Generalized Low Rank Approximations of Matrices Revisited

Jun Liu, Songcan Chen*, Zhi-Hua Zhou, *Senior Member, IEEE,* and Xiaoyang Tan, *Member, IEEE*

***Abstract*—Compared to Singular Value Decomposition (SVD), Generalized Low Rank Approximations of Matrices (GLRAM) can consume less computation time, obtain higher compression ratio, and yield competitive classification performance. GLRAM has been successfully applied to applications such as image compression and retrieval, and quite a few extensions have been successively proposed. However, in literature, some basic properties and crucial problems with regard to GLRAM have not been explored or solved yet. For this sake, we revisit GLRAM in this paper. First, we reveal such a close relationship between GLRAM and SVD that GLRAM's objective function is identical to SVD's objective function except the imposed constraints. Second, we derive a lower-bound of GLRAM's objective function, and discuss when the lower-bound can be touched. Moreover, from the viewpoint of minimizing the lower-bound, we answer one open problem raised by Ye (Machine Learning, 2005), i.e., a theoretical justification of the experimental phenomenon that, under given number of reduced dimension, the lowest reconstruction error is obtained when the left and right transformations have equal number of columns. Third, we explore when and why GLRAM can perform well in terms of compression, which is a fundamental problem concerning the usability of GLRAM.**

***Index Terms*—Dimensionality Reduction, Singular Value Decomposition, Generalized Low Rank Approximations of Matrices, Reconstruction Error**

I. Introduction

To obtain a compact representation of data, one usually employs dimensionality reduction, with Singular Value Decomposition (SVD) [1] being one of the most well-known methods. SVD has the appealing property that it can achieve the smallest reconstruction error among all the rank-k approximations, and has been successfully applied to face recognition [2], [3], information retrieval [4], [5], etc.

Applications of SVD to high-dimensional data, such as images and videos, quickly run up against practical computational limits, mainly due to the high time and space complexities of the SVD computation for large matrices [1], [6]. To deal with the problem of high space complexity, incremental algorithms have been proposed in [7], [8], [9]. To speed the computation of SVD, random sampling has been employed in [10], [11], [12]. Like the traditional SVD, these improvements on SVD all treat data as one-dimensional vector patterns.

Recently, Ye [6], [13] proposed the Generalized Low Rank Approximations of Matrices (GLRAM) method, which treats data as the native two-dimensional matrix patterns. Benefited by the two-dimensional representation of data, GLRAM was reported to consume less computation time and yield higher compression ratio than SVD in applications such as image compression and retrieval. Compared to the two-dimensional methods such as two-dimensional Principal Component Analysis [14], GLRAM employs two-sided transformations rather than single-sided one, and yields higher compression ratio. Moreover, GLRAM is applied in [6] as a pre-processing step for SVD to devise the GLRAM+SVD method, i.e., performing SVD after the intermediate dimensionality reduction stage using GLRAM. Compared to GLRAM, GLRAM+SVD achieves a significant reduction of the reconstruction error, while keeping the computation cost small. GLRAM has been successfully applied in applications such as image compression and retrieval, achieving competitive classification performance to SVD. Since the main goal of GLRAM itself is to obtain compact representation of data, we study GLRAM from the viewpoint of compression in this paper.

It is generally believed that GLRAM does not admit a closed-form solution, and thus Ye employed an iterative procedure to compute the so-involved two-sided transformations in [6], [13]. Following Ye's pioneer work, researchers have carried out quite a few studies that focus on non-iterative extensions for accelerating the computation of the two-sided transformations. Ding and Ye [15] proposed a two-dimensional SVD (2dSVD) based on row-row and column-column covariance matrices and studied the optimality properties of the 2dSVD; Zhang et al. [16] adopted a natural representation for images using eigenimages; Inoue and Urahama [17] proposed a dyadic SVD based on the higher-order SVD presented by Lathauwer et al. [18]; Liang and Shi [19] claimed to obtain an analytical GLRAM algorithm, but as pointed out in [20], [21], such a claim is incorrect and the so-called analytical GLRAM is in fact a variant of 2dSVD; Liu and Chen [21] proposed a Non-Iterative GLRAM (NIGLRAM) to approximately optimize the objective function of GLRAM; and Inoue and Urahama [20] proved that some non-iterative methods are in fact equivalent to some extent. Compared to GLRAM, these non-iterative extensions can be computed more efficiently and meanwhile can obtain competitive compression ratio and classification performance.

GLRAM has also been extended to the scenario of tensor representation [22], [23], [24], [25]. For example, the MPCA (Multilinear Principal Component Analysis) proposed by Lu et al. [22] extends GLRAM to tensor with modes over 2; and the

Jun Liu, Songcan Chen, and Xiaoyang Tan are with the Department of Computer Science & Engineering, Nanjing University of Aeronautics & Astronautics Nanjing, 210016, P.R. China. Emails: {j.liu, s.chen, x.tan}@nuaa.edu.cn.

Zhi-Hua Zhou is with the National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, P.R. China. Email: zhouzh@nju.edu.cn.

*Corresponding author (S.C. Chen), Tel: +86-25-84892452, Fax: +86-25-84498069.

element rearrangement technique proposed by Yan et al. [25] can be used to improve the approximation performance of GLRAM.

In a recent study, Liang and Shi [26] conducted a theoretical analysis on GLRAM. Firstly, based on the covariance matrix of Principal Component Analysis (PCA) [2], they gave the lower and upper bounds of GLRAM's objective function, and showed that the reconstruction error of GLRAM is not smaller than that of PCA when the reduced dimension is same. Secondly, they proposed a non-iterative GLRAM algorithm to compute the transformation matrices in a similar way to NIGLRAM [21].

Despite of the successes achieved by GLRAM and its extensions, some basic properties and crucial problems with regard to GLRAM have not been explored or solved yet in literature. For example, as Ye pointed out in [6], there is not a theoretical justification of the experimental phenomenon that the lowest reconstruction error is obtained when the left and right transformation matrices have equal number of columns. Likewise, we also concern the following two points: 1) the relationship between GLRAM and SVD has not been well studied theoretically, although GLRAM has been extensively compared with SVD empirically in [6]; and 2) when and why GLRAM can perform well in terms of compression, which is a fundamental problem concerning the usability of GLRAM.

In this paper, we revisit GLRAM to answer the aforementioned problems. The studies presented in this paper can be extended to benefit quite a few two-dimensional and tensor-based methods [18], [22], [23], [24], [25], [27], [28], which have attracted much attention from researches in areas of machine learning, computer vision, neural computation, etc.

In what follows, we briefly review SVD and GLRAM in Section II, discuss on the relationship between GLRAM and SVD in Section III, give a new lower-bound of GLRAM's objective function to answer the problem why the lowest reconstruction error can be obtained when the left and right transformation matrices have equal number of columns in Section IV, explore when and why GLRAM can perform well in terms of compression by studying the relationship among some defined criteria in Section V, conduct experiments in Section VI, and conclude this paper in Section VII.

II. A Brief Review of SVD and GLRAM

A. Notations

The major notations employed in this paper are summarized in Table I. Suppose the dataset is composed of N matrix patterns $A_i \in \mathbb{R}^{e\times f}, i = 1, 2, \ldots, N$, and $\boldsymbol{a}_i = vec(A_i)$ denotes the d-dimensional vector pattern obtained by sequentially concatenating the columns of A_i. Let I_q denote a $q \times q$ identity matrix, $\otimes$ denote the *Kronecker* product, and $||.||_F$ and $||.||_2$ denote the *Frobenius* norm and the *Euclidean* vector norm, respectively. For subsequent discussion convenience, we give the following equations [1]:

$$||A||_F^2 = ||vec(A)||_2^2, \tag{1}$$

$$vec(ABC) = (C^{\mathrm{T}} \otimes A)vec(B), \tag{2}$$

$$(AC) \otimes (BD) = (A \otimes B)(C \otimes D), \tag{3}$$

$$(A \otimes B)^{\mathrm{T}} = A^{\mathrm{T}} \otimes B^{\mathrm{T}}, \tag{4}$$

$$I_n \otimes I_m = I_m \otimes I_n = I_{mn}, \tag{5}$$

where A, B, C and D are matrices of proper sizes.

B. Singular Value Decomposition

In applications such as face recognition, the samples A_i's are naturally represented as matrix patterns. When applying SVD for dimensionality reduction (a well-known method that applies SVD for dimensionality reduction is the Eigenfaces [2]), one usually converts the matrix patterns A_i's to the concatenated vectors $\boldsymbol{a}_i, i = 1, 2, \ldots, N$. Based on this vector representation, SVD computes k orthonormal projection vectors in $P_{svd} = [\boldsymbol{p}_1^{svd}, \boldsymbol{p}_2^{svd}, \ldots, \boldsymbol{p}_k^{svd}] \in \mathbb{R}^{d\times k}$ to extract $\boldsymbol{a}_i^{svd} = P_{svd}^{\mathrm{T}}\boldsymbol{a}_i$ for $\boldsymbol{a}_i$. P_{svd} corresponds to the first k columns of U in

$$X = U\Lambda V^{\mathrm{T}}, \tag{6}$$

where $X = [\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_N]$ is the data matrix, $\Lambda = diag(\lambda_1, \lambda_2, \ldots, \lambda_N)$ contains the singular values in a non-increasing order, and U and V contain the left and right singular vectors of X, respectively.

The reconstructed sample of $\boldsymbol{a}_i$ by SVD can be denoted as $\tilde{\boldsymbol{a}}_i^{svd} = P_{svd}\boldsymbol{a}_i^{svd}$, and it is easy to get that P_{svd} provides the globally optimal solution to

$$\min_{P^{\mathrm{T}}P=I_k} J_{svd}(P) = \sum_{i=1}^{N} ||\boldsymbol{a}_i - PP^{\mathrm{T}}\boldsymbol{a}_i||_2^2, \tag{7}$$

where $J_{svd}(P_{svd})$ is the reconstruction error brought by SVD's rank-k approximation. Therefore, SVD has the property that it can achieve the smallest reconstruction error among all the rank-k approximations.

C. Generalized Low Rank Approximations of Matrices

In contrast to SVD that converts matrices A_i's to vectors $\boldsymbol{a}_i$'s, GLRAM directly manipulates A_i's, computes two transformations $L = [\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_m] \in \mathbb{R}^{e\times m}$ and $R = [\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_n] \in \mathbb{R}^{f\times n}$ with orthonormal columns, and extracts $A_i^{glram} = L^{\mathrm{T}}A_iR$ for A_i, with the reconstructed sample being $\tilde{A}_i^{glram} = LA_i^{glram}R^{\mathrm{T}}$.

To compute the bilateral transformations L and R, GLRAM minimizes the following reconstruction error:

$$\min_{\substack{L^{\mathrm{T}}L=I_m \\ R^{\mathrm{T}}R=I_n}} J_{glram}(L, R) = \sum_{i=1}^{N} ||A_i - LL^{\mathrm{T}}A_iRR^{\mathrm{T}}||_F^2. \tag{8}$$

Generally speaking, Eq. (8) does not admit a closed-form solution, and therefore [6], [13] employed an iterative procedure whose key is to optimize L under fixed R and vice versa. More specifically, under given L, R is composed of the n eigenvectors of

$$M_R = \sum_{i=1}^{N} A_i^{\mathrm{T}}LL^{\mathrm{T}}A_i \tag{9}$$

TABLE I
NOTATIONS

Notations	Descriptions	Notations	Descriptions
A_i	the i-th matrix pattern	N	number of samples
e	number of rows in A_i	f	number of columns in A_i
d	sample dimension ($d = ef$)	L	the left transformation
R	the right transformation	$\boldsymbol{l}_i$	the i-th column in L
$\boldsymbol{r}_i$	the i-th column in R	m	number of columns in L
n	number of columns in R	s	common value for m and n
$\boldsymbol{a}_i$	concatenated vector pattern of A_i	P_{svd}	the projection matrix obtained by SVD
P_{glram}	the projection matrix obtained by GLRAM	k	the number of columns in P_{svd} or P_{glram}
$\boldsymbol{a}_i^{svd}$	extracted features by SVD	A_i^{glram}	extracted features by GLRAM
$NMSE$	Normalized Mean Square Error	$NMLB$	Normalized Mean Lower-Bound
$MSLS$	Mean Similarities of Left Subspaces	$MSRS$	Mean Similarities of Right Subspaces

corresponding to the largest n eigenvalues; and under given R, L is composed of the m eigenvectors of

$$M_L = \sum_{i=1}^{N} A_i R R^{\mathrm{T}} A_i^{\mathrm{T}} \tag{10}$$

corresponding to the largest m eigenvalues.

III. Relationship between GLRAM and SVD

Although SVD and GLRAM extract features in different forms (i.e., SVD manipulates one-dimensional vectors to extract $\boldsymbol{a}_i^{svd} = P_{svd}^{\mathrm{T}} \boldsymbol{a}_i$, while GLRAM manipulates two-dimensional matrices to extract $A_i^{glram} = L^{\mathrm{T}} A_i R$), we shall reveal an essential relationship between GLRAM and SVD in the following theorem.

Theorem 1: GLRAM's objective function is identical to SVD's objective function except the imposed constraints.

Proof: One one hand, SVD's objective function Eq. (7) can be written as:

$$\min_{P_{svd}^{\mathrm{T}} P_{svd} = I_k} J_{svd}(P_{svd}) = \sum_{i=1}^{N} ||\boldsymbol{a}_i - P_{svd} P_{svd}^{\mathrm{T}} \boldsymbol{a}_i||_2^2. \tag{11}$$

On the other hand, we denote P_{glram} as

$$P_{glram} = R \otimes L = [\boldsymbol{p}_1^{glram}, \boldsymbol{p}_2^{glram}, \ldots, \boldsymbol{p}_{mn}^{glram}], \tag{12}$$

where for any $i = 1, \ldots, n$ and $j = 1, \ldots, m$,

$$\boldsymbol{p}_{(i-1)m+j}^{glram} = \boldsymbol{r}_i \otimes \boldsymbol{l}_j. \tag{13}$$

Employing Eqs. (2-5), we have

$$P_{glram}^{\mathrm{T}} P_{glram} = (R \otimes L)^{\mathrm{T}} (R \otimes L) = I_k, \tag{14}$$

$$\boldsymbol{a}_i^{glram} = vec(A_i^{glram}) = (R \otimes L)^{\mathrm{T}} \boldsymbol{a}_i = P_{glram}^{\mathrm{T}} \boldsymbol{a}_i, \tag{15}$$

where $\boldsymbol{a}_i^{glram}$ is the concatenated vector of A_i^{glram}, and $k = mn$ is the number of reduced dimension. Summarizing the results of Eqs. (12-15), GLRAM's objective function Eq. (8) can be written as

$$\begin{aligned} \min_{L,R,P_{glram}} \quad & J_{glram}(L,R) = \sum_{i=1}^{N} ||\boldsymbol{a}_i - P_{glram} P_{glram}^{\mathrm{T}} \boldsymbol{a}_i||_2^2 \\ \text{subject to} \quad & L^{\mathrm{T}} L = I_m, R^{\mathrm{T}} R = I_n, \\ & P_{glram} = R \otimes L, P_{glram}^{\mathrm{T}} P_{glram} = I_k. \end{aligned} \tag{16}$$

Comparing GLRAM's objective function Eq. (16) with SVD's Eq. (11), we can find that, GLRAM just optimizes the same objective function as SVD except the imposed constraints, i.e., SVD's P_{svd} is only subjected to the orthonormal constraint $P_{svd}^{\mathrm{T}} P_{svd} = I_k$, while, besides the orthonormal constraint, GLRAM's P_{glram} is subjected to additional *Kronecker* product constraints $P_{glram} = R \otimes L$, $L^{\mathrm{T}} L = I_m$ and $R^{\mathrm{T}} R = I_n$. This ends the proof. □

A. Remarks

First, by using Theorem 1, we can explain such experimental phenomenon reported in [6] that, GLRAM achieves higher reconstruction error than SVD under the same number of reduced dimension. This phenomenon attributes to the facts that: 1) both GLRAM and SVD in fact optimizes the same objective function, and 2) GLRAM is subjected to additional *Kronecker* product constraints $P_{glram} = R \otimes L$, $L^{\mathrm{T}} L = I_m$ and $R^{\mathrm{T}} R = I_n$ than SVD. It is worthwhile to note that, [26] also answered the question why GLRAM achieves higher reconstruction error than Principal Component Analysis [2] under the same number of reduced dimension. However, the employed methodologies are different, i.e., we answered this problem by revealing that GLRAM's objective function is identical to SVD 's objective function except the imposed constraint, while [26] answered the problem by explicitly formulating the objective function value of PCA as the lower-bound of GLRAM's objective function.

Second, GLRAM does not need to explicitly compute or store $P_{glram} \in \mathbb{R}^{d \times k}$, but instead computes and stores $L \in \mathbb{R}^{e \times m}$ and $R \in \mathbb{R}^{f \times n}$, consuming $em + fn$ elements; while SVD needs to explicitly compute and store the projection matrix $P_{svd} \in \mathbb{R}^{d \times k}$, consuming $dk = efmn$ elements. As a result, compared to SVD, GLRAM not only consumes less computation time but also obtains higher compression ratio when applied to the high-dimensional applications such as image compression.

Third, $\boldsymbol{a}_i^{svd}$ by SVD can be denoted as

$$\boldsymbol{a}_i^{svd} = [\boldsymbol{a}_i^{\mathrm{T}} \boldsymbol{p}_1^{svd}, \boldsymbol{a}_i^{\mathrm{T}} \boldsymbol{p}_2^{svd}, \ldots, \boldsymbol{a}_i^{\mathrm{T}} \boldsymbol{p}_k^{svd}]^{\mathrm{T}}, \tag{17}$$

and we have $\sum_{i=1}^{N} (\boldsymbol{a}_i^{\mathrm{T}} \boldsymbol{p}_j^{svd})^2 = \lambda_j^2$, since P_{svd} corresponds to the first k columns of U in Eq. (6). Moreover, considering the fact that λ_j's are in a non-increasing order, we have

$$\sum_{i=1}^{N} (\boldsymbol{a}_i^{\mathrm{T}} \boldsymbol{p}_j^{svd})^2 \geq \sum_{i=1}^{N} (\boldsymbol{a}_i^{\mathrm{T}} \boldsymbol{p}_{j+1}^{svd})^2. \tag{18}$$

From Eq. (18), we know that the projection vector $\boldsymbol{p}_j^{svd}$ preserves more information than $\boldsymbol{p}_{j+1}^{svd}$, and thus the information preserving abilities of SVD's projection vectors are in a non-increasing order. However, unlike (18) for SVD, the following inequality

$$\sum_{i=1}^{N}(\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{p}_j^{glram})^2 \geq \sum_{i=1}^{N}(\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{p}_{j+1}^{glram})^2 \tag{19}$$

generally does not hold for GLRAM (see Section IV-B for a counter-example, and Section VI-A for experimental verification). As a result, despite of GLRAM's close relationships with SVD presented in Theorem 1, GLRAM is still quite different from SVD.

IV. A New Lower-bound

As reported in [6], [13], under given number of reduced dimension $k = mn$, GLRAM always obtains the lowest reconstruction error when setting $s = m = n$, but "a rigorous theoretical justification behind this is still not available"[6].

In this section, we try to solve this problem, before which, we first prove a lemma.

Lemma 1: For any matrix $A \in \mathbb{R}^{e\times f}$, $L \in \mathbb{R}^{e\times m}$ and $R \in \mathbb{R}^{f\times n}$ that satisfy $L^{\mathrm{T}}L = I_m$ and $R^{\mathrm{T}}R = I_n$, we have

$$||L^{\mathrm{T}}AR||_F^2 \leq \sum_{i=1}^{\min(m,n)} \delta_i(A)^2, \tag{20}$$

where $\delta_i(A)$ denotes the i-th singular value of matrix A.

Proof: Let $\tilde{R} \in \mathbb{R}^{f\times(f-n)}$ be a matrix whose columns are the orthonormal complement of those in R, namely, $[R \quad \tilde{R}][R \quad \tilde{R}]^{\mathrm{T}} = I_f$. Since $A\tilde{R}\tilde{R}^{\mathrm{T}}A^{\mathrm{T}}$ is positive semi-definite, we have $\mathrm{trace}(L^{\mathrm{T}}A\tilde{R}\tilde{R}^{\mathrm{T}}A^{\mathrm{T}}L) \geq 0$, where $\mathrm{trace}(.)$ denotes the trace of a square matrix. Therefore, we have

$$\begin{aligned}||L^{\mathrm{T}}AR||_F^2 =&\mathrm{trace}(L^{\mathrm{T}}ARR^{\mathrm{T}}A^{\mathrm{T}}L)\\ \leq&\mathrm{trace}(L^{\mathrm{T}}A[R \quad \tilde{R}][R \quad \tilde{R}]^{\mathrm{T}}A^{\mathrm{T}}L)\\ =&\mathrm{trace}(L^{\mathrm{T}}AA^{\mathrm{T}}L)\\ \leq&\sum_{i=1}^{m}\delta_i(A)^2.\end{aligned} \tag{21}$$

By a similar derivation, we have

$$||L^{\mathrm{T}}AR||_F^2 \leq \sum_{i=1}^{n}\delta_i(A)^2. \tag{22}$$

From inequalities (21) and (22), we can easily get the inequality (20). This ends the proof. □

A. A Lower-bound of $J_{glram}(L, R)$ and a Theoretical Justification of $m = n$

Theorem 2: Let the SVD of A_i be denoted as:

$$A_i = U_i\Lambda_iV_i^{\mathrm{T}}, \tag{23}$$

where $\Lambda_i = diag(\delta_1(A_i), \delta_2(A_i), \dots, \delta_{min(e,f)}(A_i))$ contains the singular values in a non-increasing order, U_i and V_i contain the left and right singular vectors, respectively. Let $k = mn$ be the reduced dimension. Then

$$J_{glram}^{LB}(m, n) = \sum_{i=1}^{N}\sum_{j=\min(m,n)+1}^{\min(e,f)}\delta_j(A_i)^2 \tag{24}$$

is a lower-bound of $J_{glram}(L, R)$. Moreover, for given reduced dimension k, $J_{glram}^{LB}(m, n)$ achieves the minimum when $m = n$.

Proof: Employing Lemma 1, $J_{glram}(L, R)$ satisfies

$$\begin{aligned}J_{glram}(L, R) =&\sum_{i=1}^{N}||A_i - LL^{\mathrm{T}}A_iRR^{\mathrm{T}}||_F^2\\ =&\sum_{i=1}^{N}||A_i||_F^2 - \sum_{i=1}^{N}||L^{\mathrm{T}}A_iR||_F^2\\ \geq&\sum_{i=1}^{N}||A_i||_F^2 - \sum_{i=1}^{N}\sum_{j=1}^{min(m,n)}\delta_j(A_i)^2\\ =&\sum_{i=1}^{N}\sum_{j=\min(m,n)+1}^{\min(e,f)}\delta_j(A_i)^2\\ =&J_{glram}^{LB}(m, n),\end{aligned} \tag{25}$$

where the second equality follows from $L^{\mathrm{T}}L = I_m$ and $R^{\mathrm{T}}R = I_n$. Therefore, $J_{glram}^{LB}(m, n)$ offers a lower-bound of $J_{glram}(L, R)$. Furthermore, for given A_i's and reduced dimension k, the lower-bound $J_{glram}^{LB}(m, n)$ is not dependent on the computed L and R, but only dependent on m or n. It is easy to observe that $J_{glram}^{LB}(m, n)$ achieves the minimum when $m = n$. □

From Theorem 2, we can offer a theoretical justification of $m = n$ from the viewpoint of minimizing $J_{glram}^{LB}(m, n)$. Note that, although the justification is derived from the lower-bound, it is generally applicable to $J_{glram}(L, R)$ when GLRAM can perform well in terms of compression, since, as will be empirically proven in Section VI-B, $J_{glram}(L, R)$ is close to its lower-bound in this case.

Prior to our work in this paper, researches have come up with some different lower-bounds for GLRAM. The lower-bound provided in [15] is dependent on the obtained L and R and thus is quite different from ours. The lower-bound given in [26] is the objective function value of PCA, which is depicted by the eigenvalues of the covariance matrix of all the vector patterns a_i's. Note that, the lower-bound given by us is depicted by the singular values of the individual matrix patterns A_i's, and therefore it is quite different from the one given in [26]. Moreover, to the best of our knowledge, the lower-bounds given in [15], [26] can not be directly used to answer why GLRAM can obtain the lowest reconstruction error when $m = n$ for given reduced dimension $k = mn$.

In the end of this subsection, it is worthwhile to note that, since the justification of $m = n$ is not from $J_{glram}(L, R)$, but its lower-bound, it is possible that $m = n$ is not a good choice in some cases. We shall point out in Section V-C that, when e and f are extremely imbalanced, $m = n$ might not be a good choice, and suggest to deal with this problem by a technique employed in [29].

B. When the Lower-bound Can Be Touched

In this subsection, we explore when the lower-bound given in Theorem 2 can be touched. In the following, we assume $s = m = n$, and denote U_i^s and V_i^s respectively as the first s columns in U_i and V_i. The result in this subsection is given in the following theorem.

Theorem 3: If the following two conditions are satisfied:

- $U_i^s, i = 1, 2, \dots, N$ span the same projection subspace, i.e.,
$$U_1^s = U_i^s Q_i, \tag{26}$$
where $Q_i \in \mathbb{R}^{s \times s}$ is an orthonormal transformation,
- $V_i^s, i = 1, 2, \dots, N$ span the same projection subspace, i.e.,
$$V_1^s = V_i^s W_i, \tag{27}$$
where $W_i \in \mathbb{R}^{s \times s}$ is an orthonormal transformation,

then the lower-bound Eq. (24) can be touched by setting $L = U_1^s$ and $R = V_1^s$.

Proof: Let $L = U_1^s$, $R = V_1^s$, and employing Eqs. (26) and (27), we have

$$\begin{aligned} ||L^{\mathrm{T}} A_i R||_F^2 &= ||{U_1^s}^{\mathrm{T}} A_i V_1^s||_F^2 \\ &= ||Q_i^{\mathrm{T}} {U_i^s}^{\mathrm{T}} A_i V_i^s W_i||_F^2 \\ &= ||{U_i^s}^{\mathrm{T}} A_i V_i^s||_F^2 \\ &= \sum_{j=1}^{s} \delta_j(A_i)^2, \end{aligned} \tag{28}$$

$$\begin{aligned} J_{glram}(L, R) &= \sum_{i=1}^{N} (||A_i||_F^2 - ||L^{\mathrm{T}} A_i R||_F^2) \\ &= \sum_{i=1}^{N} \sum_{j=s+1}^{min(e,f)} \delta_j(A_i)^2. \end{aligned} \tag{29}$$

Obviously, the lower-bound Eq. (24) is touched. □

A special case of Theorem 3 is that, $U_i^s = U_j^s, V_i^s = V_j^s, i, j = 1, 2, \dots, N$, where $L = U_1^s$ and $R = V_1^s$ enable GLRAM to touch the lower-bound. In this special case, $L^{\mathrm{T}} A_i R = diag(\delta_1(A_i), \delta_2(A_i), \dots, \delta_s(A_i))$, and incorporating Eqs. (12), (13) and (15), we have

$$\sum_{t=1}^{N} (\boldsymbol{a}_t^{\mathrm{T}} \boldsymbol{p}_{(i-1)s+j}^{glram})^2 = \left\{ \begin{array}{ll} \sum_{t=1}^{N} \delta_i(A_t)^2 & i = j \\ 0 & i \neq j \end{array} \right. . \tag{30}$$

In this special case, we have that: 1) the inequality (19) in Section III-A does not hold, 2) the GLRAM projection vectors that benefit reconstruction are in fact $\boldsymbol{p}_{(i-1)s+i}^{glram}, i = 1, 2, \dots, s$, while the remaining ones are useless, and 3) there are projection vectors outside P_{glram} that can contain useful information, so long as the rank of any A_i is over s.

Generally speaking, the conditions offered in Theorem 3 are too strong to be satisfied in real applications. However, these two conditions remind us to measure the relationship between A_i's by the similarities between the subspaces spanned by U_i^s's (and V_i^s's), which will be discussed in detail in the next section.

V. When and Why GLRAM Can Perform Well in Terms of Compression

In this section, we explore a fundamental problem considering the usability of GLRAM, namely, when and why GLRAM can perform well in terms of compression. This is quite important, since it will help practitioners to decide whether to use GLRAM or not for compression purpose.

A. Criteria Related to the Compression Performance of GLRAM

An important criterion that measures the compression performance of GLRAM is the reconstruction error, $J_{glram}(L, R)$. It will be nice if the value of a criterion is between [0 1], which is easier to use in many tasks, say, empirical evaluation. For this sake, we employ the *Normalized Mean Square Error* (NMSE) as

$$NMSE = \frac{J_{glram}(L, R)/N}{\sum_{i=1}^{N} ||A_i||_F^2 / N}. \tag{31}$$

Obviously, $NMSE$ is between [0 1], and is proportional to $J_{glram}(L, R)$. The larger $NMSE$ is, the worse the compression performance is, and vice versa. Based on $NMSE$, it is easy to define the *Normalized Mean Retained Information* (NMRI) as $NMRI = 1 - NMSE$. In contrast to $NMSE$, the larger $NMRI$ is, the better the compression performance is. One key parameter in $NMSE$ is the value of s, and we want to obtain low $NMSE$ at small s.

The remaining problem is to define some proper criteria that can help determine when and why low $NMSE$ can be obtained at small s. Our motivation comes from the analysis of the normalized reconstruction error of a given matrix A. According to Lemma 1, we have

$$\frac{||A||_F^2 - ||L^T A R||_F^2}{||A||_F^2} \geq \frac{\sum_{i=1}^{min(r,c)} \delta_i(A)^2 - \sum_{i=1}^{s} \delta_i(A)^2}{\sum_{i=1}^{min(r,c)} \delta_i(A)^2},$$

from which we can observe that

- The lower-bound of the normalized reconstruction error is determined by the distribution of the singular values. Specifically, if the leading s singular values of A does not dominate, the normalized reconstruction error can not be small, and vice versa. Since this lower-bound is only determined by the given sample itself, we call this factor as the *within-samples factor*.
- When the lower-bound is small, it is meaningful to consider whether there exist L and R that can make the normalized reconstruction error close to the lower-bound. Obviously, when L and R correspond to the leading s left and right singular vectors of A, respectively, the lower-bound is touched. Moreover, considering the fact that we are looking for L and R for compressing N samples A_i's, then L (and R) should be the tradeoff among U_i^s's (and V_i^s's). Since L and R are tradeoff among all the matrix samples, we call this factor as the *between-samples factor*.

To depict the *within-samples factor*, we make use of the singular values of the matrices A_i's to compute the lower-bound $J_{glram}^{LB}(m,n)$ defined in Eq. (24). Similar to the introduction of $NMSE$, which can bring convenience to tasks such as evaluation, we define the *Normalized Mean Lower-Bound* (NMLB) as

$$NMLB = \frac{J_{glram}^{LB}(m,n)/N}{\sum_{i=1}^{N} ||A_i||_F^2/N}. \tag{32}$$

Like the relationship between $J_{glram}(L,R)$ and $J_{glram}^{LB}(m,n)$ revealed in Eq. (25), it is easy to get the relationship between $NMSE$ and $NMLB$ as

$$NMSE \geq NMLB. \tag{33}$$

Moreover, based on $NMLB$, it is easy to define the *Normalized Mean Upper-Bound* (NMUB) as $NMUB = 1 - NMLB$, and it is easy to get the relationship between $NMRI$ and $NMUB$ as

$$NMRI \leq NMUB. \tag{34}$$

which says that, the upper-bound of $NMRI$ is $NMUB$.

To depict the *between-samples factor*, we measure the relationship between A_i's by the similarities between the subspaces spanned by U_i^s's (and V_i^s's). Specifically, we define the *Mean Similarities of Left Subspaces* (MSLS) as

$$MSLS = \frac{2}{N(N-1)} \sum_{i=1}^{N} \sum_{j=i+1}^{N} SLS(U_i^s, U_j^s), \tag{35}$$

where

$$SLS(U_i^s, U_j^s) = \sqrt{\frac{1}{s}||{U_i^s}^{\mathrm{T}} U_j^s||_F^2} \tag{36}$$

is the similarity of subspaces spanned by U_i^s and U_j^s. Similarly, we define the *Mean Similarities of Right Subspaces* (MSRS) as

$$MSRS = \frac{2}{N(N-1)} \sum_{i=1}^{N} \sum_{j=i+1}^{N} SRS(V_i^s, V_j^s), \tag{37}$$

where

$$SRS(V_i^s, V_j^s) = \sqrt{\frac{1}{s}||{V_i^s}^{\mathrm{T}} V_j^s||_F^2} \tag{38}$$

is the similarity of subspaces spanned by V_i^s and V_j^s. It is easy to get that, both $MSLS$ and $MSRS$ are between [0 1]. Moreover, in the case described in Theorem 3, both $MSLS$ and $MSRS$ obtain the maximal value, i.e., 1.

With the criteria defined above, we are ready to discuss when and why GLRAM can perform well in terms of compression in the coming subsections.

B. Discussion on the Normalized Mean Lower-Bound Criterion

Considering the definition of $NMLB$ of $NMSE$ and their relationship revealed in inequality (33), it is easy to get the following proposition

Proposition 1: For given s:

- When $NMLB$ is large, $NMSE$ is large too, and therefore GLRAM can not work well in terms of compression.
- When $NMLB$ is small, it is possible that GLRAM can work well in terms of compression, depending on whether $NMSE$ can be close to $NMLB$.

Note that, $NMLB$ decreases with increasing s, and in the extreme case, $NMLB$ decreases to zero when $s = \min(e,f)$. However, the purpose of GLRAM is to obtain compact representation at low reconstruction error, and thus low $NMLB$ should be obtained at relatively small s. Therefore, we have the following proposition

Proposition 2: For varying s: To ensure that GLRAM can work well in terms of compression (obtaining compact representation at low reconstruction error), the value of $NMLB$ should decrease sharply with increasing s, so that $NMLB$ can be low when s is small.

To elaborate Proposition 2, we analyze the following two special cases. The first one is that, the singular values of A_i ($i = 1, \ldots, N$) are the same, so that $NMLB(s) = \frac{\min(e,f)-s}{\min(e,f)}$. Obviously, $NMLB$ is large when s is far smaller than $\min(e,f)$, and GLRAM can not compress well by obtaining low $NMSE$ at low s. The second one is that, $\delta_1(A_i)$, the first singular value of A_i ($i = 1, \ldots, N$) is extremely larger than the rest singular values, so that for every s, $NMLB$ is equal to 0 numerically. In this case, one can choose $s = 1$ to ensure small $NMLB$ ($= 0$ numerically). The difference between these two special cases are that, with increasing s, $NMLB$ decreases slowly (linearly) in the first case while extremely sharply in the second one. Of course, for real databases, the distribution of singular values is between these two special case, and obviously GLRAM is suitable for the case that the leading few singular values dominate.

When $NMLB$ is low at small s, it is worthwhile to consider whether the obtained $NMSE$ can be close to its lower-bound $NMLB$. We will discuss this in the next subsection.

C. Discussion on the Similarities Among Subspaces Criteria

In looking for L and R that compress the N samples A_i's, it is obvious that for a given sample A_i, if L (and R) spans the same subspace as U_i^s (and V_i^s), the reconstruction error of A_i can touch the lower-bound, i.e., $||A_i - LL^{\mathrm{T}}A_iRR^{\mathrm{T}}||_F^2 = \sum_{s+1}^{\min(e,f)} \delta_j(A_i)^2$. Since L (and R) is a tradeoff among the U_i^s's (and V_i^s's), then it is reasonable that, when U_i^s's (and V_i^s's) span close subspaces, the obtained L (and R) is likely to be close to the given U_i^s (and V_i^s) so that the reconstruction error of A_i is close to its lower-bound. Based on this analysis, we give the following proposition:

Proposition 3: MSLS and MSRS Criteria:
The larger $MSLS$ and $MSRS$ are, the more likely $NMSE$ is close to $NMLB$.

We elaborate this proposition by two examples. For the first one, considering the case presented in Theorem 3, it is obvious that $MSLS = MSRS = 1$. In this example, $NMSE$ equals $NMLB$. For the second one, suppose that there are N rank 1 matrices $A_i = \boldsymbol{u}_i\boldsymbol{v}_i^{\mathrm{T}}$, where $\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{u}_j = 0$ ($\forall i \neq j$), $\boldsymbol{v}_i^{\mathrm{T}}\boldsymbol{v}_j = 0$ ($\forall i \neq j$), $\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{u}_i = 1$, and $\boldsymbol{v}_i^{\mathrm{T}}\boldsymbol{v}_i = 1$. We set $s = m = n = 1$ and apply GLRAM to compress these N matrices. Obviously, in this example, $NMLB = 0$, and $MSLS = MSRS = 0$. By some careful mathematical deductions, we can get $NMSE =$

$\frac{N-1}{N}$, which is far larger than $NMLB = 0$, especially for large N.

From Proposition 3, we know that, when $MSLS$ and $MSRS$ are large, $NMSE$ is likely to be close to its lower-bound $NMLB$. For better revealing the relationship among $NMSE$, $NMLB$, $MSLS$ and $MSRS$, we make use of $\frac{NMRI}{NMUB} = \frac{1-NMSE}{1-NMLB}$, which depicts the ratio between the retained information and the upper-bound $NMUB$. Obviously, the larger $\frac{NMRI}{NMUB}$ is, the closer $NMSE$ is to $NMLB$. Our experiments in Section VI-B show that, with increasing $MSLS$ and $MSRS$, $\frac{NMRI}{NMUB}$ generally increases as well.

It is meaningful to point out that, there are some special cases when $MSLS$ and $MSRS$ are low, but $NMSE$ is possibly close to $NMLB$. Here we also consider the two special cases employed in Section V-B. For the first one, we assume that A_i's are matrices with equal singular values, and that the spaces spanned by U_i^s's (and V_i^s) are not close so that $MSLS$ and $MSRS$ are small. In this case, it is possible that $NMSE$ can be close to $NMLB$. For the second case, we assume that, $\delta_1(A_i)$, the first singular value of A_i $(i = 1, \ldots, N)$ is extremely larger than the rest singular values, and the first left (and right) singular vectors U_i^1's (and V_i^1's) are the same, but the remaining left (and right) singular vectors differ quite a lot. When when setting $s > 1$, it is possible that $NMSE$ and $NMLB$ are both 0 numerically, but that $MSLS$ and $MSRS$ are small. However, these two cases can be addressed by using the distribution of $NMLB$ under varying s. For the first case, $NMLB$ decreases linearly with varying s, and therefore GLRAM does perform well in terms of compression due to high $NMSE$ at low s. For the second one, since $NMLB$ decreases extremely sharply, then $s = 1$ is enough for compact compression, with $NMSE$ being close to $NMLB$ (both equal 0 numerically), and $MSLS = MSRS = 1$.

In the end of this subsection, it is worthwhile to make use of Proposition 3 to point out that, when e and f are extremely imbalanced, i.e., $e \gg f$ or $f \gg e$, $m = n$ might not be a good choice for GLRAM. Without loss of generality, we assume $f = 2$ and $e \gg f$. In this case, although we can set $s = m = n = 2$ (note that, $m \leq e$ and $n \leq f$) achieving $NMLB = 0$, $NMSE$ can be typically large when the subspaces spanned by U_i^s's differ greatly so that $MSLS$ is very low. From this special case, we can say that GLRAM should be applied to databases with balanced e and f. Moreover, a possible way to deal with imbalanced e and f is to reshape the imbalanced matrices to the balanced ones, and this technique has been employed in [29].

From the discussion in the previous subsections, we can say that when 1) $NMLB$ decreases sharply with increasing s, and can achieve a low value at small s, and meanwhile 2) $MSLS$ and $MSRS$ are relatively large, GLRAM can work well in terms of compression.

VI. Experiments

To study the information preserving abilities of GLRAM projection vectors, and to empirically discuss the relationship among the criteria defined in Section V, we conduct experiments on the following three datasets (one synthetic and two real-world face datasets):

- RAND is a synthetic dataset consisting of 400 matrices of size 112 × 92. The 10304 entries in each matrix are randomly generated from the normal distribution with mean 0 and standard derivation 1;
- ORL[1] contains the face images of 40 persons, for a total of 400 images. The resolution of the face images is 112 × 92;
- FERET [30] is a database that consists of a total of 14,051 gray-scale images representing 1,199 subjects. In this paper, we conduct experiments on a subset of the FERET test 1996, *fa*, which contains 1,196 face images. The resolution of the face images is 150×130.

When implementing the GLRAM algorithm, we follow the similar setting as in [6], namely, the parameter η (which is defined as the ratio between the values of Root Mean Square Reconstruction Error [6] corresponding to two adjacent iterative steps, and controls the convergence precision) is set to 10^{-6}, and $s = m = n = 10$.

A. On Information Preserving Abilities of $\boldsymbol{p}_j^{glram}$'s

In this subsection, we explore the information preserving abilities of $\boldsymbol{p}_j^{glram}$'s defined in (13). We conduct experiments on RAND and ORL, and report results in Fig. 1, where the x-axis corresponds to j, the number of projection vector, and the y-axis corresponds to the logarithmic preserved information $\ln(\sum_{i=1}^{N}(\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{p}_j^{glram})^2)$. From Fig. 1, we can clearly observe that, unlike SVD, the information preserving abilities of GLRAM projection vectors are not in a non-increasing order. We would like to point out that, the preserved information of the projection vector $\boldsymbol{p}_j^{glram}$ measured by $\sum_{i=1}^{N}(\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{p}_j^{glram})^2$ has an equivalence relationship with the reconstruction error by this projection vector measured by $\sum_{i=1}^{N}\|\boldsymbol{a}_i - \boldsymbol{p}_j^{glram}\boldsymbol{p}_j^{glram\,\mathrm{T}}\boldsymbol{a}_i\|_2^2$, as we can easily verify $\sum_{i=1}^{N}\|\boldsymbol{a}_i - \boldsymbol{p}_j^{glram}\boldsymbol{p}_j^{glram\,\mathrm{T}}\boldsymbol{a}_i\|_2^2 = \sum_{i=1}^{N}\|\boldsymbol{a}_i\|_2^2 - \sum_{i=1}^{N}(\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{p}_j^{glram})^2$. We choose the former measurement for better distinguishing the information preserving abilities of each individual projection vector.

Next, we explore whether there are certain projection vectors that on one hand are orthogonal to GLRAM projection vectors, and on the other hand can preserve more information than the GLRAM projection vectors. For this sake, we compute 1) $\boldsymbol{r}_i, i = n+1, \ldots, f$ that are the $f - n$ eigenvectors of M_R corresponding to the rest $f - n$ eigenvectors in a non-increasing order, and 2) $\boldsymbol{l}_j, j = m+1, \ldots, e$ that are the $e - m$ eigenvectors of M_L corresponding to the rest $e - m$ eigenvectors in a non-increasing order. Obviously, the projection vectors $\boldsymbol{r}_i \otimes \boldsymbol{l}_j, \forall\, i = n+1, \ldots, f, j = 1, \ldots, e$ and $i = 1, \ldots, f, j = m+1, \ldots, e$ are orthogonal to the GLRAM projection vectors $\boldsymbol{r}_i \otimes \boldsymbol{l}_j$, $i = 1, \ldots, n, j = 1, \ldots, m$. Our experimental results on RAND and ORL show that, there are respectively 86 and 106 projection vectors that on one hand are orthogonal to GLRAM projection vectors in P_{glram}, and on

[1] http://www.uk.research.att.com/facedatabase.html

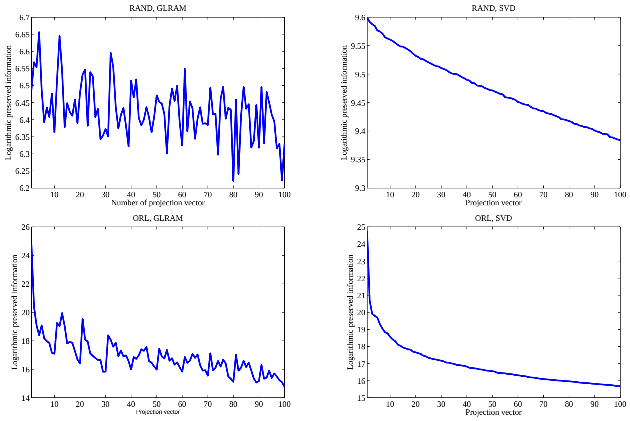

Fig. 1. The information preserving abilities of $\boldsymbol{p}_j^{glram}$'s. The x-axis corresponds to j, the number of projection vector, and the y-axis corresponds to the logarithmic preserved information $\ln(\sum_{i=1}^{N}(\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{p}_j^{glram})^2)$.

the other hand can preserve more information than the worst GLRAM projection vector.

B. On Different Criteria

First, we verify $s = m = n$ from the viewpoint of minimizing the lower-bound of GLRAM's objective function, when the GLRAM solution is close to its lower-bound. For this sake, we fix the reduced dimension $k = mn = 100$, and try the following five combinations for $m \times n$: 1×100, 5×20, 10×10, 20×5, and 100×1, and conduct experiments on FERET. The results are reported in Fig. 2, from which we can see that, 1) $NMSE$ is close to $NMLB$, and 2) both $NMLB$ and $NMSE$ obtain the minimal values when $s = m = n = 10$.

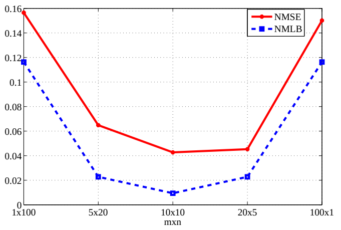

Fig. 2. $NMSE$ and $NMLB$ under varying $m \times n$. The x-axis corresponds to different sizes of $m \times n$, and the y-axis denote the values of $NMSE$ or $NMLB$.

Second, we report the values of the criteria on different datasets in Table II. From this table, we can observe that

- GLRAM performs poorly on the synthetic dataset RAND, with the criterion $NMSE = 0.9849$ being typically high. The underlying reasons are: 1) $NMLB = 0.6877$ is very high, which leads to $NMSE \geq NMLB = 0.6877$; and 2) $MSLS = 0.2982$ and $MSRS = 0.3290$ are very small, and thus $NMSE$ is much larger than $NMLB$. Therefore, if the value of $NMLB$ is relatively large and meanwhile the values of $MSLS$ and $MSRS$ are relatively small, GLRAM can not achieve relatively low $NMSE$.
- GLRAM performs well on the real image datasets ORL and FERET, with $NMSE < 0.05$. The underlying reasons are that: 1) the criterion $NMLB$ is very small; and 2) the criteria $MSLS$ and $MSRS$ are very large. Therefore, if the value of $NMLB$ is relatively small and meanwhile the values of $MSLS$ and $MSRS$ are relatively large, GLRAM can achieve relatively low $NMSE$.
- Looking at $\frac{NMRI}{NMUB} = \frac{1-NMSE}{1-NMLB}$, which depicts the ratio between the retained information and the upper-bound $NMUB$, we can see that the values on FERET and ORL is very high (over 0.9), while the value on RAND is very small (less than 0.1). These results are in accordance withe results that the $MSLS$ and $MSRS$ criteria are high on FERET and ORL, while low on RAND.

TABLE II
VALUES OF THE CRITERIA ON DIFFERENT DATASETS.

Criterion	RAND	ORL	FERET
$NMSE$	0.9849	0.02453	0.04272
$NMLB$	0.6877	0.007127	0.009400
$MSLS$	0.2982	0.7348	0.7172
$MSRS$	0.3290	0.7307	0.7808
$\frac{NMRI}{NMUB}$	0.04822	0.9825	0.9664

Third, we have a look at the distribution of the $NMLB$ criterion with varying s. We report results on ORL and RAND in Fig. 3, from which we can see that 1) when $s = 1$, $NMLB$ is already very low (less than 0.1) on ORL but quite high (over 0.9) on RAND, and 2) with increasing s, $NMLB$ decreases quickly to below 0.01 when $s \geq 8$ on ORL, but $NMLB$ remains over 0.1 when $s \leq 50$. Therefore, to ensure that GLRAM can perform well in terms of compression, $NMLB$ should decrease sharply with increasing s.

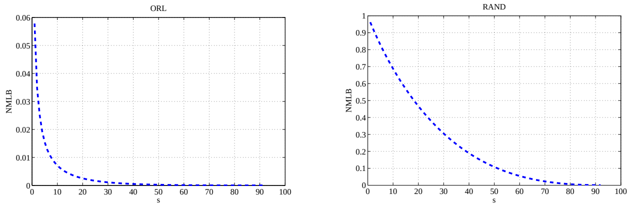

Fig. 3. The distribution of $NMLB$ under varying s.

Fourth, we explore the relationship between $NMSE$ an $NMLB$ under fixed $MSLS$ and $MSRS$. For this sake, we make use of the fractional order singular value representation of each matrix pattern A_i as [31]

$$A_i^{\alpha} = U_i \Lambda_i^{\alpha} V_i^{\mathrm{T}}, \tag{39}$$

where $\Lambda_i^{\alpha} = diag(\delta_1(A_i)^{\alpha}, \ldots, \delta_{min(e,f)}(A_i)^{\alpha})$, U_i, V_i and $\delta_j(A_i)$ are defined in Eq. (23), and α is a nonnegative parameter. Obviously, A_i^{α} under different nonnegative α shares the same left and right singular vectors as A_i. Therefore, by setting different nonnegative α's, we will generate different datasets $\{A_i^{\alpha}\}_{i=1}^{N}$'s with identical $MSLS$ and $MSRS$. Moreover, for any given A_i, $\delta_j(A_i)$'s are positive and in a non-increasing order, and thus the larger α is, the higher the criterion $NMLB$ is. We report the results in the first row of Fig. 4, from which we can observe

- On RAND, the value of $NMLB$ decreases with increasing α, but the value of $NMSE$ remains very high (over

0.9). Specifically, when $\alpha = 10$, $NMLB = 0.02535$ is relatively small, but $NMSE = 0.9688$ is far larger than $NMLB$. This attributes to the fact that $MSLS$ and $MSRS$ are very small. Therefore, we can say that, when $MSLS$ and $MSRS$ are relatively small, it is likely that GLRAM can not obtain relatively low $NMSE$.

- On ORL and FERET, since the values of $MSLS$ and $MSRS$ are relatively large, $NMSE$ is very close to $NMLB$. In other words, $J_{glram}(L,R)$ is close to its lower-bound $J^{LB}_{glram}(m,n)$ in Eq. (24). Therefore, the justification of $m = n$ (in Section IV-A) from the viewpoint of minimizing $J^{LB}_{glram}(m,n)$ is applicable to $J_{glram}(L,R)$ for image datasets such as ORL and FERET.
- On ORL and FERET, despite that the values of $MSLS$ and $MSRS$ are very high, when setting α to 0, the value of $NMSE$ becomes typically high, which attributes to the fact that the value of $NMLB$ is typically large (over 0.9). Therefore, we can say that, GLRAM can not obtain relatively low $NMSE$, when $NMLB$ is relatively large.

Fifth, we explore the relationship between $\frac{NMRI}{NMUB}$ and the similarities among subspaces criteria $MSLS$ and $MSRS$. Here, $\frac{NMRI}{NMUB} = \frac{1-NMSE}{1-NMLB}$ denotes the ratio between the retained information and the upper-bound $NMUB$. Obviously, the larger $\frac{NMRI}{NMUB}$ is, the closer $NMSE$ is to $NMLB$. To generate varying $MSLS$ and $MSRS$, we let s change from 1 to $\min(e, f)$. We conduct experiments on RAND, ORL and FERET, and report results in the second row of Fig. 4. We can observe that,

- On RAND, when $s = 1$, $MSLS = 0.0758$ and $MSRS = 0.0836$ are very small, i.e., the first left (and right) singular vectors of the matrices A_i's differ greatly, and therefore $\frac{NMRI}{NMUB} = 0.0105$ is quite small, which leads to that $NMSE$ is quite larger than $NMLB$. When $s = 92$, $MSRS$ touches 1 since the V_i^s are 92×92 orthonormal matrices, but $MSLS$ does not since the 112×92 matrices U_i^s's do not span the same subspace. Finally, it is easy to get that, $MSLS$ and $MSRS$ consistently increase with increasing s, and $\frac{NMRI}{NMUB}$ consistently increases when $MSLS$ and $MSRS$ increase.
- On ORL, when $s = 1$, $MSLS = 0.9707$ and $MSRS = 0.9844$ are very large, which are quite different from those on RAND. Moreover, benefited by the large $MSLS$ and $MSRS$ values, $\frac{NMRI}{NMUB} = 0.9544$ is quite close to 1, which leads to that $NMSE = 0.0456 + 0.9544 \times NMLB$ is quite close to $NMLB$. Similar observation can be obtained from FERET. It should be noted that, although $NMSE$ is close to $NMLB$ when $s = 1$, $NMLB$ is not small enough yet, and thus the s we choose is larger than 1 in practice.
- On ORL and FERET, it is interesting to note that, when s increases from 1 to about 10, $MSLS$ and $MSRS$ generally decrease; and when s increases from 11 to $\min(e, f)$, $MSLS$ and $MSRS$ consistently increase. For $\frac{NMRI}{NMUB}$, it achieves the minimum when $s = 2$, and increases with increasing s when $s > 2$. The reason that $\frac{NMRI}{NMUB}$ does not decrease when s increases from 2 to about 10 might be that, the large values of $MSLS$ and $MSRS$ at $s = 1$ benefit the compression performance at $s > 1$. Finally, $\frac{NMRI}{NMUB}$ increases as $MSLS$ and $MSRS$ increase when $s > 10$.
- Comparing the curves on RAND with those on ORL and FERET, we can find that, the curve of $\frac{NMRI}{NMUB}$ is generally below those of $MSLS$ and $MSRS$ on RAND, but is generally above those of $MSLS$ and $MSRS$ on ORL and FERET. Moreover, the values of $MSLS$ and $MSRS$ are always relatively large (over 0.7) on FERET and ORL, which attributes to the fact that the matrices share a common characteristic, i.e., face image. In contrast, $MSLS$ and $MSRS$ are quite low when s is below 20, since the matrices are random and thus do not share much common characteristics.

Summarizing the results from the above experiments, we can say that, 1) when $NMLB$ does not decrease sharply with increasing s, $NMLB$ can not be low at small s, and therefore GLRAM can not perform well in terms of compression performance; 2) when $NMLB$ is low at small s, but $MSLS$ and $MSRS$ are very low, GLRAM can not work well in terms of compression; and 3) when $NMLB$ is low at small s and meanwhile $MSLS$ and $MSRS$ are large (say, > 0.7 as on ORL and FERET), GLRAM can obtain good compression performance.

VII. Conclusion

In this paper, we revisit GLRAM to reveal its properties, answer an open problem raised by [6], and explore when and why GLRAM can perform well in terms of compression. Our main contributions are that:

1) We reveal the close relationship between GLRAM and SVD that GLRAM optimizes the same objective function as SVD except the imposed constraints. Based on the revealed relationship, we can theoretically answer why GLRAM achieves higher reconstruction error than SVD under the same number of reduced dimension. Moreover, we show that the information preserving abilities of the projection vectors $\boldsymbol{p}_j^{glram}$'s are not in a non-increasing order as those of SVD.

2) We offer a lower-bound of GLRAM's objective function, based on which we answer an open problem raised by [6], i.e., giving a theoretical justification of $m = n$ from the viewpoint of minimizing the lower-bound of $J_{glram}(L,R)$ in Theorem 2.

3) We explore a fundamental problem with the usability of GLRAM, and argue that, when $NMLB$ is low at small s and meanwhile $MSLS$ and $MSRS$ are large, GLRAM can obtain good compression performance.

The arguments made in this paper are verified by theoretical proofs and empirical evaluations on one synthetic and two real-world face datasets. In our viewpoint, the following four aspects are worthy of future study:

1) Since the information preserving abilities of GLRAM projection vectors are not in a non-increasing order as SVD,

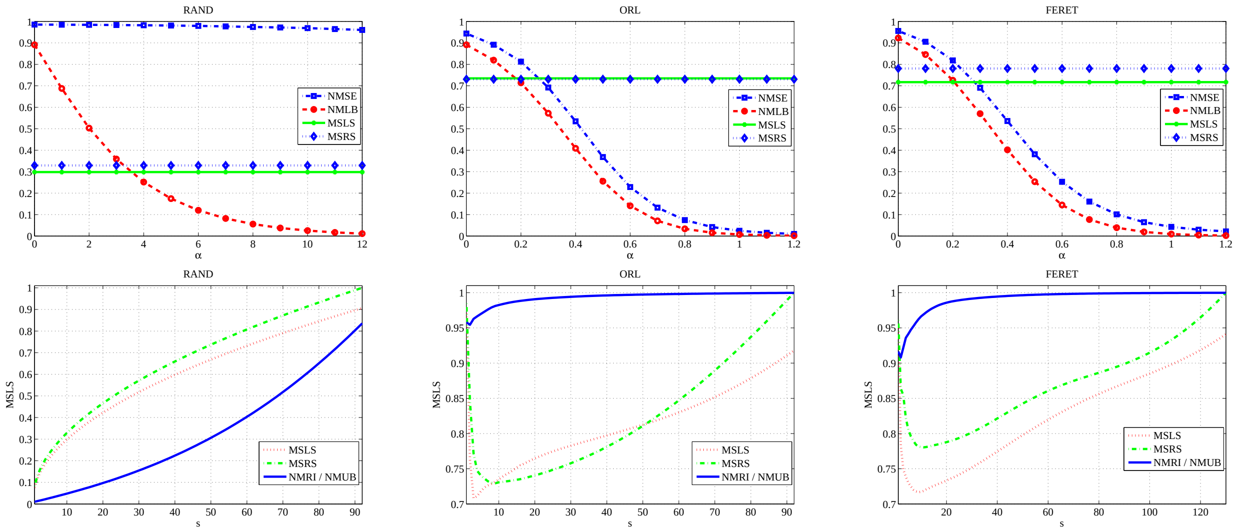

Fig. 4. Relationships among different criteria. First row: $NMSE$ versus $NMLB$ under fixed $MSLS$ and $MSRS$. Second row: $\frac{NMRI}{NMUB}$'s relationship with $MSLS$ and $MSRS$. Please refer to Table I for explanation of the notations.

we can generalize the GLRAM objective function Eq. (8) to

$$\min_{\substack{L^{\mathrm{T}}L=I_e,\\ R^{\mathrm{T}}R=I_f,\\ ||G||_F^2=k}} J(L,R,G)=\sum_{i=1}^{N}||G\cdot(A_i-\tilde{A}_i)||_F^2, \tag{40}$$

where $\tilde{A}_i=LL^{\mathrm{T}}A_iRR^{\mathrm{T}}$, G an $e\times f$ binary matrix, and $\cdot$ is the element-by-element matrix multiplication. The underlying motivation is to employ the k projection vectors that have the strongest abilities in preserving information. We can also extend the study in this paper to benefit the extensions of GLRAM, the two-dimensional discriminant methods [27], the tensor based methods [24], [23], etc.

2) It is worthwhile to give some probabilistic interpretations for the two-dimensional and tensor based methods (one related work is [28]). It is also meaningful to derive some simplified rules for testing when and why GLRAM can obtain good compression performance, and explore whether there exists a nice empirical model with which GLRAM outperforms the SVD, and $m=n$ is the optimal choice.

3) Based on the revealed relationship between GLRAM and SVD, it is meaningful to explore whether and how the two-dimensional and tensor-based representation can incorporate the prior (spatial) information of the data for image classification, and a similar study can be carried out on the MatPCA and MatFLDA methods [29], which convert one-dimensional data to two-dimensional ones for dimensionality reduction.

4) Since the optimization problem in Eq. (8) is non-convex, we can not assure that the solution to GLRAM is globally optimal. Therefore, it is worthwhile to further consider this open problem raised in [6], i.e., when GLRAM can have the global convergence property.

Acknowledgement

We would like to thank the associate editor and the four anonymous reviewers for the constructive comments that improved this paper. This work was supported by the National Science Foundation of China under Grant Nos. 60635030, 60773060, 60973097, 60905035, the National Fundamental Research Program of China under Grant No. 2010CB327903, the Jiangsu Science Foundation under Grant Nos. BK2008018, BK2009369, and the Higher Education University's Doctors' Foundation under grant No. 200802870003.

References

[1] G. Golub and C. Van Loan, *Matrix Computations*, 3rd ed. Baltimore, MD: The Johns Hopkins University Press, 1996.

[2] M. Turk and A. Pentland, “Eigenfaces for recognition,” *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–96, 1991.

[3] P. Belhumeur, J. Hespanha, and D. Kriegman, “Eigenfaces vs. fisherfaces: Recognition using class specific linear projection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 711–720, 1997.

[4] M. Berry, S. Dumais, and G. O'Brie, “Using linear algebra for intelligent information retrieval,” *SIAM Review*, vol. 37, no. 4, pp. 573–595, 1995.

[5] S. Deerwester, S. Dumais, T. Landauer, G. Furnas, and R. Harshman, “Indexing by latent semantic analysis,” *Journal of the Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[6] J. Ye, “Generalized low rank approximations of matrices,” *Machine Learning*, vol. 61, no. 1-3, pp. 167–191, 2005.

[7] M. Brand, “Incremental singular value decomposition of uncertain data with missing values,” in *Procceedings of the Seventh European Conference on Computer Vision*, 2002, pp. 707–720.

[8] M. Gu and S. Eisenstat, “A fast and stable algorithm for updating the singular value decomposition,” Department of Computer Science, Yale University, Tech. Rep., 1993.

[9] K. Kanth, D. Agrawal, and A. Singh, “Dimensionality reduction for similarity searching in dynamic databases,” *Computer Vision and Image Understanding*, vol. 75, no. 1-2, pp. 59–72, 1999.

[10] D. Achlioptas and F. McSherry, “Fast computation of low rank matrix approximations,” in *Procceedings of the Thirty-third annual ACM symposium on the theory of computing*, 2001, pp. 611–618.

[11] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay, “Clustering in large graphs and matrices,” in *Procceedings of the Tenth Annual ACM-SIAM Symposium on Discrete Algorithms*, 1999, pp. 291–299.

[12] A. Frieze, R. Kannan, and S. Vempala, “Fast monte-carlo algorithms for finding low-rank approximations,” *Journal of the ACM*, vol. 51, no. 6, pp. 1025–1041, 2004.

[13] J. Ye, “Generalized low rank approximations of matrices,” in *Procceedings of the Twenty-first International Conference on Machine learning*, Banff, Canada, 2004, pp. 887–894.

[14] J. Yang, D. Zhang, A. Frangi, and J. Yang, "Two-dimensional PCA: A new approach to appearance-based face representation and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 131–137, 2004.

[15] C. Ding and J. Ye, "2-dimensional singular value decomosition for 2d maps and images," in *Procceedings of the SIAM International Conference on Data Mining*, Newport Beach, CA, 2005, pp. 22–34.

[16] D. Zhang, S. Chen, and J. Liu, "Representing image matrices: Eigenimages versus eigenvectors," in *Procceedings of the Second International Symposium on Neural Networks*, Chongqing, China, 2005, pp. 659–664.

[17] K. Inoue and K. Urahama, "DSVD: A tensor-based image compression and recognition method," in *Procceedings of the IEEE International Symposium on Circuits and Systems*, Kobe, Japan, 2005, pp. 6308–6311.

[18] L. Lathauwer, B. Moor, and J. Vandewalle, "A multi multilinear singular value decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.

[19] Z. Liang and P. Shi, "An analytical algorithm for generalized low-rank approximations of matrices," *Pattern Recognition*, vol. 38, no. 11, pp. 2217–2219, 2005.

[20] K. Inoue and K. Urahama, "Equivalence of non-iterative algorithms for simultaneous low rank approximations of matrices," in *Procceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, New York, NY, 2006, pp. 154–159.

[21] J. Liu and S. Chen, "Non-iterative generalized low rank approximation of matrices," *Pattern Recognition Letters*, vol. 27, no. 9, pp. 1002–1008, 2006.

[22] H. Lu, K. N. Plataniotis, and V. A. N., "Mpca: Multilinear principal component analysis of tensor objects," *IEEE Transactions on Neural Networks*, vol. 19, no. 1, pp. 18–39, 2008.

[23] D. Xu, S. Yan, L. Zhang, H. Zhang, Z. Liu, and H. Shum, "Concurrent subspaces analysis," in *Procceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Diego, CA, 2005, pp. 203–208.

[24] H. Wang and N. Ahuja, "Rank-r approximation of tensors using image-as-matrix representation," in *Procceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Diego, CA, 2005, pp. 346–353.

[25] S. Yan, D. Xu, S. Lin, T. Huang, and S. Chang, "Element rearrangement for tensor-based subspace learning," in *Procceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Minneapolis, MN, 2007.

[26] Z. Liang, D. Zhang, and P. Shi, "The theoretical analysis of glram and its applications," *Pattern Recognition*, vol. 40, no. 3, pp. 1032–1041, 2007.

[27] J. Ye, R. Janardan, and Q. Li, "Two-dimensional linear discriminant analysis," in *Advances in Neural Information Processing Systems 17*, L. Saul, Y. Weiss, and L. Bottou, Eds. Cambridge, MA: MIT Press, 2004, pp. 354–363.

[28] S. Yu, J. Bi, and J. Ye, "Probabilistic interpretations and extensions for a family of 2d pca-style algorithms," in *Procceedings of the KDD'2008 Workshop on Data Mining using Matrices and Tensors*, 2008.

[29] S. Chen, Y. Zhu, D. Zhang, and J. Yang, "Feature extraction approaches based on matrix pattern: Matpca and matflda," *Pattern Recognition Letters*, vol. 26, no. 8, pp. 1157–1167, 2005.

[30] P. J. Phillips, H. Wechsler, J. Huang, and P. Rauss, "The feret database and evaluation procedure for face recognition algorithms," *Image and Vision Computing*, vol. 16, no. 5, pp. 295–306, 1998.

[31] J. Liu, S. Chen, and X. Tan, "Fractional order singular value decomposition representation for face recognition," *Pattern Recognition*, vol. 41, no. 1, pp. 378–395, 2008.

Jun Liu received his B.S. degree from Nantong Institute of Technology (now Nantong University) in 2002, and his Ph.D. degree from Nanjing University of Aeronautics and Astronautics (NUAA) in November, 2007. He joined the Department of Computer Science & Engineering, NUAA, as a Lecturer in 2007. He is currently a postdoc in the Biodesign Institute and the Department of Computer Science & Eigineering of Arizona State University. His research interests include Dimensionality Reduction, Sparse Learning, and Large-Scale Optimization. He has authored or coauthored over 20 scientific papers.

Songcan Chen received the B.Sc. degree in mathematics from Hangzhou University (now merged into Zhejiang University) in 1983. In Dec. 1985, he completed the M.Sc. degree in computer applications at Shanghai Jiaotong University and then worked at NUAA in Jan. 1986 as an assistant lecturer. There he received a Ph.D. degree in communication and information systems in 1997. Since 1998, as a full professor, he has been with the Department of Computer Science and Engineering at NUAA. His research interests include pattern recognition, machine learning and neural computing. In these fields, he has authored or coauthored over 70 scientific journal papers.

Xiaoyang Tan received his B.S. and M.S. degree in computer applications from NUAA in 1993 and 1996, respectively. Then he worked at NUAA in June 1996 as an assistant lecturer. He received a Ph.D. degree from Department of Computer Science and Technology of Nanjing University, China, in 2005. From Sept.2006 to OCT.2007, he worked as a postdoctoral researcher in the LEAR (Learning and Recognition in Vision) team at INRIA Rhone- Alpes in Grenoble, France. His research interests are in face recognition, machine learning, pattern recognition, and computer vision. In these fields, he has authored or coauthored over 20 scientific papers.

Zhi-Hua Zhou (S'00-M'01-SM'06) received the BSc, MSc and PhD degrees in computer science from Nanjing University, China, in 1996, 1998 and 2000, respectively, all with the highest honors. He joined the Department of Computer Science & Technology at Nanjing University as an assistant professor in 2001, and is currently Cheung Kong Professor and Director of the LAMDA group. His research interests are in artificial intelligence, machine learning, data mining, pattern recognition, information retrieval, evolutionary computation and neural computation. In these areas he has published over 70 papers in leading international journals or conference proceedings. Dr. Zhou has won various awards/honors including the National Science & Technology Award for Young Scholars of China (2006), the Award of National Science Fund for Distinguished Young Scholars of China (2003), the Microsoft Young Professorship Award (2006), etc. He is an Associate Editor-in-Chief of *Chinese Science Bulletin*, Associate Editor of *IEEE Transactions on Knowledge and Data Engineering* and *ACM Transactions on Intelligent Systems and Technology*, and on the editorial boards of various journals. He is a co-founder of ACML, Steering Committee member of PAKDD and PRICAI, Program Committee Chair/Co-Chair of PAKDD'07, PRICAI'08 and ACML'09, vice Chair or area Chair of conferences including IEEE ICDM'06, IEEE ICDM'08, SIAM DM'09, ACM CIKM'09, etc., and General Chair/Co-Chair or Program Committee Chair/Co-Chair of a dozen of native conferences in China. He is the chair of the Machine Learning Society of the Chinese Association of Artificial Intelligence (CAAI), vice chair of the Artificial Intelligence & Pattern Recognition Society of the China Computer Federation (CCF), and chair of the IEEE Computer Society Nanjing Chapter.