Medical Image Processing for Interventional Applications

SVD in Optimization - Part 2

Online Course – Unit 5

Andreas Maier, Joachim Hornegger, Frank Schebesch
Pattern Recognition Lab (CS 5)

virtuelle hochschule bayern

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

Optimization Problem III

Another quite important optimization problem in image processing and pattern recognition is the following:

Problem: Given a matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$.
Compute the matrix $\widehat{\boldsymbol{B}} \in \mathbb{R}^{n \times n}$ of rank $k < n$ that minimizes:

$$\widehat{\boldsymbol{B}} = \arg\min_{\boldsymbol{B}} \|\boldsymbol{A} - \boldsymbol{B}\|_2 , \quad \text{subject to} \quad \text{rank}(\boldsymbol{B}) = k.$$

Solution: Using SVD, the solution can be computed easily by:

$$\widehat{\boldsymbol{B}} = \sum_{i=1}^{k} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^{\mathsf{T}}.$$

Optimization Problem III

Example

The SVD can be used to compute the image matrix of rank 1 that approximates an image best w. r. t. $\|.\|_2$. Figure 1 shows an image $\boldsymbol{I}$ and its rank 1 - approximation $\boldsymbol{I}' = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^{\mathsf{T}}$.

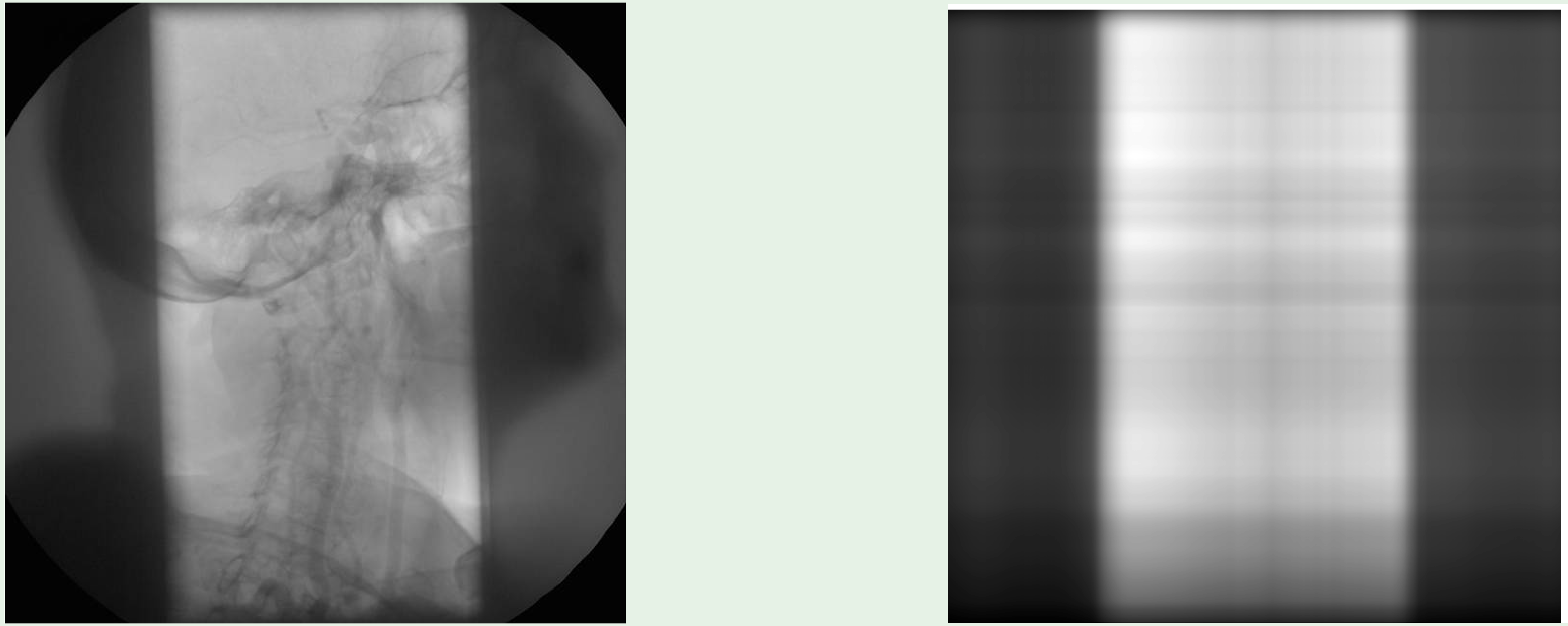

Figure 1: Original X-ray image (left) and its rank 1 - approximation (right)

Optimization Problem IV

Problem: The ***Moore–Penrose pseudoinverse*** is required to find the solution to the following optimization problem:

$$\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2 \rightarrow \min .$$

Solution: The least squares solution of this optimization problem is given by

$$\boldsymbol{x} = \boldsymbol{A}^{\dagger}\boldsymbol{b},$$

where we get $\boldsymbol{A}^{\dagger} \in \mathbb{R}^{n\times m}$ based on the SVD of $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ by:

$$\boldsymbol{A}^{\dagger} = (\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A})^{-1}\boldsymbol{A}^{\mathsf{T}} = \boldsymbol{V}\boldsymbol{\Sigma}^{\dagger}\boldsymbol{U}^{\mathsf{T}}.$$

Optimization Problem IV

Derivation: We start with the optimization problem:

$$\frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2^2 \rightarrow \min,$$

which can be solved analytically by derivation of this functional:

$$\begin{aligned} & \boldsymbol{A}^{\mathrm{T}}\left(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\right)=0 \\ \Leftrightarrow \quad & \boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}-\boldsymbol{A}^{\mathrm{T}}\boldsymbol{b}=0 \\ \Leftrightarrow \quad & \boldsymbol{x}=\left(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{b}. \end{aligned}$$

Optimization Problem IV

The diagonal matrix $\boldsymbol{\Sigma}^\dagger$ in the SVD of the pseudo-inverse of $\boldsymbol{A}$ is given by:

$$\boldsymbol{\Sigma}^\dagger = \begin{pmatrix} \frac{1}{\sigma_1} & & & & & 0 & \dots & 0 \\ & \ddots & & & & & & \\ & & \frac{1}{\sigma_r} & & & \vdots & & \vdots \\ & & & 0 & & & & \\ & & & & \ddots & & & \\ & & & & & 0 & \dots & 0 \end{pmatrix} \in \mathbb{R}^{n \times m},$$

where $\sigma_r > 0$ is the smallest nonzero singular value of $\boldsymbol{A}$.

Optimization Problem IV

Example

Compute the regression line through the following 2-D points:

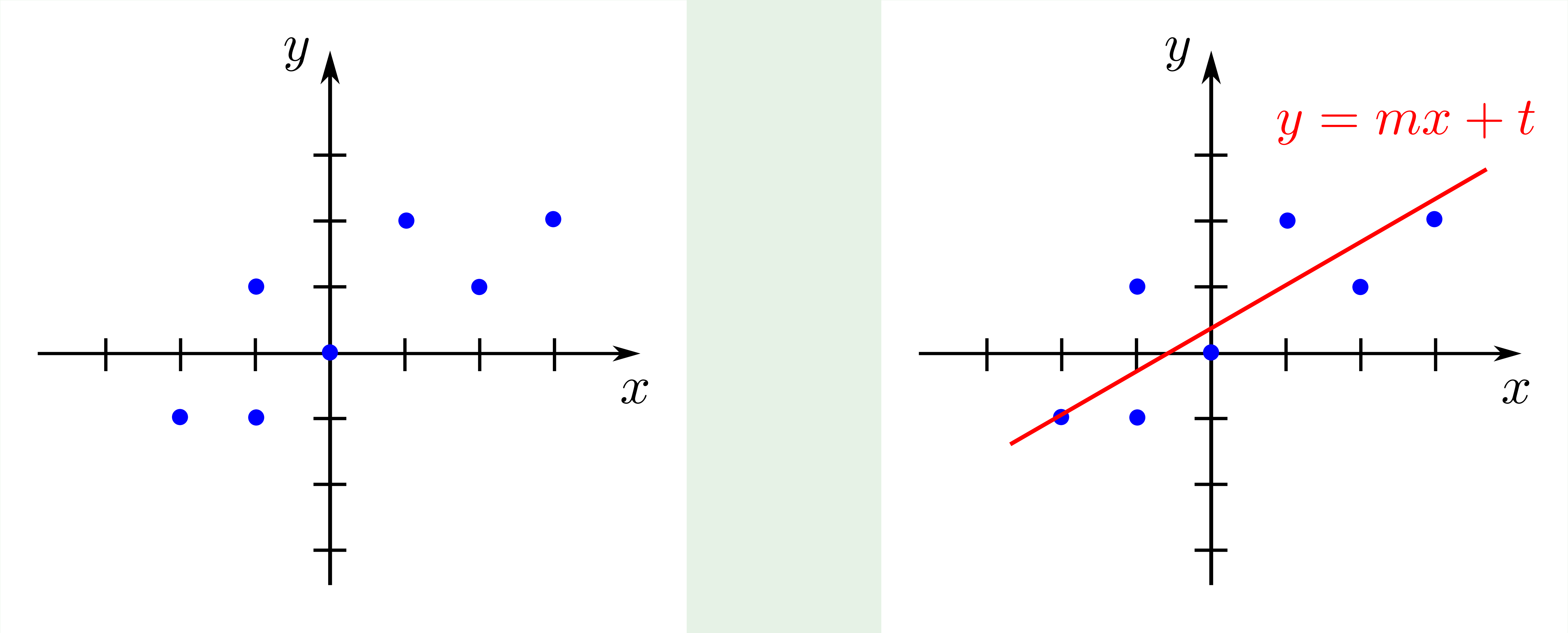

Figure 2: Regression line through a set of 2-D points

Optimization Problem IV

All points (x_i, y_i), $i = 1, \ldots, 7$, have to fulfill the line equation:

$$y_i = mx_i + t, \quad \text{for} \quad i = 1, \ldots, 7.$$

Thus we get the following system of linear equations:

$$\begin{pmatrix} 3 & 1 \\ 2 & 1 \\ 1 & 1 \\ 0 & 1 \\ -1 & 1 \\ -1 & 1 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} m \\ t \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \\ 2 \\ 0 \\ 1 \\ -1 \\ -1 \end{pmatrix}.$$

Optimization Problem IV

The Moore-Penrose pseudo-inverse for this particular problem is:

$$\boldsymbol{A}^{\dagger} = \begin{pmatrix} 0.14 & 0.09 & 0.04 & -0.01 & -0.07 & -0.07 & -0.12 \\ 0.11 & 0.12 & 0.13 & 0.15 & 0.16 & 0.16 & 0.18 \end{pmatrix}.$$

Therefore, for the regression line we get the equation:

$$y = 0.56x + 0.41.$$

Remarks on SVD Computation

- SVD can be computed for every matrix.
- SVD is numerically robust.
- In most practical situations we have more rows than columns:

$$m \gg n.$$

- The time complexity to decompose $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ is:

$$4m^2n + 8mn^2 + 9n^3.$$

- For us, the SVD is a black box. We do not consider algorithms to compute the SVD numerically.

Take Home Messages

- We have studied two additional applications (see also previous unit):
  - low-rank approximation,
  - fitting of a regression line.
- SVD is *the* tool for linear equations – it cannot fail (but in many special cases there may exist better solutions).
- SVD is provided by all standard libraries.

Further Readings

Read the original:

Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. 3rd ed. Johns Hopkins Studies in the Mathematical Sciences. Baltimore: The Johns Hopkins University Press, Oct. 1996

A very detailed and easy to follow introduction of the SVD can be found in:

Carlo Tomasi's class notes, chapter 3 (a **must-read**).

The theory is described in an easy to read format in:

Lloyd N. Trefethen and David Bau III. *Numerical Linear Algebra*. Philadelphia: SIAM, 1997

For details about the numerical computation of SVD see:

William H. Press et al. *Numerical Recipes – The Art of Scientific Computing*. 3rd ed. Cambridge University Press, 2007. Get at `http://numerical.recipes/` (August 2016).

A good reference for properties of matrices is the following script:

Kaare Brandt Petersen and Michael Syskind Pedersen. *The Matrix Cookbook*. Online. Technical University of Denmark, Nov. 2012. URL: `http://www2.imm.dtu.dk/pubdb/p.php?3274`