)$ in (5) has only one root $\lambda < s_1 = 1$, which lies in the interval $(0, 1 - |c_1|)$.*

Lemma 3 *The solution to the problem in (5) is the minimum root of the first derivative of $f(\lambda)$.*
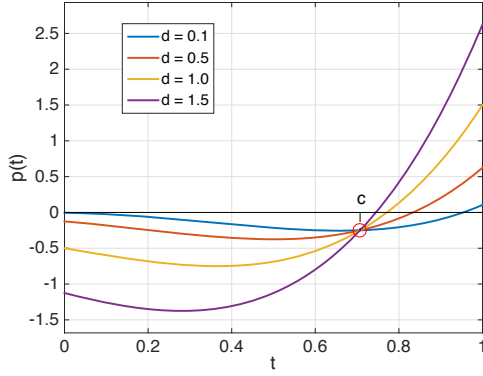
Fig. 2 Illustration of the polynomial $p(t) = t^4 + 2dt^3 + (d^2 - 1)t^2 - 2c^2dt - c^2d^2$ with $c^2 = 0.5$ and various d, and its unique roots in the interval $[c, 1]$.

Proofs of Lemmas 2 and 3 are given in Appendices B-C. We proceed to show that $\lambda^\star$ can be found with a tighter bound.

Lemma 4 *The function $f(\lambda)$ in (5) has a unique global minimizer in the interval $(1 - t_1, 1 - t_2)$, where t_1 and t_2 are the roots which lie in the interval of $[|c_1|, 1]$ of two degree-4 polynomials $p_1(t)$ and $p_2(t)$, given by*

$$p_i(t) = t^4 + 2d_i t^3 + (d_i^2 - 1) t^2 - 2c_1^2 d_i t - c_1^2 d_i^2$$

where $d_1 = s_2 - s_1$ and $d_2 = s_K - s_1$.

We provide proof of Lemma 4 in Appendix D, and illustrate the polynomials $p_i(t)$ in $[0, 1]$ for various d_i in Fig. 2. The roots t_i approach $|c_1|$ when d_i are large, and 1 when d_i are small.

If $s_K = s_2$, i.e., $(K - 1)$ eigenvalues $s_2, s_3, ..., s_K$ are identical, then $t_1 = t_2 = t^\star$, and $\lambda = 1 - t^\star$ is a root of $f'(\lambda) = 0$. When s_2 and s_K are relatively close, the bound $[1 - t_1, 1 - t_2]$ is tight and provides a good approximation to the root λ as illustrated in Fig. 2.

When $d_1 \geq 1$, it follows that $d_2 \geq 1$, and the bound width $(t_1 - t_2)$ is relatively small. For example, when $d_2 = 2d_1$ and $d_1 > 1$, the bound width $(t_1 - t_2)$ is often less than 0.1, while the width is even less than 0.01 when d_1 exceeds 3, and is less than 0.001 when $d_1 \geq 10$, despite of values of s_K, as seen in Fig 3 for the cases $d_2 = 2d_1$ and $d_2 = 1000d_1$.

In general, the bound width $(t_1 - t_2)$ is tight when $d_1 \approx d_2$, i.e., the eigenvalues $s_2, \ldots, s_K$ are located in a narrow range, or when d_1 exceeds 1, i.e., $s_2 > 2$. However, the bound width is not sufficiently good when $d_1 < 1 < d_2$, especially when d_1 is small, t_1 approaches 1, and t_2 approaches $|c_1|$. Hence, there is no much improvement on the bound for λ, compared to the obvious bound $[0, 1 - |c_1|]$.

In order to improve the bound of the minimiser $\lambda^\star$, when $s_2 - s_1 < 1$, we propose to solve a similar equation to (4) but with a smaller number of terms. We shall refer it to as
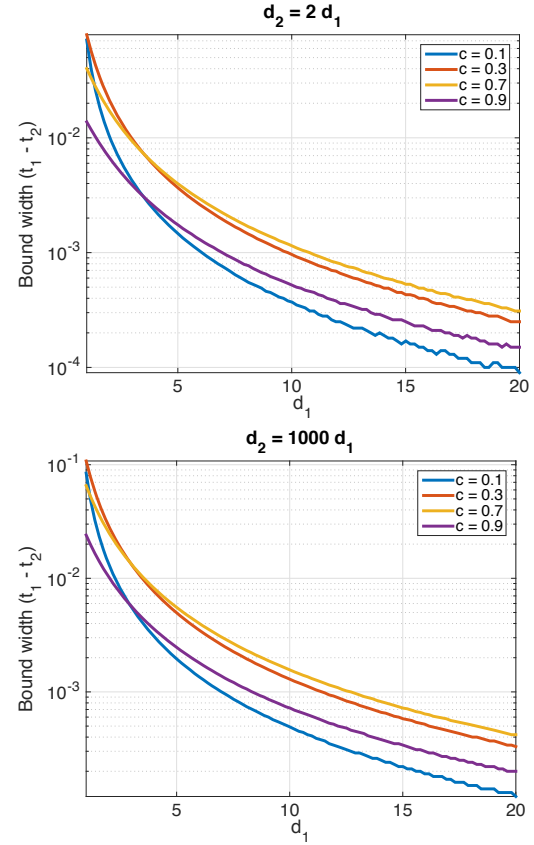

Fig. 3 Illustration of the bound width $(t_1 - t_2)$ when $d_2 = 2d_1$ and $d_2 = 1000d_1$ for various values of $c_1 = c$.

the *truncated problem*. Let $\tilde{c}_L = \sqrt{\sum_{k=L}^{K} c_k^2}$. We define a set of equations $f_l^{(L)}(\lambda)$ and $f_u^{(L)}(\lambda)$ constructed from the first L terms of the equation $f'(\lambda)$ in (4)

$$f_l^{(L)}(\lambda) = 1 - \sum_{l=1}^{L} \frac{c_l^2}{(s_l - \lambda)^2} - \frac{\tilde{c}_{L+1}^2}{(s_{L+1} - \lambda)^2},$$

$$f_u^{(L)}(\lambda) = 1 - \sum_{l=1}^{L} \frac{c_l^2}{(s_l - \lambda)^2} - \frac{\tilde{c}_{L+1}^2}{(s_K - \lambda)^2}.$$

Lemma 5 *The roots $\lambda_{l,L}^\star$ of $f_l^{(L)}(\lambda)$ and the roots $\lambda_{u,L}^\star$ of $f_u^{(L)}(\lambda)$ in $[0, 1 - |c_1|]$ are unique, and form the lower and upper bounds of the root $\lambda^\star$ of $f'(\lambda)$ in (4)*

$$\lambda_{l,1}^\star \leq \lambda_{l,2}^\star \leq \cdots \leq \lambda_{l,K-2}^\star \leq \lambda^\star \leq \lambda_{u,K-2}^\star \leq \cdots \leq \lambda_{u,2}^\star \leq \lambda_{u,1}^\star \quad (6)$$

The proof is given in Appendix E. We note that the bound derived in Lemma 4 is a particular case of Lemma 5 with $\lambda_{l,1}^\star = 1 - t_1$ and $\lambda_{u,1}^\star = 1 - t_2$.

Lemma 5 states that we can obtain a tighter bound for the minimiser $\lambda^\star$ of f' by solving a truncated secular equation with only a few terms $\frac{c_l^2}{(s_l - \lambda)^2}$. The method is particularly useful when the first L eigenvalues, $s_1, s_2, ..., s_L$, are very close to each other, while s_{L+1} exceeds 1 significantly.
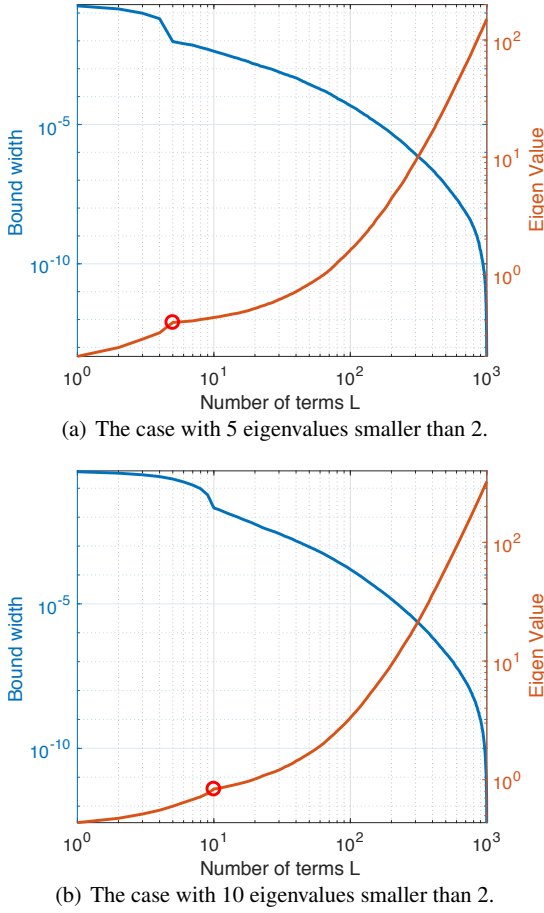
(a) The case with 5 eigenvalues smaller than 2.

(b) The case with 10 eigenvalues smaller than 2.

Fig. 4 Illustration of a bound width of λ by solving the reduced problem using L terms. The bound width can be less than 0.001 when solving the truncated equations with dozens of terms.

Example 1 In Fig. 4, we demonstrate good estimates of the minimiser $\lambda^\star$ of the equation $f'(\lambda)$ which has $K = 1000$ terms. The eigenvalues s_k are randomly generated such that some of the first T eigenvalues, s_k, are smaller than 2, where $T = 5$ or 10. The eigenvalues, s_k, are plotted in Fig. 4. The bound width $(\lambda_{u,L}^\star - \lambda_{l,L}^\star)$ is computed for various $L = 1, 2, \ldots, K - 1$. For the first case, we can obtain a bound of less than 0.01 when solving the truncated problem of only 4 or 5 terms. For the second case, a bound of less than 0.01 is achieved when solving a truncated equation with $L = 11$ terms. The bound is tighter, less than 10^{-3} when the truncated equation has 20-40 terms. Moreover, solving the truncated problems with 200 terms provides good approximation to the global minimiser λ with an error less than 10^{-5}.

2.3 The case when more than one coefficients c_n are zeros

Assume that there are more than one zero coefficients $c_l = 0$, we then denote their index set by $\mathcal{I}_0 = \{l : c_l = 0\}$, and

by n the smallest index of this set, i.e., $c_n = 0$. We shall first show that the entries $\tilde{x}_l^\star$ of the minimiser $\tilde{x}^\star$ are zeros, where $l \in \mathcal{I}_0$ and $l \neq n$, and the optimization problem can be converted to the case with only one zero coefficient $c_n = 0$.

The objective function (2) can be rewritten as

$$\min \left(\frac{1}{2} \sum_{k \notin \mathcal{I}_0} s_k \tilde{x}_k^2 + \sum_{k \notin \mathcal{I}_0} c_k \tilde{x}_k \right) + \frac{1}{2} \sum_{l \in \mathcal{I}_0} \tilde{x}_l^2 s_l, \quad \text{s.t.} \quad \tilde{x}^T \tilde{x} = 1$$

and achieves a minimum when the subset of the variables $\tilde{x}_{\mathcal{I}_0} = [\tilde{x}_l : l \in \mathcal{I}_0]$ is a minimiser of the following problem

$$\min \quad \sum_{l \in \mathcal{I}_0} \tilde{x}_l^2 s_l \quad \text{s.t.} \quad \tilde{x}_{\mathcal{I}_0}^T \tilde{x}_{\mathcal{I}_0} = r$$

where $r = 1 - \sum_{k \notin \mathcal{I}_0} x_k^2 > 0$. The problem now boils down to finding an eigenvector associated with the smallest eigenvalue, i.e., s_n, of the diagonal matrix $\text{diag}(s_{\mathcal{I}_0})$. This implies that $\tilde{x}_n^2 = r$, and the other entries $\tilde{x}_l$ are zeros, where $l \in \mathcal{I}_0$, $l \neq n$. The problem is now simplified into a problem formulated for s_k and c_k where $k \in \{1, \ldots, K\} \setminus \mathcal{I}_0 \cup \{n\}$, which has at most one zero coefficient $c_n = 0$. We will next show that x_n is also zero if $n > 1$.

2.4 The case when only one coefficient c_n is zero with $n > 1$

Lemma 6 *When there is only one $c_n = 0$ with $n > 1$, the n-th variable of the minimiser is zero, i.e., $\tilde{x}_n^\star = 0$.*

The proof of this case is given in Appendix F. In summary, as shown in this and previous sub-sections, if the coefficients c_n, with $n > 1$, are zeros, the corresponding parameters of the minimiser $\tilde{x}_n$ are zeros as well, and the remaining variables are a solution to a similar problem but with a reduced number of parameters.

2.5 The case when $c_1 = 0$

When $c_1 = 0$, we consider the two sub-cases, when $d = \sum_{k>1}^{K} \frac{c_k^2}{(s_k-1)^2}$ is less than or greater than 1.

Lemma 7 *Consider the case $c_1 = 0$, let $d = \sum_{k>1}^{K} \frac{c_k^2}{(s_k-1)^2}$.*

– *If $d \leq 1$, then the following $\tilde{x}^\star$ is a minimiser to the problem in (2)*

$$\tilde{x}_k^\star = \frac{c_k}{1 - s_k}, \quad k > 1$$

and $\tilde{x}_1^\star$ can take one of the two values $\pm \sqrt{1 - d}$.

– *Otherwise, the minimiser has $\tilde{x}_1 = 0$, and the remaining $(K-1)$ variables $[\tilde{x}_2, \ldots, \tilde{x}_K]$ are a solution to a reduced problem*

$$\min \quad \sum_{k>1} \frac{1}{2} s_k \tilde{x}_k^2 + c_k \tilde{x}_k, \quad \text{s.t.} \quad \sum_{k>1} \tilde{x}_k^2 = 1. \quad (7)$$

Proof of Lemma 7 is presented in Appendix G.

2.6 Algorithm

Steps to solve the QP over a sphere are summarised in Algorithm 1. The algorithm first normalizes the parameters b and Q, and converts the considered problem to a QP problem with a diagonal matrix $\mathrm{diag}(s)$, $s_1 = 1$ and $c^T c = 1$.

Zero coefficients c_l, where $l > 1$, are verified in order to simplify the problem to that with a fewer number of parameters of s_I and c_I, where $I = \{1\} \cup \{l : c_l \neq 0, l > 1\}$ is the index set of 1 and non-zeros c_l.

Next, identical eigenvalues, s_k, are identified and the problem is simplified again to the one with distinct eigenvalues.

For the reduced problem with $\tilde{s}$ and $\tilde{c}$, the solution can be found in closed-form in the following particular cases

- $\tilde{c}_1 = 0$
- $s_2 > s_1 = 1$
- $d = \sum_{k>1} \frac{\tilde{c}_k^2}{(\tilde{s}_k - 1)^2} < 1$.

In other cases, we find the lower and upper bounds of the minimiser $\lambda^\star$ by finding roots of two polynomials of degree-4 or by solving truncated equations with a few rational terms. The global minimiser is then found using an iterative algorithm in the estimated bounds.

Example 2 (SCQP as a tool for image deconvolution)

This example demonstrates an application of the SCQP to image deconvolution. Consider a grayscale image of size 64×64 (see Fig. 5(top)), where each pixel is blurred by vertical motion of the width of 5 pixels above and below

$$y = \mathbf{H}\, s$$

where s and y are vectorisations of the original and blurred images, respectively, and $\mathbf{H}$ is a sparse blurring matrix. We note that this matrix is of size 4096×4096, and has rank of 4094. In order to reconstruct the image s, one can apply the Wiener filtering or equivalently solve an optimisation problem which minimises the approximation error and the difference between each pixel and those surrounding it, i.e., to enhance smoothness in the image [28], as

$$\min_{\hat{x}} \quad \|y - \mathbf{D}\hat{x}\|_2^2 + \mu \|\mathbf{L}\,\hat{x}\|_2^2$$

where $\mu > 0$, and $\mathbf{L}$ is the discrete Laplacian, which plays a role of a high-pass filter. Different from the regularization filter, we express the estimated image as $\hat{x} = \alpha x$ where x is a unit-length vector, $x^T x = 1$, and minimise the reconstruction error

$$\min_{\alpha, x} \|y - \alpha \mathbf{D} x\|_2^2 = \|y\|_2^2 + \alpha^2\, x^T \mathbf{Q} x - 2\alpha b^T x, \quad \text{s.t.} \quad \|x\|_2^2 = 1,$$

where $\mathbf{Q} = \mathbf{D}^T \mathbf{D}$ and $b = \mathbf{D}^T y$. It is obvious that the optimal $\alpha^\star$ is given by

$$\alpha^\star = \frac{b^T x}{x^T \mathbf{Q} x}, \tag{8}$$

Algorithm 1: Spherically Constrained Quadratic Programming (SCQP)

Input: Q and b
Output: x minimises $\frac{1}{2} x^T \mathbf{Q} x + c^T x$, s.t., $x^T x = 1$
begin

1 Eigenvalue decomposition $\mathbf{Q} = \mathbf{U}\,\mathrm{diag}(\sigma)\,\mathbf{U}^T$

2 $s = \frac{\sigma - \sigma_1}{\|b\|} + 1$, $c = \mathbf{U}^T \frac{b}{\|b\|}$

3 $I_0 = \{l : c_l = 0, l > 1\}$, $I = \{1, \ldots, K\} \setminus I_0$, $\tilde{K} = \mathrm{card}(I)$

4 $\tilde{s} = s_I$, $\tilde{c} = c_I$

   if $\tilde{c}_1 = 0$ **then**

5       $d = \sum_{k \in I} \frac{\tilde{c}_k^2}{(\tilde{s}_k - 1)^2}$

     if $d < 1$ *and* $\tilde{s}_2 > \tilde{s}_1 = 1$ **then**

6         $\tilde{x}_1 = 1 \pm \sqrt{d}$, $\tilde{x}_k = \frac{\tilde{c}_k}{1 - \tilde{s}_k}$, $k = 2, \ldots, \tilde{K}$

     else

7         $\tilde{x} = \mathrm{qps\_nnz}(\tilde{s}(2 : \tilde{K}) - \tilde{s}(2) + 1, \tilde{c}(2 : \tilde{K}))$

8         $\tilde{x} = [0, \tilde{x}]$;

   else

9       $\tilde{x} = \mathrm{qps\_nnz}(\tilde{s}, \tilde{c})$

10 $x_{I_0} = 0$, $x_I = \tilde{x}$

11 $x \leftarrow \mathbf{U} x$

function $x = \mathrm{qps\_nnz}(s, c)$
Input: $s = [s_1 = 1 \leq s_2 \leq \ldots \leq s_K]$, unit-length vector c,
    $c^T c = 1$ and $c_k \neq 0$
Output: x minimises $\frac{1}{2} x^T \mathrm{diag}(s) x + c^T x$, s.t., $x^T x = 1$
begin

12 Compute the roots t_1 and t_2 in $[|c_1|, 1)$ of the polynomials
    $p_i(t) = t^4 + 2d_i\, t^3 + (d_i^2 - 1)\, t^2 - 2c_1^2 d_i\, t - c_1^2 d_i^2$, where
    $d_1 = s_2 - s_1$ and $d_2 = s_K - s_1$

13 Find a root λ in $(1 - t_1, 1 - t_2)$ of $f(\lambda) = 1 - \sum_k \frac{c_k^2}{(\lambda - s_k)^2}$

14 $x = [x_1, \ldots, x_K]$ where $x_k = \frac{c_k}{\lambda - s_k}$

---

and x is a solution to the following SCQP

$$\min \quad x^T \mathbf{Q} x + \frac{2}{\alpha} b^T x, \quad \text{s.t.} \quad x^T x = 1 . \tag{9}$$

Following this, we perform an alternating estimation process between x and α. We first initialize a unit-length vector x, compute α as in (8), then update x by solving (9), and update α again. The process is executed until there is no significant change in the object function value.

In Fig. 5, we show the reconstructed image using the regularisation filtering with $\mu = 0.0025$. The image achieved a PSNR = 24.6 dB. The image reconstructed using the SCQP based method obtained a PSNR = 49.05 dB after 672 iterations, as shown in Fig. 5(bottom). We note that the performance of the regularization filtering is affected by the choice of the regularisation parameter μ.

2.7 QP with inequality constraint $x^T x \leq 1$

For completeness of this section, we now present the QP with an inequality quadratic constraint

$$\min \quad \frac{1}{2} x^T \mathbf{Q} x + b^T x, \quad \text{s.t.} \quad x^T x \leq 1 . \tag{10}$$
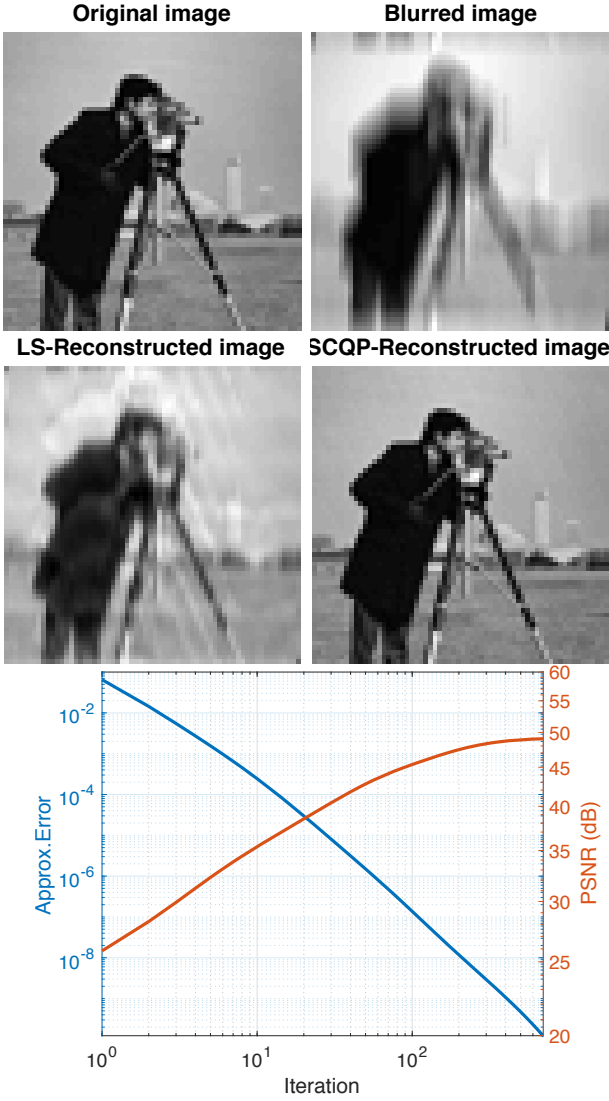
Fig. 5 Image deconvolution in Example 2. (Top) Comparison between reconstructed images using the regularization filtering and the SCQP-based reconstruction method. (Bottom) Objective function values and PSNR of the estimated image change over the iterations.

First, the vector x is expanded by an extra parameter z_1, where $z_1^2 = 1 - x^T x$, to yield a new unit length vector $z = [z_1, x_1, \ldots, x_K]^T$. The vector z is a global minimiser to the following SCQP

$$\min \quad \frac{1}{2} z^T \begin{bmatrix} 0 & \\ & \mathbf{Q} \end{bmatrix} z + b_z^T [0, b^T], \quad \text{s.t.} \quad z^T z = 1.$$

Or, in other words, $\tilde{z} = \begin{bmatrix} z_1 \\ \tilde{x} \end{bmatrix}$ is a minimiser to a simplified problem

$$\min \quad \frac{1}{2} \tilde{z}^T \operatorname{diag}(s_z) \tilde{z} + c_z^T \tilde{z}, \quad \text{s.t.} \quad \tilde{z}^T \tilde{z} = 1$$

where $c_z = [0, c^T]^T$ and $s_z = \frac{1}{\|b\|}[0, \sigma] + 1$. Since the first entry $c_z(1)$ is zero, the problem falls into the case stated in

Lemma 7. This can happen in two cases for

$$d = \sum_{k=1}^K \frac{c_k^2}{(s_z(k+1) - 1)^2} = \|b\|^2 \sum_{k=1}^K \frac{c_k^2}{\sigma_k^2}$$

– If $d \leq 1$, we obtain a solution $\tilde{x}_k = \|b\| \frac{c_k}{\sigma_k}$,
– Otherwise, $\tilde{z}_1 = z_1 = 0$, and we solve a QP with the equality constraint $x^T x = 1$.

Example 3 (**A toy example**) We replicate Example 1 in [11] for a minimization problem (10) with $\mathbf{Q} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$, $b = \begin{bmatrix} 0 \\ 1.8 \end{bmatrix}$. In order to solve the problem, we expand $\mathbf{Q}$ with one row and column of zeros, and b with one zero entry. The newly expanded matrix of $\mathbf{Q}$ has eigenvalues $[-1, 0, 1]$. After the normalisation of b and $\mathbf{Q}$, we obtain the expanded parameters

$$c = [0, \frac{b}{\|b\|}] = [0, 0, 1]^T,$$

and the shifted eigenvalues

$$s = \frac{[-1, 0, 1]^T + 1}{\|b\|} + 1 = [1, 1.5556, 2.1111]^T.$$

Since c_2 is zero, $\tilde{x}_2 = 0$, the problem boils down to finding the two variables $[\tilde{x}_1, \tilde{x}_3]$. Since c_1 is zero, and $d = \frac{c_3^2}{(s_3-1)^2} = 0.81 < 1$, according to Lemma 7,

$$\tilde{x}_3 = \frac{c_3}{s_3 - 1} = -0.9$$

and $\tilde{x}_1$ can take one of two values, $\tilde{x}_1 = \pm\sqrt{1 - \tilde{x}_3^2} = \pm 0.4359$. Finally, we convert $\tilde{x} = [\pm 0.4359, 0, -0.9]^T$ to the original space of the expanded vector x by multiplying it with the eigenvectors of $\mathbf{Q}$ to give

$$x = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tilde{x} = [0, \pm 0.4359, -0.9]^T.$$

That is, there are two global minimisers $[\pm 0.4359, -0.9]^T$. Fig. 6 illustrates the solution of the problem, where the shaded region shows the objective function when the points $[x_1, x_2]$ are on a unit circle.

Example 4 We now change the vector b in the previous example to $[0, 3]^T$. With this setting, the vector is still $c = [0, 0, 1]$, but the eigenvalues are $s = [1, 1.3333, 1.6667]$. Again, since $c_2 = 0$, we still have $\tilde{x}_2 = 0$. However, because $d = \frac{c_3^2}{(s_3-1)^2} = 2.25 > 1$, according to Lemma 7, $\tilde{x}_1 = 0$. We need to find only $\tilde{x}_3$. For this particular case, it turns out that $\tilde{x}_3 = -1$. Finally the global minimiser is $[0, -1]^T$.
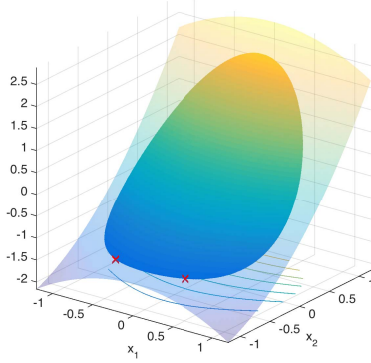
Fig. 6 Minimization of a quadratic function in a sphere in Example 3 has two global minimisers $[\pm 0.4359, -0.9]$, which are red "cross" points.

3 SCQP with Matrix-variates

Consider an extension of the SCQP in (1) for a matrix-variate. The problem can be formulated for a matrix $\mathbf{X}$ of size $I \times R$ as

$$\min \quad f(\mathbf{X}) = \frac{1}{2} \operatorname{tr}(\mathbf{X}^T \mathbf{Q} \mathbf{X}) + \operatorname{tr}(\mathbf{B}^T \mathbf{X}) \quad \text{s.t.} \quad \|\mathbf{X}\|_F^2 = 1 \quad (11)$$

where $\mathbf{Q}$ is a psd matrix of size $I \times I$ and $\mathbf{B}$ is of size $I \times R$. A straightforward approach to (11) is to rewrite it in the form of an ordinary SCQP for the vectorisation $\operatorname{vec}(\mathbf{X})$,

$$\min \quad f(\mathbf{X}) = \frac{1}{2} \operatorname{vec}(\mathbf{X})^T (\mathbf{I}_R \otimes \mathbf{Q}) \operatorname{vec}(\mathbf{X}) + \operatorname{vec}(\mathbf{B})^T \operatorname{vec}(\mathbf{X})$$
$$\text{s.t.} \quad \|\mathbf{X}\|_F = 1 \,,$$

and then apply the algorithm in the previous section to find $\mathbf{X}$. The symbol "$\otimes$" stands for the Kronecker product.

An alternative method would be to rewrite the objective function in a form similar to (2), as

$$f(\mathbf{X}) = \frac{1}{2} \boldsymbol{x}^T (\mathbf{I}_R \otimes \operatorname{diag}(\boldsymbol{\sigma})) \boldsymbol{x} + \boldsymbol{v}^T \boldsymbol{x}$$

where $\boldsymbol{x} = \operatorname{vec}(\mathbf{X}^T \mathbf{U})$, $\boldsymbol{v} = \operatorname{vec}(\mathbf{B}^T \mathbf{U})$ and $\mathbf{Q} = \mathbf{U} \operatorname{diag}(\boldsymbol{\sigma}) \mathbf{U}^T$ is an EVD of $\mathbf{Q}$. Due to the Kronecker product, each eigenvalue σ_i, $i = 1, \dots, I$, is replicated R times. Hence, according to Lemma 1, we can deduce the minimiser to (11) from the minimiser $\boldsymbol{z}^\star$ to an SCQP of a smaller scale

$$\min \quad \frac{1}{2} \boldsymbol{z}^T \operatorname{diag}(\boldsymbol{\sigma}) \boldsymbol{z} + \boldsymbol{c}^T \boldsymbol{z} \quad \text{s.t.} \quad \boldsymbol{z}^T \boldsymbol{z} = 1 \,,$$

where $\boldsymbol{c} = [c_1, \dots, c_I]$, $c_i = \|\mathbf{B}^T \boldsymbol{u}_i\|$. More specifically, $\boldsymbol{x}_i = \frac{z_i}{c_i} \mathbf{B}^T \boldsymbol{u}_i$ for a nonzero coefficient c_i. Otherwise, $\boldsymbol{x}_i$ is often a zero vector for a zero c_i, except only the case $c_1 = 0$.

4 SCQP for large scale data

The most computationally demanding step in the closed-form method for SCQP is the EVD of the matrix $\mathbf{Q}$ of size

$K \times K$. When the vector $\boldsymbol{x}$ comprises hundreds of thousands of entries, this computation may not be executed in a computer. To this end, we convert the large scale SCQP to sub-problems of smaller scale, each of which can be solved in closed-form.

First we partition the index set $I = [1, 2, \dots, K]$ into L disjoint segments I_l of size K_l, $K = K_1 + K_2 + \cdots + K_L$, such that EVDs of matrices of size $K_l \times K_l$ can be performed on a computer. For each sub-vector $\boldsymbol{x}_l = \boldsymbol{x}(I_l)$, we denote by α_l and $\tilde{\boldsymbol{x}}_l$ its ℓ_2-norm and normalized vector, $l = 1, \dots, L$, respectively, i.e., $\boldsymbol{x}(I_l) = \alpha_l \tilde{\boldsymbol{x}}_l$ where $\tilde{\boldsymbol{x}}_l^T \tilde{\boldsymbol{x}}_l = 1$. We also denote a complement set by $I_{\bar{l}} = \{1, \dots, K\} \setminus I_l$. Similarly, we define $\alpha_{\bar{l}}$ and sub-vectors $\tilde{\boldsymbol{x}}_{\bar{l}}$, $\boldsymbol{b}_l$ and $\boldsymbol{b}_{\bar{l}}$. Note that $\boldsymbol{x}^T \boldsymbol{x} = \alpha_l^2 + \alpha_{\bar{l}}^2 = 1$ and $\alpha_{\bar{l}} = \|\boldsymbol{\alpha}_{m \neq l}\|_2$. Hence, the vector $[\alpha_l, \alpha_{\bar{l}}]^T$ also has a unit length.

For convenience, we consider again the SCQP problem for $\boldsymbol{x}$

$$\min_{\boldsymbol{x}} \quad y = \frac{1}{2} \boldsymbol{x}^T \mathbf{Q} \boldsymbol{x} + \boldsymbol{b}^T \boldsymbol{x} \quad \text{s.t.} \quad \boldsymbol{x}^T \boldsymbol{x} = 1 \quad (12)$$

and rewrite it as SCQP sub-problems for unit-length vectors $[\alpha_l, \alpha_{\bar{l}}]^T$ and $\tilde{\boldsymbol{x}}_l$, for $l = 1, \dots, L$. For example, an SCQP for only two parameters $[\alpha_l, \alpha_{\bar{l}}]^T$ is given by

$$\min \quad y = \frac{1}{2} [\alpha_l, \alpha_{\bar{l}}] \mathbf{T}_l [\alpha_l \, \alpha_{\bar{l}}]^T + [\alpha_l, \alpha_{\bar{l}}] \boldsymbol{u} \quad (13)$$
$$\text{s.t.} \quad \alpha_l^2 + \alpha_{\bar{l}}^2 = 1$$

where $\mathbf{T}_l = \begin{bmatrix} \tilde{\boldsymbol{x}}_l^T \mathbf{Q}_{l,l} \tilde{\boldsymbol{x}}_l & \tilde{\boldsymbol{x}}_l^T \mathbf{Q}_{l,\bar{l}} \tilde{\boldsymbol{x}}_{\bar{l}} \\ \tilde{\boldsymbol{x}}_l^T \mathbf{Q}_{l,\bar{l}} \tilde{\boldsymbol{x}}_{\bar{l}} & \tilde{\boldsymbol{x}}_{\bar{l}}^T \mathbf{Q}_{\bar{l},\bar{l}} \tilde{\boldsymbol{x}}_{\bar{l}} \end{bmatrix}$ is of size 2×2 and $\boldsymbol{u} = [\boldsymbol{b}_l^T \tilde{\boldsymbol{x}}_l, \boldsymbol{b}_{\bar{l}}^T \tilde{\boldsymbol{x}}_{\bar{l}}]^T$. The above problem can be straightforwardly solved in a closed-form, while keeping $\tilde{\boldsymbol{x}}_l$ fixed. Once α_l and $\alpha_{\bar{l}}$ are updated, the other scaling coefficients α_m for $m \neq l$ are then scaled by a factor of $\dfrac{\alpha_{\bar{l}}}{\|\boldsymbol{\alpha}_{m \neq l}\|_2}$

$$\alpha_{m \neq l} \leftarrow \frac{\alpha_{\bar{l}}}{\|\boldsymbol{\alpha}_{m \neq l}\|_2} \alpha_{m \neq l} \,. \quad (14)$$

Next we rewrite the SCQP for $\boldsymbol{x}$ in (12) as an SCQP for $\tilde{\boldsymbol{x}}_l$, for $l = 1, \dots, L$, while keeping the other parameters fixed as

$$\min_{\tilde{\boldsymbol{x}}_l} \quad y = \frac{\alpha_l^2}{2} \tilde{\boldsymbol{x}}_l^T \mathbf{Q}_{l,l} \tilde{\boldsymbol{x}}_l + \alpha_l (\boldsymbol{b}_l + \mathbf{Q}_{l,\bar{l}} \boldsymbol{x}_{\bar{l}})^T \tilde{\boldsymbol{x}}_l + c_l \quad (15)$$
$$\text{s.t.} \quad \tilde{\boldsymbol{x}}_l^T \tilde{\boldsymbol{x}}_l = 1 \,,$$

where c_l is independent of $\tilde{\boldsymbol{x}}_l$. Because $\mathbf{Q}_{l,l}$ are of relatively small sizes $K_l \times K_l$, update of $\tilde{\boldsymbol{x}}_l$ can be proceeded in closed-form.

Finally, by alternating between the updates in (13), (14) and (15), we can update entire parameters α_l and $\tilde{\boldsymbol{x}}_l$. We summarise the update procedure in Algorithm 2. For each partitioning of $[1, \dots, K]$, EVDs of $\mathbf{Q}_{l,l}$ are computed only once, then we perform an inner loop to update $\tilde{\boldsymbol{x}}_l$ and α_l until there is no further improvement.

Algorithm 2: Block update for large scale SCQP

Input: Q and b
Output: x minimises $\frac{1}{2}\,x^T\mathbf{Q}x + c^T x$, s.t., $x^T x = 1$
begin
    repeat
1   Partition $\{1, 2, \ldots, K\}$ into L disjoint segments in random
2   Precompute EVD of $\mathbf{Q}_{l,l}$ for $l = 1, \ldots, L$
        repeat
            for $l = 1, \ldots, L$ **do**
3   Solve SCQP in (15) to update $\tilde{x}_l$ of length K_l
4   Solve SCQP in (13) to update $[\alpha_l, \alpha_{\bar{l}}]$
5   Adjust $\alpha_{m\neq l} \leftarrow \frac{\alpha_{\bar{l}}}{\|\alpha_{m\neq l}\|_2}\,\alpha_{m\neq l}$
        until *a stopping criterion is met*
    until *a stopping criterion is met*

---

5 Linear Regression with Bound Constraint

Another problem, which can be formulated as SCQP, is the linear regression with a constrained bound on the regression error

$$\min_{x} \quad \|x\|^2 \quad \text{subject to} \quad \|y - \mathbf{A}x\| \leq \delta\,, \qquad (16)$$

where y is a vector of length I of dependent variables, $\mathbf{A}$ is a regressor matrix of size $I \times K$ and δ a nonnegative regression bound.

It is obvious that if $\delta \geq \|y\|$, then the zero vector $x = \mathbf{0}$ is a minimiser to (16). Therefore, in order to achieve a meaningful regression, the regression bound δ needs to be in the following range.

Lemma 8 (Range of the bound δ) *The problem (16) has a minimiser of nonzero entries when*

$$\|\boldsymbol{\Pi}_{\mathbf{A}}^{\perp} y\| \leq \delta < \|y\|$$

where $\boldsymbol{\Pi}_{\mathbf{A}}^{\perp}$ is an orthogonal complement of the column space of $\mathbf{A}$.

Proof Let $\mathbf{U}$ be an orthogonal basis for the column space of $\mathbf{A}$. Then

$$\delta^2 \geq \|y - \mathbf{A}x\|^2 = \|\mathbf{U}^T y - \mathbf{U}^T \mathbf{A}x\|_F^2 + \|\boldsymbol{\Pi}_{\mathbf{A}}^{\perp} y\|^2 \geq \|\boldsymbol{\Pi}_{\mathbf{A}}^{\perp} y\|^2\,.$$

For simplicity, we assume that $\mathbf{A}$ is full rank matrix, otherwise, we solve the problem with a compressed regressor matrix with a smaller bound

$$\min \quad \|x\|^2 \quad \text{subject to} \quad \|\hat{y} - \hat{\mathbf{A}}x\| \leq \hat{\delta}$$

where $\hat{y} = \mathbf{U}^T y$, $\hat{\mathbf{A}} = \mathbf{U}^T \mathbf{A}$, and $\hat{\delta}^2 = \delta^2 - \|\boldsymbol{\Pi}_{\mathbf{A}}^{\perp} y\|^2$.

We shall now derive an equivalent SCQP to the problem in (16). We first show that the inequality sign in (16) can be replaced by the equal sign.

Lemma 9 *The minimiser to (16) is the minimiser to the following problem*

$$\min_{x} \quad \|x\|^2 \quad \text{s.t.} \quad \|y - \mathbf{A}x\| = \delta \qquad (17)$$

See the proof in Appendix H.

The proposed algorithms to solve the problem in (17) are presented for two cases, when the length of x does not exceed the number of dependent variables, $K \leq I$, and when $K > I$.

5.1 The case when $K \leq I$

We first consider the case when the matrix of regressors $\mathbf{A}$ is of full column rank, $K \leq I$. Let $\mathbf{A} = \mathbf{U}\,\text{diag}(s)\mathbf{V}^T$ be an SVD of $\mathbf{A}$, where $\mathbf{V}$ is an orthonormal matrix of size $K \times K$, and $s = [s_1, \ldots, s_K] > 0$. Hence $\boldsymbol{\Pi}_{\mathbf{A}}^{\perp} = \mathbf{I} - \mathbf{U}\,\mathbf{U}^T$.

Let $\hat{y} = \mathbf{U}^T y$, $\hat{\delta} = \sqrt{\delta^2 - \|\boldsymbol{\Pi}_{\mathbf{A}}^{\perp} y\|^2}$, $z = \frac{1}{\hat{\delta}}(\hat{y} - \text{diag}(s)\mathbf{V}^T x)$, then

$$x = \mathbf{V}\,\text{diag}(s^{-1})(\hat{y} - \hat{\delta}z)$$
$$\|x\|_F^2 = (\hat{y} - \hat{\delta}z)^T\,\text{diag}(s^{-2})(\hat{y} - \hat{\delta}z)$$
$$\|y - \mathbf{A}x\|^2 = \|\mathbf{U}^T y - \text{diag}(s)\mathbf{V}^T x\|_F^2 + \|\boldsymbol{\Pi}_{\mathbf{A}}^{\perp} y\|^2$$
$$= \hat{\delta}^2\,\|z\|^2 + \|\boldsymbol{\Pi}_{\mathbf{A}}^{\perp} y\|^2\,.$$

By this reparameterization, the problem in (17) becomes an SCQP which can be solved in closed-form

$$\min_{z} \quad z^T\,\text{diag}(\hat{\delta}s^{-2})z - 2\,\hat{y}^T\,\text{diag}(s^{-2})z \quad \text{s.t.} \quad z^T z = 1.$$

5.2 The case when $K > I$

For this case, we develop an iterative algorithm, at each iteration, wherebt the problem (17) is rewritten as a subproblem with an invertible regressor matrix. To this end, we first generate an initial feasible point $x^{(0)}$ such that $\|y - \mathbf{A}x^{(0)}\| = \delta$.

We then select a sub matrix $\mathbf{A}_{\mathcal{I}}$ of $\mathbf{A}$ such that $\mathbf{A}_{\mathcal{I}}$ is invertible, $2 \leq \text{card}(\mathcal{I}) \leq I$ and at least one entry $x_{k\in\mathcal{I}}^{(0)}$ is non-zero. Let $z_{\mathcal{I}}^{(0)} = y - \sum_{k\notin\mathcal{I}} a_k x_k^{(0)}$, then

$$\|y - \mathbf{A}x\| = \|z_{\mathcal{I}}^{(0)} - \mathbf{A}_{\mathcal{I}}\,x_{\mathcal{I}}^{(0)}\| = \delta.$$

By fixing the parameters x_k, $k \notin \mathcal{I}$, the new estimate $x_{\mathcal{I}}^{(1)}$ of $x_{\mathcal{I}}$ is a minimiser to the following problem

$$x_{\mathcal{I}}^{(1)} = \arg\min_{x_{\mathcal{I}}} \quad \|x_{\mathcal{I}}\|^2 \quad \text{subject to} \quad \|z_{\mathcal{I}}^{(0)} - \mathbf{A}_{\mathcal{I}}x_{\mathcal{I}}\| = \delta.$$

Since $\mathbf{A}_{\mathcal{I}}$ is invertible and $x_{\mathcal{I}}^{(0)}$ is a non-zero point which holds the constraint, the above constrained QP has a non-zero global minimiser, which can be solved in closed-form as in the case in Section 5.1. As a result of this update, the new estimate $x^{(1)}$ still holds the constraint, and

$$\|x^{(1)}\|^2 = \|x^{(0)}\|^2 - \|x_{\mathcal{I}}^{(0)}\|^2 + \|x_{\mathcal{I}}^{(1)}\|^2 \leq \|x^{(0)}\|^2\,.$$

The algorithm then selects another index set $\mathcal{I}$, and continues updating the entries $x_{\mathcal{I}}$ by a non-zero $x_{\mathcal{I}}^{(t)}$. This alternating update scheme generates a sequence of estimates $x^{(t)}$,

which preserve the constraint $\|\boldsymbol{y} - \mathbf{A}\boldsymbol{x}^{(t)}\| = \delta$, while keeping their norm non-increasing, $\|\boldsymbol{x}^{(t)}\| \leq \|\boldsymbol{x}^{(t-1)}\| \leq \ldots \leq \|\boldsymbol{x}^{(0)}\|$.

The linear regression with a bound error constraint has found novel applications in the error preserving correction methods for the Canonical Polyadic tensor Decomposition or the CPD with bounded norm of rank-1 tensors [26].

6 Quadratic Programming with Elliptic Constraints

Consider a QP with multiple quadratic constraints, each representing an ellipsoid, so that the feasible set is an intersection of the ellipsoids. This problem has been extensively studied in the literature, and arise in many applications in phase recovery, power flow, MIMO detection, quadratic-assignment, sensor-network localization, max-cut problems. For comprehensive review of the problem and its applications, we refer to [23, 25].

Definition 2 (Quadratic programming over ellipsoids) Consider a positive semi-definite matrix $\mathbf{Q}$ of size $K \times K$, a vector $\boldsymbol{b}$ of length K, and a set of M positive semi-definite matrices $\mathbf{H}_m$. The quadratic programming over ellipsoids solves the optimisation problem

$$\min \quad \frac{1}{2}\boldsymbol{x}^T\mathbf{Q}\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{x} \tag{18}$$
$$\text{s.t.} \quad \boldsymbol{x}^T\mathbf{H}_m\boldsymbol{x} = 1, \quad m = 1, \ldots, M\,.$$

The constraints in the above programming are given in a simple form without linear terms as in the objective function. In practice, however, the full quadratic forms can be converted to the homogenised form of the parameter vector $\left[1, \boldsymbol{x}^T\right]^T$, e.g., $\left[1, \boldsymbol{x}^T\right]\begin{bmatrix} 0 & \boldsymbol{b}^T \\ \boldsymbol{b} & \mathbf{Q} \end{bmatrix}\begin{bmatrix} 1 \\ \boldsymbol{x} \end{bmatrix}$ [25].

In addition, the case with inequality constraints, i.e., $\boldsymbol{x}^T\mathbf{H}_m\boldsymbol{x} \leq 1$, can also be converted to the equality constraints by introducing additional variables $\boldsymbol{s} = [s_1, s_2, \ldots, s_M]$ such that

$$1 = \boldsymbol{x}^T\mathbf{H}_m\boldsymbol{x} + s_m^2 = [\boldsymbol{x}^T, \boldsymbol{s}^T]\begin{bmatrix} \mathbf{H}_m & \\ & \boldsymbol{e}_m\boldsymbol{e}_m^T \end{bmatrix}\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{s} \end{bmatrix}\,.$$

For the above quadratically constrained quadratic programming (QCQP) problem, we can apply relaxations to find approximate solutions, e.g., the Lagrangian and Semidefinite Programming (SDP) based relaxations. The SDP relaxation introduces a symmetric matrix of rank-1, $\mathbf{X} = \boldsymbol{x}\boldsymbol{x}^T$, and relaxes the condition to the semidefinite condition $\mathbf{X} \succeq \boldsymbol{x}\boldsymbol{x}^T$. The quadratic objective and constraint functions can then be rewritten in linear form, see [4, 5, 19] for relaxations for QCQP.

Different from the existing methods, we introduce an augmented Lagrangian based algorithm for the problem in (18). The constraints over multiple ellipsoids are interpreted as a constraint over a sphere and an orthogonal projection. In order to achieve this, we define symmetric matrices

---

Algorithm 3: Augmented Lagrangian Algorithm for QCQP

Input: Matrices $\mathbf{Q}, \mathbf{D}_1, \ldots, \mathbf{D}_N$ and $\boldsymbol{b}$
Output: $\boldsymbol{x}$ minimises $\quad \frac{1}{2}\boldsymbol{x}^T\mathbf{Q}\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{x}, \quad$ s.t., $\boldsymbol{x}^T\boldsymbol{x} = 1$,
$\quad\quad\quad \boldsymbol{x}^T\mathbf{D}_n\boldsymbol{x} = 0$
begin
1   Initialize $\boldsymbol{y}$ and $\boldsymbol{z}$ as zero vectors and $\gamma > 0$
   repeat
2      $\mathbf{T} = \texttt{reshape}(\boldsymbol{z} - \frac{\boldsymbol{y}}{\gamma}, [K \times K]), \mathbf{T}_x = \frac{1}{2}(\mathbf{T} + \mathbf{T}^T)$
3      $\boldsymbol{x} = \texttt{QPS}(\mathbf{Q} + \gamma\mathbf{I} - 2\gamma\mathbf{T}_x, \boldsymbol{b})$
4      $\boldsymbol{z} = \Pi_{\mathbf{D}}^\perp(\boldsymbol{x} \otimes \boldsymbol{x} + \frac{\boldsymbol{y}}{\gamma})$
5      $\boldsymbol{y} \leftarrow \boldsymbol{y} + \gamma(\boldsymbol{x} \otimes \boldsymbol{x} - \boldsymbol{z})$
6   Adjust $\gamma \leftarrow \alpha\gamma$ if the objective function tents to a slow convergence
   until *a stopping criterion is met*

$\mathbf{D}_n = \mathbf{H}_1 - \mathbf{H}_{n+1}$, and rewrite the optimisation problem in (18) in the form of

$$\min \quad \frac{1}{2}\boldsymbol{x}^T\mathbf{Q}\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{x}$$
$$\text{s.t.} \quad \boldsymbol{x}^T\mathbf{H}_1\boldsymbol{x} = 1, \quad \boldsymbol{x}^T\mathbf{D}_n\boldsymbol{x} = 0, \quad n = 1, \ldots, N$$

or in the following form

$$\min \quad \frac{1}{2}\tilde{\boldsymbol{x}}^T\tilde{\mathbf{Q}}\tilde{\boldsymbol{x}} + \tilde{\boldsymbol{b}}^T\tilde{\boldsymbol{x}} \tag{19}$$
$$\text{s.t.} \quad \tilde{\boldsymbol{x}}^T\tilde{\boldsymbol{x}} = 1 \text{ and } \tilde{\boldsymbol{x}}^T\tilde{\mathbf{D}}_n\tilde{\boldsymbol{x}} = 0, \quad n = 1, \ldots, N$$

after a reparameterization

$$\tilde{\boldsymbol{x}} = \mathbf{F}^T\boldsymbol{x}, \quad\quad\quad\quad \tilde{\mathbf{Q}} = \mathbf{F}^{-1}\mathbf{Q}\mathbf{F}^{-1\,T},$$
$$\tilde{\boldsymbol{b}} = \mathbf{F}^{-1}\boldsymbol{b}, \quad\quad\quad\quad \tilde{\mathbf{D}}_n = \mathbf{F}^{-1}\mathbf{D}_n\mathbf{F}^{-1\,T},$$

where $\mathbf{F}$ is the Cholesky factor matrix of $\mathbf{H}_1 = \mathbf{F}\mathbf{F}^T$.

For simplicity of notation, we will solve the problem in (19) with parameters $\mathbf{Q}$, $\mathbf{Z}$ and $\boldsymbol{b}$ and variables $\boldsymbol{x}$ with $\boldsymbol{x}^T\boldsymbol{x} = 1$, that is

$$\min \quad \frac{1}{2}\boldsymbol{x}^T\mathbf{Q}\boldsymbol{x} + \boldsymbol{b}^T\tilde{\boldsymbol{x}} \tag{20}$$
$$\text{s.t.} \quad \boldsymbol{x}^T\boldsymbol{x} = 1,$$
$$\quad\quad \boldsymbol{x}^T\mathbf{D}_n\boldsymbol{x} = 0, \quad n = 1, \ldots, N\,.$$

6.1 An Augmented-Lagrangian algorithm for QCQP

In order to solve the QCQP in (20), we split the problem into two subproblems, each with a single constraint, by introducing an additional variable $\boldsymbol{z}$

$$\min \quad f(\boldsymbol{x}) + g(\boldsymbol{z}) \quad \text{s.t.} \quad \boldsymbol{z} - \boldsymbol{x} \otimes \boldsymbol{x} = \mathbf{0} \tag{21}$$

where $f(\boldsymbol{x})$ is the function of $\boldsymbol{x}$ over a sphere for the optimization problem

$$\min \frac{1}{2}\boldsymbol{x}^T\mathbf{Q}\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{x}, \quad \text{subject to} \quad \boldsymbol{x}^T\boldsymbol{x} = 1,$$

and $g(z)$ represents the projection onto subspace span by orthogonal complement of $\mathbf{D} = [\text{vec}(\mathbf{D}_1), \ldots, \text{vec}(\mathbf{D}_N)]$, $\mathbf{D}^T z = \mathbf{0}$.

The augmented Lagrangian function of the problem (21) is given by

$$\mathcal{L}(x, y, z) = f(x) + g(z) + y^T(x \otimes x - z) + \frac{\gamma}{2}\|x \otimes x - z\|^2$$

where $\gamma > 0$. The algorithm consists of update rules for the variables x, z and y

$$x = \arg\min \quad f(x) + y^T(x \otimes x - z) + \frac{\gamma}{2}\|x \otimes x - z\|^2 \,, \quad (22)$$

$$z = \arg\min \quad g(z) + y^T(x \otimes x - z) + \frac{\gamma}{2}\|x \otimes x - z\|^2 \,, \quad (23)$$

$$y \leftarrow y + \gamma(x \otimes x - z) \,. \quad (24)$$

6.2 Estimation of x

The optimisation problem in (22) can indeed be written as an SCQP, as follows

$$x = \arg\min \ f(x) + y^T(x \otimes x - z) + \frac{\gamma}{2}\|x \otimes x - z\|^2$$

$$= \arg\min \ f(x) + \frac{\gamma}{2}\|x \otimes x - z + \frac{y}{\gamma}\|^2 \quad (25)$$

$$= \arg\min \ f(x) + \frac{\gamma}{2}(x^Tx)^2 - \gamma(x \otimes x)^T(z - \frac{y}{\gamma})$$

$$= \arg\min \ \frac{1}{2}x^T(\mathbf{Q} + \gamma\mathbf{I} - 2\gamma\mathbf{T}_x)x + b^Tx \quad \text{s.t. } x^Tx = 1 \,,$$
$$(26)$$

where $\mathbf{T}_x = \frac{\mathbf{T} + \mathbf{T}^T}{2}$ is a symmetric matrix of size $K \times K$, $\mathbf{T} = \texttt{reshape}(z - \frac{y}{\gamma}, [K \times K])$. At each iteration to update x, we reshape the vector $(z - \frac{y}{\gamma})$ to a matrix of size $K \times K$, and then construct a symmetric matrix $\mathbf{T}_x$. The matrix $\mathbf{Q}$ is changed by a term $\gamma\mathbf{T}_x$, while the vector b is preserved. The last equation indicates that the vector x can be found in closed-form using Algorithm 1.

6.3 Estimation of z

The vector z is updated as a minimiser to the following problem

$$z = \arg\min \quad g(x) + y^T(x \otimes x - z) + \frac{\gamma}{2}\|x \otimes x - z\|^2$$

$$= \arg\min \quad g(x) + \frac{\gamma}{2}\|z - x \otimes x - \frac{y}{\gamma}\|^2$$

$$= \arg\min \quad \|z - x \otimes x - \frac{y}{\gamma}\|^2 \qquad \text{s.t. } \mathbf{D}^T z = 0$$

$$= \Pi_{\mathbf{D}}^{\perp}(x \otimes x + \frac{y}{\gamma}) \,, \quad (27)$$

where $\Pi_{\mathbf{D}}^{\perp}(z)$ is the orthogonal projection of the vector z onto the orthogonal complement of the column space of $\mathbf{D}$, e.g.,

$$\Pi_{\mathbf{D}}^{\perp}(z) = z - \mathbf{D}(\mathbf{D}^T\mathbf{D})^{-1}\mathbf{D}^T z \,.$$

6.4 Algorithm for QCQP

An augmented Lagrangian based algorithm for QCQP including the updates in (26) and (27) is summarised in Algorithm 3. The vectors y and z are initialised as zeros, while the parameter γ is set to a sufficiently high value. Experiments show that running the algorithm with a small γ at the beginning will decrease the objective function quickly, but it may make the algorithm unstable after several to a dozen of iterations. However, setting γ to a too large value will slow down the convergence of the algorithm. In order to obtain a good setting, we should run the algorithm for a few iterations for various values of γ, then choose the setting which gives a good convergence result. The algorithm is then executed using the chosen parameters. In our experience, γ can be set to a fraction of the minimum condition number of $\mathbf{H}_m$, while the associated matrix $\mathbf{H}_m$ should be chosen to present the quadratic constraint, i.e., $x^T\mathbf{H}_m x = 1$.

During the estimation process, the high value of parameter γ should be reduced if the objective function becomes stable, or yields a slow convergence. However, reducing γ too much can cause a divergence, and the parameters should be corrected.

6.5 Linearisation for the update of x

As per derivation in (26), x is updated as a minimiser of an SCQP. The algorithm iterates to update x over z and y. As in Algorithm 1, at each iteration to update x, one needs to compute the eigenvalue decomposition of the matrix $\mathbf{Q} + \gamma\mathbf{I} - 2\gamma\mathbf{T}_x$, which is changed by a term $\mathbf{T}_x$. In order to accelerate the update rule, we perform the following linearisation which can bypass the matrix $\mathbf{T}_x$ in the quadratic term.

From (25), x is updated as a minimiser to an SCQP

$$x = \arg\min \quad f(x) + \frac{\gamma}{2}\|x \otimes x - z + \frac{y}{\gamma}\|^2$$

$$= \arg\min \quad f(x) + \frac{\gamma}{2}\|xx^T - \mathbf{T}_x\|_F^2 \,.$$

Now, we replace the second term in the above problem by its linearization at the previous update denoted by x_o

$$x = \arg\min \ f(x) + \frac{\gamma}{2}\left(c(x_o) + \left(\frac{\partial c(x)}{\partial x}(x_o)\right)^T(x - x_o)\right.$$
$$\left. + \mu\|x - x_o\|_F^2\right)$$

$$= \arg\min \ f(x) + \frac{\gamma}{2}\left(4(x_o x_o^T - \mathbf{T}_x)x_o\right)^T x + \frac{\gamma\mu}{2}\|x - x_o\|_F^2$$

$$= \arg\min \ f(x) + 2\gamma x_o^T(x_o x_o^T - \mathbf{T}_x)x + \frac{\gamma\mu}{2}\|x - x_o\|_F^2$$

$$= \arg\min \ \frac{1}{2}x^T(\mathbf{Q} + \gamma\mu\mathbf{I})x + (b + \gamma(2 - \mu)x_o - 2\gamma\mathbf{T}_x x_o)^T x$$
$$(28)$$

$$\text{s.t.} \quad x^Tx = 1$$

where $c(\boldsymbol{x}) = \|\boldsymbol{x}\boldsymbol{x}^T - \mathbf{T}_x\|_F^2$ and $\mu > 0$.

The optimisation in (28) shows that the quadratic term adjusts $\mathbf{Q}$ by a term $\gamma\mu\mathbf{I}$, i.e., shifting its eigenvalues by $\gamma\mu$. So we need not decompose the matrix $\mathbf{Q}$ again. Different from the update in (26), the linear term changes by the previous estimate of $\boldsymbol{x}$.

For this new update (28), the algorithm is computationally cheaper, and still preserves the convergence as that in (26).

6.6 Generating a positive definite matrix $\mathbf{H}_1$

In the conversion of the QCQP problem in (18) to the problem (20), the constraints over multiple ellipsoids are interpreted as a constraint over a sphere and orthogonality constraints. Choosing a psd matrix $\mathbf{H}_k$ from the set of matrices $\mathbf{H}_n$ plays an important role and affects the entire estimating process. Here, a simple condition is that the selected matrix $\mathbf{H}_k$ should have a low condition number. In some cases, it is better to generate a new psd matrix $\mathbf{H}_\alpha$ rather than choosing one among $\mathbf{H}_n$. The new matrix $\mathbf{H}_\alpha$ should have a condition number as small as possible by solving an eigenvalue problem (EVP) [8]

$$\min_{\alpha,\gamma} \quad \gamma, \quad \text{s.t.} \quad \mathbf{I} < \mathbf{H}_\alpha < \gamma\mathbf{I}$$

where $\mathbf{H}_\alpha = \sum_{n=1}^N \mathbf{H}_n\alpha_n$.

The problem can also be formulated as a semidefinite programming (SDP) problem, which can be solved using the SEDUMI or TFOCS toolboxes. The generated matrix $\mathbf{H}_\alpha$ is then scaled by a factor of $\left(\sum_{n=1}^N \alpha_n\right)$, so that it satisfies the quadratic constraint $\boldsymbol{x}^T\mathbf{H}_\alpha\boldsymbol{x} = 1$.

Alternatively, the matrix $\mathbf{H}_\alpha$ can be generated so that its Frobenius norm is minimum, i.e.,

$$\min \quad \|\mathbf{H}_\alpha\|_F^2$$
$$\text{subject to} \quad \mathbf{H}_\alpha = \sum_{n=1}^N \mathbf{H}_n\alpha_n, \quad \sum_{n=1}^N \alpha_n = 1, \quad \alpha_n \geq 0.$$

For the latter problem, we define a matrix $\mathbf{H} = [\text{vec}(\mathbf{H}_1),\dots,\text{vec}(\mathbf{H}_N)]$, and find a vector $\boldsymbol{\alpha} = [\alpha_1,\dots,\alpha_N]$ in a quadratic programming with a linear constraint

$$\min \quad \boldsymbol{\alpha}^T(\mathbf{H}^T\mathbf{H})\boldsymbol{\alpha}, \quad \text{s.t.} \quad \mathbf{1}^T\boldsymbol{\alpha} = 1, \quad \alpha_n \geq 0.$$

Example 5 (**Effects of the condition number of the psd matrix involving in the quadratic constraint**)

In this example, we consider a QP problem of a psd matrix $\mathbf{Q}$ of size 10×10 over quadratic constraints for three psd matrices $\mathbf{H}_1$, $\mathbf{H}_2$ and $\mathbf{H}_3$. The linear term in the QP problem is with a zero vector $\boldsymbol{b}$. The matrices are randomly generated, and the condition numbers of $\mathbf{H}_n$ are 185.7, 12403 and 1000.1, respectively. Since the second matrix $\mathbf{H}_2$ has a high condition number, we generate a new psd matrix $\tilde{\mathbf{H}}_2$ from $\mathbf{H}_1$, $\mathbf{H}_2$ and $\mathbf{H}_3$ as described in Section 6.6. The new matrix has a low condition number of 8.9, and is used in place of the matrix $\mathbf{H}_2$.

In the first analysis, we compare the performances and convergence behaviour of the QCQP algorithm (Algorithm 3) when each matrix, either $\mathbf{H}_i$ or $\tilde{\mathbf{H}}_2$, is selected to represent the quadratic constraint, i.e., $\boldsymbol{x}^T\mathbf{H}_i\boldsymbol{x} = 1$, and the remaining two matrices represent the orthogonality constraints of the parameter vector $\boldsymbol{z}$. There are four possible selections of the matrix. The parameter γ is fixed to 0.1 in the test. Fig. 7 shows the objective values to illustrate the convergence and final performance, and the ℓ_2-norm of the orthogonality constraints $\|\mathbf{D}^T(\boldsymbol{x} \otimes \boldsymbol{x})\|_2^2$ to verify if all the quadratic constraints $\boldsymbol{x}^T\mathbf{H}_i\boldsymbol{x} = 1$ are achieved. The results indicate that when the matrix with a high condition number, $\mathbf{H}_2$ or $\mathbf{H}_3$, plays as a quadratic constraint, the algorithm converges to a false local minima, which do not satisfy the orthogonality constraints $\mathbf{D}^T(\boldsymbol{x} \otimes \boldsymbol{x}) = \mathbf{0}$.

When running the optimisation with a quadratic constraint over the matrix $\mathbf{H}_1$ or $\tilde{\mathbf{H}}_2$, the algorithm converges to the same value of 4.8659×10^{-4} with a norm $\|\mathbf{D}^T(\boldsymbol{x}\otimes\boldsymbol{x})\|_2^2$ at level of 10^{-7} and 10^{-10}, respectively. An important result is that the algorithm needs only 526 iterations to achieve such high accuracy with the matrix $\tilde{\mathbf{H}}_2$, while it needs at least 10000 iterations for the problem with a quadratic constraint over the matrix $\mathbf{H}_1$.

The results imply that when the matrix involved in the quadratic constraint has a large condition number, the optimisation becomes hard, and the algorithm demands a large number of iterations. It even can converge to a local minimum if the step size γ is not chosen properly. This is clear because the QCQP conversion requires the Cholesky decomposition of an ill-conditioned matrix. For this case, generating a new psd matrix with a lower condition number is suggested to replace the one of ill condition. An alternative method is to run the algorithm with a relatively higher step size. For example, when running the algorithm with $\gamma = 100$, the algorithm converges to the (global) minimum, but it needs 200.000 iterations as illustrated in Fig. 8.

We note that the optimization problem in (20) can be solved using the interior-point algorithm. We verify this method in three optimisation problems, each corresponding to a matrix $\mathbf{H}_i$. The results show that the method converges twice to a false local minimum with an objective value of 0.0017.

Example 6 (**Effect of the step size γ**)

As shown in the previous example, a large step size γ can be useful when the QCQP problem is hard. In this example, we use the same matrices as in Example 5, and compare convergence behaviour of the proposed algorithm when γ is varied
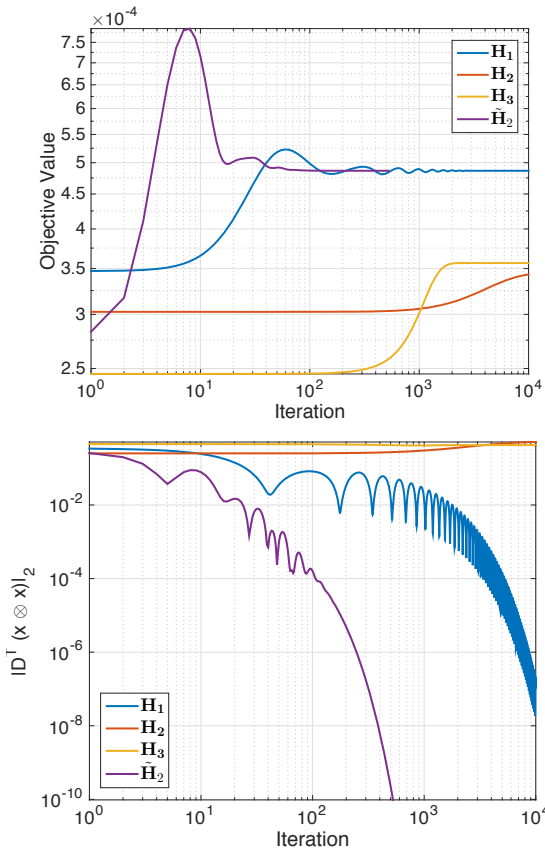
Fig. 7 Illustration of performances of the proposed algorithm in Example 5. The algorithm demands a huge number of iterations, but converges to local minima, when the quadratic constraint $x^T H x = 1$ is constrained over the matrix H_2 or H_3. The algorithm quickly converges when the quadratic constraint is accompanied by the matrix $\tilde{H}_2$.
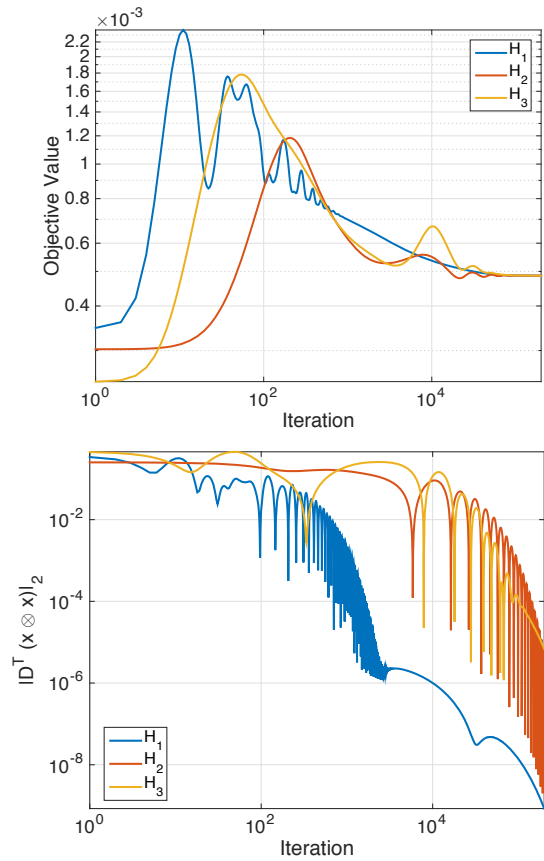

Fig. 8 Illustration of performances of the proposed algorithm in Example 5. The algorithm converges slowly after 200000 iterations when running with a high step size $\gamma = 100$.

in the range of $[0.001, 1000]$. We plot the objective function values, i.e., $x^T Q x$ to illustrate the convergence of the proposed algorithm, and the ℓ_2-norms of the orthogonality constraints on $x \otimes x$, i.e., $\|D^T(x \otimes x)\|_2$ in Fig. 9. A relatively large step size, e.g., $\gamma = 10, 100, 1000$, enforces the orthogonality constraints on the parameter vector quickly, but it reduces the objective value slowly; hence, it slows down the overall convergence. However, a very small step size, e.g., $\gamma = 0.01$, may make the algorithm diverge, while the orthogonality constraints cannot be enforced on $x \otimes x$. Selection of an appropriate step size affects the overall convergence. Fortunately, we can choose the step sizes in quite a wide range. In this example, $\gamma = 0.01$ is possibly the best selection, but $\gamma = 1$ and 10 are also good choices, although the algorithm may need a more iterations.

Example 7 (**Performance of the linearisation method**)

In this example, we verify performance of Algorithm 3, but x is updated using the linearization method in Section 6.5. The parameters are initialised as in the previous examples. The step-size μ is set to the step size γ, and is varied in the

same range as in Example 6. The objective values and norm of the constraints are plotted in Fig. 10. Compared to the results shown in Fig. 9, there is not much difference in the convergence of the proposed algorithm using the two update rules for x.

Example 8

In this example, we present results from 100 simulations with a similar settings to those in Example 5. In each run, the matrices are randomly generated. The step size γ is set to $\alpha \kappa_H$, where α is in a range of 10^{-4} to 10^4, and κ_H denotes the smallest condition number of the matrices H_m. We verify performance of the constrained QP problems in which the matrix H is generated to have a minimum condition number. In addition, we compare the performance of the proposed algorithm with those using the interior-point algorithm for constrained nonlinear minimization. In order to assess the performance, we compute the relative objective errors, i.e., a relative error between the objective value and the best (smallest) objective value among all objective values obtained by the considered methods in each run, error of the constraints, and the number of iterations. Fig. 11 shows empirical cumulative distribution functions of the measures
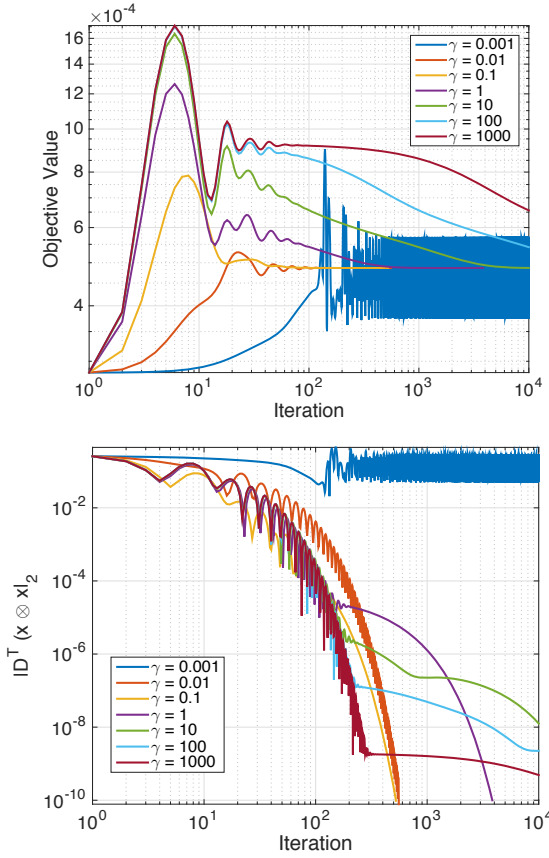
Fig. 9 Performance of the algorithm in Example 5 over a range of γ. A relatively large γ enforces the orthogonality constraints quickly, but varies the objective value slowly; hence, it slows down the overall convergence. A very small γ makes the algorithm diverge.

Fig. 10 Performance of the proposed algorithm using the linearisation update rule for x in Example 5 for a range of values of γ. A relatively large value of γ enforces the orthogonality constraints quickly, but varies the objective value slowly; hence, it slows down the overall convergence. A very small γ makes the algorithm diverge.

for two cases, with and without minimisation of the condition number. A setting of γ is considered good, if the algorithm achieves a small relative error, e.g., less than 10^{-3}, and a small constraint error, e.g., $\leq 10^{-8}$.

As shown in Fig. 11, when the matrices $\tilde{\mathbf{H}}$ have minimum condition numbers, the algorithm achieves good results with small relative errors, less than 10^{-4}, with $\alpha = \{10^{-4}, 10^{-3}, \dots, 1\}$. Fig. 11(b-center) shows that in some runs the outcome vectors x may not satisfy the constraints when $\alpha = 10^{-4}$ and 10^{-3}. With the settings $\alpha = \{0.01, 0.1, 1\}$, the algorithm not only converges to the desired solution but also requires a fewer iterations, especially when $\alpha = 0.01$. We note that when setting α to high values, e.g., ≥ 10, the small constraint errors indicate that the outcome satisfies the constraints, but the algorithm does not converge to the global minimum within a predefined 100000 iterations, e.g., $\alpha = 10$ and 100, or it stops because the objective function does not appear to improve significantly, e.g., for $\alpha \geq 1000$.

For the case without the correction of the condition number, although the algorithm converges with $\alpha = 10^{-3}, 10^{-2}$,

it often demands a huge number of iterations, as illustrated in Fig. 11(a-right).

Compared to the performance of the interior point algorithm (IP), the results indicate that the IP algorithm attains a convergence ratio of 75% to converge to the best solutions. The augmented Lagrangian algorithm with appropriate stepsizes, i.e., when $\alpha = \{0.001, 0.01, 0.1\}$, attains a convergence ratio of 89%.

7 Best Rank-1 Tensor Approximation to Symmetric Tensor of Order-4

We now present a novel application of the quadratic minimisation over a sphere to finding a best rank-1 tensor approximation of an order-4 symmetric tensor. The concept of the symmetric tensor is extended from the symmetric matrix, i.e., invariant under any permutation of its indices. Symmetric tensors can be cumulant tensors, or derivative tensors of the second Generalised Characteristic Functions

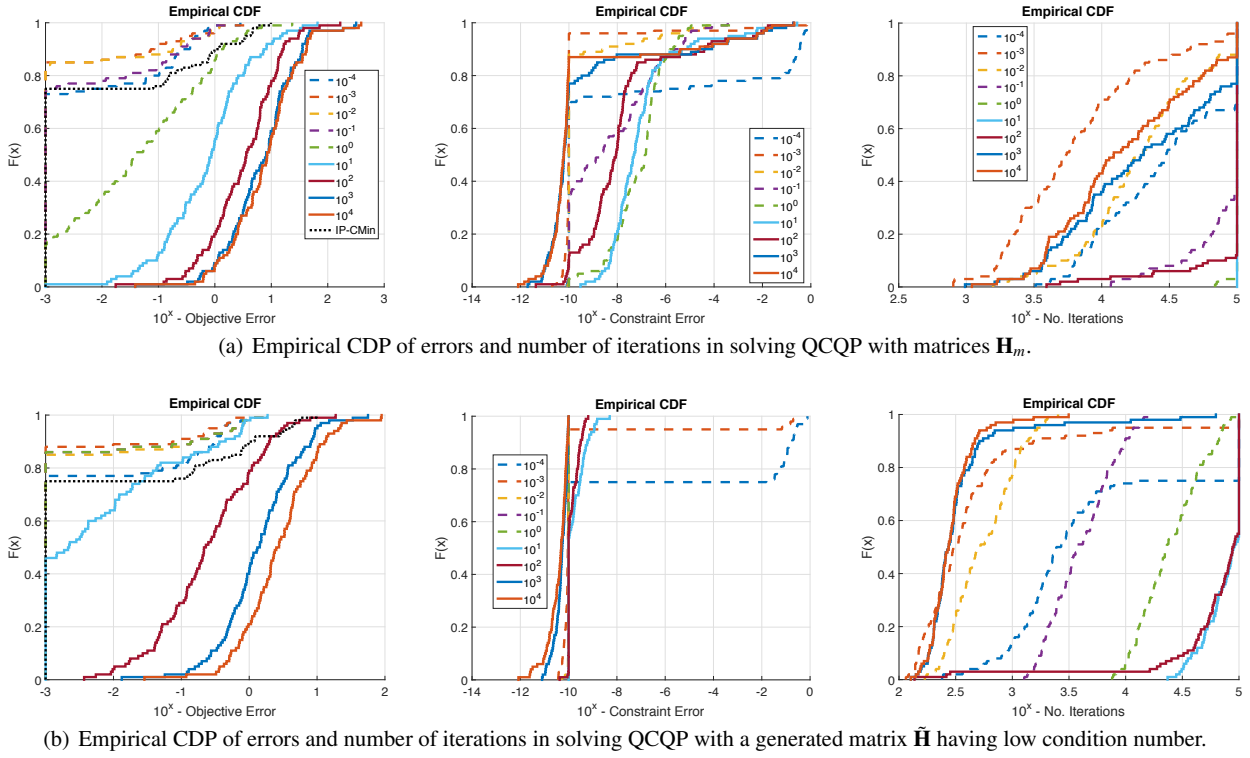(a) Empirical CDP of errors and number of iterations in solving QCQP with matrices $\mathbf{H}_m$.

(b) Empirical CDP of errors and number of iterations in solving QCQP with a generated matrix $\tilde{\mathbf{H}}$ having low condition number.

Fig. 11 Performance comparison in solving the QP with three quadratic constraints.

[1, 10, 20, 34], or tensors representing similarity or interaction between groups of identities used for clustering [24, 31].

We consider an order-4 tensor $\mathcal{Y}$ which is symmetric, i.e., $y(i_1, i_2, i_3, i_4) = y(j_1, j_2, j_3, j_4)$, where $[j_1, j_2, j_3, j_4]$ is any permutation of indices $[i_1, i_2, i_3, i_4]$. The best rank-1 tensor approximation to the tensor $\mathcal{Y}$ is to minimize the following approximation error

$$\min_{\lambda, \boldsymbol{x}} \qquad \|\mathcal{Y} - \lambda \boldsymbol{x} \circ \boldsymbol{x} \circ \boldsymbol{x} \circ \boldsymbol{x}\|_F^2 \qquad (29)$$

where $\lambda \boldsymbol{x} \circ \boldsymbol{x} \circ \boldsymbol{x} \circ \boldsymbol{x}$ represents the best rank-1 tensor to approximate $\mathcal{Y}$, and $\boldsymbol{x}$ is a unit-length vector, $\boldsymbol{x}^T \boldsymbol{x} = 1$. For shorthand notation, we denote $\boldsymbol{x} \circ \boldsymbol{x} \circ \boldsymbol{x} \circ \boldsymbol{x} = \boldsymbol{x}^{(4)}$. By expanding the Frobenious norm (29) as

$$\|\mathcal{Y} - \lambda \boldsymbol{x}^{(4)}\|_F^2 = \|\mathcal{Y}\|_F^2 + \lambda^2 - 2\lambda \langle \mathcal{Y}, \boldsymbol{x}^{(4)} \rangle$$

it is straightforward to see that the optimal weight $\lambda^\star$ is the inner product between the tensor $\mathcal{Y}$ and the rank-1 tensor $\boldsymbol{x}^{(4)} = \boldsymbol{x} \circ \boldsymbol{x} \circ \boldsymbol{x} \circ \boldsymbol{x}$, that is, $\lambda^\star = \langle \mathcal{Y}, \boldsymbol{x}^{(4)} \rangle$. Hence, the objective function is rewritten as

$$\min_{\boldsymbol{x}} \qquad \|\mathcal{Y}\|_F^2 - 2\left(\mathcal{Y} \bullet \boldsymbol{x}^{(4)}\right)^2.$$

For a positive λ, we maximise the inner product to give

$$\max \qquad \langle \mathcal{Y}, \boldsymbol{x}^{(4)} \rangle \quad \text{s.t.} \quad \boldsymbol{x}^T \boldsymbol{x} = 1$$

and minimise the inner product for a negative λ, that is

$$\min \qquad \langle \mathcal{Y}, \boldsymbol{x}^{(4)} \rangle \quad \text{s.t.} \quad \boldsymbol{x}^T \boldsymbol{x} = 1.$$

The final solution λ is that with the largest absolute value. Both problems can be solved on a Riemannian or Stiefel manifold using e.g., the Trust-Region solver [7]. Here, we propose another method to solve the two above problems.

Let $\boldsymbol{z} = \boldsymbol{x} \otimes \boldsymbol{x}$, then the minimisation problem in (30) becomes

$$\min \qquad \boldsymbol{z}^T \mathbf{Q} \boldsymbol{z} \quad \text{s.t.} \quad \boldsymbol{z} = \boldsymbol{x} \otimes \boldsymbol{x}, \quad \text{and} \quad \boldsymbol{z}^T \boldsymbol{z} = 1$$

where $\mathbf{Q}$ is a mode-(1,2) matricization of $\mathcal{Y}$.

The augmented Lagrangian function of the above problem now becomes

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = f(\boldsymbol{z}) + \boldsymbol{y}^T(\boldsymbol{z} - \boldsymbol{x} \otimes \boldsymbol{x}) + \frac{\gamma}{2}\|\boldsymbol{z} - \boldsymbol{x} \otimes \boldsymbol{x}\|^2 \qquad (30)$$

where $f(\boldsymbol{z})$ is the objective function of the minimization of $\boldsymbol{z}^T \mathbf{Q} \boldsymbol{z}$ subject to $\boldsymbol{z}^T \boldsymbol{z} = 1$. Variables $\boldsymbol{x}$, $\boldsymbol{z}$ and $\boldsymbol{y}$ are sequentially updated following the sequence

$$\boldsymbol{z} = \arg\min \quad f(\boldsymbol{z}) + \boldsymbol{y}^T(\boldsymbol{z} - \boldsymbol{x} \otimes \boldsymbol{x}) + \frac{\gamma}{2}\|\boldsymbol{z} - \boldsymbol{x} \otimes \boldsymbol{x}\|^2$$

$$= \arg\min \quad \frac{1}{2}\boldsymbol{z}^T \mathbf{Q} \boldsymbol{z} + (\boldsymbol{y} - \gamma \boldsymbol{x} \otimes \boldsymbol{x})^T \boldsymbol{z} \qquad (31)$$

$$\text{subject to} \quad \boldsymbol{z}^T \boldsymbol{z} = 1$$

$$\boldsymbol{x} = \arg\min \quad \boldsymbol{y}^T(\boldsymbol{z} - \boldsymbol{x} \otimes \boldsymbol{x}) + \frac{\gamma}{2}\|\boldsymbol{z} - \boldsymbol{x} \otimes \boldsymbol{x}\|^2$$

$$= \arg\min \quad \|\boldsymbol{z} + \frac{\boldsymbol{y}}{\gamma} - \boldsymbol{x} \otimes \boldsymbol{x}\|^2 \qquad (32)$$

$$\boldsymbol{y} \leftarrow \boldsymbol{y} + \gamma(\boldsymbol{z} - \boldsymbol{x} \otimes \boldsymbol{x}). \qquad (33)$$

The unit-length vector $\boldsymbol{z}$ is a minimiser to a SCQP, whereas $\boldsymbol{x}$ is the eigenvector associated with the largest eigenvalue of

the symmetric matrix $\mathbf{T}_s = \frac{1}{2}(\mathbf{T} + \mathbf{T}^T)$, where

$$\mathbf{T} = \mathbf{Z} + \frac{1}{\gamma}\mathbf{Y},$$

$z = \text{vec}(\mathbf{Z})$ and $y = \text{vec}(\mathbf{Y})$.

Example 9 (**Best rank-1 tensor approximation to a symmetric tensor of order-4**)

This example compares performance of our algorithm for best rank-1 tensor approximation for symmetric tensor of order-4, and the Riemannian trust-region solver in the Manopt toolbox [7]. We generate 1000 random tensors of size $I \times I \times I \times I$, where $I = 10$ or 20, then matricize them so that they will become symmetric tensors of order-4. The tensors are normalized to have unit Frobenius norm. For each run, the best approximation error $\varepsilon^\star$ is defined as the smallest error among approximation errors of the two methods: Augmented Lagrangian method and the Riemannian trust-region

$$\varepsilon = \|\mathcal{Y} - \lambda \, \boldsymbol{x}^{(4)}\|_F^2 = 1 - \lambda^2 \,.$$

Relative errors to the best approximation error $\frac{\varepsilon - \varepsilon^\star}{\varepsilon^\star}$ is then assessed to measure performance of the approximation.

Fig. 12 shows the empirical cumulative distribution functions of 1000 relative errors. The results indicate that our algorithm based on the quadratic optimisation over sphere achieves a higher success rate. For example, for the case when $I = 20$, our algorithm attains an error less than 0.001 with a rate of 96.8%, whereas the trust-region solver achieves a rate of 73.1% for the same error range. When $I = 10$, the Augmented Lagrangian algorithm has a success rate of 92.5% for a similar accuracy of 0.001, while the trust-region algorithm has a quite low rate of 47.4%.

8 Generalized Eigenvalue Decomposition with Eigen matrix of low rank structure

We now address a constrained generalised eigenvalue decomposition which exploits the QCQP to derive an algorithm. The considered problem is stated below.

Definition 3 (GEVD with eigen matrix having a low rank structure) Consider a positive semi-definite matrix $\mathbf{Q}$ of size $IJ \times IJ$ and a positive definite matrix $\mathbf{S}$ of size $IJ \times IJ$. We solve the following optimisation problem

$$\min \quad \text{tr}(\mathbf{X}^T \mathbf{Q} \mathbf{X}) \quad \text{s.t.} \quad \mathbf{X}^T \mathbf{B} \mathbf{X} = \mathbf{I}_R$$

to find a matrix $\mathbf{X}$ of size $IJ \times R$, where each column of $\mathbf{X}$ is a vectorisation of a product of two matrices

$$\boldsymbol{x}_r = \text{vec}\left(\mathbf{G}_r \mathbf{A}^T\right) \tag{34}$$

and $\mathbf{G}_r$ are matrices of size $I \times S$ and $\mathbf{A}$ is of size $J \times S$.
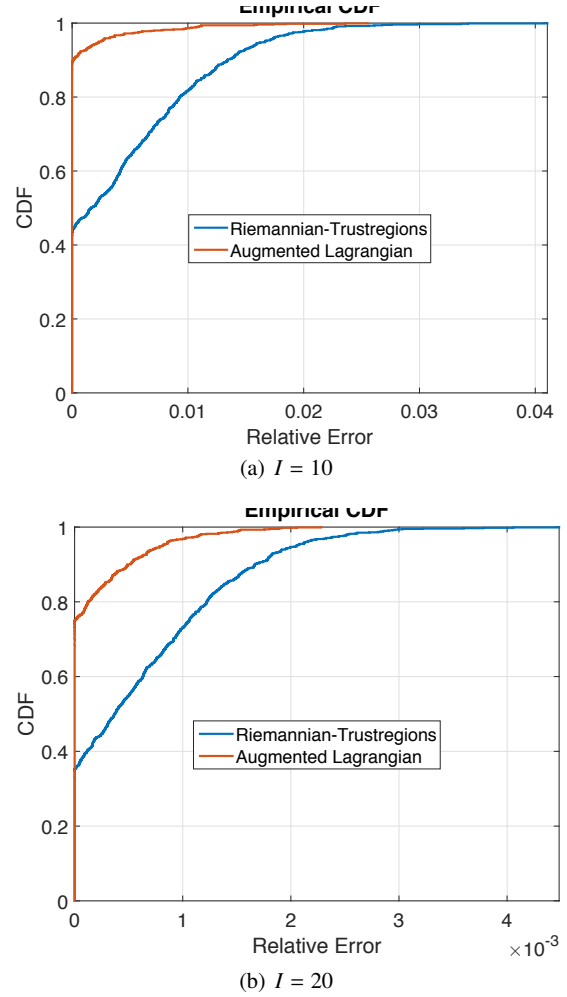

(a) $I = 10$

(b) $I = 20$

Fig. 12 The empirical cumulative distribution functions of the relative errors of two algorithms based on augmented Lagrangian and Riemannian trust-region methods.

If we concatenate the matrices $\mathbf{G}_r$ into an order-3 tensor of size $I \times S \times R$, the factor matrix $\mathbf{X}$ is a mode-(1,2) matricization of an order-3 tensor $\mathcal{X}$ of size $I \times J \times R$, defined as

$$\mathcal{X} = \mathcal{G} \times_2 \mathbf{A} \,.$$

Because of scaling and rotation ambiguities, the matrix $\mathbf{A}$ can always be normalized to be an orthogonal matrix. However, we do not exploit the orthogonality constraint on $\mathbf{A}$ in its estimation, but perform orthogonal normalisation after each update. With the above interpretation, the matrix of eigenvectors, $\mathbf{X}$, is considered a block matrix in the tensor train format of only two cores. For the GEVD in which the matrix $\mathbf{X}$ is a block TT-matrix composed from more cores, the problem in (3) becomes a local problem in an alternating algorithm to estimate the cores. The problem in (34) is a constrained GEVD.

For this simple case, we can express the factor matrix as $\mathbf{X} = [\mathcal{X}]_{(1,2)} = [\mathcal{X}]_{(3)}^T$, and

$$\mathbf{X} = (\mathbf{A} \otimes \mathbf{I}_I)\mathbf{G} \tag{35}$$

where $\mathbf{G} = [\text{vec}(\mathbf{G}_1), \ldots, \text{vec}(\mathbf{G}_R)]$ and

$$\mathbf{X} = [\mathbf{I}_J \otimes \mathbf{G}_1, \ldots, \mathbf{I}_J \otimes \mathbf{G}_R] \left(\mathbf{I}_R \otimes \text{vec}(\mathbf{A}^T)\right).$$

We now show that $\mathcal{G}$ can be estimated using a GEVD, and $\mathbf{A}$ is a solution to a quadratic programming problem with quadratic constraints. Our proposed algorithm alternates the estimation of $\mathcal{G}$ and $\mathbf{A}$.

8.1 Update of $\mathbf{G}_r$

By exploiting the expression in (35), while fixing the matrix $\mathbf{A}$, we can find $\mathbf{G}$ in a GEVD, given by

$$\min \quad \text{tr}(\mathbf{G}^T \mathbf{Q}_G \mathbf{G}) \quad \text{s.t.} \quad \mathbf{G}^T \mathbf{B}_G \mathbf{G} = \mathbf{I}_R$$

where

$$\mathbf{Q}_G = (\mathbf{A}^T \otimes \mathbf{I}_I) \mathbf{Q} (\mathbf{A} \otimes \mathbf{I}_I),$$
$$\mathbf{B}_G = (\mathbf{A}^T \otimes \mathbf{I}_I) \mathbf{B} (\mathbf{A} \otimes \mathbf{I}_I).$$

8.2 Update of $\mathbf{A}$

In order to derive the update rule for $\mathbf{A}$, from (36), we can rewritte the objective function as

$$\text{tr}(\mathbf{X}^T \mathbf{Q} \mathbf{X}) = \sum_{r=1}^{R} \mathbf{x}_r^T \mathbf{Q} \mathbf{x}_r$$
$$= \sum_{r=1}^{R} \text{vec}(\mathbf{A}^T)^T (\mathbf{I}_J \otimes \mathbf{G}_r)^T \mathbf{Q} (\mathbf{I}_J \otimes \mathbf{G}_r) \text{vec}(\mathbf{A}^T)$$
$$= \text{vec}(\mathbf{A}^T)^T \mathbf{Q}_A \text{vec}(\mathbf{A}^T) \tag{36}$$

where

$$\mathbf{Q}_A = \sum_{r=1}^{R} (\mathbf{I}_J \otimes \mathbf{G}_r)^T \mathbf{Q} (\mathbf{I}_J \otimes \mathbf{G}_r).$$

Similarly, the quadratic constraint is rewritten for each pair of columns $\mathbf{x}_r$ and $\mathbf{x}_s$ as

$$\delta_{r,s} = \mathbf{x}_r^T \mathbf{B} \mathbf{x}_s$$
$$= \text{vec}(\mathbf{A}^T)^T (\mathbf{I}_J \otimes \mathbf{G}_r)^T \mathbf{B} (\mathbf{I}_J \otimes \mathbf{G}_s) \text{vec}(\mathbf{A}^T)$$
$$= \text{vec}(\mathbf{A}^T)^T \mathbf{B}_{r,s} \text{vec}(\mathbf{A}^T). \tag{37}$$

for $r, s = 1, \ldots, R$, where

$$\mathbf{B}_{r,s} = (\mathbf{I}_J \otimes \mathbf{G}_r)^T \mathbf{B} (\mathbf{I}_J \otimes \mathbf{G}_s). \tag{38}$$

As a result of (36) and (37), the vector $\text{vec}(\mathbf{A}^T)$ is a minimiser of a quadratic programming problem with $R(R+1)/2$ quadratic constraints

$$\min \quad \text{vec}(\mathbf{A}^T)^T \mathbf{Q}_A \text{vec}(\mathbf{A}^T)$$
$$\text{subject to} \quad \text{vec}(\mathbf{A}^T)^T \mathbf{B}_{r,r} \text{vec}(\mathbf{A}^T) = 1, \quad r = 1, \ldots, R$$
$$\text{vec}(\mathbf{A}^T)^T \mathbf{B}_{r,s} \text{vec}(\mathbf{A}^T) = 0, \quad 1 \le s < r \le R.$$

Following the method in Section 6.6, we can generate a matrix $\mathbf{B}$ with a low condition number from the matrices $\mathbf{B}_{r,s}$, or choose a matrix with the smallest condition number among them, e.g., $\mathbf{B}_{1,1}$. Denote by $\mathbf{F}$ the factor matrix in the Cholesky decomposition of the matrix $\mathbf{B}_{1,1} = \mathbf{F}\mathbf{F}^T$, and introduce the following symmetric matrices of size $JS \times JS$

$$\mathbf{D}_{r,r} = \mathbf{F}^{-1}(\mathbf{B}_{1,1} - \mathbf{B}_{r,r})\mathbf{F}^{-1T}, \quad r = 2, \ldots, R$$
$$\mathbf{D}_{r,s} = \mathbf{F}^{-1}(\mathbf{B}_{r,s} + \mathbf{B}_{r,s}^T)\mathbf{F}^{-1T}, \quad 1 \le s < r \le R$$

and

$$\tilde{\mathbf{Q}}_A = \mathbf{F}^{-1} \mathbf{Q}_A \mathbf{F}^{-1T}.$$

We can then find $\boldsymbol{a} = \mathbf{F}^T \text{vec}(\mathbf{A}^T)$ in the following constrained optimization based on Algorithm 3

$$\min \quad \boldsymbol{a}^T \tilde{\mathbf{Q}}_A \boldsymbol{a}$$
$$\text{subject to} \quad \boldsymbol{a}^T \boldsymbol{a} = 1,$$
$$\boldsymbol{a}^T \mathbf{D}_{r,r} \boldsymbol{a} = 0, \quad r = 2, \ldots, R,$$
$$\boldsymbol{a}^T \mathbf{D}_{r,s} \boldsymbol{a} = 0, \quad 1 \le s < r \le R.$$

Example 10 (**Discriminant analysis of hand-written digits**)

In this example, we illustrate an application of the proposed algorithm for solving the constrained GEVD in (34). More specifically, we perform a discriminant analysis on the training samples, which comprise handwritten images for digits 0, 1 and 2. The data is taken from the MNIST dataset. Images are of size 28×28, and their Gabor features are computed for 8 orientations and 4 scales. The Gabor images are scaled down to size 16×16, then vectorized and concatenated into a matrix of size 256×32. All digit images construct an order-3 tensor of size $256 \times 32 \times 900$, 300 images for each digit. Ten random 10-fold cross-validations are performed on 900 samples: 810 for training and 90 for the test set.

Denote the matrix of training samples by $\mathbf{Y}_{tr}$, which is of size 8192×810. We seek a projection matrix $\mathbf{X}$ of size 8192×2 to extract 2 feature vectors $\mathbf{F} = \mathbf{Y}_{tr}^T \mathbf{X}$ which maximises the Fisher score, a ratio of the between and with-in distances

$$\max \quad \frac{\text{tr}(\mathbf{F}^T \mathbf{S}_b \mathbf{F})}{\text{tr}(\mathbf{F}^T \mathbf{S}_w \mathbf{F})}$$

where $\mathbf{S}_b$ and $\mathbf{S}_w$ are the between and with-in scattering matrices constructed for the training samples. Alternating to the maximisation of the trace-ratio, we solve the GEVD problem

$$\max \quad \text{tr}(\mathbf{X}^T \mathbf{Y}_{tr} \mathbf{S}_b \mathbf{Y}_{tr}^T \mathbf{X}),$$
$$\text{subject to} \quad \mathbf{X}^T \mathbf{Y}_{tr} \mathbf{S}_w \mathbf{Y}_{tr}^T \mathbf{X} = \mathbf{I}_2.$$

Because each digit is represented by a vector of length $8192 (= 256 \times 32)$, which exceeds the number training samples of 810, the above ordinary linear discriminant analysis often leads to over-fitting, and it is not applicable. To this
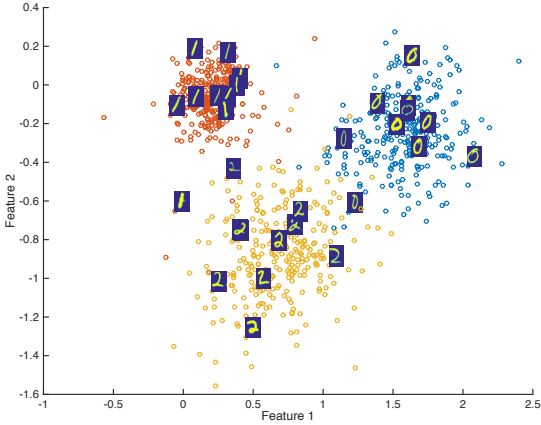
Fig. 13 Scattering plot of hand-written digits using two extracted features in Example 10.

end, we apply the constrained GEVD in (34). Columns of the matrix $\mathbf{X}$ are constrained with a structure

$$\boldsymbol{x}_r = \text{vec}\left(\mathbf{G}_r \mathbf{A}^T\right).$$

In our example, $\mathbf{G}_1$ and $\mathbf{G}_2$ are of size 256×2 and $\mathbf{A}$ is of size 32×2. The extracted features for training samples are computed and used to train a simple LDA classifier. Using the proposed algorithm, we obtain a classification accuracy of 97.36% averaged over 10×10-fold cross-validations.

Fig 13 shows the scatter plot of samples plotted using the two feature vectors, demonstrating that digits 0, 1 and 2 are distinguished.

9 Conclusions

We have introduced a robust solution to the SCQP problem by imposing an error bound on the root of the underlying secular equation. The method has been initially derived as SCQP for matrix variate data, together with the related linear regression with an error bound constraint. In addition, we proposed an algorithm for the QCQP problem which treats QCQP as SCQP and an orthogonal projection. In the process, the quadratic term within the quadratic constraint is correctted by a term with a minimum condition number. This correction method has been shown to improve the convergence of the proposed algorithm. Applications of the SCQP and QCQP have been presented for image deconvolution, tensor decomposition and constrained GEVD.

A Proof of Lemma 1

Proof We consider a simple case when some eigenvalues are identical, e.g., $s_1 = s_2 = \cdots = s_L < s_{L+1} < \cdots < s_K$. If $\boldsymbol{c}_{1:L}$ are all zeros, the objective function is independent of $\tilde{\boldsymbol{x}}_{1:L} = [\tilde{x}_1, \tilde{x}_2 \dots, \tilde{x}_L]$. Hence, $\tilde{\boldsymbol{x}}_{1:L}$ can be any point on the ball $\|\tilde{\boldsymbol{x}}_{1:L}\|^2 = d^2 = 1 - \sum_{k=L+1}^{K} \tilde{x}_k^2$. Otherwise,

$\tilde{\boldsymbol{x}}_{1:L}$ is a minimiser to a constrained linear programming while fixing the other parameters $\tilde{x}_{L+1}, \dots, \tilde{x}_K$, in the form

$$\min \quad \boldsymbol{c}_{1:L}^T \tilde{\boldsymbol{x}}_{1:L} \quad \text{subject to} \quad \|\tilde{\boldsymbol{x}}_{1:L}\| = d$$

which yields

$$\tilde{\boldsymbol{x}}_{1:L} = \frac{-d}{\|\boldsymbol{c}_{1:L}\|} \boldsymbol{c}_{1:L}.$$

For both cases, we can define

$$\boldsymbol{z} = [-d, \tilde{x}_{L+1}, \dots, \tilde{x}_K],$$
$$\tilde{\boldsymbol{c}} = [\|\boldsymbol{c}_{1:L}\|, c_{L+1}, \dots, c_K],$$
$$\tilde{\boldsymbol{s}} = [s_1, s_{L+1}, \dots, s_K],$$

and perform a reparameterization to estimate $\boldsymbol{z}$ from a similar SCQP but with distinct eigenvalues $\tilde{\boldsymbol{s}}$, as

$$\min \quad \frac{1}{2} \boldsymbol{z}^T \text{diag}(\tilde{\boldsymbol{s}}) \boldsymbol{z} + \tilde{\boldsymbol{c}}^T \boldsymbol{z} \quad \text{subject to} \quad \boldsymbol{z}^T \boldsymbol{z} = 1.$$

B Proof of Lemma 2

Proof It follows from the second derivative of $f(\lambda)$, given by

$$f''(\lambda) = -2 \sum_k \frac{c_k^2}{(s_k - \lambda)^3}$$

that $f''(\lambda) < 0$ for all $\lambda < s_1 = 1$. That is, $f'(\lambda)$ monotonically decreases with $\lambda < s_1 = 1$.

In addition, since $s_k \geq 1$, for all k, we have

$$f'(0) = 1 - \sum_{k=1}^{K} \frac{c_k^2}{s_k^2} \geq 1 - \sum_{k=1}^{K} c_k^2 = 0$$

and

$$f'(1 - |c_1|) = 1 - \sum_{k=1}^{K} \frac{c_k^2}{(1 - |c_1| - s_k)^2} = 1 - \frac{c_1^2}{c_1^2} - \sum_{k=2}^{K} \frac{c_k^2}{(1 - |c_1| - s_k)^2}$$

$$\leq - \sum_{k=2}^{K} \frac{c_k^2}{(1 - |c_1| - s_k)^2} \leq 0.$$

This implies that $f'(\lambda)$ has a unique root smaller than 1. Moreover, the root lies in the interval $[0, 1 - |c_1|)$.

C Proof of Lemma 3

Proof Let λ_1 be a root which is smaller than $s_1 = 1$, and λ_2 be another root of $f'(\lambda)$. Then, according to Lemma 2, $\lambda_2 > 1 > \lambda_1$, and

$$\sum_{k=1}^{K} \frac{c_k^2}{(\lambda_1 - s_k)^2} = \sum_{k=1}^{K} \frac{c_k^2}{(\lambda_2 - s_k)^2} = 1. \tag{39}$$

It can be shown that

$$f(\lambda_2) - f(\lambda_1) = \lambda_2 - \lambda_1 + \sum_{k=1}^{K} \frac{c_k^2}{\lambda_2 - s_k} - \frac{c_k^2}{\lambda_1 - s_k}$$

$$= (\lambda_2 - \lambda_1)\left(1 - \sum_{k=1}^{K} \frac{|c_k|}{s_k - \lambda_1} \frac{|c_k|}{s_k - \lambda_2}\right)$$

$$\geq (\lambda_2 - \lambda_1)\left(1 - \sqrt{\sum_{k=1}^{K} \frac{c_k^2}{(\lambda_1 - s_k)^2}} \sqrt{\sum_{k=1}^{K} \frac{c_k^2}{(\lambda_2 - s_k)^2}}\right)$$

$$= (\lambda_2 - \lambda_1)(1 - 1 \times 1) = 0. \tag{40}$$

This inequality is obtained by applying the Cauchy-Schwarz inequality, whereas (40) is obtained after replacing the optimal conditions in (39). The equality case does not occur because of $s_1 - \lambda_2 < 0$, that is, $f(\lambda_2) > f(\lambda_1)$ and the minimiser $\lambda^\star$ of $f(\lambda)$ is the minimum root λ_1 of $f'(\lambda)$.

D Proof of Lemma 4

Proof We first show that the polynomials $p_i(t)$ have unique roots in $[|c_1|, 1]$. The second-derivative of $p_i(t)$ is given by

$$p_i''(t) = 12t^2 + 12d_i t + 2(d_i^2 - 1)$$

and has two roots $\bar{t}_{1,2} = \dfrac{-3d_i \mp \sqrt{3d_i^2 + 6}}{6}$.

If $d_i > 1$, the roots $\bar{t}_{1,2}$ are negative. Hence, the first derivative $p_i'(t)$ monotonically increases in $[0, +\infty)$. In addition, since

$$p_i'(0) = -2c_1^2 d_i \leq 0$$

$p_i'(t)$ has only one root in $[0, +\infty)$. Together with the fact that

$$p_i(0) = -c_1^2 d_i^2 \leq 0$$
$$p_i(|c_1|) = c_1^2(c_1^2 - 1) \leq 0$$
$$p_i(1) = d_i(d_i + 2)(1 - c_1^2) \geq 0,$$

the polynomial $p_i(t)$ has a unique root in $[|c_1|, 1]$.

If $d_i \leq 1$, the second root $\bar{t}_2$ is non-negative, $\bar{t}_2 \geq 0$. However, since $p_i'(0) = -2c_1^2 d_i \leq 0$, the first derivative $p_i'(t)$ has only one root in $[0, +\infty)$. Again as for the case $d_1 > 1$, the polynomial $p_i(t)$ also has unique root in $[|c_1|, 1]$.

As the definition of the root t_2, we can prove that derivative $f'(s_1 - t_2)$ does not exceed zero, that is

$$f'(s_1 - t_2) = 1 - \frac{c_1^2}{t_2^2} - \sum_{k=2}^{K} \frac{c_k^2}{(s_k - s_1 + t_2)^2} \leq 1 - \frac{c_1^2}{t_2^2} - \frac{\sum_{k=2}^{K} c_k^2}{(s_K - s_1 + t_2)^2}$$

$$= 1 - \frac{c_1^2}{t_2^2} - \frac{1 - c_1^2}{(d_2 + t_2)^2} = \frac{p_2(t_2)}{t_2^2(d_2 + t_2)^2}$$

$$= 0. \tag{41}$$

Similarly, we have

$$f'(s_1 - t_1) = 1 - \frac{c_1^2}{t_1^2} - \sum_{k=2}^{K} \frac{c_k^2}{(s_k - s_1 + t_1)^2} \geq 1 - \frac{c_1^2}{t_1^2} - \frac{\sum_{k=2}^{K} c_k^2}{(d_1 + t_1)^2}$$

$$= 1 - \frac{c_1^2}{t_1^2} - \frac{1 - c_1^2}{(d_1 + t_1)^2} = \frac{p_1(t_1)}{t_1^2(d_1 + t_1)^2}$$

$$= 0. \tag{42}$$

From (41) and (42), it follows that $f'(t)$ has a root in $[1 - t_1, 1 - t_2]$. This root is unique and also the global minimiser of $f(\lambda)$ in (5). This completes the proof.

E Proof of Lemma 5

Proof First, similar to Lemma 2, the roots $\lambda_{l,L}^\star$ and $\lambda_{u,L}^\star$ are unique in the interval $[0, 1 - |c_1|]$. Taking into account that $\sum_{k=1}^{K} c_k^2 = 1$, and $s_K \geq s_k$ for all k, we have

$$f'(\lambda) = 1 - \sum_{l=1}^{L} \frac{c_l^2}{(s_l - \lambda)^2} - \sum_{k=L+1}^{K} \frac{c_k^2}{(s_k - \lambda)^2}$$

$$\leq 1 - \sum_{l=1}^{L} \frac{c_l^2}{(s_l - \lambda)^2} - \frac{\sum_{k=L+1}^{K} c_k^2}{(s_K - \lambda)^2} = 1 - \sum_{l=1}^{L} \frac{c_l^2}{(s_l - \lambda)^2} - \frac{\tilde{c}_{L+1}^2}{(s_K - \lambda)^2}$$

$$= f_u^{(L)}(\lambda).$$

Similarly, we can derive $f'(\lambda) \geq f_l^{(L)}(\lambda)$. It appears that the function values of $f'(\lambda)$ at $\lambda_{l,L}^\star$ and $\lambda_{u,L}^\star$ are nonnegative and non-positive, respectively,

$$f'(\lambda_{l,L}^\star) \geq f_l^{(L)}(\lambda_{l,L}^\star) = 0,$$
$$f'(\lambda_{u,L}^\star) \leq f_u^{(L)}(\lambda_{u,L}^\star) = 0,$$

thus implying that

$$\lambda_{l,L}^\star \leq \lambda^\star \leq \lambda_{u,L}^\star.$$

The sequence of inequalities in (6) can be proved in a similar way.

F Proof of Lemma 6

Proof By contradiction, assume that the variable $\tilde{x}_n^\star$ is non-zero. Since there is only one $c_n = 0$, from (3), the multiplier $\lambda^\star$ must be equal to s_n, that is

$$\lambda^\star = s_n,$$

and the minimiser $\tilde{x}^\star$ is given by

$$\tilde{x}_k^\star = \frac{c_k}{s_n - s_k}, \quad k \neq n$$

and $\tilde{x}_n^\star$ is derived from the unit-length condition of $\tilde{x}^\star$

$$\tilde{x}_n^2 = 1 - \sum_{k \neq n} \tilde{x}_k^2$$

with an additional assumption that

$$\sum_{k \neq n} \frac{c_k^2}{(s_n - s_k)^2} < 1.$$

The objective function in (2) at $\tilde{x}^\star$, as well as the Lagrangian function at $(\tilde{x}^\star, \lambda^\star = s_n)$ are given by

$$\mathcal{L}(\tilde{x}^\star, \lambda^\star) = \frac{1}{2}\left(s_n - s_n \sum_{k \neq n} \frac{c_k^2}{(s_n - s_k)^2} + \sum_{k \neq n} \frac{c_k^2 s_k}{(s_n - s_k)^2}\right) + \sum_{k \neq n} \frac{c_k^2}{s_n - s_k}$$

$$= \frac{1}{2}\left(s_n + \sum_{k \neq n} \frac{c_k^2}{s_n - s_k}\right). \tag{43}$$

Now, we consider a vector $\bar{x}$ whose n-th entry is zero, $\bar{x}_n = 0$, and the rest $(K - 1)$ coefficients $\bar{x}_n = [\bar{x}_1, \ldots, \bar{x}_{n-1}, \bar{x}_{n+1}, \ldots, \bar{x}_K]$ are minimiser to a reduced problem

$$\min \quad \frac{1}{2}\sum_{k \neq n} s_k \tilde{x}_k^2 + \sum_{k \neq n} c_k \tilde{x}_k$$

$$\text{subject to} \quad \sum_{k \neq n} \tilde{x}_k^2 = 1.$$

According to the results in Section 2.2, when c_k, $k \neq n$, are non-zeros, the Lagrangian function for this reduced problem at the minimiser $\bar{x}_n$ is given by

$$\mathcal{L}_n(\bar{x}_n, \lambda_n^\star) = \frac{1}{2}\left(\lambda_n^\star + \sum_{k \neq n} \frac{c_k^2}{\lambda_n^\star - s_k}\right), \tag{44}$$

where the optimal multiplier $\lambda_n^\star < s_1 = 1$. From (43) and (44), it is apparent that

$$\mathcal{L}(\tilde{x}^\star, \lambda^\star) > \mathcal{L}_n(\bar{x}_n, \lambda_n^\star) = \mathcal{L}(\bar{x}, \lambda_n^\star),$$

which contradicts with the claim that $\tilde{x}^\star$ is the minimiser to the problem (2). This implies that the n-th variable of the minimiser must be zero, i.e., $\tilde{x}_n^\star = 0$.

G Proof of Lemma 7

Proof When $c_1 = 0$, from the first optimality condition in (3), we have

$$(s_1 - \lambda)\, \tilde{x}_1 = 0\,.$$

Assume that $\tilde{x}^\star$ is a minimiser to the problem in (2) with a non-zero $\tilde{x}_1^\star$, then $\lambda = s_1 = 1$ and

$$\tilde{x}_k^\star = \frac{c_k}{\lambda - s_k} = \frac{c_k}{1 - s_k}$$

for $k > 1$. From the unit-length constraint, it follows that $\tilde{x}_1$ can be deduced as

$$(\tilde{x}_1^\star)^2 = 1 - \sum_{k>1}(\tilde{x}_k^\star)^2 = 1 - d$$

which requires the condition $d \leq 1$. Implying that, if $d > 1$, $\tilde{x}_1^\star$ must be zero, and the rest $(K-1)$ variables $[\tilde{x}_2^\star, \ldots, \tilde{x}_K^\star]$ are minimiser to the reduced problem of (2).

When $d \leq 1$, there exists $\tilde{x}_1^\star$, and the objective function at $\tilde{x}^\star$ is given by

$$\mathcal{L}(\tilde{x}^\star, s_1) = \frac{1}{2}\left(1 - \sum_{k>2}\frac{c_k^2}{s_k - 1}\right).$$

Now, we consider a vector $\tilde{x}$ whose $\tilde{x}_1 = 0$, and $\bar{x} = [\tilde{x}_2, \ldots, \tilde{x}_K]$ is a minimiser to the reduced problem (7). Similar to the analysis in Section 2.2, the objective function of the reduced problem (7) achieves a global minimum at the minimum root $\bar{\lambda}$ of the first derivative of the Lagrangian function

$$\mathcal{L}_1(\bar{x}, \bar{\lambda}) = \frac{1}{2}\left(\bar{\lambda} - \sum_{k>2}\frac{c_k^2}{s_k - \bar{\lambda}}\right),$$

where $\bar{\lambda}$ is smaller than s_2.

Since the second derivative of $\mathcal{L}_1(\bar{x}, \lambda)$ w.r.t. λ is negative for all $\lambda < s_2$, the function $\mathcal{L}_1(\bar{x}, \lambda)$ is concave in $(-\infty, s_2)$. It then follows that

$$\mathcal{L}(\tilde{x}^\star, s_1) < \mathcal{L}_1(\bar{x}, \bar{\lambda})\,,$$

and $\tilde{x}^\star$ is the global minimiser. Note that $\tilde{x}_1^\star$ can be $\sqrt{1-d}$ or $-\sqrt{1-d}$.

H Proof of Lemma 9

Proof Let $x^\star$ be a minimiser to the problem (16)

$$x^\star = \arg\min_x \quad \|x\|^2 \quad \text{s.t.} \quad \|y - \mathbf{A}x\| \leq \delta\,.$$

It is obvious that if there are zero entries in $x^\star$, we can omit columns of $\mathbf{A}$ corresponding to these entries, and the regression problem formulated for the remaining sub matrix of $\mathbf{A}$ has a non-zero minimiser. Hence, we can assume that entries of $x^\star$ are nonzeros.

Let $z = y - \sum_{k=2}^K a_k x_k^\star$, then $x_1^\star$ is a minimiser to the optimisation w.r.t. x_1, that is

$$x_1^\star = \arg\min_{x_1} \quad x_1^2 \quad \text{s.t.} \quad \|z - a_1 x_1\| \leq \delta\,. \tag{45}$$

The constraint function can be written as

$$c(x_1) = \|z - a_1 x_1\|^2 - \delta^2 = \|a_1\|^2 x_1^2 - 2(a_1^T z)\, x_1 + \|z\|^2 - \delta^2\,.$$

Since $c(x_1^\star) \leq 0$, $c(x_1)$ must have two roots t_- and t_+. Moreover, it is clear from (45) that $\|z\|^2 > \delta^2$ otherwise $x_1^\star = 0$. Hence, the two roots t_- and t_+ have the same signs because

$$t_- t_+ = \frac{\|z\|^2 - \delta^2}{\|a_1\|^2} > 0\,.$$

As a result, the minimiser to (45) must be one of the two roots, $x_1^\star = \min(|t_-|, |t_+|)$, and the inequality condition becomes the equality one.

References

1. de Almeida, A.L.F., Luciani, X., Stegeman, A., Comon, P.: CONFAC decomposition approach to blind identification of underdetermined mixtures based on generating function derivatives. IEEE Transactions on Signal Processing **60**(11), 5698–5713 (2012)

2. ApS, M.: The MOSEK optimization toolbox for MATLAB manual. Version 7.1 (Revision 28). (2015). URL http://docs.mosek.com/7.1/toolbox/index.html

3. Arima, N., Kim, S., Kojima, M.: A quadratically constrained quadratic optimization model for completely positive cone programming. SIAM Journal on Optimization **23**(4), 2320–2340 (2013). DOI 10.1137/120890636. URL http://dx.doi.org/10.1137/120890636

4. Bao, X., Sahinidis, N.V., Tawarmalani, M.: Semidefinite relaxations for quadratically constrained quadratic programming: A review and comparisons. Mathematical Programming **129**(1), 129 (2011). DOI 10.1007/s10107-011-0462-2.

5. Baron, D.P.: Quadratic programming with quadratic constraints. Naval Research Logistics Quarterly **19**(2), 253–260 (1972)

6. Ben-Tal, A., Teboulle, M.: Hidden convexity in some nonconvex quadratically constrained quadratic programming. Mathematical Programming **72**(1), 51–63 (1996). DOI 10.1007/BF02592331.

7. Boumal, N., Mishra, B., Absil, P.A., Sepulchre, R.: Manopt, a Matlab toolbox for optimization on manifolds. Journal of Machine Learning Research **15**, 1455–1459 (2014). URL http://www.manopt.org

8. Boyd, S., El Ghaoui, L., Feron, E., Balakrishnan, V.: Linear Matrix Inequalities in System and Control Theory, *Studies in Applied Mathematics*, vol. 15. SIAM, Philadelphia, PA (1994)

9. Burer, S., Kim, S., Kojima, M.: Faster, but weaker, relaxations for quadratically constrained quadratic programs. Computational Optimization and Applications **59**(1), 27–45 (2014). DOI 10.1007/s10589-013-9618-8.

10. Cardoso, J.F.: Super-symmetric decomposition of the fourth-order cumulant tensor. blind identification of more sources than sensors. In: Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP91), vol. 5, pp. 3109–3112. Toronto, Canada (1991)

11. Chen, Y., Gao, D.Y.: Global solutions to large-scale spherical constrained quadratic minimization via canonical dual approach. ArXiv e-prints (2013)

12. Dostál, Z.: Optimal Quadratic Programming Algorithms: With Applications to Variational Inequalities, 1st edn. Springer Publishing Company, Incorporated (2009)

13. Dostál, Z., Kozubek, T.: An optimal algorithm and superrelaxation for minimization of a quadratic function subject to separable convex constraints with applications. Mathematical Programming **135**(1), 195–220 (2012). DOI 10.1007/s10107-011-0454-2.

14. Gander, W., Golub, G.H., von Matt, U.: A constrained eigenvalue problem. Special Issue Dedicated to Alan J. Hoffman, Linear Algebra and its Applications **114**, 815 – 839 (1989). DOI http://dx.doi.org/10.1016/0024-3795(89)90494-1.

15. Goemans, M.X., Williamson, D.P.: Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. J. ACM **42**(6), 1115–1145 (1995). DOI 10.1145/227683.227684.

16. Hager, W.W.: Minimizing a quadratic over a sphere. SIAM Journal on Optimization **12**(1), 188–208 (2001). DOI 10.1137/S1052623499356071.

17. Holmström, K.: TOMLAB – an environment for solving optimization problems in MATLAB. In: Proceedings for the Nordic Matlab conference '97, pp. 27–28 (1997)

18. Kim, S., Kojima, M.: Second order cone programming relaxation of nonconvex quadratic optimization problems. Optimization Methods and Software **15**, 201–224 (2000)

19. Kim, S., Kojima, M.: Exact solutions of some nonconvex quadratic optimization problems via sdp and socp relaxations. Computational Optimization and Applications **26**(2), 143–154 (2003). DOI 10.1023/A:1025794313696.

20. Lim, L.H., Comon, P.: Blind multilinear identification. CoRR **abs/1212.6663** (2012, preprint)

21. Linderoth, J.: A simplicial branch-and-bound algorithm for solving quadratically constrained quadratic programs. Mathematical Programming **103**(2), 251–282 (2005). DOI 10.1007/s10107-005-0582-7.

22. Locatelli, M.: Some results for quadratic problems with one or two quadratic constraints. Oper. Res. Lett. **43**(2), 126–131 (2015). DOI 10.1016/j.orl.2014.12.002.

23. Luo, Z., Ma, W., So, A.M., Ye, Y., Zhang, S.: Semidefinite relaxation of quadratic optimization problems. IEEE Signal Process. Mag. **27**(3), 20–34 (2010). DOI 10.1109/MSP.2010.936019.

24. Muti, D., Bourennane, S.: Multiway filtering based on fourth order cumulantsh. Applied Signal Processing EURASIP **7**, 1147–1159 (2005)

25. Nesterov, Y., Wolkowicz, H., Ye, Y.: Semidefinite Programming Relaxations of Nonconvex Quadratic Optimization, pp. 361–419. Springer US, Boston, MA (2000). DOI 10.1007/978-1-4615-4381-7_13

26. Phan, A.H., Tichavský, P., Cichocki, A.: Error preserving correction method for CPD and bounded-norm CPD ArXiv e-prints (2017)

27. Phan, A.H., Yamagishi, M., Cichocki, A.: An augmented lagrangian algorithm for decomposition of symmetric tensors of order-4. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2547–2551 (2017). DOI 10.1109/ICASSP.2017.7952616

28. Reeves, S.J.: Chapter 6 - image restoration: Fundamentals of image restoration. In: J. Trussell, A. Srivastava, A.K. Roy-Chowdhury, A. Srivastava, P.A. Naylor, R. Chellappa, S. Theodoridis (eds.) *Academic Press Library in Signal Processing*, vol. 4, pp. 165 – 192. Elsevier (2014). DOI https://doi.org/10.1016/B978-0-12-396501-1.00006-6.

29. Rendl, F., Wolkowicz, H.: A semidefinite framework for trust region subproblems with applications to large scale minimization. Math. Program. **77**, 273–299 (1997). DOI 10.1007/BF02614438.

30. Rojas, M., Santos, S.A., Sorensen, D.C.: Algorithm 873: LSTRS: Matlab software for large-scale trust-region subproblems and regularization. ACM Trans. Math. Softw. **34**(2), 11:1–11:28 (2008). DOI 10.1145/1326548.1326553

31. Shashua, A., Zass, R., Hazan, T.: Multi-way clustering using super-symmetric non-negative tensor factorization. In: European Conference on Computer Vision (ECCV). Graz, Austria (2006).

32. Sorensen, D.C.: Minimization of a large-scale quadratic function subject to a spherical constraint **7**(1), 141–161 (1997). DOI http://dx.doi.org/10.1137/S1052623494274374.

33. Wen, Z., Yin, W.: A feasible method for optimization with orthogonality constraints. Mathematical Programming pp. 1–38 (2012). DOI 10.1007/s10107-012-0584-1.

34. Yuen, N., Friedlander, B.: Asymptotic performance analysis of blind signal copy using fourth order cumulant. Int. Journal of Adaptative Control Signal Processing **10**(2–3), 239–265 (1996)