Fachbereich Mathematik und Informatik
Universität Marburg

Sommersemester 2024

**Exercises to „Matrix Methods in Data Analysis“**
– Sheet 8 –
Submission till Sunday, June 16th, 23:50, via Ilias

If you need a German translation of any of the following, please ask Dr. Lochmann.

**Exercise 1** (9 + 2 = 11 points)

a) In Ilias, you will find a set of 100 3-dimensional data points. Apply the $k$-means algorithm with $k = 3$ to determine centroids of 3 clusters. As initial centroids, use:

$$c_1 = \begin{pmatrix} 320 \\ 320 \\ 320 \end{pmatrix}, \quad c_2 = \begin{pmatrix} 340 \\ 340 \\ 340 \end{pmatrix}, \quad c_3 = \begin{pmatrix} 360 \\ 360 \\ 360 \end{pmatrix}$$

Iterate till nothing changes anymore. Enter the resulting centroids in Ilias, rounded to the nearest integer.

b) What is the quality of the clustering you achieved in exercise 1?

**Exercise 2** (7,5 + 1,5 = 9 points)

a) Let

$$A = \begin{pmatrix} 2 & 0 & 1 \\ 3 & 1 & 3 \\ 0 & 4 & 1 \\ 0 & 0 & 11 \\ 7 & 1 & 1 \end{pmatrix} \quad \text{and} \quad W^{(1)} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}.$$

Apply the alternating nonnegative least squares algorithm (page 106), combined with the “cheaper alternative” using QR decomposition (page 107), to get a rank-2-nonnegative factorization of $A$. Iterate 50 times, then enter $W^{(50)} \cdot R^{(50)}$ into Ilias. (It should be near $A$, but not exactly.) Round to 2 digits after the comma.

b) Answer the questions in Ilias about (a).

**Note 1:** Don’t forget to normalize $W^{(j)}$ after each step. Remember that “normalization” in this context means something different than before. Use the normalization Elden proposes on page 106 (i.e. columnwise normalization using the max-norm). Remember to shift the factors from $W$ to $R$: When you divide column number $a$ of $W$ by a factor $x$, do you have to divide or multiply, row or column $a$ of $R$ by $x$? Test this with some examples first! Remember that $W^{(j)}H^{(j)} \approx A$.

**Note 2:** There is a possible misunderstanding in Elden’s alternating nonnegative least squares algorithm: The $k$ he references in step 2 is not the rank $k$ he chose before and uses for the shapes of $W$ and $H$! Best is to replace every $k$ in his algorithm’s step 2 by a $j$, and you’ll be on the safe side.

*Bitte wenden.*

**Project IRIS** (Project for 4 people)

In this project we want to classify flowers ... no, wait, we already covered botany three weeks ago. Not very many of you were interested in that project. Maybe try something cooler? Like ... detection of nuclear weapons testing using infrasound?

IRIS stands for "Incorporated Research Institutions for Seismology" and was a consortium of several research institutes, until it merged with another one to form UNAVCO. Both collect(ed) seismographic and athmospheric data, which could then be used to guide vulcanology and geophysics, but also to identify and locate nuclear explosions when they happen.

One of their instruments is a grid of infrasound detectors, and the data IRIS collected this way from 2011 to 2015 is available here:

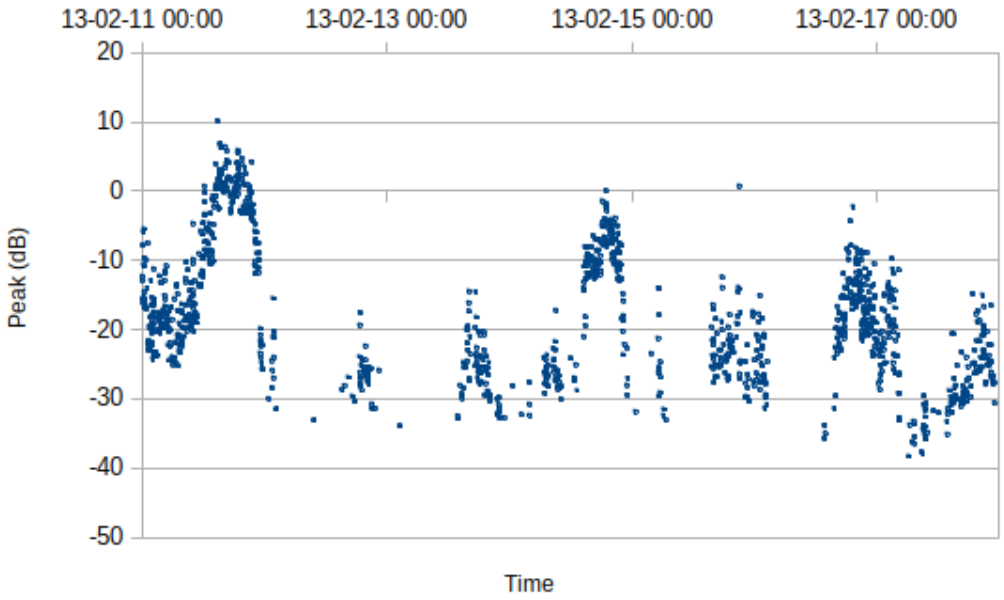
`https://ds.iris.edu/ds/products/infrasound/`

A nice explanatory video about how these detectors work can be found here:

`https://www.youtube.com/watch?v=GVWOA5pZG6o&list=PL10C6C45B27E1638A`

In Ilias, you will find the data for Feb. 11th 2013 to Feb. 17th 2013. Understand it, present how it works and load it. Each row represents one event in one location for some short interval of time. Convert the starting, ending, and peak times into something usable (like "seconds since midnight Feb. 11th").

The different stations have different sensitivities, and we will have to account for this later. For now, sort all events by station. Choose a station, then create a scatter plot for its events for peak vs. peak time. You should see a graph similar to this:




Check this with a few stations, just to convince yourself and your audience. Present two or three of these graphs.

Obviously, those events cluster (for the station above it would be something like $k = 6$ to $k = 9$ clusters, depending on how one counts), and if we want to understand the overall situation for *all* stations, we need to reduce information – so the idea is: Calculate a few clusters per station instead of hundreds of events! Then visualize the centroids for all stations, instead of all events for all stations.

For each of the 415 stations, use a $k$-means algorithm to calculate $k$ clusters. For each cluster, calculate meaningful data, like duration, maximum value, mean value, or some

*Bitte wenden.*

more sensible measure of "height". Normalize this "height" to account for the fact that each station has different sensitivity.

You will have to choose $k$ and the initial centroids without seeing each of the 415 stations ... we are looking for an automated solution here, after all! So think hard about how to make your algorithms robust and fool-proof. Present your code and explain your choices. If you did some experiments which did not work, feel free to present them, too! (You need to fill 40 minutes, after all.)

You also need to choose a scaling for your axis. The $x$- and $y$-axis have totally different units, and as I have hinted on during the lecture on Thursday, the result of $k$-means strongly depends on the exact scaling. See this as a way to tinker: Multiply the $y$-axis with any number you want, to get different results. Experiment and choose a setting that works with several stations. Present your experiments and explain your choice. I suggest not to scale the $x$-axis, because it is needed intact later in the process.

*Optional:* Signal strength is shown in dezibel. This is a logarithmized value! Maybe the clusters can be calculated more easily when you apply exp first, then scale?

We now want to combine the results of the various stations into a single animation. In Ilias, you will find a second file, holding the geographical coordinates of each of the stations. They are all located in continental USA and Canada. Super-imposed on a map of North America, show each station at its coordinates, and create an animation, where time represents time (just faster, say 5 minutes for all 7 days of data). For each cluster show when it appears, where it appears, and with which strength, by just adding a bright dot at that position and time; brightness depending on strength of the cluster.

You might be able to see some kind of "waves" running over the continent, similar to this video: `https://www.youtube.com/watch?v=vxJTQuWISbk`, but with infrasound, rather than lightnings. (You might just be seeing noise however. Would be fine as well.)

*Optional:* Instead of all that cluster stuff, try to smooth the data instead (maybe some moving average?), and create a second animation with that. Is this approach superior?

**Presentation:** Presentation by 4 people on Wednesday, June 26th, Thursday, June 27th, or Monday, July 1st, with two teams on each day; 6 teams in total. Remember that everyone of your team should present for at least 10 minutes, so plan accordingly.

**Application:** Application is due to June 16th, via Ilias. Write who your team consists of, and on which of the days you are able to present. You may add a preference, if you wish.