Numerical Linear Algebra

- SVD Decomposition.
- Schur factorization.
- Eigenvalue finding.

Singular Value Decomposition

- Singular Value Decomposition (SVD) is the generalization of the eigendecomposition of a positive semidefinite normal matrix to any $m \times n$ matrix.

- Singular Value Decomposition (SVD) is the generalization of the eigendecomposition of a positive semidefinite normal matrix to any $m \times n$ matrix.
- It is a factorization of a matrix M into:

$$M = U \Sigma V^*$$

where U is $m \times m$ a unitary matrix, Σ is a $m \times n$ rectangular diagonal matrix with non-negative real numbers on the diagonal and V is a $n \times n$ unitary matrix.

Singular Value Decomposition

- Singular Value Decomposition (SVD) is the generalization of the eigendecomposition of a positive semidefinite normal matrix to any $m \times n$ matrix.
- It is a factorization of a matrix M into:

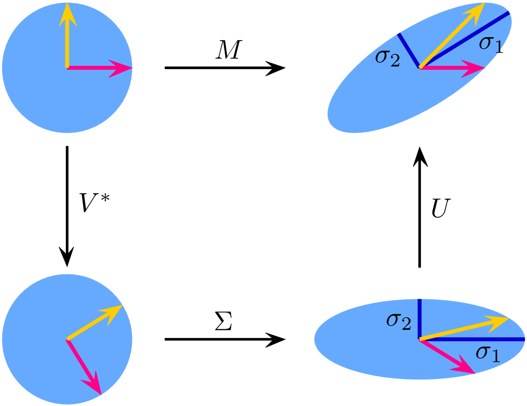
$$M = U\Sigma V^*$$

where U is $m \times m$ a unitary matrix, Σ is a $m \times n$ rectangular diagonal matrix with non-negative real numbers on the diagonal and V is a $n \times n$ unitary matrix.

- The diagonal entries σ_i of Σ are known as the singular values of M . The columns of U and the columns of V are called the left-singular vectors and right-singular vectors of M, respectively.

Physical meaning of SVD

Singular Value Decomposition 4/11/17 4:06 AM



$$M = U \cdot \Sigma \cdot V^*$$

- The left-singular vectors of M are a set of orthonormal eigenvectors of MM^*.

- The left-singular vectors of M are a set of orthonormal eigenvectors of MM^*.
- The right-singular vectors of M are a set of orthonormal eigenvectors of M^*M.

- The left-singular vectors of M are a set of orthonormal eigenvectors of MM^*.
- The right-singular vectors of M are a set of orthonormal eigenvectors of M^*M.
- The non-zero singular values of M (found on the diagonal entries of Σ) are the square roots of the non-zero eigenvalues of both M^*M and MM^*.

Rank, Null space and Range

- The right-singular vectors corresponding to vanishing singular values of M span the null space of M

Rank, Null space and Range

- The right-singular vectors corresponding to vanishing singular values of M span the null space of M
- The left-singular vectors corresponding to the non-zero singular values of M span the range of M.

Rank, Null space and Range

- The right-singular vectors corresponding to vanishing singular values of M span the null space of M
- The left-singular vectors corresponding to the non-zero singular values of M span the range of M.
- The rank of M equals the number of non-zero singular values which is the same as the number of non-zero diagonal elements in Σ.

SVD and eigen decomposition

- Eigenvalue decomposition of a square matrix:

$$A = X\Lambda X^{-1}$$

where Λ is a diagonal matrix and X contains linearly independent eigenvectors of A.

SVD and eigen decomposition

- Eigenvalue decomposition of a square matrix:

$$A = X\Lambda X^{-1}$$

where Λ is a diagonal matrix and X contains linearly independent eigenvectors of A.

- SVD is the generalization of eigen decomposition to rectangular matrices.

SVD and eigen decomposition

- Eigenvalue decomposition of a square matrix:

$$A = X\Lambda X^{-1}$$

  where Λ is a diagonal matrix and X contains linearly independent eigenvectors of A.
- SVD is the generalization of eigen decomposition to rectangular matrices.
- SVD uses two bases (left and right singular vectors) while eigenvalue decomposition uses only one (just the eigenvectors).

SVD and eigen decomposition

- Eigenvalue decomposition of a square matrix:

$$A = X\Lambda X^{-1}$$

  where Λ is a diagonal matrix and X contains linearly independent eigenvectors of A.
- SVD is the generalization of eigen decomposition to rectangular matrices.
- SVD uses two bases (left and right singular vectors) while eigenvalue decomposition uses only one (just the eigenvectors).
- In applications, SVD is relevant for problems involving the matrix itself where as eigen decomposition is useful to compute iterated forms of the matrix – such as matrix powers or exponentials etc.

Schur Factorization

- One final factorization:

$$A = QTQ^*$$

where Q is unitary and T is upper-triangular.

Schur Factorization

- One final factorization:

$$A = QTQ^*$$

  where Q is unitary and T is upper-triangular.
- Since A and T are similar, eigenvalues of A neccesarily appear on the diagonal of T.

Schur Factorization

- One final factorization:

$$A = QTQ^*$$

  where Q is unitary and T is upper-triangular.
- Since A and T are similar, eigenvalues of A neccesarily appear on the diagonal of T.
- Diagonalization algorithms use this factorization.

- Any eigenvalue solver has to be iterative!

Diagonalization and Schur Factorization

- Any eigenvalue solver has to be iterative!
- Most of the general purpose eigenvalue algorithms proceed by computing the Schur factorization:

$$\underbrace{Q_j^* \cdots Q_2^* Q_1^*}_{Q^*} A \underbrace{Q_1 Q_2 \cdots Q_j}_{Q}$$

converges to an upper triangular matrix T as $j \to \infty$.

Two phases of eigenvalue computations

- Whether or not A is hermitian, the sequence is usually split into two phases – first a direct method is applied to produce a upper-Hessenberg matrix H, that is, a matrix with zeros below the first subdiagonal.

Two phases of eigenvalue computations

- Whether or not A is hermitian, the sequence is usually split into two phases – first a direct method is applied to produce a upper-Hessenberg matrix H, that is, a matrix with zeros below the first subdiagonal.
- In the second phase, an iteration is used to generate a formally infinite sequence of Hessenberg matrices that converge to a triangular form.

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix} \xrightarrow{Phase 1} \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \end{bmatrix}$$

$$\xrightarrow{Phase 2} \begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \\ & & & & \times \end{bmatrix}$$

Reduction to Hessenberg/Tridiagonal form

- Use Householder reflectors to introduce zeros, but leave the first row as it is!

Reduction to Hessenberg/Tridiagonal form

- Use Householder reflectors to introduce zeros, but leave the first row as it is!
- Upon applying with Q_1 on the right, it will not destroy the zeroes that you have!

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix} \xrightarrow{Q_1^*} \begin{bmatrix} \times & \times & \times & \times & \times \\ \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \end{bmatrix}$$

$$\xrightarrow{Q_1} \begin{bmatrix} \times & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \times & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ 0 & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ 0 & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ 0 & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \end{bmatrix}$$

Householder reduction to Hessenberg Form

The following algorithm computes the Householder reduction of A to Hessenberg form:

Algorithm 1 Householder reduction to Hessenberg form

```
for k = 1 to m − 2 do
    x = A_{k+1:m,k}
    v_k = sign(x_1)||x||_2 e_1 + x
    v_k = v_k/||v_k||_2
    A_{k+1:m,k:n} = A_{k+1:m,k:n} − 2v_k(v_k^* A_{k+1:m,k:n})
    A_{1:m,k+1:n} = A_{1:m,k+1:n} − 2v_k(v_k^* A_{1:m,k+1:n})
end for
```

Householder reduction to Tridiagonal form

- Reduces a symmetric/hermitian matrix to tridiagonal form.

Householder reduction to Tridiagonal form

- Reduces a symmetric/hermitian matrix to tridiagonal form.
- Work done $\sim \frac{4}{3}m^3 Flops$