Matrix Computations

Singular Value Decomposition

Fengpei Li

Supervisors: Jie Lu, Ziping Zhao, ShanghaiTech University

Table of contents

Singular Value Decomposition Algorithms

Power iteration

The SVD decomposition: $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$

- assume $m \geq n$ and $\sigma_1 > \sigma_2 > \cdots > \sigma_n > 0$

The power iteration can be used to compute the thin SVD, and the idea is as follows.

- form $\mathbf{A}^T\mathbf{A} = \mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^T$
- apply the power iteration to $\mathbf{A}^T\mathbf{A}$ to obtain $\mathbf{v}_1$ and $\lambda_1\left(\mathbf{A}^T\mathbf{A}\right) = \mathbf{v}_1\mathbf{A}^T\mathbf{A}\mathbf{v}_1$
- obtain $\mathbf{u}_1 = \mathbf{A}\mathbf{v}_1 / \left\|\mathbf{A}\mathbf{v}_1\right\|_2$, $\sigma_1 = \left\|\mathbf{A}\mathbf{v}_1\right\|_2$
- do deflation $\mathbf{A} \coloneqq \mathbf{A} - \sigma_1\mathbf{u}_1\mathbf{v}_1^T$, and repeat the above steps until all singular components are found

Exercise 1

Implement the power iteration for SVD decomposition.

QR iteration

The QR iteration can be used to compute the SVD, and the idea is as follows.

1. form $\mathbf{A}^T\mathbf{A}$
2. apply the (symmetric) QR iteration to obtain the eigendec. $\mathbf{A}^T\mathbf{A} = \mathbf{V}_1\tilde{\mathbf{\Sigma}}^2\mathbf{V}_1^T$
3. solve $\mathbf{U}_1\tilde{\mathbf{\Sigma}} = \mathbf{A}\mathbf{V}_1$ via QR factorization with column pivoting where $\tilde{\mathbf{\Sigma}} \in \mathbb{R}^{r\times r}$ is a diagonal matrix with diagonal entries being the nonnegative square root of diagonal entries of $\tilde{\mathbf{\Sigma}}^2$

Remark

This approach is numerically unstable which depends on the $(\kappa(\mathbf{A}))^2$ (just as the issue in using the methods of normal equations for certain least squares problems)

Exercise 2

Compute the SVD decomposition of a 3-by-2 matrix by the QR iteration

$$\mathbf{B} = \begin{bmatrix} 3 & 2 \\ 1 & 4 \\ 0.5 & 0.5 \end{bmatrix}$$

Are the results right? If not, can you get the correct answer through the results of QR iteration?

SVD via Symmetric QR Iteration

- Associated with any $\mathbf{A}$ is the real symmetric matrix $\mathbf{A}^T\mathbf{A}$, whose eigenvalues tell us what the singular values of $\mathbf{A}$ are, but the relationship between the eigenvalues of $\mathbf{A}^T\mathbf{A}$ and the singular values of $\mathbf{A}$ is nonlinear.
- another real symmetric matrix assoc. with $\mathbf{A}$ has better properties in this regard
- let $\mathbf{A} \in \mathbb{R}^{m \times n}$ and define the real symmetric matrix

$$\mathbf{J} = \begin{bmatrix} \mathbf{0} & \mathbf{A}^T \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \in \mathbb{S}^{m+n}$$

  - matrix $\mathbf{J}$ is called the Jordan-Wielandt matrix
  - eigenvalues of $\mathbf{J}$ are $\pm\sigma_1(\mathbf{A}), \ldots, \pm\sigma_p(\mathbf{A})$ together with $|m-n|$ zeros
  - eigenvector of $\mathbf{J}$ associated with $\pm\sigma_i(\mathbf{A})$ $(i = 1, \ldots, p)$ is $\frac{1}{\sqrt{2}}\begin{bmatrix} \mathbf{v}_i^T & \pm\mathbf{u}_i^T \end{bmatrix}^T$

SVD via Symmetric QR Iteration

- if $m \geq n$, $\mathbf{J}$ obtains an eigendecomposition given by

$$\mathbf{J} = \mathbf{Q}\mathrm{Diag}\left(\sigma_1(\mathbf{A}), \ldots, \sigma_p(\mathbf{A}), -\sigma_1(\mathbf{A}), \ldots, -\sigma_p(\mathbf{A}), \underbrace{0, \ldots, 0}_{m-n \text{ zeros}}\right)\mathbf{Q}^T$$

  where $\mathbf{Q}$ is

$$\mathbf{Q} = \frac{1}{\sqrt{2}}\begin{bmatrix} \mathbf{V} & \mathbf{V} & \mathbf{0} \\ \mathbf{U}_1 & -\mathbf{U}_1 & \sqrt{2}\mathbf{U}_2 \end{bmatrix}$$

- **Fact:** by applying symmetric QR iteration to $\mathbf{J}$ to find $\mathbf{U}$ and $\mathbf{V}$, we are *implicitly* computing the QR iteration of $\mathbf{A}^T\mathbf{A}$
- standard method to compute SVD from results for eigenvalues of real symmetric matrices

SVD via Symmetric QR Iteration

Algorithm 1: SVD via Symmetric QR Iteration

Input: $\mathbf{A} \in \mathbb{R}^{m \times n}$ $(m \geq n)$
form $\mathbf{J}$
$[\mathbf{Q}, \mathbf{\Lambda}] = \text{SymQRIteration}(\mathbf{J})$ % symmetric QR iteration
obtain $\mathbf{U}$ and $\mathbf{V}$ from $\mathbf{Q}$
obtain $\mathbf{\Sigma}$ from $\mathbf{\Lambda}$
Output: $\mathbf{U}$, $\mathbf{\Sigma}$, $\mathbf{V}$

Exercise 3

Implement the Symmetric QR Iteration for SVD decomposition.

SVD via Symmetric Tridiagonal QR Iteration

- Fact: any $\mathbf{A} \in \mathbb{R}^{m \times n}$ can be unitarily transformed to an upper bidiagonal form as $\mathbf{B} = \mathbf{U}_B^T \mathbf{A} \mathbf{V}_B$ where $\mathbf{B}$ is upper bidiagonal, but a diagonal form is not attainable
- it is easy to show if $\mathbf{B}$ is bidiagonal then $\mathbf{B}^T\mathbf{B}$ is symmetric tridiagonal
  - the **bidagonal reduction** of $\mathbf{A}$ is related to the tridiagonal reduction of $\mathbf{A}^T\mathbf{A}$
- for $\mathbf{A} \in \mathbb{R}^{m \times n}$ $(m \geq n)$, the standard method for SVD computation is
  1. apply orthogonal transformations to abtain a upper bidiagonal form
  2. diagonalize the bidiagonal form

$$\mathbf{A} = \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix} \xrightarrow{\text{stage 1}} \begin{bmatrix} \times & \times & & \\ & \times & \times & \\ & & \times & \times \\ & & & \times \\ & & & \end{bmatrix} \xrightarrow{\text{stage 2}} \begin{bmatrix} \times & & & \\ & \times & & \\ & & \times & \\ & & & \times \\ & & & \end{bmatrix}$$

SVD via Symmetric Tridiagonal QR Iteration

- **Bidiagonal reduction**: applying Householder reflectors alternately on the left and right
  - left reflector introduces zeros below the diagonal
  - right reflector introduces a row of zeros to the right of the first superdiagonal

$$\mathbf{A} = \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix} \xrightarrow{\mathbf{U}_1^T} \underbrace{\begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \end{bmatrix}}_{\tilde{\mathbf{A}}_1 = \mathbf{U}_1^T \mathbf{A}} \xrightarrow{\mathbf{V}_1} \underbrace{\begin{bmatrix} \times & \times & 0 & 0 \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \end{bmatrix}}_{\mathbf{A}_1 = \mathbf{U}_1^T \mathbf{A} \mathbf{V}_1} \longrightarrow \dots$$

  - $\mathbf{U}_1^T$ is the Householder reflector that reflects $\mathbf{A}(1:m,1)$
  - $\mathbf{V}_1 = \begin{bmatrix} 1 & 0 \\ 0 & \tilde{\mathbf{V}}_1 \end{bmatrix}$ with $\tilde{\mathbf{V}}_1$ the Householder reflector that reflects $\tilde{\mathbf{A}}_1(1, 2:n)$

SVD via Symmetric Tridiagonal QR Iteration

- finally, we obtain

$$\underbrace{\mathbf{U}_n^T \mathbf{U}_{n-1}^T \cdots \mathbf{U}_1^T}_{\mathbf{U}_B^T} \mathbf{A} \underbrace{\mathbf{V}_1 \mathbf{V}_2 \cdots \mathbf{V}_{n-2}}_{\mathbf{V}_B} = \mathbf{B}$$

  where $\mathbf{B}$ is a bidiagonal matrix that has the form

$$\mathbf{B} = \begin{bmatrix} \alpha_1 & \beta_1 & & \\ & \alpha_2 & \ddots & \\ & & \ddots & \beta_{n-1} \\ & & & \alpha_n \\ & & & \\ & & & \end{bmatrix} \in \mathbb{R}^{m \times n}$$

  and it can be verified that $\alpha_i \geq 0$ and $\beta_i \geq 0$
- complexity: $\mathcal{O}\left(4mn^2\right)$
- also called Golub-Kahan bidiagonalization

SVD via Symmetric Tridiagonal QR Iteration

- SVD of bidiagonal form $\mathbf{B}$: the task is to solve a real symmetric eigenvalue problem for $\mathbf{B}^T\mathbf{B}$, $\mathbf{B}\mathbf{B}^T$, or $\mathbf{J}_B = \begin{bmatrix} \mathbf{0} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{bmatrix}$
    - permutations are applied so that $\mathbf{\Pi} \begin{bmatrix} \mathbf{0} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{bmatrix} \mathbf{\Pi}^T$ is symmetric tridiagonal, and then methods for symmetric tridiagonal eigenvalue problems such as divideand-conquer (cf. Chapter 8.3-8.5 of [Golub-Van Loan'13]) can be used
    - implicit QR iteration for $\mathbf{B}^T\mathbf{B}$ or $\mathbf{B}\mathbf{B}^T$ by directly working on $\mathbf{B}$ (cf. Chapter 8.6.3 of [Golub-Van Loan'13])
- after we get the SVD

$$\mathbf{B} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$$

- the SVD for $\mathbf{A}$ is given by

$$\mathbf{A} = \underbrace{\mathbf{U}_B\tilde{\mathbf{U}}}_{\mathbf{U}}\mathbf{\Sigma}\underbrace{\tilde{\mathbf{V}}^T\mathbf{V}_B^T}_{\mathbf{V}^T}$$

SVD via Symmetric Tridiagonal QR Iteration

Algorithm 2: SVD via Symmetric Tridiagonal QR Iteration

Input: $\mathbf{A} \in \mathbb{R}^{m \times n}$

$\mathbf{B} = \mathbf{U}_B^T \mathbf{A} \mathbf{V}_B$ % bidiagonal reduction for $\mathbf{A}$

form $\mathbf{J}_B$

$[\mathbf{Q}, \mathbf{\Lambda}] = \text{SymTriQRIteration}\left(\mathbf{\Pi} \mathbf{J}_B \mathbf{\Pi}^T\right)$ % symmetric tridiagonal QR iteration

obtain $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{V}}$ from $\mathbf{Q}$

obtain $\mathbf{\Sigma}$ from $\mathbf{\Lambda}$

$\mathbf{U} = \mathbf{U}_B \tilde{\mathbf{U}}$

$\mathbf{V} = \mathbf{V}_B \tilde{\mathbf{V}}$

Output: $\mathbf{U}$, $\mathbf{\Sigma}$, $\mathbf{V}$

SVD via iterative QR algorithm

Since the SVD decomposition consists of two orthogonal matrices $\mathbf{U}$, $\mathbf{V}$ and one diagonal matrix $\mathbf{\Sigma}$. We can repeatedly perform QR decomposition on $\mathbf{A}$ to get its SVD

- The idea is to use the QR decomposition on $\mathbf{A}$ to gradually "pull" $\mathbf{U}$ out from the left and then use QR on $\mathbf{A}^T$ to "pull" $\mathbf{V}$ out from the right.
- This process makes $\mathbf{A}$ lower triangular and then upper triangular alternately.
- Eventually, $\mathbf{A}$ becomes both upper and lower triangular at the same time, (i.e. Diagonal) with the singular values on the diagonal.

SVD via iterative QR

Algorithm 3: SVD via iterative QR Iteration

Input: $\mathbf{A} \in \mathbb{R}^{m \times n}$

Initialize $\mathbf{U}^{(0)} = \mathbf{V}^{(0)} = \mathbf{I}$, $\mathbf{\Sigma}^{(0)} = \mathbf{A}$

for $i = 1, \ldots, n$ **do**

$\quad \left[\mathbf{Q}, \mathbf{\Sigma}^{(i+0.5)}\right] = \mathrm{QR}\left(\mathbf{\Sigma}^{(i)}\right)$

$\quad$ Update $\mathbf{U}^{(i)} = \mathbf{U}^{(i-1)}\mathbf{Q}$

$\quad \left[\mathbf{Q}, \left(\mathbf{\Sigma}^{(i+1)}\right)\right] = \mathrm{QR}\left(\left(\mathbf{\Sigma}^{(i+0.5)}\right)^T\right)$

$\quad$ Update $\mathbf{V}^{(i)} = \mathbf{V}^{(i-1)}\mathbf{Q}$

end

Output: $\mathbf{U}$, $\mathbf{\Sigma}$, $\mathbf{V}$

Exercise 4

Implement an iterative algorithm for SVD decomposition using QR.

Applications of SVD

Least Squares via SVD

- consider solving the linear system $\mathbf{y} = \mathbf{A}\mathbf{x}$ when $\mathbf{A}$ is fat
- this is an underdetermined problem: we have more unknowns n than the number of equations m
- assume that $\mathbf{A}$ has full row rank. By now we know that any

$$\mathbf{x} = \mathbf{A}^{\dagger}\mathbf{y} + \boldsymbol{\eta}, \text{ for any } \boldsymbol{\eta} \in \mathcal{R}(\mathbf{V}_2) = \mathcal{N}(\mathbf{A})$$

- Idea: discard $\boldsymbol{\eta}$ and take $\mathbf{x} = \mathbf{A}^{\dagger}\mathbf{y}$ as our solution
- Question: does discarding $\boldsymbol{\eta}$ make sense ?
- Answer: it makes sense under the minimum 2-norm problem formulation

$$\min_{\mathbf{x} \in \mathbb{R}^n} \quad \|\mathbf{x}\|_2^2 \qquad \text{s.t. } \mathbf{y} = \mathbf{A}\mathbf{x}.$$

  It can be shown that the solution is uniquely given by $\mathbf{y} = \mathbf{A}^{\dagger}\mathbf{x}$.

Least Squares via SVD

- consider the least squares problem

$$\min_{\mathbf{x}\in\mathbb{R}^n} \quad \|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2^2$$

  for general $\mathbf{A} \in \mathbb{R}^{m\times n}$

- we have, for any $\mathbf{x} \in \mathbb{R}^n$,

$$\begin{aligned}
\|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2^2 &= \left\|\mathbf{y}-\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{x}\right\|_2^2 = \left\|\mathbf{U}^T\mathbf{y}-\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{x}\right\|_2^2 \\
&= \left\|\begin{bmatrix} \mathbf{U}_1^T \\ \mathbf{U}_2^T \end{bmatrix}\mathbf{y} - \begin{bmatrix} \tilde{\boldsymbol{\Sigma}}\mathbf{V}_1^T \\ \mathbf{0} \end{bmatrix}\mathbf{x}\right\|_2^2 \\
&= \left\|\mathbf{U}_1^T\mathbf{y}-\tilde{\boldsymbol{\Sigma}}\mathbf{V}_1^T\mathbf{x}\right\|_2^2 + \left\|\mathbf{U}_2^T\mathbf{y}\right\|_2^2 \\
&\geq \left\|\mathbf{U}_2^T\mathbf{y}\right\|_2^2
\end{aligned}$$

- the equality above is attained if $\mathbf{x}$ satisfies $\mathbf{U}_1^T\mathbf{y} = \tilde{\boldsymbol{\Sigma}}\mathbf{V}_1^T\mathbf{x}$, and that leads to an least squares solution

$$\begin{aligned}
\mathbf{U}_1^T\mathbf{y} = \tilde{\boldsymbol{\Sigma}}\mathbf{V}_1^T\mathbf{x} &\Longleftrightarrow \mathbf{V}_1^T\mathbf{x} = \tilde{\boldsymbol{\Sigma}}^{-1}\mathbf{U}_1^T\mathbf{y} \\
&\Longleftrightarrow \mathbf{x} = \mathbf{V}_1\tilde{\boldsymbol{\Sigma}}^{-1}\mathbf{U}_1^T\mathbf{y} + \boldsymbol{\eta}, \text{ for any } \boldsymbol{\eta} \in \mathcal{R}(\mathbf{V}_2) = \mathcal{N}(\mathbf{A})
\end{aligned}$$

Low-Rank Matrix Approximation

truncated SVD provides the best approximation in the least squares sense

Theorem (Eckart-Young-Mirsky)
Consider the following problem

$$\min_{\mathbf{A}\in\mathbb{R}^{m\times n}, rank(\mathbf{B})\le k} \|\mathbf{A}-\mathbf{B}\|_F^2 \, ,$$

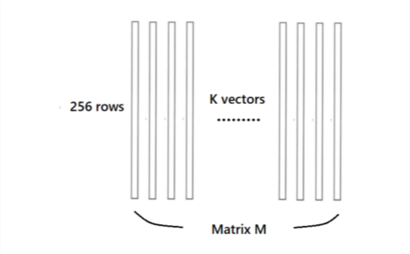
where $\mathbf{A} \in \mathbb{R}^{m\times n}$ *and* $k \in \{1, \ldots, p\}$ *are given. The truncated SVD* $\mathbf{A}_k$ *is an optimal solution to the above problem and the minimum is* $\sum_{i=k+1}^{p} \sigma_i^2$.

Exercise: Image Compression via SVD

Exercise: Handwritten Digits Classification via SVD

In this exercise, we will discuss the application of SVD in handwritten digits classification. In the MNIST dataset, each digit picture is represented as a 16-by-16 matrix, we vectorize them into $\mathbb{R}^{256}$ vectors and use these vectors as columns to construct the data matrix.

- Construct the training matrices $\mathbf{M}_0, \mathbf{M}_1, \ldots, \mathbf{M}_9$ corresponding to the 10 digits.
- Perform SVD decomposition on the 10 matrices to obtain the corresponding 10 orthogonal matrices $\mathbf{U}_0, \ldots, \mathbf{U}_9$, which are the orthonormal basis for the column space of matrix $\mathbf{M}$.



Exercise: Handwritten Digits Classification via SVD

Proposition *Suppose* $\mathbf{M} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$ *, then* $\{\mathbf{u}_1, \ldots, \mathbf{u}_r\}$ *form an orthonormal basis of the column space of* $\mathbf{M}$*, where* r *is the rank of* $\mathbf{M}$ *and* $\mathbf{u}_i$ *is the* i*th column of* $\mathbf{U}$*.*

- For any unknown digit vector $\mathbf{q}$, we define residual between $\mathbf{q}$ and the orthonormal basis of the z-th digit as the distance between $\mathbf{q}$ and $\mathrm{Proj}_{\mathbf{U}_z}(\mathbf{q})$, which is given as

$$\left\| \mathbf{q} - \sum_{i=1}^{r} \langle \mathbf{q}, \mathbf{u}_{z,i} \rangle \mathbf{u}_{z,i} \right\|_2 .$$

- Since there are 10 digits in total, we want to compute the residual between $\mathbf{q}$ and each of these 10 orthonormal basis and classify the unknown digit as z, where the residual between $\mathbf{q}$ and the orthonormal basis for digit z is smallest among all the ten digits.

Now, try to fill in the code in *exercise6_digit_classification.m* to implement handwritten digit classification.