Foundations of data science, summer 2020

JONATHAN LENNARTZ, MICHAEL NÜSKEN, ANNIKA TARNOWSKI

7. Exercise sheet
Hand in solutions until Thursday, 4 June 2020, 12:00

Exercise 7.1 (Toy Example of SVD). (15 points)

You can use the numpy command `numpy.linalg.svd()` for this exercise.

(i) Take $A = \begin{bmatrix} 1 & 2 \\ 3 & 1 \\ 2 & 3 \end{bmatrix}$ (from the course). $\boxed{6}$

(a) Use python to compute its SVD and its 1-truncated SVD.

(b) Plot the rows of the data matrix A and the rows of its 1-truncated SVD A_1 as points.

(c) Compare to the first Lemma about the k-truncated SVD.

Lemma. *The rows of A_k are the projections of the rows of A to the best-fit k-subspace $\mathcal{V}_k$ spanned by the first k singular vectors of A.*

Hint: You may want to repeat the previous with a few other matrices A.

(ii) Write python routines: $\boxed{3}$

(a) For a given set of points and a line, the sum of the squared distances from the points to the line.

(b) For a given set of points, the best-fit 1- and 2-subspace to approximate the points.

Solution. You can best check your own solution by comparing the following exercise with your own.

Please note that for the best-fit *subspace* you must not center the data. This is because a subspace must always contain the origin.

To compute the sum of the squared distances from the points to a given line, you probably want to compute the distance vector between the point and the projection of the point. Therefore you will need that (if input your line as a vector) the projection of a vector z to the line given by v takes the form of

$$\frac{|v\rangle \langle v \,|\, z\rangle}{\langle v \,|\, v\rangle}.$$

If you input your line as an equation of the form $y = mx$, you implicitly use the vector $|(1, m)\rangle$ to describe your line. Then your formula for the x-coordinate will look like

$$\frac{z_1 + mz_2}{1 + m^2}$$

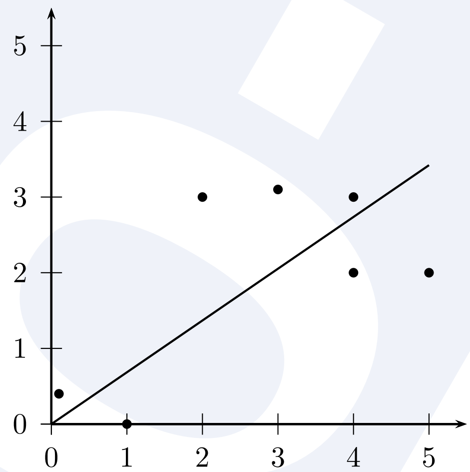
and you obtain the y-coordinate by multiplication with m. ○

$\boxed{4}$ (iii) Apply the two routines to the data sets given in the file `07-sol-2d.csv` you find in sciebo.

(a) Plot the points with the corresponding best-fit line so we can visually check the correctness of your routine.

(b) Compare the sum of the squared distances of the best-fit line to the sum of the squared distances of the line given by $y = \frac{2}{3}x$.

Solution. The picture you obtain should look like the following:



- The line should point in the direction of approximately

$$|(0.83\text{Ⴤ}, 0.56\text{ሐ})\rangle$$

or in the other version of implementation, have steepness of $m = 0.68\text{ሐ}$.

- The value for the sum of the squared distances to the best-fit line is 4.75Ⴤ.

- The value for the sum of the squared distances to the line given by $y = \frac{2}{3}x$ is 4.76⊥.

Here are some other values that might help for debugging:

- If you centered your data (which you should not) you obtain 4.66Ч.
- If you computed the sum of the squared projection lengths you obtain: 102.03⅃ (best-fit) and 102.02⊤ ($y = \frac{2}{3}x$).

(iv) Plot the data set and best-fit 2-subspace given in `07-sol-3d.csv`. 2

(v) Repeat some of the previous with a random data sets containing 500 points...