Fast approximation of orthogonal matrices and application to PCA

Cristian Rusu

Faculty of Automatic Control and Computers, University Politehnica Bucharest

cristian.rusu@upb.ro

Lorenzo Rosasco

LCSL, Universitá di Genova

Massachusetts Institute of Technology and Istituto Italiano di Tecnologia

Abstract

We study the problem of approximating orthogonal matrices so that their application is numerically fast and yet accurate. We find an approximation by solving an optimization problem over a set of structured matrices, that we call extended orthogonal Givens transformations, including Givens rotations as a special case. We propose an efficient greedy algorithm to solve such a problem and show that it strikes a balance between approximation accuracy and speed of computation. The approach is relevant to spectral methods and we illustrate its application to PCA.

1 Introduction

Orthonormal transformations play a key role in most matrix decomposition techniques and spectral methods [4]. As such, manipulating them in an efficient manner is essential to many practical applications. While general matrix-vector multiplications with orthogonal matrices take $O(d^2)$ space and time, it is natural to ask whether faster approximate computations (say $O(d \log d)$) can be achieved while retaining enough accuracy.

Approximating an orthonormal matrix with just a few building blocks is hard in general. The standard decomposition technique meant to reduce complexity is a low-rank approximation. Unfortunately, for an orthonormal matrix, which is perfectly conditioned, this approach is meaningless.

In this work, we are inspired by the fact that several orthonormal/unitary transformations that exhibit low numerical complexity are known. The typical example is the discrete Fourier transform with its efficient implementation as the fast Fourier transform [47] together with other Fourier-related algorithms: fast Walsh-Hadamard transforms [17], fast cosine transforms [33], and fast Hartley transforms [9]. Other approaches include fast wavelet transforms [5], banded orthonormal matrices [40, 44] and fast Slepian transforms [27]. Decomposition of orthogonal matrices into $O(d)$ Householder reflectors or $O(d^2)$ Givens rotations [20][Chapter 5.1] are known already. These basic building blocks have been further extended, for example we have fast Givens rotations [36] and two generalizations of the Givens rotations [6] and [35]. In theoretical physics, unitary decompositions parametrize symmetry groups [45], and they are compactly parametrized using σ-matrices [42] or symmetric positive definite matrices [3]. To the best of our knowledge, none of these factorizations focus on reducing the computational complexity of using the orthogonal/unitary transformations but rather they model properties of physical systems.

Our idea is to approximately factor any orthonormal matrix into a product of a fixed number of sparse matrices such that their application (to a vector) has linearithmic complexity. In this paper, we derive structured approximations to orthonormal matrices that can be found efficiently and applied remarkably fast. We pose the search for an efficient approximation as an optimization problem over a product of structured matrices, the extended orthogonal Givens transformations. These structures extend Givens rotations to also include reflectors with no computational drawback and suggest a

decomposition of the main optimization problem into sub-problems that are easy to understand and solved via a greedy approach. The theoretical properties of the obtained solution are characterized in terms of approximation bounds while the empirical properties are studied extensively.

We illustrate our approach considering dimensionality reduction with principal component analysis (PCA). Here the goal is not to propose a new fast algorithm to compute the principal directions, plenty of efficient algorithms for this task [15][20][Chapter 10] [22]. Rather, we aim at constructing fast dimensionality reduction operators. While the calculation of the principal components is a one-off computation, a numerically efficient projection operation is critical since it is required multiple times in downstream applications. The problem of deriving fast projections has also been previously studied. Possible approaches include: fast wavelet transforms [49], sparse PCA [48, 52], structured transformations such as circulant matrices [24], Kronecker products [21], Givens rotations [12, 32, 34] or structured random projections [1, 18]. Compared to these works, we propose a new way to factorize any orthogonal matrix, including PCA directions, into simple orthogonal structures that we call extended orthogonal Givens transformations and which naturally lead to optimization problems that have closed-form solutions and are therefore efficiently computed.

We note that our approach provides new perspectives on the structure of the orthogonal group and how to coarsely approximate it, which might have an impact on other open research questions.

The paper is organized as follows: Section 2 describes the basic building blocks and algorithm that we propose, Section 3 gives the theoretical guarantees for our contributions, Section 4 details the application of our method to PCA projections and Section 5 shows the numerical experiments.

2 The proposed algorithm

Given a $d \times d$ orthonormal $\mathbf{U}$, the matrix-vector multiplication $\mathbf{U}\mathbf{x}$ takes $O(d^2)$ operations. We want to build $\bar{\mathbf{U}}$ such that $\mathbf{U} \approx \bar{\mathbf{U}}$ and $\bar{\mathbf{U}}\mathbf{x}$ takes $O(d \log d)$ operations. Parametrizations of orthonormal matrices [2] are known, but to be best of our knowledge, the problem of accurately approximating $\mathbf{U}$ as product of only a few $O(d \log d)$ transformations is open. Given a $d \times p$ diagonal $\mathbf{\Sigma}_p$, in the spirit of previous work minimizing Frobenius norm approximations [30, 31], we consider the problem

$$\underset{\bar{\mathbf{U}}, \ \bar{\mathbf{\Sigma}}_p}{\text{minimize}} \ \|\mathbf{U}\mathbf{\Sigma}_p - \bar{\mathbf{U}}\bar{\mathbf{\Sigma}}_p\|_F^2 \ \text{subject to} \ \bar{\mathbf{U}} \in \mathcal{F}_g, \tag{1}$$

where $\bar{\mathbf{U}}$ is $d \times d$ and $\bar{\mathbf{\Sigma}}_p$ is a $d \times p$ diagonal matrix. Choosing $p = d$ while $\mathbf{\Sigma}_p$ and $\bar{\mathbf{\Sigma}}_p$ to be the identity, we simply approximate $\mathbf{U}$. We will also use $\mathbf{U}_p$ do denote the first $p \leq d$ columns of $\mathbf{U}$. The above general formulation allows to also consider cases where different directions might have different importance. Then, $\mathcal{F}_g$ is a set of orthogonal matrices – defined next – that can be applied fast and allow to efficiently but approximately solve (1).

2.1 The basic building blocks

Classic matrix building blocks that are numerically efficiency include circulant/Toeplitz matrices or Kronecker products. These choices are inefficient as they depend on $O(d)$ free parameters but their matrix-vector product cost is $O(d \log d)$ or even $O(d\sqrt{d})$, i.e., they do not scale linearly with the number of parameters they have. Consider the sparse orthogonal matrices

$$\mathbf{G}_{ij} = \begin{bmatrix} \mathbf{I}_{i-1} & & & & \\ & * & & * & \\ & & \mathbf{I}_{j-i-1} & & \\ & * & & * & \\ & & & & \mathbf{I}_{d-j} \end{bmatrix}, \tilde{\mathbf{G}}_{ij} \in \left\{ \begin{bmatrix} c & -s \\ s & c \end{bmatrix}, \begin{bmatrix} c & s \\ s & -c \end{bmatrix} \right\}, \text{ such that } c^2 + s^2 = 1, \tag{2}$$

where the non-zero part (denoted by $*$ and $\tilde{\mathbf{G}}_{ij}$) on rows and columns i and j. The transformation in (2), with the first option in $\tilde{\mathbf{G}}_{ij}$, is a Givens (or Jacobi) rotation. With the second option, we have a very sparse Householder reflector. These transformations were first used by Rusu and Thompson [37] to learn numerically efficient sparsifying dictionaries for sparse coding. The $\mathbf{G}_{ij}$s have the following advantages: i) they are orthogonal; ii) they are sparse and therefore fast to manipulate: matrix-vector multiplications $\mathbf{G}_{ij}\mathbf{x}$ take only 6 operations; iii) there are two degrees of freedom to learn: c (or s) and the binary choice; and iv) allowing both sub-matrices in $\tilde{\mathbf{G}}_{ij}$ enriches the structure and as we will see, leads to an easier (closed-form solutions) optimization problem.

We propose to consider matrices $\bar{\mathbf{U}} \in \mathcal{F}_g$ that are products of g transformations from (2), that is

$$\bar{\mathbf{U}} = \prod_{k=1}^{g} \mathbf{G}_{i_k j_k} = \mathbf{G}_{i_1 j_1} \dots \mathbf{G}_{i_g j_g}. \tag{3}$$

Matrix-vector multiplication with $\bar{\mathbf{U}}$ takes $6g$ operations – when g is $O(d \log d)$ this is significantly better than $O(d^2)$, while the coding complexity of each $\mathbf{G}_{ij}$ is approximately $2 \log_2 d + C$: $2 \log_2 d - 1$ bits to encode the choice of the two indices, a constant factor C for the pair (c, s) and 1 bit for the choice between the rotation and reflector. The coding complexity of $\bar{\mathbf{U}}$ scales linearly with g.

We note that Givens rotations have been used extensively to build numerically efficient transformations [11, 19, 30–32]. However, 2×2 reflector was not used before. This may be because in linear algebra (e.g. in QR factorization) and in optimization [39] considering also the reflector has no additional benefit: the rotation alone introduces each zero in the QR factorization and the reflector does not have an exponential mapping on the orthogonal manifold, respectively. As we will show, considering both the rotation and the reflector has the advantage of providing a closed-form solution to our problem.

2.2 The proposed greedy algorithm

We propose to solve the optimization problem in (1) with a greedy approach: we keep $\bar{\mathbf{\Sigma}}_p$ and all variables fixed except for a single $\mathbf{G}_{i_k j_k}$ from $\bar{\mathbf{U}}$ and minimize the objective function. When optimizing w.r.t. $\mathbf{G}_{i_k j_k}$ it is convenient to write

$$\|\mathbf{U}\mathbf{\Sigma}_p - \bar{\mathbf{U}}\bar{\mathbf{\Sigma}}_p\|_F^2 = \| \underbrace{\prod_{t=1}^{k-1} \mathbf{G}_{i_t j_t}^T \mathbf{U}\mathbf{\Sigma}_p}_{\mathbf{L}^{(k)}} - \mathbf{G}_{i_k j_k} \underbrace{\prod_{t=k+1}^{g} \mathbf{G}_{i_t j_t} \bar{\mathbf{\Sigma}}_p}_{\mathbf{N}^{(k)}} \|_F^2 = \|\mathbf{L}^{(k)} - \mathbf{G}_{i_k j_k} \mathbf{N}^{(k)}\|_F^2. \tag{4}$$

The next result characterizes the Givens transformation $\mathbf{G}_{i_k j_k}$ minimizing the above norm. We drop the dependence on k for ease of notation.

Theorem 1 (Locally optimal $\mathbf{G}_{ij}$). *Let $\mathbf{L}$ and $\mathbf{N}$ be two $d \times p$ matrices. Further, let $\mathbf{Z} = \mathbf{L}\mathbf{N}^T$ and* $\mathbf{Z}_{\{i,j\}} = \begin{bmatrix} Z_{ii} & Z_{ij} \\ Z_{ji} & Z_{jj} \end{bmatrix}$ *then we have*

$$C_{ij} = \|\mathbf{Z}_{\{i,j\}}\|_* - tr(\mathbf{Z}_{\{i,j\}}) = \begin{cases} \sqrt{(Z_{ii} + Z_{jj})^2 + (Z_{ij} - Z_{ji})^2} - Z_{ii} - Z_{jj}, & \text{if } \det(\mathbf{Z}_{\{i,j\}}) \geq 0 \\ \sqrt{(Z_{ii} - Z_{jj})^2 + (Z_{ij} + Z_{ji})^2} - Z_{ii} - Z_{jj}, & \text{if } \det(\mathbf{Z}_{\{i,j\}}) < 0 \end{cases} \tag{5}$$

Let $\mathbf{Z}_{\{i^\star, j^\star\}} = \mathbf{V}_1 \mathbf{S} \mathbf{V}_2^T$ be the SVD of $\mathbf{Z}_{\{i^\star, j^\star\}}$, with

$$(i^\star, j^\star) = \underset{(i,j),\ j>i}{\arg\max}\ C_{ij}. \tag{6}$$

Then, the extended orthogonal Givens transformation that minimizes $\|\mathbf{L} - \mathbf{G}_{ij}\mathbf{N}\|_F^2$ is given by $\tilde{\mathbf{G}}_{i^\star j^\star}^\star = \mathbf{V}_1 \mathbf{V}_2^T$.

The above theorem derives a locally optimal way to construct an approximation $\bar{\mathbf{U}}$. We iteratively apply the result to find, for each component k in (3), the extended orthogonal Givens transformation that best minimizes the objective function (1). The full procedure is in Algorithm 1 and can be viewed in two different ways: i) a coordinate minimization algorithm; or ii) a hierarchical decomposition where each stage is extremely sparse. The proposed algorithm is guaranteed to converge, in the sense of the objective function (1), to a stationary point. Indeed, no step in the algorithm can increase the objective function, since the sub-problems are minimized exactly: we choose the best indices and then perform the best 2×2 transformation. We note three remarks on the properties of Algorithm 1.

Remark 1 (Complexity of Algorithm 1). *The computational complexity of the iterative part of Algorithm 1 is $O(dg)$ and the initialization is dominated by the computation of all the scores C_{ij} which takes $O(d^2)$. Note that, the C_{ij}s are computed from scratch only once in the initialization phase. After that, at each step k we need to recompute the C_{ij} (redo the 2×2 singular value decompositions) only for the indices (i_k, j_k) currently used (the Givens transformations act on two coordinates at a time). All other scores are update by the same quantity: in (6), the C_{ij} are the same except when i or j belong to the set (i_k, j_k). This observation substantially reduces the running time.*

Algorithm 1 Approximate orthonormal matrix factorization with extended Givens transformations

---

Input: The p orthogonal components $\mathbf{U}_p$ and their weights (singular values) $\mathbf{\Sigma}_p$, the size g of the approximation (3), the update rule for $\bar{\mathbf{\Sigma}}_p$ in { 'identity', 'original', 'update' } and the stopping criterion ϵ (default taken to be $\epsilon = 10^{-2}$).
Output: The linear transformation $\bar{\mathbf{U}}\bar{\mathbf{\Sigma}}_p$, the approximate solution to (1).
Initialize: $\mathbf{G}_{i_k j_k} = \mathbf{I}_{d \times d}, k = 1, \ldots, g$ and compute all scores C_{ij} according to (5) with $\mathbf{Z} = \mathbf{U}_p \bar{\mathbf{\Sigma}}_p^T$, where $\bar{\mathbf{\Sigma}}_p = \left[\mathbf{I}_{p \times p}; \quad \mathbf{0}_{(d-p) \times p} \right]$ if the update rule is 'identity' and $\bar{\mathbf{\Sigma}}_p = \mathbf{\Sigma}_p$ otherwise.
repeat
    Set $\mathbf{L}^{(0)} = \mathbf{U}_p \mathbf{\Sigma}_p$ and set $\mathbf{N}^{(0)} = \mathbf{G}_{i_1 j_1} \ldots \mathbf{G}_{i_g j_g} \bar{\mathbf{\Sigma}}_p$.
    for $k = 1$ **to** g **do**
        Update $\mathbf{N}^{(k)} = \mathbf{G}_{i_k j_k}^T \mathbf{N}^{(k-1)}$ and find best score according to (6).
        Compute the best k^{th} transformation by Theorem 1.
        Update $\mathbf{L}^{(k)} = \mathbf{G}_{i_k j_k}^T \mathbf{L}^{(k-1)}$ and then update all scores C_{ij} in (5) but only for indices $i, j \in \{i_k, j_k\}$ with $\mathbf{Z} = \mathbf{L}^{(k)} (\mathbf{N}^{(k)})^T$ – all other scores are unchanged.
    end for
    Set $\bar{\mathbf{\Sigma}}_p = \left[\text{diag}(\mathbf{L}^{(g)}); \quad \mathbf{0}_{(d-p) \times p} \right]$ if rule is 'update', $i \leftarrow i + 1$ and $\epsilon_i = \|\mathbf{L}^{(g)} - \bar{\mathbf{\Sigma}}_p\|_F^2$.
until $|\epsilon_{i-1} - \epsilon_i| < \epsilon$, if $i > 1$.

Remark 2 (Complexity of applying $\bar{\mathbf{U}}\bar{\mathbf{\Sigma}}_p$). *When $p < d$ the computational complexity of $6g$ operations is an upper bound. Since we keep only p components, we need be careful not to perform operations whose result is thrown away by the mask $\bar{\mathbf{\Sigma}}_p$. Consider for example a transformation $\mathbf{G}_{1d}$ applied to a vector of size d projected to a $p < d$ dimensional space. The three operations that take place on the dth component are unnecessary. Then, after computing $\bar{\mathbf{U}}$, a pass is made through each of the g transforms to decide which of two coordinates the computations are necessary for the final result. As we will show, this further improves the numerical efficiency of our method.*

Remark 3 (On the choice of indices). *Algorithm 1 greedily chooses at each step k the indices according to (6). Other factors might be considered: i) choosing indices based on previous choices so that only a select group of indices are used throughout the algorithm, or ii) make multiple choices at each step in order to speed up the algorithm.*

3   Analysis of the proposed algorithm

We consider $p = d$, i.e., $\bar{\mathbf{\Sigma}}_p = \mathbf{I}_{d \times d}$ and therefore $\mathbf{Z} = \mathbf{U}$. We model the $\mathbf{U}$ as a random orthonormal matrix with Haar measure [25] updated so that the diagonal is positive. We perform this update because multiplication by a diagonal matrix with ± 1 entries has no computational cost but it brings $\mathbf{U}$ closer to $\mathbf{I}_{d \times d}$. The goal of this section is to establish upper bounds for the distance between $\mathbf{U}$ and $\bar{\mathbf{U}}$, as a function of d and g. We first comment on the inherent difficulty of the problem.

Remark 4 (The approximation gap). *Since the orthogonal group has size $O(d^2)$, by the pigeonhole principle a random orthogonal matrix as (3) and only $g \ll d^2$ degrees of freedom cannot be exactly approximated with less than $O(d^2)$ operations. For our purposes, think g either $O(d)$ or $O(d \log d)$. Our goal is to show that the fast structures we propose can perform well in practice and have theoretical bounds that guarantee worse case or average accuracy.*

Next, we show two approximations bounds depending on the number of Givens transformations (2).

Theorem 2 (A special bound). *Given a random $d \times d$ orthonormal $\mathbf{U}$, for large d, its approximation $\bar{\mathbf{U}}$ from (1) with $g = d/2$ transformations from (2) obeys*

$$\mathbb{E}[\|\mathbf{U} - \bar{\mathbf{U}}\|_F^2] \leq 2d - \sqrt{2\pi d}. \tag{7}$$

Theorem 3 (A general bound). *Given a random $d \times d$ orthonormal $\mathbf{U}$, for large d, its approximation $\bar{\mathbf{U}}$ from (1) with $g \leq d(d-1)/2$ transformations from (2) is bounded by*

$$\mathbb{E}\left[\|\mathbf{U} - \bar{\mathbf{U}}\|_F^2\right] \leq 2(d - \lfloor r \rfloor) - \frac{2\sqrt{2}}{\sqrt{\pi}} \sqrt{d - \lfloor r \rfloor}, \text{ where } r = d - \frac{1 + \sqrt{(2d-1)^2 - 8g}}{2}. \tag{8}$$

Theorem 2 shows that, on average, the performance might degrade with increasing d. As stated in Remark 4, this is not surprising since the orthonormal group is much larger than the structure we are trying to approximate it with. The next result provides a bound for other values of g. In Theorem 3, taking $g = c_1 d \log d$ for some positive constant c_1 we have that $r \approx c_1 \log d$. This means that whenever $p \ll d$ we will roughly need $O(d)$ Givens transformations from (2) to improve the $\lfloor r \rfloor$ term. Since the proof of the theorem uses only rotations (and furthermore, in a particular order of indices (i_k, j_k)) we expect our algorithm to perform much better than the bound indicates as it allows for a richer structure (2) and uses greedy steps that maximally improve the accuracy at each step. The previous theorems consider the Frobenius norm. In the Jacobi iterative process for diagonalizing a symmetric matrix with Givens rotations [20][Chapter 8.4] the progress of the procedure (convergence) is measured using the off-diagonal "norm" $\text{off}(\mathbf{U}) = \sqrt{\sum_t^d \sum_{q \neq t}^d U_{tq}^2}$.

Theorem 4 (Convergence in the off-diagonal norm). *Given a $d \times d$ orthonormal $\mathbf{U}$ and a single Givens transformation $\mathbf{G}_{ij}$, assuming $\det(\mathbf{U}_{\{i,j\}}) \geq 0$ we have*

$$\text{off}(\mathbf{U}\mathbf{G}_{ij}^T)^2 \leq \text{off}(\mathbf{U})^2 + \frac{1}{2}((U_{ii} - U_{jj})^2 - (U_{ij} - U_{ji})^2). \tag{9}$$

This result shows that, unlike with the Jacobi iterations, monotonic convergence in this quantity is not guaranteed and depends on the relative differences between the diagonal and the off-diagonal entries of $\mathbf{U}_{\{i,j\}}$. Our method convergence monotonically to a stationary point when we measure the progress in the Frobenius norm.

Remark 5 (The effect of a single $\mathbf{G}_{ij}$). *Given a $d \times d$ orthonormal $\mathbf{U}$ and a Givens transformation $\mathbf{G}_{ij}$ from (2) we have that: i) $\mathbf{U}\mathbf{G}_{ij}^T$ is closer to the identity matrix in the sense that $\tilde{\mathbf{G}}_{ij}$ makes a positive contribution to the diagonal elements, i.e., $\text{tr}(\mathbf{U}\mathbf{G}_{ij}^T) = \text{tr}(\mathbf{U}) + C_{ij}$; and ii) $\mathbb{E}[C_{ij}] \approx 0.6956 d^{-1/2}$ if $\mathbf{U}$ is random with Haar measure and positive diagonal for large d.*

The above remark suggests a metric to study the convergence of the proposed method: each Givens transformation adds the score C_{ij} to the diagonal entries of the current approximation (and therefore ensures that $\mathbf{U}\bar{\mathbf{U}}^T$ converges to the identity – the only diagonal orthonormal matrix). By choosing the maximum C_{ij} we are taking the largest step in this direction.

Remark 6 (Evolution of $C_{i_k j_k}$ with k). *Given a fixed $0 < u < 1$, consider the toy construction $\mathbf{U}_{\{i,j\}} = \begin{bmatrix} u & z_2 \\ z_1 & u \end{bmatrix}$, i.e., a 2×2 sub-matrix of a $d \times d$ orthonormal matrix where diagonal elements are equal and off-diagonals are two independent truncated standard normal random variables in the interval $[-\sqrt{1-u^2}, \sqrt{1-u^2}]$ (since rows and columns of $\mathbf{U}$ are ℓ_2 normalized). Then, for large d, by direct calculation we have that the expected score $\mathbb{E}[C_{ij}(u)]$, i.e., C_{ij} as a function of u, obeys*

$$\mathbb{E}[C_{ij}(u)] \propto (1-u)^2, \tag{10}$$

i.e., the expected C_{ij} decreases on average quadratically with the increase in the diagonal elements. The remark is intuitive: as k increases $\bar{\mathbf{U}}$ is more accurate and $\mathbf{U}\bar{\mathbf{U}}^T$ becomes diagonally dominant, i.e., $\mathbf{U}\bar{\mathbf{U}}^T \rightarrow \mathbf{I}_d$ as $k \rightarrow O(d^2)$, and we do expect to reach lower scores $C_{i_k j_k}$, i.e., few Givens transformations from (2) provide a rough estimation while very good approximations require $k \approx d^2$.

3.1 Other ways to measure the approximation error

Throughout this paper we use the Frobenius norm to measure and study the approximation error we propose. In this section we discuss this choice and explore a few alternatives.

In the linear algebra literature, a natural way to measure approximation error is through the operator norm. This is especially true in the randomized linear algebra field (where matrix concentration inequalities which bound the operator norm play a central role). Moreover, the review manuscript [46] deals explicitly with some potential issues that might arise from using the Frobenius, instead of the operator, norm in matrix approximations: the discussion in Chapter 6 of [46] entitled "Warning: Frobenius–Norm Bounds". The text highlights situations where the matrix to be approximated is low rank and corrupted by noise and/or scaling issues are present. While that discussion holds true, in our case we deal with a given perfectly conditioned matrix $\mathbf{U}$ and its approximation displays the same scaling ($\bar{\mathbf{U}}$ has normalized columns just as $\mathbf{U}$). Consider the following remark.

Remark 7. [Simplifying the Frobenius norm objective function] *Consider for simplicity $p = d$ and $\bar{\boldsymbol{\Sigma}}_d = \boldsymbol{\Sigma}_d$ in (1) and see that $\|(\mathbf{U} - \bar{\mathbf{U}})\boldsymbol{\Sigma}_d\|_F^2 = 2\sum_{i=1}^d \sigma_i(1 - \mathbf{u}_i^T\bar{\mathbf{u}}_i) = 2\sum_{i=1}^d \sigma_i(1 - \cos(\theta_i))$, where θ_i is the angle between column vectors $\mathbf{u}_i$ and $\bar{\mathbf{u}}_i$ (the columns of $\mathbf{U}$ and $\bar{\mathbf{U}}$, respectively), and $\sigma_i > 0$ are the diagonal elements of $\boldsymbol{\Sigma}_d$.* □

Therefore, the proposed optimization objective function minimizes the weighted sum of the cosines of the angles between the original columns and their approximations. As such, low-rank or scaling issues cannot arise. The only potential problem is that different columns might have very different approximation errors (i.e., there could exist i such that $\cos(\theta_i) \approx 1$ while there could be some j for which $\cos(\theta_j) \ll 1$). This issue can be mitigated by increasing g in (3) or choosing carefully the indices (i_k, j_k) where the proposed transformations operate.

Assume that we consider the operator norm as the approximation error, i.e., we want to minimize $\|\mathbf{U} - \bar{\mathbf{U}}\|_2$ in (1). We have the following two results.

Remark 8. [The spectrum of the error matrix] *Consider $p = d$ and $\bar{\boldsymbol{\Sigma}}_d = \boldsymbol{\Sigma}_d = \mathbf{I}_d$ in (1) then for any orthonormal $\mathbf{U}$ and $\bar{\mathbf{U}}$ the error matrix $\mathbf{U} - \bar{\mathbf{U}}$ is normal and has all its eigenvalues on a circle of radius one centered at $(1, 0)$ in the complex plane. As such, we have that $\|\mathbf{U} - \bar{\mathbf{U}}\|_2 \leq 2$.* □

Theorem 5. [A bound on the operator norm of the error matrix] *Consider $p = d$ and $\bar{\boldsymbol{\Sigma}}_d = \boldsymbol{\Sigma}_d = \mathbf{I}_d$ in (1) and that $\mathbf{u}_i^T\bar{\mathbf{u}}_i \geq 0$ for all $i = 1, \ldots, d$, then the operator norm of the error matrix obeys $\|\mathbf{U} - \bar{\mathbf{U}}\|_2 \leq 1 - \epsilon_{min} + \sqrt{(d-1)(1 - \epsilon_{min}^2)}$ where $\epsilon_{min} = \min_i \mathbf{u}_i^T\bar{\mathbf{u}}_i = \min_i \cos(\theta_i)$. The bound in Remark 8 is met when $\epsilon_{min} \geq (d-2)/d$.* ∎

Remark 8 describes the full spectrum of the error matrix. It is interesting to notice that the upper bound $\|\mathbf{U} - \bar{\mathbf{U}}\|_2 \leq 2$ coincides with the expectation result from [13]: as $d \to \infty$ if both $\mathbf{U}$ and $\bar{\mathbf{U}}$ are chosen uniformly at random with Haar measure then almost surely $\|\mathbf{U} + \bar{\mathbf{U}}\| \to 2$. Theorem 5 describes an upper bound on the operator norm of the error matrix. The bound is tight only for very high values of ϵ_{min} and therefore is meant to give a qualitative measure of the approximation. As in Remark 7, the key quantity is $\cos(\theta_i)$ but now the bound depends on the worst approximation: while the Frobenius norm objective function minimizes the sum of the pairwise distances between the columns of $\mathbf{U}$ and $\bar{\mathbf{U}}$, when using the operator norm the approximation accuracy depends on the largest distance between the same pairwise columns. The assumption that $\mathbf{u}_i^T\bar{\mathbf{u}}_i \geq 0$ is not restrictive at all as we expect $\mathbf{U}^T\bar{\mathbf{U}}$ to be diagonally dominant with positive elements on the diagonal (see Remark 5 and the discussion after Remark 6). We note that in this context the Frobenius norm approach could be used to minimize a proxy for the operator norm. As in Remark 7 we have the choice of σ_i we could update these values with each iteration of the proposed algorithm such that $\sigma_i^{(\text{new})} \leftarrow \frac{\mathbf{u}_{i_{\max}}^T\bar{\mathbf{u}}_{i_{\max}}}{\mathbf{u}_i^T\bar{\mathbf{u}}_i}$ where $i_{\max} = \arg\max_i \mathbf{u}_i^T\bar{\mathbf{u}}_i$. This choice will encourage the algorithm to improve upon the worst pairwise column approximation (i.e., the highest $\mathbf{u}_i^T\bar{\mathbf{u}}_i$). This approach would be in the spirit of an iteratively reweighted least squares algorithm such as [14].

Finally, the last measure of accuracy we consider is the angle between two subspaces as described by [7]. This value can be non-zero only when $p < d$, a case which is interesting for PCA projections and which we detail in the next section. Following [29], given $\mathbf{U}_p$ and $\bar{\mathbf{U}}_p$, whose columns span two d-dimensional subspaces of size p, we compute the angles between the two subspaces as

$$\beta_i = \arccos(\tau_i), \ i = 1, \ldots, p, \tag{11}$$

where $0 \leq \tau_i \leq 1$ are the singular values of $\mathbf{U}_p^T\bar{\mathbf{U}}_p$ and where $0 \leq \beta_1 \leq \cdots \leq \beta_p \leq \pi/2$. We take the principal angle to be the largest angle above, i.e., $\beta = \beta_p = \arccos(\tau_p)$.

Finally, we would like to note that all the approximation errors we have previously discussed measure (in different ways) how well $\mathbf{U}_p^T\bar{\mathbf{U}}_p$ approaches the identity matrix.

4   Application: fast PCA projections

Consider a training set of d-dimensional points $\{\mathbf{x}_i\}_{i=1}^N$ and the $d \times N$ matrix $\mathbf{X} = [\mathbf{x}_1 \ \ldots \ \mathbf{x}_N]$. Given $1 \leq p < d$, PCA provides the optimal p-dimensional projection that minimally distorts, on average, the data points. The projection is given by the eigenvectors of the p largest eigenvalues of $\mathbf{X}\mathbf{X}^T$ or, equivalently, the left singular vectors of the p largest singular values of $\mathbf{X}$, that is

$$\mathbf{X}\mathbf{X}^T \approx \mathbf{U}_p(\boldsymbol{\Sigma}_p\boldsymbol{\Sigma}_p^T)\mathbf{U}_p^T \text{ and } \mathbf{X} \approx \mathbf{U}_p\boldsymbol{\Sigma}_p\mathbf{V}_p^T. \tag{12}$$

Given the above decompositions we can approximate $\mathbf{X}$ by $\bar{\mathbf{X}} = \bar{\mathbf{U}}\bar{\boldsymbol{\Sigma}}_p \mathbf{V}_p^T$, i.e., we keep $\mathbf{V}_p$ but we modify the principal components and their singular values, such that we minimize the error given by

$$\|\mathbf{U}_p \boldsymbol{\Sigma}_p \mathbf{V}_p^T - \bar{\mathbf{U}}\bar{\boldsymbol{\Sigma}}_p \mathbf{V}_p^T\|_F^2 = \|\mathbf{U}_p \boldsymbol{\Sigma}_p - \bar{\mathbf{U}}\bar{\boldsymbol{\Sigma}}_p\|_F^2, \tag{13}$$

where $\mathbf{U}_p$ is $d \times p$, the diagonal matrix $\boldsymbol{\Sigma}_p$ is $p \times p$, $\bar{\mathbf{U}}$ is of size $d \times d$, $\bar{\boldsymbol{\Sigma}}_p$ is $d \times p$ and is zero except for its main $p \times p$ diagonal. In this paper, we work with $\mathbf{X}$, as opposed to $\mathbf{X}\mathbf{X}^T$, to keep the relationship with $\mathbf{U}_p$ linear, rather than quadratic. In the context of applying our approach to PCA, we use our decomposition on the principal components $\mathbf{U}_p$ which we assume are already calculated together with the associated singular values $\boldsymbol{\Sigma}_p$ which we may use as weights in (1).

Note that in (13), $\mathbf{V}_p$, which has size N, is not necessary and that the two-step procedure is equivalent to computing the projections $\bar{\mathbf{U}}$ directly from $\mathbf{X}$. Also note that based on (12), we could factor $\mathbf{X} \approx \bar{\mathbf{U}}\boldsymbol{\Sigma}\bar{\mathbf{V}}^T$ where $\bar{\mathbf{U}}$ and $\bar{\mathbf{V}}^T$ are approximations in $\mathcal{F}_{O(d \log d)}$ and $\mathcal{F}_{O(N \log N)}$, respectively. The difficulty here is the dependency of $\bar{\mathbf{V}}$ on $N \gg d$ which would require a large running time.

With $\bar{\mathbf{U}}$ fixed, for $\bar{\boldsymbol{\Sigma}}_p$ we have several strategies: i) set it to the identity, i.e., flatten the spectrum; ii) keep it to the original singular values $\boldsymbol{\Sigma}_p$; or iii) continuously update it to minimize the Frobenius norm, i.e., get the new "singular values" that are optimal with the approximation $\bar{\mathbf{U}}$. The first approach favors the accurate reconstruction of all components while the other approach favors mostly the few leading components only (depending on the decay rate of the corresponding singular values).

4.1 Comparison with the symmetric diagonalization by Givens rotations approach

Because of the locally optimal way the Givens transformations are chosen (see Theorem 1), our proposed factorization algorithm is computationally slower than the Jacobi diagonalization process which chooses the Givens rotations on indices (i, j) corresponding to the largest off-diagonal entry of the covariance matrix. Furthermore, the Jacobi decomposition uses each rotation to zero the largest absolute value off-diagonal entry and because of this sub-optimal choice needs $O(d^2 \log d)$ Givens rotations [10] to complete de diagonalization (more than the $\frac{d(d-1)}{2}$ needed to fully represent the orthonormal group).

In all other aspects, our approach provides advantages over the Jacobi approach: i) we define a clear objective function that we locally optimize exactly; ii) it is known that the Jacobi process converges slowly when the number of rotations is low [28], which is exactly the practically relevant scenario we have, i.e., $g \ll d^2$; iii) with the same computational complexity, i.e., g terms in the factorization, our proposed approach is always more accurate since we include as a special case the Givens rotations.

4.2 Comparison with structured matrix factorization

Our approach requires the explicit availability of the orthonormal principal directions $\mathbf{U}_p$. Previous methods that factor using only Givens rotations are not applied directly on an orthogonal matrix. These methods rely on receiving as input a symmetric object (e.g., $\mathbf{X}\mathbf{X}^T$) and then using a variant of Jacobi iterations for matrix diagonalization [31] or multiresolution factorizations [30] to find the good rotations that approximate the orthonormal eigenspace. Applying Givens transformations directly to $\mathbf{X}$ on the left, i.e., $\mathbf{G}_{ij}\mathbf{X}$, cannot lead to the computation of the PCA projections $\mathbf{U}$ but only to the polar decomposition. On the other hand, when applying Givens rotations on both sides of the covariance matrix, i.e., $\mathbf{G}_{ij}\mathbf{X}\mathbf{X}^T\mathbf{G}_{ij}^T$, then the right eigenspace $\mathbf{V}$ cancels out in the product (12) and we are able to directly recover $\mathbf{U}$ (but we need $\mathbf{X}\mathbf{X}^T$ explicitly). Finally, note that the diagonalization process approximates the full eigenspace $\mathbf{U}$ and cannot separate from the start the p principal components $\mathbf{U}_p$ because they are solving the following problem

$$\underset{\bar{\mathbf{U}}, \, \bar{\boldsymbol{\Lambda}}}{\text{minimize}} \, \|\mathbf{X}\mathbf{X}^T - \bar{\mathbf{U}}\bar{\boldsymbol{\Lambda}}\bar{\mathbf{U}}\|_F^2 \text{ subject to } \bar{\mathbf{U}} \in \mathcal{F}_g. \tag{14}$$

This formulation is useful to approximate the whole symmetric matrix $\mathbf{X}\mathbf{X}^T$ (or $\mathbf{U}$), but not necessarily the p principal eigenspace $\mathbf{U}_p$. To get these we would need to complete the diagonalization process, find the p largest entries on the diagonal of $\bar{\boldsymbol{\Lambda}}$ and then work backward to identify the rotations that contributed diagonalizing those largest elements. This procedure would be prohibitively expensive. Previous work, e.g. [30, 32], deals with approximating $\mathbf{X}\mathbf{X}^T$ rather than computing PCA. In the same line of work, the one-sided Jacobi algorithm for SVD [16] can also be applied.
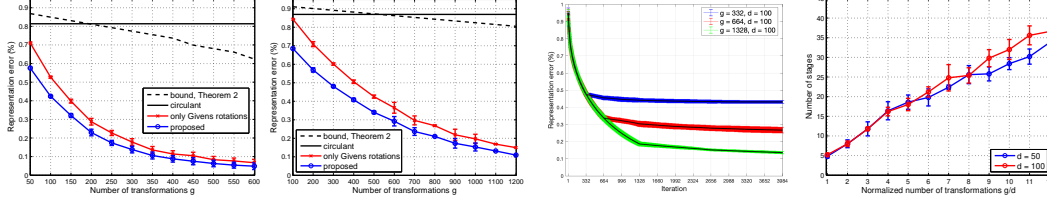
Figure 1: Average approximation errors and standard deviations over 100 realizations of random orthonormal matrices of size $d = 50$ (left) and $d = 100$ (right). For reference we show the bound developed in Theorem 3, the approximation accuracy of the circulant [24] (Toeplitz performed just marginally better) and that of the using the factorization (3) but allowing only Givens rotations. As expected, performance degrades with large d.

Figure 2: Left: for $d = 100$ and $g \in \{332, 664, 1328\}$ we show the evolution (mean and standard deviation) of the objective function with the number of iterations. In each case, the first g iterations are the initialization process. Right: for $d \in \{50, 100\}$ we show the number of stages in each transformation we learrn with the proposed algorithm as a function of g. For both plots results are averaged over 100 realizations.

5   Experimental results

Given the rather pessimistic guarantees, we tackle problems: how well does Algorithm 1 recover random orthogonal matrices and principal components such that we benefit from the computational gains but do not significantly impact the approximation/classification accuracy. Source code available.[1]

5.1   Synthetic experiments

For fixed d we generate random orthonormal matrices from the Haar measure [25]. Figure 1 shows the representation error $(2d)^{-1}\|\mathbf{U} - \bar{\mathbf{U}}\|_F^2$ for the proposed method. The plot shows that allowing for the Givens transformations $\tilde{\mathbf{G}}_{ij}$ in (2) brings a 17% relative benefit as compared to using only the Givens rotations while, for the same g, the computational complexity is the same. The circulant approximation performs worst because it has the lowest number of degrees of freedom, only d (computationally, it is comparable with the Givens and proposed approaches for $g = 100$). Lastly, we can observe that the bound is very pessimistic, especially for these values of g. In Figure 2 (left) we show for fixed number of transformations g the progress that the proposed algorithm makes with each iteration. It is interesting to observe that the initialization steps (first g steps) significantly decrease the approximation error while the other step make only moderate improvements. This indicates that convergence is slow and might take a large number of iterations with little progress made by the latter steps. Also in Figure 2 (right) we show the number of stages in each transformation. A stage is a set of extended orthogonal Givens transformations that can be applied in parallel, i.e., they do not share any indices (i_k, j_k) among them. This is important from an implementation perspective as parallel processing can be exploited to speedup the transformations.

5.2   MNIST digits and fashion

We now turn to a classification problem. We use the MNIST digits and fashion datasets. The points have size $d = 400$ (we trimmed the bordering whitespace) and we have $N = 6 \times 10^4$ training and $N_{\text{test}} = 10^4$ test points. In all cases, we use the k-nearest neighbors (k-NN) algorithm with $k = 10$, and we are looking to correctly classify the test points. Before k-NN we apply PCA and our proposed method. Results are shown in Figure 3. We deploy two variants of the proposed method: approximate the principal components as if they had equal importance and approximate the principal components while simultaneously also updating estimates of the singular values.

For comparison, we also show the sparse JL [26]. In this case, the target dimension is $p \in \{15, 30\}$ while the random transformation of size $p \times d$ only has three non-zero entries per column. More non-zeros did not have any significant effect on the classification accuracy while increasing (doubling, in this case) the target dimension p increases the classification accuracy by 10%. The results reported in the plots are averages for 100 realizations and the standard deviation is below 1%. Of course, the significant advantage of the JL approach over PCA is that no training is needed. The disadvantage
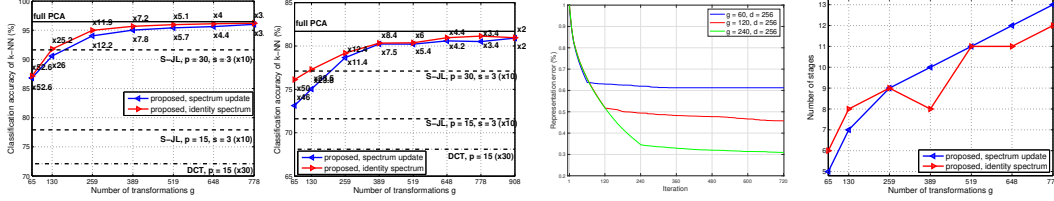
---

[1]https://github.com/cristian-rusu-research/fast-orthonormal-approximation

Figure 3: Classification accuracy obtained by the k-NN algorithm for the MNIST digits (left) and fashion (right) datasets as a function of the complexity of the proposed projections. Dimensionality reduction was done with $p = 15$ principal components. The bold text represents the speedup (FLOPS) compared to the cost of projecting with the unstructured, optimal, PCA components which takes $2pd$ operations. For the sparser JL the variable s is the number of non-zeros in each column.

Figure 4: For the USPS dataset ($d = 256$) we have, left: and $g \in \{60, 120, 240\}$ we show the evolution (mean and standard deviation) of the objective function with the number of iterations – again, in each case, the first g iterrations are the initialization steps; right: the number of stages in each transformation created by the proposed method as a function of the number of extended orthogonal Givens transformations g.

Table 1: Average classification accuracies for k-NN when using PCA projections and our approximations without spectrum update, for various datasets. We also show the speedup (FLOPS and actual running time) and the number of features selected in the calculations as a proportion out of the total d (see also Remark 2). Results are averaged over 100 random realizations (train/test splits).

DATASET	FULL PCA	PROPOSED ALGORITHM			
	ACCURACY	ACCURACY	SPEEDUP(FLOPS)	SPEEDUP(TIME)	SELECTION
PENDIGITS	95 ± 0.7	91 ± 2.1	$\times 1.6$	$\times 1.1$	1
ISOLET	92 ± 0.4	90 ± 1.0	$\times 12$	$\times 10.1$	1
USPS	95 ± 1.2	94 ± 0.8	$\times 7.7$	$\times 4.7$	0.64
UCI	90 ± 1.9	87 ± 1.5	$\times 2.5$	$\times 1.6$	0.72
20NEWS	80 ± 3.1	77 ± 2.1	$\times 3.1$	$\times 2.5$	0.3
EMNIST DIGITS	97 ± 2.5	95 ± 1.8	$\times 13$	$\times 11.3$	0.37
MNIST 8M	96 ± 2.0	94 ± 0.9	$\times 15$	$\times 13.7$	0.28

is that if we choose greedily the target projection dimension, i.e., low p, the accuracy degrades significantly for JL. Results are identical for PCA when p is 15 of 30.

Since we are dealing with image data, we also project using the discrete cosine transform (DCT). For the digits dataset, the performance is poor but, surprisingly, for the fashion dataset, this approach is competitive given the large speedup it reports (we used the sparse fast Fourier transform [23] as we want only the largest p components). We have also performed the projection by fast wavelet transforms with 'haar' and several of the Daubechies 'dbx' filters but the results were always similar to that of the DCT. For clarity of exposition, we did not add these results to the figures.

Our proposed methods report a clear trade-off between the classification accuracy and the numerical complexity of the projections. If we insist on an accuracy level close (within 1–2%) of the full PCA then the speedup is only about x3. Reasonable accuracy is obtained for a speedup of x4–x5 after which the results degrade quickly. For $p = 15$ better performance seems impossible via randomization. In this figure, the speedup is measured in terms of the number of operations (FLOPS).

Finally, in Figure 6 (left) we compare our proposed methods against the sparse PCA on MNIST digits. sPCA performs exceptionally well in terms of the classification accuracy given the computational budget (a similar result is replicated for MNIST fashion). On other datasets where the principal components capture some global features (not local like in our example) we expect this performance to degrade. The training time of sPCA exceeds by 60% the running time of PCA plus that of our method. We used the implementation of Wang et al. [48]. Because of ideas in Remark 1, we perform only the finally useful calculations and further reduce the computational cost on average by one third (these are accounted for already in the numbers in the plots).
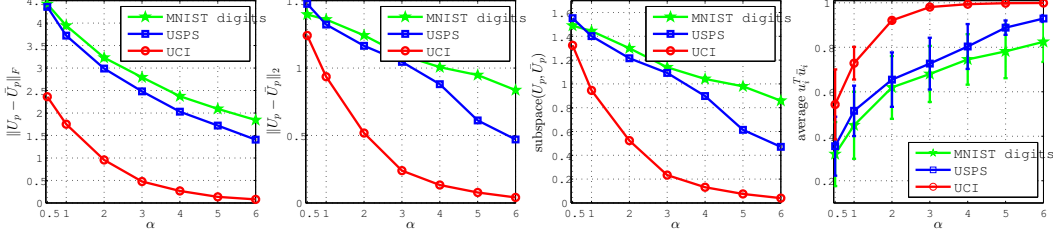
9

Figure 5: We show for three datasets used in Section 5.3, as a function of the number of Givens transformations g in $\bar{\mathbf{U}}_p$ that goes like $g = \alpha p \log_2 d$, the Frobenius (left most) and operator norms errors (second from the left), angle between subspaces distance measured in radians (third from the left) and correlations (right most shows the average correlations but also minimum and maximum values). The results are averaged over 10 realizations (random training/testing data samples) but the variance is always below 0.05.

5.3 Experiments on other datasets

The 20-newsgroups dataset consists of 18827 articles from 20 newsgroups (approximately 1000 per class). The data set was tokenized using the rainbow package (www.cs.cmu.edu/ mccallum/bow/rainbow). Each article is represented by a word-count vector for the $d = 2 \times 10^4$ common words in the vocabulary. For this dataset we have $N_{\text{test}} = 5648$, $p = 200$, and as shown in Figure 6 (right) in this case we outperform sparse PCA.

We also apply our algorithm to several other popular dataset from the literature: PENDIG-ITS (www.ics.uci.edu/~mlearn/MLRepository.html) with 10 classes and $d = 16$, $N = 7494$, $N_{\text{test}} = 3498$, $p = 4$; ISOLET (archive.ics.uci.edu/ml/datasets/isolet) with 26 classes and $d = 617$, $N = 6238$, $N_{\text{test}} = 1559$, $p = 150$; USPS (github.com/darshanbagul/USPS_Digit_Classification) with 10 classes and $d = 256$, $N = 7291$, $N_{\text{test}} = 2007$, $p = 12$; UCI (ftp.ics.uci.edu/pub/machine-learning-databases/optdigits) with 10 classes and $d = 64$, $N = 3823$, $N_{\text{test}} = 1797$, $p = 6$; EMNIST digits (nist.gov/itl/iad/image-group/emnist-dataset) with 10 classes and $d = 784$, $N = 24 \times 10^4$, $N_{\text{test}} = 4 \times 10^4$, $p = 15$; MNIST 8m (leon.bottou.org/papers/loosli-canu-bottou-2006) with 10 classes and $d = 784$, $N = 6.4 \times 10^6$, $N_{\text{test}} = 1.7 \times 10^6$, $p = 15$. All the results are shown in Table 1. Here we provide two measures for the speedup: the FLOPS (number of arithmetic operations, additions and multiplications) and the actual running time (in seconds). For the time speedup, we first computed $\bar{\mathbf{U}}$ and then implemented the matrix-vector multiplication and compared it to the generic matrix-vector multiplication with $\mathbf{U}$, both compiled in C using the gcc compiler and –O3 flag. Once computed by the proposed algorithm, the transformation could also be hardcoded and a further speedup improvement could be achieved. The running time speedups are slightly below the FLOPS speedups due to overhead in modern CPUs (fetching data, register loading etc.). Aside the computational benefits of the proposed transformations we note that while complex instruction set computing machines are closing the speedup gap in terms of running time the price is higher energy consumption and more complex execution pipelines/circuitry.

In Figure 4, for the USPS dataset, we provide insights into the behavior of the proposed algorithm with each iteration and the number of stages in each transformation that we construct. Results are similar to the ones shown for the random orthogonal approximation. We observe again that the initialization step is very efficient in reducing the approximation error (especially when compared to the iterative process that follows) and that the number of stages that have to be applied sequentially can further improve the running time when implementing the proposed transformations.

Finally, one of the most famous applications of PCA is in the field of computer vision for the problem of human face recognition. The eigenfaces [41] approach was used successfully for face recognition and classification tasks. Here, we want to reproduce the famous eigenfaces by using the proposed methods. The original eigenfaces and their approximations (sparse eigenfaces [51]), with different g and therefore different levels of detail, are shown in Figure 7.

5.4 Results on other approximation errors

In Section 3.1 we have explored several other ways to measure the approximation accuracy of the proposed algorithm. We note that in the proposed method we keep the Frobenius norm objective function but we also measure the operator norm, the angle between subspaces distance and the
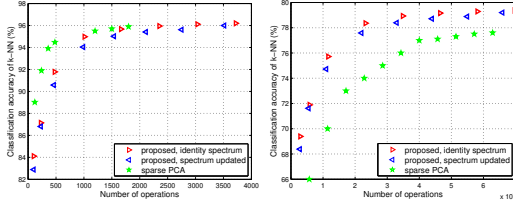
Figure 6: Classification accuracy versus number of operations on the MNIST digits (left) and 20NEWS (right) datasets for our proposed methods and the sparse PCA method [48, 52].

Figure 7: A few eigenfaces obtained by the optimal PCA (top) and by our proposed method with $g = 3059$ and $g = 1020$ (middle and bottom, respectively). The projection speedup (FLOPS) is x4.1 and x13.9, respectively.

Table 2: Speed-up achieved with the proposed approximations $\bar{\mathbf{U}}_p$ against a vanilla C implementation of dense matrix-vector multiplication and BLAS Level 2 functions, in both cases with no parallelism.

DATASET:	ISOLET	USPS	EMNIST DIGITS	MNIST 8M
C LANGUAGE (TIME)	×10.1	×4.7	×11.3	×13.7
BLAS (TIME)	×4.8	×2.5	×5.7	×6.8

average/minimum/maximum correlations. All results for three datasets are shown in Figure 5, the choices of parameters are the same as in Section 5.3. The first observation is that the approximation errors increase with larger d (as already clear from Remark 4 and Theorem 3), i.e, best results are obtained for UCI ($d = 64$) and the worst for MNIST ($d = 400$). Second, note that improving the Frobenius norm error we also improve all the other error measures. Thirdly, note the similar behavior between the operator norm and subspace distance errors. It is interesting to observe that the two plots seem to suggest experimentally that $\arccos(\sigma_{\min}(\mathbf{U}_p^T \bar{\mathbf{U}}_p)) \approx 1 - \sigma_{\max}(\mathbf{I}_p - \mathbf{U}_p^T \bar{\mathbf{U}}_p)$ which hold over a large array of number of Givens transformations g. Lastly, the two right-most plots highlight the connection between the operator norm optimization and the maximization of the lowest coherence between columns and their approximations (as per Theorem 5).

Depending on the application at hand we can choose how to measure the approximation error.

5.5 Details of the implementation

The source code attached to this paper is written for the Matlab environment. Besides the implementation of Algorithm 1, we also provide the code to efficiently perform the matrix-vector multiplication with the proposed $\bar{\mathbf{U}}$ and $\bar{\mathbf{U}}_p$, respectively (also taking into account Remark 3). Still, due to the characteristics of Matlab, our implementation is not faster than the dense matrix-vector multiplication "*". To provide an appropriate comparison, we implement the matrix-vector multiplication with our structures in the compiled lower-level programming language C. We provide comparisons of this implementation against two scenarios: a vanilla matrix-vector multiplication (using the appropriate ordering of the loops to exploit locality) and the Basic Linear Algebra Subprograms (BLAS) Level 2 routines for matrix-vector multiplication (SGEMV). We use the single thread variant of BLAS as for the dimensions d we consider the overhead of the parallel implementation is significant.

Applying the proposed transformations on batch features, i.e., matrix-matrix multiplications, would require careful application of the proposed transformations (2) to fully use the computing architecture and maximize performance.

6 Conclusions

This paper proposes a new matrix factorization algorithm for orthogonal matrices based on a class of structured matrices called extended orthogonal Givens transformations. We show that there is a trade-off between the computational complexity and accuracy of the approximations created by our approach. We apply our method to the approximation of a fixed number of principal components and show that, with a minor decrease in performance, we can reach significant computational benefits.

Future research directions include strengthening the theoretical guarantees since they are way above what we observe experimentally. Further, it would be of interest to improve the complexity of the proposed algorithm either by a parallel implementation or using randomization (e.g. computing a random subset of the $O(d^2)$ scores). As an immediate application, it would be interesting to apply our decomposition to the recently proposed unitary recurrent neural networks [50].

Acknowledgment

References

[1] N. Ailon and B. Chazelle. Approximate nearest neighbors and the fast Johnson-Lindenstrauss transform. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 557–563, 2006.

[2] T. Anderson, I. Olkin, and L. Underhill. Generation of random orthogonal matrices. *SIAM Journal on Scientific and Statistical Computing*, 8(4):625–629, 1987.

[3] A. Barvinok. Approximating orthogonal matrices by permutation matrices. *Pure and Applied Mathematics Quarterly*, 2:943–961, 2006.

[4] M.-A. Belabbas and P. J. Wolfe. Spectral methods in machine learning and new strategies for very large datasets. *Proceedings of the National Academy of Sciences*, 106(2):369–374, 2009.

[5] G. Beylkin, R. Coifman, and V. Rokhlin. Fast wavelet transforms and numerical algorithms I. *Communications on Pure and Applied Mathematics*, 44(2):141–183, 1991.

[6] R. Biloti, L. C. Matioli, and J. Yuan. A short note on a generalization of the Givens transformation. *Computers & Mathematics with Applications*, 66(1):56–61, 2013.

[7] Å. Björck and G. H. Golub. Numerical methods for computing angles between linear subspaces. *Mathematics of Computation*, 27(123):579–594, 1973.

[8] E. Borel. *Introduction geometrique a quelques theories physiques*. Gauthier-Villars, 1906.

[9] R. N. Bracewell. The fast Hartley transform. *Proceedings of the IEEE*, 72(8):1010–1018, 1984.

[10] R. P. Brent and F. T. Luk. The solution of singular value and symmetric eigenvalue problems on multiprocessor arrays. *SIAM J. Sci. Stat. Comput.*, 6:69–84, 1985.

[11] G. Cao, L. R. Bachega, and C. A. Bouman. The sparse matrix transform for covariance estimation and analysis of high dimensional signals. *IEEE Transactions on Image Processing*, 20(3):625–640, 2011.

[12] H. Chen and B. Zeng. New transforms tightly bounded by DCT and KLT. *IEEE Signal Processing Letters*, 19(6):344–347, 2012.

[13] B. Collins and C. Male. The strong asymptotic freeness of Haar and deterministic matrices. *Annales Scientifiques de l'Ecole Normale Superieure*, 47, 2011. doi: 10.24033/asens.2211.

[14] I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk. Iteratively reweighted least squares minimization for sparse recovery. *Communications on Pure and Applied Mathematics*, 63(1):1–38, 2010.

[15] P. Drineas, R. Kannan, and M. Mahoney. Fast Monte Carlo algorithms for matrices II: Computing a low-rank approximation to a matrix. *SIAM Journal on Computing*, 36(1):158–183, 2006.

[16] Z. Drmac and K. Veselic. New fast and accurate Jacobi SVD algorithm. II. *SIAM Journal on Matrix Analysis and Applications*, 29(4):1343–1362, 2008.

[17] B. J. Fino and V. R. Algazi. Unified matrix treatment of the fast Walsh-Hadamard transform. *IEEE Transactions on Computers*, C-25(11):1142–1146, 1976.

[18] C. B. Freksen and K. G. Larsen. On using Toeplitz and circulant matrices for Johnson-Lindenstrauss transforms. In *28th International Symposium on Algorithms and Computation*, pages 32:1–32:12, 2017.

[19] T. Frerix and J. Bruna. Approximating orthogonal matrices with effective Givens factorization. *arXiv:1905.05796*, 2019.

[20] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 1996.

[21] K. H. Greenewald and A. O. Hero. Kronecker PCA based spatio-temporal modeling of video for dismount classification. *Proceedings of SPIE - The International Society for Optical Engineering*, 9093, 2014.

[22] N. Halko, P. Martinsson, and J. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011.

[23] P. Indyk, M. Kapralov, and E. Price. (Nearly) sample-optimal sparse Fourier transform. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 480–499, 2014.

[24] S. Jain and J. Haupt. Convolutional approximations to linear dimensionality reduction operators. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5885–5889, 2017.

[25] K. Johansson. On random matrices from the compact classical groups. *Annals of Mathematics*, 145(3): 519–545, 1997.

[26] D. M. Kane and J. Nelson. Sparser Johnson-Lindenstrauss transforms. *J. ACM*, 61(1):4:1–4:23, 2014.

[27] S. Karnik, Z. Zhu, M. B. Wakin, J. Romberg, and M. A. Davenport. The fast Slepian transform. *Applied and Computational Harmonic Analysis*, 46(3):624 – 652, 2019.

[28] H. P. Kempen. On quadratic convergence of the special cyclic Jacobi method. *Numer. Math.*, 9:19–22, 1966.

[29] A. V. Knyazev and M. E. Argentati. Principal angles between subspaces in an A-based scalar product: algorithms and perturbation estimates. *SIAM Journal on Scientific Computing*, 23(6):2008–2040, 2002.

[30] R. Kondor, N. Teneva, and V. K. Garg. Multiresolution matrix factorization. In *Proceedings of the 31st International Conference on Machine Learning*, pages II–1620–II–1628, 2014.

[31] L. Le Magoarou, R. Gribonval, and N. Tremblay. Approximate fast graph Fourier transforms via multi-layer sparse approximations. *IEEE Transactions on Signal and Information Processing over Networks*, 4(2): 407–420, 2018.

[32] A. B. Lee, B. Nadler, and L. Wasserman. Treelets - an adaptive multi-scale basis for sparse unordered data. *Annals of Applied Statistics*, 2(2):435–471, 2008.

[33] J. Makhoul. A fast cosine transform in one and two dimensions. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(1):27–34, 1980.

[34] M. Mathieu and Y. LeCun. Fast approximation of rotations and Hessians matrices. *arXiv:1404.7195*, 2014.

[35] F. Merchant, T. Vatwani, A. Chattopadhyay, S. Raha, S. Nandy, R. Narayan, and R. Leupers. Efficient realization of Givens rotation through algorithm-architecture co-design for acceleration of QR factorization. *arXiv:1803.05320*, 2018.

[36] W. Rath. Fast Givens rotations for orthogonal similarity transformations. *Numerische Mathematik*, 40(1): 47–56, 1982.

[37] C. Rusu and J. Thompson. Learning fast sparsifying transforms. *IEEE Transactions on Signal Processing*, 65(16):4367–4378, 2017.

[38] P. Schonemann. A generalized solution of the orthogonal Procrustes problem. *Psychometrika*, 31(1):1–10, 1966.

[39] U. Shalit and G. Chechik. Coordinate-descent for learning orthogonal matrices through givens rotations. In *Proceedings of the 31st International Conference on Machine Learning*, pages I–548–I–556, 2014.

[40] B. Simon. CMV matrices: Five years after. *Journal of Computational and Applied Mathematics*, 208(1): 120–154, 2007.

[41] L. Sirovich and M. Kirby. Low-dimensional procedure for the characterization of human faces. *J. Opt. Soc. Am. A*, 4(3):519–524, 1987.

[42] C. Spengler, M. Huber, and B. C. Hiesmayr. A composite parameterization of unitary groups, density matrices and subspaces. *Journal of Physics A: Mathematical and Theoretical*, 43(38):385306, 2010.

[43] K. Stewart. Total variation approximation of random orthogonal matrices by Gaussian matrices. *Journal of Theoretical Probability*, 2019. ISSN 1572-9230. doi: 10.1007/s10959-019-00900-5.

[44] G. Strang. Fast transforms: banded matrices with banded inverses. *Proceedings of the National Academy of Sciences*, 107(28):12413–12416, 2010.

[45] T. E. Tilma and G. Sudarshan. Generalized Euler angle parametrization for SU(N). *J. Phys.*, A35: 10467–10501, 2002.

[46] J. A. Tropp. An introduction to matrix concentration inequalities. *Found. Trends Mach. Learn.*, 8(1–2): 1–230, 2015.

[47] C. Van Loan. *Computational Frameworks for the Fast Fourier Transform*. Society for Industrial and Applied Mathematics, 1992.

[48] Z. Wang, H. Lu, and H. Liu. Tighten after relax: Minimax-optimal sparse PCA in polynomial time. *Advances in neural information processing systems*, pages 3383–3391, 2014.

[49] M. V. Wickerhauser. Two fast approximate wavelet algorithms for image processing, classification, and recognition. *Optical Engineering*, 33:33 – 33 – 11, 1994.

[50] S. Wisdom, T. Powers, J. R. Hershey, J. L. Roux, and L. E. Atlas. Full-capacity unitary recurrent neural networks. In *Advances in Neural Information Processing Systems 29*, pages 4880–4888, 2016.

[51] H. Zhang, W. Liu, L. Dong, and Y. Wang. Sparse eigenfaces analysis for recognition. In *12th International Conference on Signal Processing*, pages 887–890, 2014.

[52] H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15(2):265–286, 2006.

Supplementary materials

More on Remark 3. We say that the orthonormal group has $O(d^2)$ degrees of freedom as it was established [2] that any orthonormal matrix $\mathbf{U}$ can be factored into a product as

$$\mathbf{U} = \left(\prod_{i=1}^{d} \prod_{j=i+1}^{d} \mathbf{G}_{ij}(\theta_{ij}) \right) \mathbf{D}, \tag{15}$$

where $\mathbf{D}$ is a diagonal matrix with entries only in $\{\pm 1\}$ and the $\mathbf{G}_{ij}(\theta_{ij})$ are Givens rotations, with angles θ_{ij}, i.e., we have $c = \cos\theta_{ij}$ and $s = \sin\theta_{ij}$ in (2). To generate a random orthonormal $\mathbf{U}$ we therefore need to generate random $\mathbf{D}$ (which are $\{\pm 1\}$ with equal probability) and $\frac{d(d-1)}{2}$ random angles $0 \le \theta_{ij} \le \pi/2$. These angles are mutually independent and it is known that their joint density function is a random variable

$$Z \propto \left(\prod_{k=2}^{d} \cos^{k-2}\theta_{1k} \right) \left(\prod_{k=3}^{d} \cos^{k-3}\theta_{2k} \right) \left(\prod_{k=d}^{d} \cos^{k-d}\theta_{(d-1)k} \right). \tag{16}$$

We define the beta random variable $\sqrt{y} = \cos\theta$ (and $\pm\sqrt{1-y} = \sin\theta$) with density

$$f(y, \alpha, \beta) = \frac{y^{\alpha-1}(1-y)^{\beta-1}}{B(\alpha, \beta)}, \ B(\alpha, \beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)} \text{ with } y \in [0, 1]. \tag{17}$$

Because we are interested in the computational complexity of a random orthonormal matrix we focus on the following three special cases: i) do nothing: $\tilde{\mathbf{G}}_{ij} = \begin{bmatrix} \pm 1 & 0 \\ 0 & \pm 1 \end{bmatrix}$; ii) permute coordinates: $\tilde{\mathbf{G}}_{ij} = \begin{bmatrix} 0 & \pm 1 \\ \pm 1 & 0 \end{bmatrix}$; and iii) $\tilde{\mathbf{G}}_{ij} \in \left\{ \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}, \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \right\}$. The first two cases perform no operations while the last performs only 4 (as compared to 6 for a general rotation). Unfortunately, the joint density in (16) does not seem to have a simlpe closed-form expression. As such we show in Figure 8 numerical results of the probability distribution of $\cos\theta$ over all angles θ_{ij}, which we observe numerically that approaches an exponential distribution $\lambda\exp(-\lambda c)$. Concentration around the three special cases does not occur and therefore a random orthonormal matrix will generally have computational complexity $O(d^2)$. For example, if we discretized the continuum of $c = \cos\theta$ then the probability that a random $\mathbf{U}$ is an approximate permutation matrix and therefore basically exhibits no numerical complexity is $(1 - \exp(-\lambda\epsilon))^{\frac{d(d-1)}{2}}$ for $0 < \epsilon \ll 1$, i.e., the probability that all $\frac{d(d-1)}{2}$ rotations have $\cos\theta \le \epsilon$. $\qquad\square$
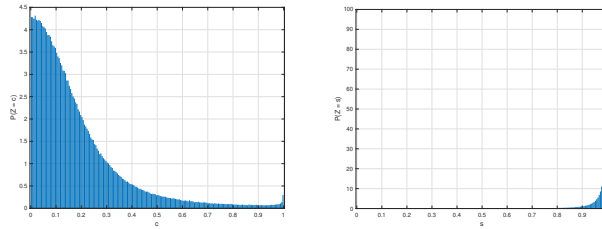

Figure 8: Experimental distribution on the entries (cosine and sine on the left and right, respectively) of the Givens rotations $\mathbf{G}_{ij}$ from (15). We observe numerically that $\cos\theta$ follows closely an exponential distribution $\lambda\exp(-\lambda c)$ with $\lambda \approx \sqrt{d}/2$ on the interval $[0, 1]$.

Proof of Theorem 1. For simplicity of exposition we will drop the sub-index k herein and therefore (4) develops to the following

$$\|\mathbf{L} - \mathbf{G}_{ij}\mathbf{N}\|_F^2 = \|\mathbf{L}\|_F^2 + \|\mathbf{G}_{ij}\mathbf{N}\|_F^2 - 2\mathrm{tr}(\mathbf{N}^T\mathbf{G}_{ij}^T\mathbf{L}) = \|\sigma_p\|_2^2 + \|\bar{\sigma}_p\|_2^2 - 2\mathrm{tr}(\mathbf{G}_{ij}^T\mathbf{L}\mathbf{N}^T), \tag{18}$$

where the Frobenius norms reduces to the ℓ_2 norms of the spectra and we have used the circular permutation property of the trace. It is convenient to denote $\mathbf{Z} = \mathbf{L}\mathbf{N}^T$ and the 2×2 matrix

$\mathbf{Z}_{\{i,j\}} = \begin{bmatrix} Z_{ii} & Z_{ij} \\ Z_{ji} & Z_{jj} \end{bmatrix}$. Given that $\mathbf{G}_{ij}$ performs operations only on rows i and j, the trace is

$$\text{tr}(\mathbf{G}_{ij}^T \mathbf{L} \mathbf{N}^T) = \sum_{k=1, k \notin \{i,j\}}^{d} Z_{kk} + \text{tr}(\tilde{\mathbf{G}}_{ij}^T \mathbf{Z}_{\{i,j\}}) = \text{tr}(\mathbf{Z}) + \text{tr}(\tilde{\mathbf{G}}_{ij}^T \mathbf{Z}_{\{i,j\}}) - \text{tr}(\mathbf{Z}_{\{i,j\}}). \quad (19)$$

To minimize the quantity in (18) we have to maximize (19) which is known as a Procrustes problem [38] whose solution is given by the polar decomposition of $\mathbf{Z}_{\{i,j\}}$ detailed in [20][Chapter 9.4.3]. Therefore, we set the optimal transformation to

$$\tilde{\mathbf{G}}_{ij}^{\star} = \mathbf{V}_1 \mathbf{V}_2^T, \ \mathbf{Z}_{\{i,j\}} = \mathbf{V}_1 \mathbf{S} \mathbf{V}_2^T, \quad (20)$$

where use the SVD of $\mathbf{Z}_{\{i,j\}}$ ($\mathbf{S} = \text{diag}(s_1, s_2)$ are the singular values). With this choice, we have

$$\max \text{tr}(\mathbf{G}_{ij}^T \mathbf{L} \mathbf{N}^T) = \text{tr}(\mathbf{Z}) + \text{tr}(\mathbf{S}) - \text{tr}(\mathbf{Z}_{\{i,j\}}) = \text{tr}(\mathbf{Z}) + \|\mathbf{Z}_{\{i,j\}}\|_* - \text{tr}(\mathbf{Z}_{\{i,j\}}) = \text{tr}(\mathbf{Z}) + C_{ij}. \quad (21)$$

We denote the nuclear norm $\|\mathbf{Z}_{\{i,j\}}\|_*$, i.e., the sum of the singular values s_1 and s_2 and we define

$$C_{ij} = \|\mathbf{Z}_{\{i,j\}}\|_* - \text{tr}(\mathbf{Z}_{\{i,j\}}). \quad (22)$$

Intuitively, the results (18), (19) follows after observing that: 1) the $\mathbf{G}_{ij}$ can be viewed as a perturbed identity matrix; 2) if $\mathbf{G}_{ij}$ is exactly $\mathbf{I}_{d \times d}$ then $\|\mathbf{L} - \mathbf{N}\|_F^2 = \|\sigma_p\|_2^2 + \|\bar{\sigma}_p\|_2^2 - 2\text{tr}(\mathbf{Z})$ while if $\mathbf{G}_{ij}$ is the optimal orthonormal transformation $\mathbf{Q}$ that minimizes (18) given by the Procrustes solution [38] then we have $\|\mathbf{L} - \mathbf{Q}\mathbf{N}\|_F^2 = \|\sigma_p\|_2^2 + \|\bar{\sigma}_p\|_2^2 - 2\|\mathbf{Z}\|_*$, where the last term is the nuclear norm of $\mathbf{Z}$; 3) therefore, we actually apply the identity transformation on all coordinates, i.e., the $\text{tr}(\mathbf{Z})$ term, while for the two chosen coordinates we apply the best (in the sense of reducing the error) orthogonal transformation whose contribution is the nuclear norm term $\|\mathbf{Z}_{\{i,j\}}\|_*$ and then correct for the trace term that was wrongly added initially in $\text{tr}(\mathbf{Z})$, by subtracting $\text{tr}(\mathbf{Z}_{\{i,j\}})$.

There are $d(d-1)/2$ quantities C_{ij} but they can be computed efficiently by noting that the singular values of $\mathbf{Z}_{\{i,j\}}$ are $s_{1,2} = \sqrt{\frac{1}{2}\left(\|\mathbf{Z}_{\{i,j\}}\|_F^2 \pm \sqrt{\|\mathbf{Z}_{\{i,j\}}\|_F^4 - 4\det(\mathbf{Z}_{\{i,j\}})^2}\right)}$. Observe that both singular values are of the form $\sigma_{1,2} = \sqrt{A \pm \sqrt{B}}$ which can be written as $\sqrt{X} \pm \sqrt{Y}$ where $X = \frac{A + \sqrt{A^2 - B}}{2}$ and $Y = A - X$. Written like this we can see that $\|\mathbf{Z}_{\{i,j\}}\|_* = \sigma_1 + \sigma_2 = 2\sqrt{X} = \sqrt{\|\mathbf{Z}_{\{i,j\}}\|_F^2 + 2|\det(\mathbf{Z}_{\{i,j\}})|}$. Depending on the sign of the determinant we have either $\|\mathbf{Z}_{\{i,j\}}\|_* = \sqrt{(Z_{ii} + Z_{jj})^2 + (Z_{ij} - Z_{ji})^2}$ or $\|\mathbf{Z}_{\{i,j\}}\|_* = \sqrt{(Z_{ii} - Z_{jj})^2 + (Z_{ij} + Z_{ji})^2}$, respectively. These give the final formulas for C_{ij} from (5). ∎

Proof of Theorem 2. Assume d is even and partition the set $\{1, \dots, d\}$ into $d/2$ pairs of indices (i_k, j_k). We therefore have

$$\sum_{k=1}^{d/2} C_{i_k j_k} = \sum_{k=1}^{d/2} (\|\mathbf{U}_{\{i_k, j_k\}}\|_* - U_{i_k i_k} - U_{j_k j_k}) = \sum_{k=1}^{d/2} \|\mathbf{U}_{\{i_k, j_k\}}\|_* - \text{tr}(\mathbf{U}). \quad (23)$$

As a side note, to maximization of this partitioned quantity is related to the weighted maximum matching algorithm (of maximum-cardinality matchings) on the graph with d nodes and with edge weights $C_{i_k j_k}$. With this choice of indices, the objective function becomes

$$\left\| \mathbf{U} - \prod_{k=1}^{d/2} \mathbf{G}_{i_k j_k} \right\|_F^2 = 2d - 2\text{tr}(\mathbf{U}) - 2\sum_{k=1}^{d/2} C_{i_k j_k} = 2d - 2\sum_{k=1}^{d/2} \|\mathbf{U}_{\{i_k j_k\}}\|_*. \quad (24)$$

We use the singular value decomposition of a generic $\mathbf{U}_{\{i,j\}} = \mathbf{V}_1 \mathbf{S} \mathbf{V}_2^T$, $\mathbf{S} = \text{diag}(\mathbf{s})$, and develop:

$$|\text{tr}(\mathbf{U}_{\{i,j\}})| = |\text{tr}(\mathbf{V}_1 \mathbf{S} \mathbf{V}_2^T)| = |\text{tr}(\mathbf{S} \mathbf{V}_2^T \mathbf{V}_1)| = |\text{tr}(\mathbf{S}\boldsymbol{\Delta})| = \left| \sum_{t=1}^{2} s_t \Delta_{tt} \right| \leq \Delta_{\max} \sum_{t=1}^{2} s_t = \Delta_{\max} \|\mathbf{U}_{\{i,j\}}\|_*,$$

$$(25)$$

where we have use the circular property of the trace and $\boldsymbol{\Delta} = \mathbf{V}_2^T \mathbf{V}_1$ where Δ_{tt} are its diagonal entries which obey $|\Delta_{tt}| \leq \Delta_{\max}$. We define the diagonal coherence as

$$\Delta_{\max} = \max\{|\Delta_{11}|, |\Delta_{22}|\}. \quad (26)$$

16

With (25), we can state that for the kth transformation that

$$\mathbb{E}[\|\mathbf{U}_{\{i_k,j_k\}}\|_*] \geq \frac{\pi}{2}\mathbb{E}[\text{tr}(\mathbf{U}_{\{i_k,j_k\}})], \tag{27}$$

where we have used the fact that $\mathbf{V}_1$ and $\mathbf{V}_2$ have the structure $\tilde{\mathbf{G}}_{ij}$ in (2) and therefore

$$\mathbb{E}[\Delta_{\max}] = \frac{1}{\pi^2}\iint_0^\pi |\cos(x)\cos(y) + \sin(x)\sin(y)|dxdy = \frac{2}{\pi}.$$

Finally, given an orthonormal $\mathbf{U}$ of size $d \times d$, $d \geq 4$, we use (24) and (27) to bound

$$\mathbb{E}\left[\left\|\mathbf{U} - \prod_{k=1}^{d/2}\mathbf{G}_{i_kj_k}\right\|_F^2\right] = 2d - 2\sum_{k=1}^{d/2}\mathbb{E}[\|\mathbf{U}_{\{i_k,j_k\}}\|_*] \leq 2d - \pi\sum_{k=1}^{d/2}\mathbb{E}[\text{tr}(\mathbf{U}_{\{i_k,j_k\}})]$$

$$\leq 2d - \pi\mathbb{E}[\text{tr}(\mathbf{U})] = 2d - \pi\mathbb{E}\left[\sum_{t=1}^d |U_{tt}|\right] = 2d - \sqrt{2\pi d}, \tag{28}$$

where we have used that $\mathbb{E}\left[\sum_{t=1}^d |U_{tt}|\right] = \sqrt{2\pi^{-1}d}$ because the diagonal elements of $\mathbf{U}$ can be viewed as Gaussian random variables with zero mean and standard deviation $d^{-1/2}$ (as the columns of $\mathbf{U}$ are normalized in the ℓ_2 norm) [43] and because the ℓ_1 norm of a standard Gaussian random vector of size d is $\sqrt{2\pi^{-1}}d$.

When d is odd, we extend the matrix with a zero column/row and "1" on the diagonal and the argument follows in the same way. In (24) we could use the expected value calculated in (37) but we reach a worse, lower, constant in (28) for the $-\sqrt{d}$ term and therefore a worse overall bound. ∎

Proof of Theorem 3. Given the orthonormal $\mathbf{U}$, by [20][Theorem 5.2.1], we can construct its QR factorization using a set of Givens rotations [20][Chapter 5.2.5]. After introducing zeros in the first r columns of $\mathbf{U}$, by left multiplication with Givens rotations, we reach its following partial triangularization

$$\mathbf{J}_{i_gj_g}\ldots\mathbf{J}_{i_1j_1}\mathbf{U} = \begin{bmatrix} \mathbf{D}'' & \mathbf{0}_{r\times(d-r)} \\ \mathbf{0}_{(d-r)\times r} & \mathbf{U}' \end{bmatrix}, \tag{29}$$

where the diagonal matrix $\mathbf{D}''$ of size $r \times r$ has entries $D''_{tt} \in \{\pm 1\}$ and $\mathbf{U}'$ of size $(d-r) \times (d-r)$ is orthonormal. To introduce the zeros on the tth column we need $(d-t)$ Givens rotations and therefore to bring $\mathbf{U}$ to the structure in (29) we need $g = \frac{r}{2}(2d - r - 1)$ Givens rotations which we have denoted $\mathbf{J}_{i_kj_k}$, $k = 1,\ldots,g$. We are exploiting the fact that the triangularization of an orthogonal matrix leads to a ± 1 diagonal. Then we might consider a good approximation to $\mathbf{U}$ the product $\bar{\mathbf{U}} = \mathbf{J}_{i_1j_1}^T\ldots\mathbf{J}_{i_gj_g}^T\mathbf{D}$. where $\mathbf{D} = \begin{bmatrix} \mathbf{D}'' & \mathbf{0}_{r\times(d-r)} \\ \mathbf{0}_{(d-r)\times r} & \mathbf{D}' \end{bmatrix}$ with $D'_{tt} = \text{sgn}(U'_{tt})$ and $\mathbf{D}''$ is taken from (29). The goal of the diagonal matrix $\mathbf{D}$ is to ensure that the product in $\bar{\mathbf{U}}$ has a nonnegative diagonal. Then, given g transforms we can bound

$$\|\mathbf{U} - \mathbf{J}_{i_1j_1}^T\ldots\mathbf{J}_{i_gj_g}^T\mathbf{D}\|_F^2 = 2(d - \lfloor r \rfloor) - 2\text{tr}(\mathbf{D}'\mathbf{U}'). \tag{30}$$

If we consider $\tilde{\mathbf{G}}_{ij}$ in (2) instead of the rotations $\mathbf{J}_{i_kj_k}$ then the quantity on the right becomes an upper bound, since Givens rotations are a special case of $\tilde{\mathbf{G}}_{ij}$ – we can always initialize the $\mathbf{G}_{i_kj_k}$ of Algorithm 1 with the $\mathbf{J}_{i_kj_k}$ defined above and the iterative procedure is guaranteed not to worsen the factorization. Therefore, the result follows after using $\mathbb{E}[\text{tr}(\mathbf{D}'\mathbf{U}')] = \mathbb{E}\left[\sum_{t=1}^{d-\lfloor r \rfloor} |U'_{tt}|\right] = \sqrt{2(d - \lfloor r \rfloor)\pi^{-1}}$. ∎

Proof of Theorem 4. First, we introduce the off-diagonal "norm", i.e., the square-root of the squared sum of the off-diagonal elements of an orthonormal matrix $\mathbf{U} \in \mathbb{R}^{d\times d}$ as

$$\text{off}(\mathbf{U})^2 = \sum_{t=1}^d \sum_{q=1,q\neq t}^d U_{tq}^2 = \|\mathbf{U}\|_F^2 - \sum_{t=1}^d U_{tt}^2 = d - \sum_{t=1}^d U_{tt}^2. \tag{31}$$

Better approximations $\bar{\mathbf{U}}$ of $\mathbf{U}$ lead to lower off$(\mathbf{U}\bar{\mathbf{U}}^T)$, as $\mathbf{U}\bar{\mathbf{U}}^T$ approaches the identity. If we use this measure, we reach the following result.
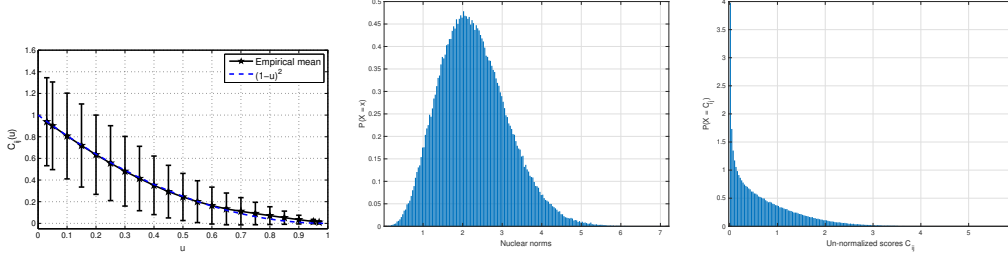
Figure 9: Left: empirical mean and standard deviation verification of Remark 5; Middle: empirical pdf for the nuclear norms of 2×2 sub-matrices from random orthonormal matrices; Right: empirical pdf for the scores for random $\mathbf{U}_{\{i,j\}}$ if the entries are standard Gaussian random variables (without the normalization factor $d^{-1/2}$).

We use the fact that C_{ij} is added to the diagonal of $\mathbf{U}$, but only to U_{ii} and U_{jj}. The quantity $f(\gamma) = (U_{ii} + \gamma C_{ij})^2 + (U_{jj} + (1-\gamma)C_{ij})^2, \gamma \in \mathbb{R}$, is minimized for $C_{ij} \neq 0$ when $\gamma_0 = \frac{1}{2} + \frac{U_{jj} - U_{ii}}{2C_{ij}}$ (and therefore $f(\gamma_0) = \frac{1}{2}\|\mathbf{U}_{\{i,j\}}\|_*^2$) which leads to

$$\text{off}(\mathbf{U}\mathbf{G}_{ij}^T)^2 = d - \sum_{t=1, t \notin \{i,j\}}^{d} U_{tt}^2 - f(\gamma) \leq d - \sum_{t=1, t \notin \{i,j\}}^{d} U_{tt}^2 - f(\gamma_0)$$

$$= d - \sum_{t=1}^{d} U_{tt}^2 + U_{ii}^2 + U_{jj}^2 - f(\gamma_0) = \text{off}(\mathbf{U})^2 + \frac{(U_{ii} - U_{jj})^2}{2} - C_{ij}(U_{ii} + U_{jj}) - \frac{C_{ij}^2}{2} \quad (32)$$

$$= \text{off}(\mathbf{U})^2 + \frac{(U_{ii} - U_{jj})^2}{2} - \frac{\|\mathbf{U}_{\{i,j\}}\|_*^2 - (U_{ii} + U_{jj})^2}{2} = \text{off}(\mathbf{U})^2 + U_{ii}^2 + U_{jj}^2 - \frac{\|\mathbf{U}_{\{i,j\}}\|_*^2}{2}.$$

We now use de explicit formulas for the singular values of a 2×2 matrix and the fact that

$$\|\mathbf{U}_{\{i,j\}}\|_*^2 = s_1^2 + s_2^2 + 2s_1 s_2 = \|\mathbf{U}_{\{i,j\}}\|_F^2 + 2|\det(\mathbf{U}_{\{i,j\}})|, \quad (33)$$

and expand this expression to get in (32)

$$\text{if } \det(\mathbf{U}_{\{i,j\}}) \geq 0: \quad \text{off}(\mathbf{U}\mathbf{G}_{ij}^T)^2 \leq \text{off}(\mathbf{U})^2 + \frac{(U_{ii} - U_{jj})^2 - (U_{ij} - U_{ji})^2}{2},$$

$$\text{if } \det(\mathbf{U}_{\{i,j\}}) < 0: \quad \text{off}(\mathbf{U}\mathbf{G}_{ij}^T)^2 \leq \text{off}(\mathbf{U})^2 + \frac{(U_{ii} + U_{jj})^2 - (U_{ij} + U_{ji})^2}{2}.$$

Therefore, to guarantee $\text{off}(\mathbf{U}\mathbf{G}_{ij}^T)^2 \leq \text{off}(\mathbf{U})^2$ we need $2(U_{ii}^2 + U_{jj}^2) \leq \|\mathbf{U}_{\{i,j\}}\|_*^2$ which is equivalent to

$$\text{if } \det(\mathbf{U}_{\{i,j\}}) \geq 0: (U_{ii} - U_{jj})^2 \leq (U_{ij} - U_{ji})^2,$$

$$\text{if } \det(\mathbf{U}_{\{i,j\}}) < 0: (U_{ii} + U_{jj})^2 \leq (U_{ij} + U_{ji})^2. \quad (34)$$

In this paper we assume that $\mathbf{U}$ is taken randomly from the Haar measure [25] and then modified to have positive diagonal. Therefore, we have $U_{tt} \geq 0$ for all t by construction, otherwise we would just consider the update $U_{tt} \leftarrow \text{sign}(U_{tt})U_{tt}$. Moreover, as the algorithm progresses we continue to have that $\det(\mathbf{U}_{\{i,j\}}) \geq 0$ because each $\mathbf{G}_{ij}$ adds a positive amount (the C_{ij} value) to the diagonal elements U_{ii} and U_{jj} thus converging towards $\mathbf{I}_{d \times d}$ in the sense of (36).

In a similar way we can construct a lower bound. Assuming w.l.o.g. that $U_{ii} \geq U_{jj} \geq 0$, the quantity $f(\gamma) = (U_{ii} + \gamma C_{ij})^2 + (U_{jj} + (1-\gamma)C_{ij})^2$ is maximized when $\gamma_0 = 1$ and therefore $f(\gamma_0) = (U_{ii} + C_{ij})^2 + U_{jj}^2$ which, similarly to (32), leads to

$$\text{off}(\mathbf{U}\mathbf{G}_{ij}^T)^2 \geq \text{off}(\mathbf{U})^2 - C_{ij}(2U_{ii} + C_{ij}). \blacksquare \quad (35)$$

More on Remark 4. Understanding the properties of these C_{ij} is of crucial importance for our approach. First, notice the effect one generalized Givens transformation has: we have $\|\mathbf{U} - \mathbf{G}_{ij}\|_F^2 = \|\mathbf{U}\mathbf{G}_{ij}^T - \mathbf{I}\|_F^2 = 2d - 2\text{tr}(\mathbf{U}\mathbf{G}_{ij}^T)$, which together with (18) leads to

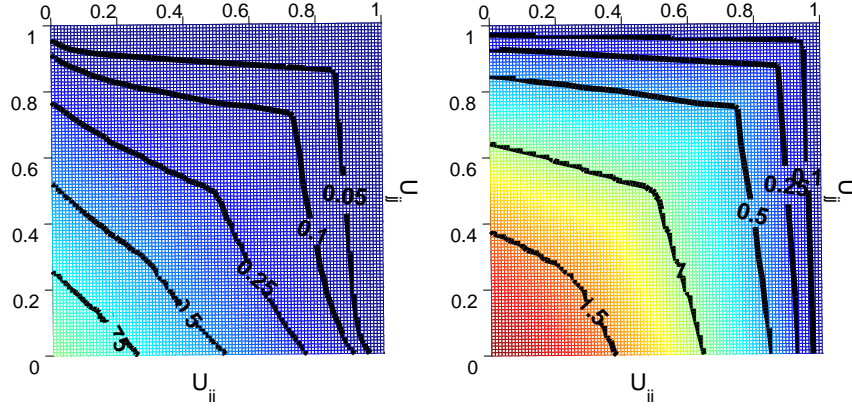$$\text{tr}(\mathbf{U}\mathbf{G}_{ij}^T) = \text{tr}(\mathbf{U}) + C_{ij} \leq d. \quad (36)$$

Figure 10: Empirical mean (left) and maximum (right) values of C_{ij} for the toy matrix $\mathbf{U}_{\{i,j\}}$ where the diagonal elements U_{ii} and U_{jj} (both in $[0,1]$) are fixed and the off-diagonals entries denoted z_1 and z_2 are uniform random variables in the interval $\left[-\sqrt{1-\beta}, \sqrt{1-\beta}\right]$ where $\beta = \max(U_{ii}^2, U_{jj}^2)$ since the columns and rows of $\mathbf{U}$ are ℓ_2 normalized. Low values (close to 0) are coded as dark blue while high values (close to 2) are coded as dark red.

In this sense, the $\mathbf{G}_{ij}$ "pushes" $\mathbf{UG}_{ij}^T$ towards the identity matrix by "contributing" C_{ij} to the diagonal of $\mathbf{U}$, i.e., we are estimating the inverse of $\mathbf{U}$ which in this case is just the transpose. Notice that $0 \leq C_{ij} \leq 4$. The minimum is achieved for symmetric positive semidefinite matrices (because in this case the eigenvalues and singular values are the same and therefore the nuclear norm equals the trace) and the maximum for $\mathbf{U}_{\{i,j\}} = -\mathbf{I}_{2\times2}$. This immediately leads to a local optimality condition for our approach: there is no $\mathbf{G}_{ij}$ to improve the approximation if all $\mathbf{U}_{\{i,j\}}$ are symmetric positive definite. Assume now we are given a random orthogonal matrix. Because the singular values $s_{1,2}$ depend on the entries of $\mathbf{U}_{\{i,j\}}$ which we model as Gaussian random variables with zero mean and standard deviation $d^{-1/2}$ [43], we have by direct calculation that $\mathbb{E}[s_1] \approx 1.7724d^{-1/2}$ and $\mathbb{E}[s_2] \approx 0.5190d^{-1/2}$ which leads to $\mathbb{E}[\|\mathbf{U}_{\{i,j\}}\|_*] \approx 2.2914d^{-1/2}$. The trace is the sum of two absolute value Gaussian random variables and therefore $\mathbb{E}[\text{tr}(\mathbf{U}_{\{i,j\}})] = 2\sqrt{2(\pi d)^{-1}}$ which leads to

$$\mathbb{E}[C_{ij}] \approx 0.6956d^{-1/2}.\square \tag{37}$$

More on Remark 5. To see how the scores C_{ij} depend on the diagonal entries of our toy model, by direct calculation with the truncated standard Gaussian random variables we have that

$$\mathbb{E}[C_{ij}(u)] = \frac{1}{2\pi\text{erf}^2\left(\sqrt{\frac{1-u^2}{2}}\right)} \iint_{-\sqrt{1-u^2}}^{\sqrt{1-u^2}} \exp\left(-\frac{z_1^2 + z_2^2}{2}\right) C_{ij}(u) \, dz_1 dz_2 \propto (1-u)^2. \tag{38}$$

We show in Figure 9 the empirical results (mean and standard deviation of C_{ij}) on the toy matrix $\mathbf{U}_{\{i,j\}} = \begin{bmatrix} u & z_2 \\ z_1 & u \end{bmatrix}$ for $0 < u < 1$. The empirical mean follows the approximation in Remark 5 (and is tight for $u \leq 0.7$) while we notice that the variance is high for almost the whole interval. In Figure 10 we show the average (left) and maximum (right) costs C_{ij} achieved for another toy model where the diagonal elements are distinct $\mathbf{U}_{\{i,j\}} = \begin{bmatrix} U_{ii} & z_2 \\ z_1 & U_{jj} \end{bmatrix}$ for $0 \leq U_{ii}, U_{jj} \leq 1$.

Finally, notice that when $\mathbf{U}_{\{i,j\}} = \begin{bmatrix} u & U_{ij} \\ U_{ij} & u \end{bmatrix}$ we have $C_{ij} = 2(U_{ij} - u)$ if $u \leq U_{ij}$ and zero otherwise, and when $\mathbf{U}_{\{i,j\}} = \begin{bmatrix} u & -U_{ij} \\ U_{ij} & u \end{bmatrix}$ we have $C_{ij} = 2\sqrt{u^2 + U_{ij}^2} - 2u$ indicating that skew symmetric sub-matrices have higher C_{ij} than symmetric ones, in general. $\square$

Proof of Remark 7. We use the fact that columns of $\mathbf{U}$ and $\bar{\mathbf{U}}$ have unit norm, and the fact that the Frobenius norm is entrywise:

$$\|(\mathbf{U} - \bar{\mathbf{U}})\boldsymbol{\Sigma}_d\|_F^2 = \sum_{i=1}^{d} \sigma_i \|\mathbf{u}_i - \bar{\mathbf{u}}_i\|_2^2 = \sum_{i=1}^{d} \sigma_i(\|\mathbf{u}_i\|_2^2 + \|\bar{\mathbf{u}}_i\|_2^2 - 2\mathbf{u}_i^T\bar{\mathbf{u}}_i) = 2\sum_{i=1}^{d}\sigma_i(1 - \cos(\theta_i)). \square$$

$$(39)$$

Proof of Remark 8. First, note that given any proper norm which is invariant to orthonormal transformations, we have that $\|\mathbf{U} - \bar{\mathbf{U}}\|_2 = \|\mathbf{I} - \mathbf{U}^T\bar{\mathbf{U}}\|_2$ and we denote the error matrix $\mathbf{E} = \mathbf{I} - \mathbf{U}^T\bar{\mathbf{U}}$. Now, start from the error matrix $\mathbf{E} = \mathbf{I} - \mathbf{U}^T\bar{\mathbf{U}}$ and use the fact that $\mathbf{U}^T\bar{\mathbf{U}}$ is orthonormal to notice by straightforward calculation that:

$$\mathbf{E}^T\mathbf{E} = (\mathbf{I} - \bar{\mathbf{U}}^T\mathbf{U})(\mathbf{I} - \mathbf{U}^T\bar{\mathbf{U}}) = \mathbf{I} - \mathbf{U}^T\bar{\mathbf{U}} - \bar{\mathbf{U}}^T\mathbf{U} + \mathbf{I} = \mathbf{E} + \mathbf{E}^T. \tag{40}$$

Denote now a complex-valued eigenvalue of $\mathbf{E}$ by $z_k = a_k + ib_k$, then because of the equality above we have that $z_k^* z_k = 2\Re(z_k)$ which in turn can be written as $a_k^2 + b_k^2 = 2a_k$ or $(a_k - 1)^2 + b_k^2 = 1$. This is the equation of a circle of radius one centered at $(1, 0)$ in the complex plane. Another way to view this result is to observe that $\mathbf{U}^T\bar{\mathbf{U}}$ is orthonormal an therefore its spectrum is on the unit circle and then $\mathbf{I}$ shifts the whole spectrum to the right by one unit.

Moreover, we also have that $\mathbf{E}\mathbf{E}^T = \mathbf{E} + \mathbf{E}^T$ and therefore $\mathbf{E}^T\mathbf{E} = \mathbf{E}\mathbf{E}^T$ which means that $\mathbf{E}$ is a normal matrix. As such, the singular values of $\mathbf{E}$ are the absolute values of its eigenvalues and therefore $\|\mathbf{E}\|_2 \le 2$. $\square$

Proof of Theorem 5. The proof is based on the Gershgorin circle theorem (detailed in Chapter 7.2 of [20]) applied to the error matrix $\mathbf{E} = \mathbf{I} - \mathbf{U}^T\bar{\mathbf{U}}$. If we denote $\mathbf{Q} = \mathbf{U}^T\bar{\mathbf{U}}$, we then have for the ith eigenvalue of $\mathbf{E}$ that

$$|\lambda_i - 1 + \mathbf{u}_i^T\bar{\mathbf{u}}_i| \le \sum_{j \ne i} |Q_{ij}| \le \sqrt{(d-1)(1 - (\mathbf{u}_i^T\bar{\mathbf{u}}_i)^2)}. \tag{41}$$

The last inequality on the right hand side comes from the $\ell_1 - \ell_2$ inequality $\|\mathbf{x}\|_1 \le \sqrt{d-1}\|\mathbf{x}\|_2$ applied to the vectors of size $d - 1$ (the rows of $\mathbf{Q}$ except for their diagonal elements and whose ℓ_2 norm is $\sqrt{1 - Q_{ii}^2}$). Returning to (41), for the term on the left hand side we have by the reverse triangle inequality ($|a - b| \ge ||a| - |b||$) that

$$||\lambda_i| - |1 - \mathbf{u}_i^T\bar{\mathbf{u}}_i|| \le |\lambda_i - (1 - \mathbf{u}_i^T\bar{\mathbf{u}}_i)|. \tag{42}$$

Finally combining (41) and (42) we have that

$$0 \le |\lambda_i| \le 1 - \mathbf{u}_i^T\bar{\mathbf{u}}_i + \sqrt{(d-1)(1 - (\mathbf{u}_i^T\bar{\mathbf{u}}_i)^2)}. \tag{43}$$

This inequality holds for $0 \le \mathbf{u}_i^T\bar{\mathbf{u}}_i \le 1$ and $d \ge 2$. The expression on the right-hand side is monotonic decreasing for $\mathbf{u}_i^T\bar{\mathbf{u}}_i \ge 0$. As such, the largest upper bound happens for $i = \arg\min_i \mathbf{u}_i^T\bar{\mathbf{u}}_i$.

And because, as shown in Remark 8, the error matrix is a normal matrix we get a bound on the operator norm (the singular values of a normal matrix are the absolute values of its eigenvalues). ∎