Basic ideas on Singular Value Decomposition (SVD)

Numerical Linear Algebra 181116

The SVD decomposition

We consider $A \in \mathbb{R}^{m \times n}$. A can be written as

$$A = USV^t \tag{1}$$

where $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ are orthogonal, $S \in \mathbb{R}^{m \times n}$ is diagonal and

AA^t is considered cause A is typically m·n, where m>n, and only square matrices are diagonalizable.

- S contains contains the singular values (singular values $= \sqrt{\text{eigenvalues of} AA^t \text{ or } A^tA}$). The singular values appear ordered arranged in descending order.
- The columns of U are the left singular vectors (i.e. the eigenvectors of AA^t are the columns of U).
- The rows of V^t are the right singular vectors (i.e. the eigenvectors of A^tA are the columns of V).

Remarks:

1. The SVD represents an expansion of the original data in a coordinate system where the covariance matrix is diagonal (it gives information on the prinipal components (PCA)).
2. The relation $AV = US$ means that there exists a special orthonormal set of vectors (i.e. the columns of V), that is mapped by the matrix A into an orthonormal set of vectors (i.e. the columns of U).
3. The singular values are always real numbers (because A^tA and/or AA^t are symmetric matrices).
4. Recall that eigenvectors of different eigenvalues of symmetric matrices are orthogonal, hence U and V are orthogonal matrices. This follows from the fact that

   $$< Ax, y >=< x, A^t y >$$

   Then, if x, y are eigenvectors of different eigenvalues of A it follows that $\lambda < x, y >=< Ax, y >=< x, A^t y >$ and if A symmetric $< x, A^t y >= \mu < x, y >$ and hence $< x, y >= 0$.
5. The SVD decomposition is not unique, it is only unique up to a reflection of each of the singular vectors (because $s_k u_k v_k = s_k(-u_k)(-v_k)$).

Ex.1 Code a simple algorithm to compute SVD decomposition of a matrix A using the eigenvalues/eigenvectors of A^tA and AA^t.

Ex.2 Use the `scipy.linalg.svd` function to get the SVD decomposition of A.

Use of SVD for some linear algebra computations

We recall that:

1. The 2-norm of A is the maximum of the spectra radius of A^tA,
2. Frobenius norm of $A =$ sqrt of the sum by rows and columns of the A of the square of the elements.
3. The pseudoinversa (Moore-Penrose) of $A = USV^t$ is the matrix $A+ = V((1/S)^t)U^t$ (for a diagonal matrix one has $S+ = (1/S)^t$).

Ex.3 Write a program that uses SVD decomposition to compute

(a) the rank(A),

(b) the 2-norm of A,

(c) the Frobenius norm of A,

(d) the condition number $k_2(A)$,

(e) the pseudoinverse A^+ of A.

Use of SVD to solve Least Square problems

Let us consider a linear *projector*, that is, a linear map $P : \mathbb{R}^n \to \mathbb{R}^n$ such that $P^2 = P$ which defines a projection onto the linear subspace $V = \mathrm{Im}(P) = \{y \in \mathbb{R}^n$ for which $\exists x \in \mathbb{R}^n$ s.t. $y = P(x)$. If the matrix representing P is symmetric the projector P is *orthogonal.*

Given a nontrivial subspace $V \subset \mathbb{R}^n$ there exists a unique orthogonal projector P_V onto V. Given $A \in \mathbb{R}^{m\times n}$ the following properties hold:

1. $P_{\mathrm{Im}(A)} = AA^+$.
2. $P_{\mathrm{Ker}(A)} = I - A^+A$.

We consider the LS problem as follows: of all the vectors x which minimize $\|Ax-\|$, which is the shortest (e.g. the one with $\|x\|_2$ minimum)? That is, we look for the minimum norm solution of the least squares problem.

Solution: One has $Ax \approx b$, but $b \notin \mathrm{Im}(A)$ in general. This motivates to use the projector onto $\mathrm{Im}(A)$ and consider the problem $Ax = AA^+b$ instead. Then

$$Ax = AA^+b \Leftrightarrow A(x - A^+b) = 0 \Leftrightarrow x - A^+b \in \mathrm{Ker}(A) \Leftrightarrow \exists w \in \mathbb{R}^n \text{ s.t. } x - A^+b = (I - A^+A)w$$

hence

$$x = A^+b + (I - A^+A)w$$

where $w \in \mathbb{R}^n$ is an arbitrary vector. Taking $w = 0$ we minimize $\|x\|_2$, then

$$\boxed{x = A^+b}$$

is the LS solution with minimum norm.

Ex.4 Write a program that uses SVD decomposition to solve the LS problem.

Computing the SVD decomposition

There are different strategies to compute the SVD of a matrix $A \in \mathbb{R}^{m\times n}$. Below we describe a method to compute the singular values.

The method consists in two main steps:

Step 1. Bidiagonalize the matrix

$$H = \begin{pmatrix} 0 & A^t \\ A & 0 \end{pmatrix}$$

to obtain $B \in \mathbb{R}^{(m+n)\times(m+n)}$ upper bidiagonal.

Step 2. Use LR-type iteration to diagonalize B and obtain the singular values in the diagonal.

Ex.1 Write a routine to check that:

(a) the eigenvalues of H are $\pm s_i$, $i = 1 \dots n$, where s_i, $i = 1, \dots, n$ are the singular values of A.

(b) if v is an eigenvector of H then $\sqrt{2}v$ is a column of

$$\begin{pmatrix} V \\ \pm U \end{pmatrix} \text{ or } \begin{pmatrix} -V \\ \pm U \end{pmatrix}$$

The bidiagonalization process can be carried out applying Householder transformations.

Ex.2 Write a function `house(x)` such that given $x \in \mathbb{R}^l$ computes $v \in \mathbb{R}^l$, with $v_1 = 1$, and $\beta \in \mathbb{R}$ so that the Householder transformation $P = I - \beta vv^t$ vis such that $P(x) = \|x\|_2 e_1$, $e_1 = (1, 0, \dots, 0)$. This can be achieved by the algorithm 5.1.1. in Matrix computations, Golub-Van Loan, 3rd ed. p210.

In practice, one never forms the matrix P explicitly. Note that one has

$$PA = (I - \beta vv^t)A = A - vw_1^t$$

where $w_1 = \beta A^t v$ and

$$AP = A(I - \beta vv^t)A = A - w_2 v^t$$

where $w_2 = \beta A v$.

Ex.3 Write functions `PA(bet,v,A)` and `AP(bet,v,A)` that perform the previous updating computations.

Ex.4 Write a function `bidiag(A)` that performs the bidiagonalization of A by applying Householder transformations. If $A \in \mathbb{R}^{m\times n}$, one step of the algorithm consist in

(a) remove the terms below the diagonal (in the column j) using a Householder transformation P,

(b) update $A = PA$,

(c) remove the terms to the right of the superdiagonal (of the row j),

(d) update $A = AP$.

This is performed for $j = 1, \ldots, n$ steps.

Ex.5 Write a program that giving a matrix A, computes the matrix H and reduce it to bidiagonal B. Write the output to a file (just the dimension $m+n$ and the two arrays containing the bidiagonal of H).

This bidiagonal matrix B will be the input of the LR-algorithm.

Ex.6 Write a program that implements the *qds* algorithm 5.10 in Applied Numerical Linear Algebra, Demmel, p244. The algorithm converges to a diagonal matrix with the square of the singular values of B in the diagonal. Let a and b are the diagonal and superdiagonal, respectively, of $B \in \mathbb{R}^{n\times n}$. Then consider $q = a^2$ and $e = b^2$. One step of the algorithm is given by

$$
\begin{aligned}
&\textit{for } j = 1 \textit{ to } n-1 \\
&\qquad \hat{q}_j = q_j + e_j - \hat{e}_{j-1} \\
&\qquad \hat{e}_j = e_j(q_{j+1}/\hat{q}_j) \\
&\textit{end for} \\
&\quad \hat{q}_n = q_n - \hat{e}_{n-1}
\end{aligned}
$$

where we assume that $b_0 = \hat{b}_0 = b_n = \hat{b}_n = 0$. Stop the iteration whenever $\|e\|_\infty < 10^{-14}$.