Project: Fitting tabulated data using singular values decomposition (SVD)

Tomasz Chwiej

8th April 2025

1 Introduction

We have a set of m two-dimensional data representing unknown function $y(x)$

$$\text{input data: } \{(x_0, y_0), (x_1, y_1), \ldots, (x_{m-1}, y_{m-1})\} \tag{1}$$

which we wish to fit with polynomial of $n-1$ degree

$$f(x) = \sum_{k=0}^{n-1} c_k x^k \tag{2}$$

with yet unknown linear coefficients c_k. By substituting sequentially x_i to the polynomial formula one expect y_i on the right side of equation, consequently we get the set of linear equations (SLE)

$$A\vec{c} = \vec{y}, \qquad A \in R^{m\times n}, \quad \vec{c} \in R^n, \quad \vec{y} \in R^m \tag{3}$$

$$\begin{bmatrix} 1 & x_0^1 & x_0^2 & \ldots & x_0^{n-1} \\ 1 & x_1^1 & x_1^2 & \ldots & x_1^{n-1} \\ 1 & x_2^1 & x_2^2 & \ldots & x_2^{n-1} \\ 1 & x_3^1 & x_3^2 & \ldots & x_3^{n-1} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & x_{m-1}^1 & x_{m-1}^2 & \ldots & x_{m-1}^{n-1} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{m-1} \end{bmatrix} \tag{4}$$

which under condition $m > n$ is overdetermined. Our task is to solve this SLE, find the coefficients and draw the polynomial to compare it with the input data.

2 Method

In order to solve the SEL we use SVD decomposition in its thin version as follows

$$A\vec{c} = \vec{y} \tag{5}$$

$$A = U\Sigma V^T, \qquad U \in R^{m\times n}, \quad \Sigma \in R^{n\times n}, \quad V \in R^{n\times n} \tag{6}$$

$$U\Sigma V^T \vec{c} = \vec{y} \tag{7}$$

$$\vec{c} = V\Sigma^{-1}U^T\vec{y} \tag{8}$$

the last equation we express as a sum of rank(1) matrices

$$\vec{c} = \sum_{k=0}^{n-1} \frac{1}{\sigma_k} \vec{v}_k \underbrace{\left(\vec{u}_k^T \vec{y}\right)}_{\substack{scalar \\ product}} \tag{9}$$

where σ_k, $\vec{u_k}$ and $\vec{v}_k$ are k-th singular value and its left and right vector, respectively. Briefly, first we must find SVD decomposition of A, then project $\vec{y}$ on the left singular vectors and finally add up all $\vec{v}_k$ scaled by the inverse of its singular value and scalar product.

3 Practical part

Perform the following tasks in order of appearance, practical hints corresponding to use of essential Lapack routine is given in next section.

1. Assume $m = 8$, $n = 6$, $x_a = 0$, $x_b = 3\pi$ and generate input data
$$\vec{x} = [x_0, x_1, \ldots, x_m] \tag{10}$$
$$\vec{y} = [y_0, y_1, \ldots, y_m] \tag{11}$$
for function
$$y(x) = x \cdot \cos(x), \qquad x \in [x_a, x_b] \tag{12}$$
Nodes x_i are distributed evenly $x_i = x_a + i \cdot (x_b - x_a)/(m-1), \quad i = 0, 1, \ldots, m-1$. Save data in arrays.

2. Fill the matrix elements accordingly to its form presented in Eq.4
$$A = [a_{i,j}], \qquad a_{i,j} = x_i^j, \quad i = 0, 1, \ldots, m-1, \quad j = 0, 1, \ldots, n-1 \tag{13}$$
Write it to file.

3. Calculate SVD with Lapacke routine *LAPACKE_dgesvd()*. Write singular values to file.

4. Reconstruct the matrix A exploiting its form as a sum of rank(1) matrices
$$A = \sum_{k=0}^{n-1} \sigma_k \underbrace{\vec{u}_k \vec{v}_k^T}_{\substack{outer\\ product}} \qquad \Longrightarrow \qquad a_{i,j} = \sum_{k=0}^{n-1} \sigma_k \left(\vec{u}_k\right)_i \left(\vec{v}_k\right)_j \tag{14}$$
to confirm that everything is ok until now.

5. Calculate the coefficients of polynomial c_k according to Eq.9. Write the coefficients to file.

6. At home repeat tasks (1-6) for other choices of m and N: $(m, n) = (30, 10)$, $(m, n) = (30, 20)$. Prepare separate plot for each of these three choices of (m,n) and show in it the input data and the polynomial. Prepare a report containing these plots, comment on discrepancy between input data and fitting function.

4 Computational hints

We find the SVD with *LAPACKE_dgesvd()* routine

```
lapack_int LAPACKE_dgesvd ( int matrix_layout , char jobu , char jobvt ,
lapack_int m , lapack_int n , double* a , lapack_int lda , double* s ,
double* u , lapack_int ldu , double* vt , lapack_int ldvt , double* superb );
```

assuming following parameters:

- *matrix_layout* = *LAPACK_ROW_MAJOR*, then matrices: A,U,V are 1D arrays filled in row-major manner

- $jobu = jobv =' S'$, then matrices U and V are filled with singular vectors only (thin version of SVD)
- m and n have values defined in Sec.3
- a is 1D array of size $m \cdot n$, $lda = n$ (for $LAPACK\_ROW\_MAJOR$), filling of elements A:

$$a_{i,j} \to a[i \cdot n + j], \quad i = 0, \ldots, m-1, \quad j = 0, \ldots, n-1 \tag{15}$$

- s is 1D array of size n and keeps the singular values in descending order
- u is 1D array of size $m \cdot n$ and keeps $\vec{u}_k$ vectors in columns (**but the storage is still ROW-MAJOR-ORDER**), $ldu = n$ (for $LAPACK\_ROW\_MAJOR$), reading the i-th element of $\vec{u}_k$:

$$(\vec{u}_k)_i \to u[i \cdot n + k], \quad i = 0, \ldots, m-1, \quad k = 0, \ldots, n-1 \tag{16}$$

- vt is 1D array of size $n \cdot n$ and keeps $\vec{v}_k$ vectors in rows (transposed matrix V), $ldvt = n$ (for $LAPACK\_ROW\_MAJOR$), reading the i-th element of $\vec{v}_k$:

$$(\vec{v}_k)_i \to vt[i + k \cdot n], \quad i = 0, \ldots, n-1, \quad k = 0, \ldots, n-1 \tag{17}$$

- $superb$ is 1D array of size n

5 Example results

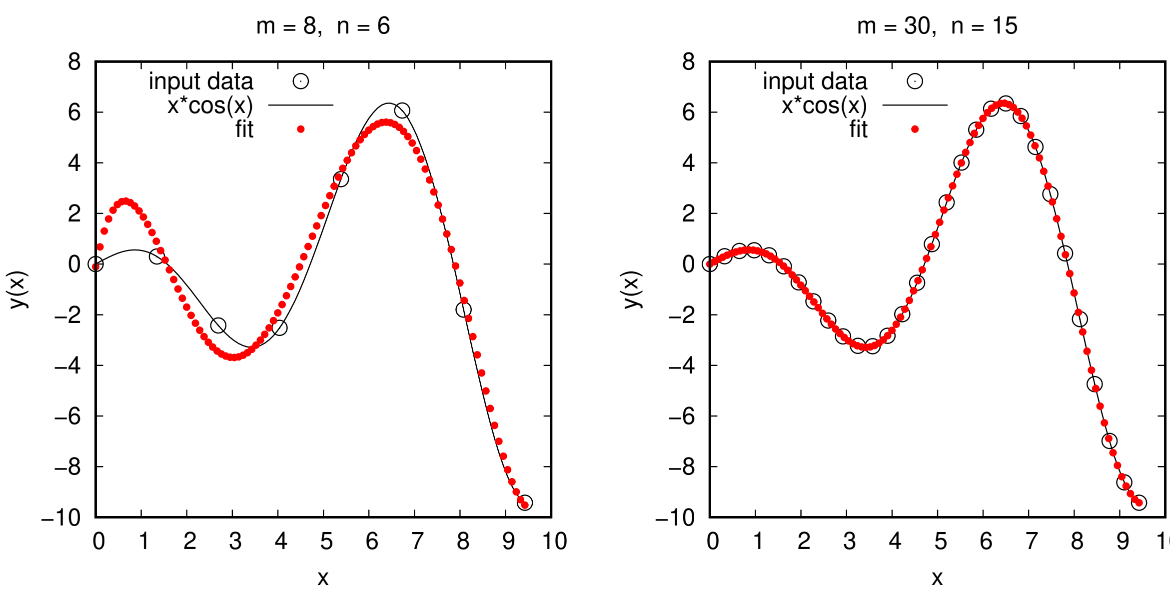

Figure 1: Example results for $(m, n) = (8, 6)$ (left) and $(m, n) = (30, 15)$ (right).