

CSE 494 CSE/CBS 598 (Fall 2007): Numerical Linear Algebra for Data  
Exploration— Two dimensional SVD and PCA  
Instructor: Jieping Ye

## 1 Introduction

- Traditional methods in information retrieval and machine learning deal with data in vectorized representation. A collection of data is then stored in a single matrix  $A \in \mathbb{R}^{N \times n}$ , where each column of  $A$  corresponds to a vector in the  $N$ -dimensional space. A major benefit of this vector space model is that the algebraic structure of the vector space can be exploited.
- For high-dimensional data, one would like to simplify the data, so that traditional machine learning and statistical techniques can be applied. However, crucial information intrinsic in the data should not be removed under this simplification. A widely used method for this purpose is to approximate the single data matrix,  $A$ , with a matrix of lower rank.

### 1.1 Problem formulation

- Let  $A_i \in \mathbb{R}^{r \times c}$ , for  $i = 1, \dots, n$ , be the  $n$  data points in the training set, where  $r$  and  $c$  denote the number of rows and columns respectively for each  $A_i$ . We aim to compute two matrices  $L \in \mathbb{R}^{r \times \ell_1}$  and  $R \in \mathbb{R}^{c \times \ell_2}$  with orthonormal columns, and  $n$  matrices  $M_i \in \mathbb{R}^{\ell_1 \times \ell_2}$ , for  $i = 1, \dots, n$ , such that  $LM_iR^T$  approximates  $A_i$ , for all  $i$ . Here,  $\ell_1$  and  $\ell_2$  are two pre-specified parameters that are best set to the same value, based on the experimental results. Mathematically, we can formulate this as the following minimization problem: Computing optimal  $L$ ,  $R$  and  $\{M_i\}_{i=1}^n$ , which solve

$$\min_{\substack{L \in \mathbb{R}^{r \times \ell_1} : L^T L = I_{\ell_1} \\ R \in \mathbb{R}^{c \times \ell_2} : R^T R = I_{\ell_2} \\ M_i \in \mathbb{R}^{\ell_1 \times \ell_2} : i = 1, \dots, n}} \sum_{i=1}^n \|A_i - LM_iR^T\|_F^2. \quad (1)$$

- The matrices  $L$  and  $R$  in the above approximations act as the two-sided linear transformations on the data in matrix form.

## 2 The main algorithm

- The following theorem shows that the  $M_i$ 's are determined by the transformation matrices  $L$  and  $R$ , which significantly simplifies the minimization problem in Eq. (1).

**Theorem 2.1.** *Let  $L$ ,  $R$  and  $\{M_i\}_{i=1}^n$  be the optimal solution to the minimization problem in Eq. (1). Then  $M_i = L^T A_i R$ , for every  $i$ .*

*Proof.* By the property of the trace of matrices,

$$\begin{aligned}
\sum_{i=1}^n \|A_i - LM_iR^T\|_F^2 &= \sum_{i=1}^n \text{trace}((A_i - LM_iR^T)(A_i - LM_iR^T)^T) \\
&= \sum_{i=1}^n \text{trace}(A_iA_i^T) + \sum_{i=1}^n \text{trace}(M_iM_i^T) \\
&\quad - 2 \sum_{i=1}^n \text{trace}(LM_iR^T A_i^T), \tag{2}
\end{aligned}$$

where the second term  $\sum_{i=1}^n \text{trace}(M_iM_i^T)$  results from the fact that both  $L$  and  $R$  have orthonormal columns, and  $\text{trace}(AB) = \text{trace}(BA)$ , for any two matrices.

Since the first term on the right hand side of Eq. (2) is a constant, the minimization in Eq. (1) is equivalent to minimizing

$$\sum_{i=1}^n \text{trace}(M_iM_i^T) - 2 \sum_{i=1}^n \text{trace}(LM_iR^T A_i^T). \tag{3}$$

It is easy to check that the minimum of (3) is achieved, only if  $M_i = L^T A_i R$ , for every  $i$ . This completes the proof of the theorem.  $\square$

- Theorem 2.1 implies that  $M_i$  is uniquely determined by  $L$  and  $R$  with  $M_i = L^T A_i R$ , for all  $i$ . Hence the key step for the minimization in Eq. (1) is the computation of the common transformations  $L$  and  $R$ . A key property of the optimal transformations  $L$  and  $R$  is stated in the following theorem:

**Theorem 2.2.** *Let  $L$ ,  $R$  and  $\{M_i\}_{i=1}^n$  be the optimal solution to the minimization problem in Eq. (1). Then  $L$  and  $R$  solve the following optimization problem:*

$$\max_{\substack{L \in \mathbb{R}^{r \times \ell_1} : L^T L = I_{\ell_1} \\ R \in \mathbb{R}^{c \times \ell_2} : R^T R = I_{\ell_2}}} \sum_{i=1}^n \|L^T A_i R\|_F^2. \tag{4}$$

*Proof.* From Theorem 2.1,  $M_i = L^T A_i R$ , for every  $i$ . Substituting this into  $\sum_{i=1}^n \|A_i - LM_iR^T\|_F^2$ , we obtain

$$\sum_{i=1}^n \|A_i - LM_iR^T\|_F^2 = \sum_{i=1}^n \|A_i\|_F^2 - \sum_{i=1}^n \|L^T A_i R\|_F^2. \tag{5}$$

Hence the minimization in Eq. (1) is equivalent to the maximization of

$$\sum_{i=1}^n \|L^T A_i R\|_F^2,$$

which completes the proof of the theorem.  $\square$

- To the best of our knowledge, there is no closed form solution for the maximization problem in Eq. (4). A key observation, which leads to an iterative algorithm for the computation of  $L$  and  $R$ , is stated in the following theorem:

**Theorem 2.3.** Let  $L$ ,  $R$  and  $\{M_i\}_{i=1}^n$  be the optimal solution to the minimization problem in Eq. (1). Then

(1). For a given  $R$ ,  $L$  consists of the  $\ell_1$  eigenvectors of the matrix

$$M_L = \sum_{i=1}^n A_i R R^T A_i^T$$

corresponding to the largest  $\ell_1$  eigenvalues.

(2). For a given  $L$ ,  $R$  consists of the  $\ell_2$  eigenvectors of the matrix

$$M_R = \sum_{i=1}^n A_i^T L L^T A_i$$

corresponding to the largest  $\ell_2$  eigenvalues.

*Proof.* By Theorem 2.2,  $L$  and  $R$  maximize

$$\sum_{i=1}^n \|L^T A_i R\|_F^2,$$

which can be rewritten as

$$\begin{aligned} \sum_{i=1}^n \text{trace}(L^T A_i R R^T A_i^T L) &= \text{trace} \left( L^T \sum_{i=1}^n (A_i R R^T A_i^T) L \right) \\ &= \text{trace} (L^T M_L L), \end{aligned} \quad (6)$$

where  $M_L = \sum_{i=1}^n A_i R R^T A_i^T$ . Hence, for a given  $R$ , the maximum of

$$\sum_{i=1}^n \|L^T A_i R\|_F^2 = \text{trace} (L^T M_L L)$$

is achieved, only if  $L \in \mathbb{R}^{r \times \ell_1}$  consists of the  $\ell_1$  eigenvectors of the matrix  $M_L$  corresponding to the largest  $\ell_1$  eigenvalues.

Similarly, by the property of the trace of matrices,

$$\sum_{i=1}^n \|L^T A_i R\|_F^2$$

can also be rewritten as

$$\begin{aligned} \sum_{i=1}^n \text{trace}(R^T A_i^T L L^T A_i R) &= \text{trace} \left( R^T \sum_{i=1}^n (A_i^T L L^T A_i) R \right) \\ &= \text{trace} (R^T M_R R), \end{aligned} \quad (7)$$

where  $M_R = \sum_{i=1}^n A_i^T L L^T A_i$ . Hence, for a given  $L$ , the maximum of

$$\sum_{i=1}^n \|L^T A_i R\|_F^2 = \text{trace} (R^T M_R R)$$

is achieved, only if  $R \in \mathbb{R}^{c \times \ell_2}$  consists of the  $\ell_2$  eigenvectors of the matrix  $M_R$  corresponding to the largest  $\ell_2$  eigenvalues. This completes the proof of the theorem.  $\square$

- Theorem 2.3 results in an iterative procedure for computing  $L$  and  $R$  as follows: for a given  $L$ , we can compute  $R$  by computing the eigenvectors of the matrix  $M_R$ ; with the computed  $R$ , we can then update  $L$  by computing the eigenvectors of the matrix  $M_L$ . The procedure can be repeated until convergence. The pseudo-code of the above iterative procedure is given in **Algorithm GLRAM** below.

---

**Algorithm GLRAM**

**Input:** matrices  $\{A_i\}_{i=1}^n$ ,  $\ell_1$ , and  $\ell_2$

**Output:** matrices  $L$ ,  $R$ , and  $\{M_i\}_{i=1}^n$

---

1. Obtain initial  $L_0$  for  $L$  and set  $i \leftarrow 1$ ;
  2. While not convergent
  3.   form the matrix  $M_R = \sum_{j=1}^n A_j^T L_{i-1} L_{i-1}^T A_j$ ;
  4.   compute the  $\ell_2$  eigenvectors  $\{\phi_j^R\}_{j=1}^{\ell_2}$  of  $M_R$  corresponding to the largest  $\ell_2$  eigenvalues;
  5.    $R_i \leftarrow [\phi_1^R, \dots, \phi_{\ell_2}^R]$ ;
  6.   form the matrix  $M_L = \sum_{j=1}^n A_j R_i R_i^T A_j^T$ ;
  7.   compute the  $\ell_1$  eigenvectors  $\{\phi_j^L\}_{j=1}^{\ell_1}$  of  $M_L$  corresponding to the largest  $\ell_1$  eigenvalues;
  8.    $L_i \leftarrow [\phi_1^L, \dots, \phi_{\ell_1}^L]$ ;
  9.    $i \leftarrow i + 1$ ;
  10. EndWhile
  11.  $L \leftarrow L_{i-1}$ ;
  12.  $R \leftarrow R_{i-1}$ ;
  13. For  $j$  from 1 to  $n$
  14.    $M_j \leftarrow L^T A_j R$ ;
  15. EndFor
- 

- Theorem 2.3 implies that the matrix updates in Lines 5 and 8 of GLRAM do not decrease the value of  $\sum_{i=1}^n \|L^T A_i R\|_F^2$ , since the computed  $R$  and  $L$  are locally optimal. Hence by Theorem 2.2, the value of  $\sum_{i=1}^n \|A_i - L M_i R^T\|_F^2$ , or

$$\text{RMSRE} \equiv \sqrt{\frac{1}{n} \sum_{i=1}^n \|A_i - L M_i R^T\|_F^2} \quad (8)$$

does not increase. Here RMSRE stands for the *Root Mean Square Reconstruction Error*. The convergence of GLRAM follows, since RMSRE is bounded from below by 0, as stated in the following Theorem:

**Theorem 2.4.** *The GLRAM Algorithm monotonically non-increases the RMSRE value as defined in Eq. (8), hence it converges in the limit.*

Table 1: Statistics of our test datasets.

Dataset	Size ( $n$ )	Dimension ( $r \times c$ )	Number of classes
RAND	500	$100 \times 100 = 10000$	—
PIX	300	$100 \times 100 = 10000$	30
ORL	400	$92 \times 112 = 10304$	40
AR	1638	$101 \times 88 = 8888$	126
PIE	6615	$32 \times 24 = 768$	63
USPS	3000	$16 \times 16 = 256$	10

### 3 Evaluation

#### 3.1 Datasets

- The statistics of all datasets are summarized in Table 1.
- RAND is a synthetic dataset, consisting of 500 data points of size  $100 \times 100$ . All the entries are randomly generated between 0 and 255 (the same range as the four face image datasets).

#### 3.2 Effect of the ratio of $\ell_1$ to $\ell_2$ on reconstruction error

- In this experiment, we study the effect of the ratio of  $\ell_1$  to  $\ell_2$  on reconstruction error, where  $\ell_1$  and  $\ell_2$  are the row and column dimensions of the reduced representation  $M_i$  in GLRAM. To this end, we run GLRAM with different combinations of  $\ell_1$  and  $\ell_2$  with a constant product  $\ell_1 \cdot \ell_2 = 400$ . The results on PIX, ORL, and AR are shown in Table 2. It is clear from the table that the RMSRE value is small, when  $\ell_1/\ell_2 \approx 1$ , and the minimum is achieved when  $\ell_1/\ell_2 = 1$  in all cases.
- To examine whether this is related to the fact that for images, the number of rows ( $r$ ) and the number of columns ( $c$ ) are comparable, we subsample the images in PIX down to a size of  $50 \times 100 = 5000$ . The result on this dataset is included in Table 2. Interestingly, we observe the same trend in this dataset. That is, the RMSRE value is small, when  $\ell_1/\ell_2 \approx 1$ . We have conducted similar experiments on other datasets and observed the same trend. This may be related to the effect of balancing between the left and right transformations involved in GLRAM.
- Finally, we examine the effect of the ratio using the synthetic dataset. The result on RAND is included in the last column of Table 2. We observe the same trend as other datasets. That is, the RMSRE value is small, when  $\ell_1/\ell_2 \approx 1$ .
- The above experiment on both the synthetic and real-world datasets suggests that choosing  $\ell_1/\ell_2 \approx 1$  may be a good strategy in practice. In all the following experiments, we set both  $\ell_1$  and  $\ell_2$  equal to a common value  $d$ .

#### 3.3 Sensitivity of GLRAM to the choice of the initial $L_0$

- In this experiment, we examine the sensitivity of GLRAM to the choice of the initial  $L_0$  for  $L$  (see Line 1 of the GLRAM algorithm). To this end, we run GLRAM with 10 different initial  $L_0$ 's. The first one is  $L_0 = (I_d, 0)^T$ , while the next nine being randomly generated.

Table 2: Effect of the ratio of  $\ell_1$  to  $\ell_2$  on reconstruction error: Row shown in bold has minimum RMSRE (where  $\ell_1 = \ell_2$ ).

Parameters		Datasets				
$\ell_1$	$\ell_2$	PIX	ORL	AR	PIX (50 × 100)	RAND
5	80	569.06	2128.8	3605.4	384.67	7189.9
8	50	441.72	1737.2	2822.0	290.97	7177.5
10	40	387.47	1580.1	2457.4	250.55	7174.1
16	25	294.90	1376.9	1978.3	180.88	7170.9
<b>20</b>	<b>20</b>	<b>278.01</b>	<b>1367.3</b>	<b>1902.8</b>	<b>169.28</b>	<b>7170.6</b>
25	16	279.81	1423.9	1965.4	172.43	7171.1
40	10	349.12	1697.6	2379.1	226.90	7174.2
50	8	406.06	1864.6	2629.4	269.79	7177.2
80	5	529.26	2366.1	3426.3	—	7190.0

- First, we study the sensitivity of GLRAM using the image datasets. The result on ORL is shown in Figure 1 (left), where the horizontal axis is the number of iterations and the vertical axis is the RMSRE value (on a log scale).  $d$  is set to be 10.

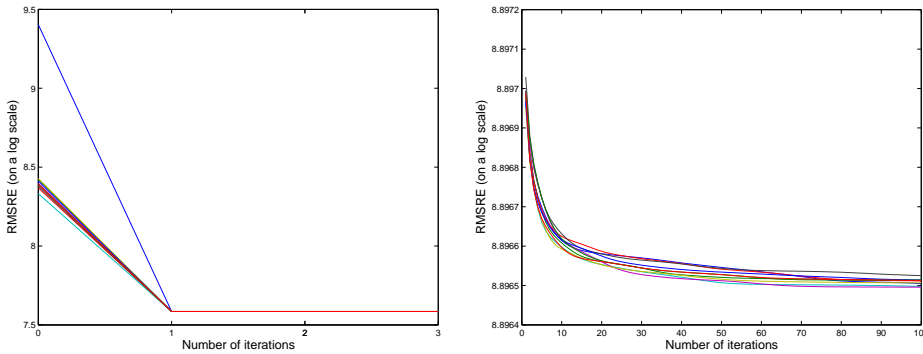


Figure 1: Sensitivity of GLRAM to the choice of the initial  $L_0$  on ORL (left) and RAND (right). The ten curves correspond to the ten runs with different initial  $L_0$ 's. The horizontal axis is the number of iterations and the vertical axis is the RMSRE value (on a log scale).

- We can observe from the figure that GLRAM converges rapidly for all ten initial choices of  $L$ . It converges within two to three iterations with the specified threshold ( $\eta = 10^{-6}$ ). For all ten different initial  $L_0$ 's, GLRAM converges to the same RMSRE value. To check whether GLRAM converges to the same solution, we compare the resulting left transformations  $L$  from the ten different runs. Two transformations can be compared by computing the largest principal angle between the column spaces of these two transformations. The angle between the left transformation resulting from the first run and the ones from other nine runs are computed (results omitted). For all cases, the angles are around  $10^{-10}$  to  $10^{-7}$ . This implies that GLRAM essentially converges to the same solution (subject to an orthogonal transformation) for the ten different runs. We observe the same trend in other four image datasets (PIX, AR, PIE, and USPS) as well as different values of  $d$  and the results are omitted.

- Next, we examine the sensitivity of GLRAM using RAND, the synthetic dataset. The result is shown in Figure 1 (right). It is clear from the figure that GLRAM converges much slower on RAND than on image datasets. We run GLRAM with the threshold  $\eta = 10^{-6}$ , and it does not converge until 78 iterations. Furthermore, GLRAM does not converge to the same solution (measured by the angle between two subspaces). Further experiments also show that the final RMSRE value may be different for different initial  $L_0$ 's, even though the difference always seems small. This is likely due to the fact that there are some similarities among the images in the same image datasets, while the data in RAND is randomly generated.
- The experiment above implies that for datasets with some hidden structures, such as faces and handwritten digits, GLRAM may converge to the global solution, regardless of the choice of the initial  $L_0$ . However, it is not true in general, as shown in the RAND dataset.

### 3.4 Compression effectiveness

- In this experiment, we examine the quality of the images compressed by the proposed algorithm and compare it with SVD and 2DPCA. Image compression is commonly applied as a pre-processing step for storage and transmission of large image data. There exists a tradeoff between quality of compressed images and compression ratio, as a high compression ratio usually leads to poor quality of compressed images.
- Figure 2 shows images of 10 different persons from the ORL dataset. The 10 images in the first row are the original images from the dataset. The 10 images in the second row are the ones compressed by the GLRAM algorithm with  $d = 10$ . The compression ratio is about 98.0. The images compressed by SVD and 2DPCA with approximately the same number of reduced dimensions as GLRAM are shown in the third and fourth rows of Figure 2 respectively. It is clear that the images compressed by our proposed algorithm have slightly better visual quality than those compressed by 2DPCA, while the ones compressed by SVD have the best visual quality. However, the compression ratio of SVD (3.85) is much smaller than that of GLRAM (98.0).
- Figure 3 shows images of 10 different digits from the USPS dataset.  $d = 5$  is used in GLRAM. The compression ratio is about 10. GLRAM and SVD perform slightly better than 2DPCA. Furthermore, the compression ratio of SVD (9.4) is close to that of GLRAM (10.2). The different behavior between ORL and USPS is related to the fact that USPS has a relatively large number of data points compared to its dimension, i.e.,  $n \gg rc$ .



Figure 2: First row: raw images from ORL dataset. Second row: images compressed by GLRAM. Third row: images compressed by SVD. Fourth row: images compressed by 2DPCA. Note that the compression ratio of SVD (3.85) is much smaller than that of GLRAM (98.0).

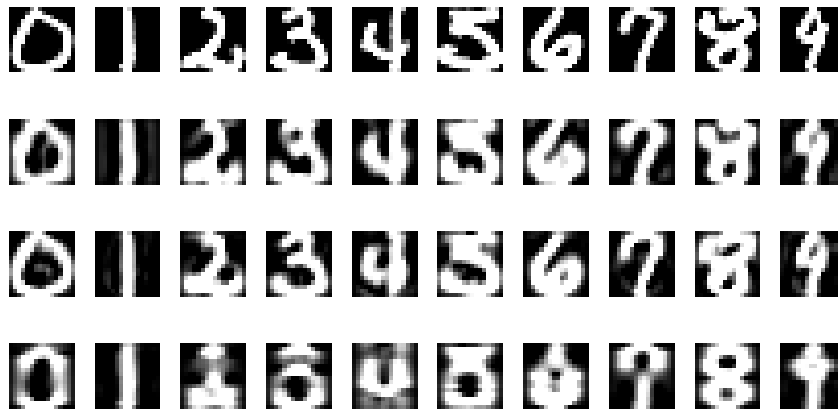


Figure 3: First row: raw images from USPS dataset. Second row: images compressed by GLRAM. Third row: images compressed by SVD. Fourth row: images compressed by 2DPCA.