GROUP ASSIGNMENT 2DRR00, FALL 2025

Register your group members on Canvas before the specified deadline.

State all names and student numbers of the group members in the report

READ THIS INFO WELL:
Write a report giving results of these assignments, to be carried out in groups.

- See Canvas for the **allowed numbers of students per group.**
- Submit the report via Canvas **before the deadline** stated on Canvas!. The format has to be PDF. You can submit ∞ many times a (preliminary) version of the report, until the deadline.
- The report should be well readable. Answer the questions completely, but rather concisely. Do not overdo the number of pages: **try to have** $\leq$ 12 **pages**, but it is OK if you need a few more.
- You may use Latex to have beautiful math, but Word or any other (online) editor is also fine. Scanned handwriting is also allowed, but please take care that the file size remains modest ($<$ 5 MB if possible). You may use the Latex template on Canvas if you want, but it is not mandatory.

You may plot matrices in **small script** $\left[\begin{smallmatrix}1 & 2\\ 3 & 4\end{smallmatrix}\right]$ (e.g., using latex's `scalebox`) and vectors in the form $[1,2,3]^T$ to save space. (This is commonly done in books as well.) As matrices tend to be space-consuming, take care of efficient page use (margins not too wide, etc).

Occasional contact with other groups is allowed, but each group has to hand in original work. These assignments are meant to have **lots of (math) fun together** with your group, learn useful linear algebra, and perform practical experiments. Of course, respect scientific integrity, as this is very important in academia.

Only use linear algebra methods seen in (or close to those in) class; do not use advanced codes or methods found on the web.

Do **NOT hand in computer codes**.

Do **not** include relatively easy computations with numbers that are not so informative for the reader and the story, and mainly take a lot of space. Instead, describe and explain well what you do, give the results, and analyze and discuss them.

Don't give too many significant digits! Usually 0.12 instead of 0.123456789 is perfectly fine. (NB: this is a frequent "flaw" in many research papers)

For good academic writing, it is recommended to read the `latex-tips` document, although this is mainly meant for research papers.

WORKING TOGETHER

Almost every year, there is at least 1 group that contacts the lecturer ca 1–2 days before the deadline: "some members have promised to finish their share, but they left and didn't do it." Do not end up in this situation! In the first place, this is meant as **group work**, to do together, **learn together**, and have lots of fun together. Every member should look into every question, also since this is very useful to get acquire skills for the final exam. Didactic studies suggest that learning as a group has benefits. So truly work together, discuss together, brainstorm together, have meetings, and have a sensible draft of each question done at least 1–2 weeks before the deadline. Do not be satisfied with promises of group members who have not handed in a draft (say) 2 weeks before the deadline. Email the lecturer in this case, as indicated above; we can split the group for you.
We will generally not split a group in the last week before the deadline.

So, two extremes:

- Examples on what is NOT the intention:
  split 6 questions among 6 team members. Hardly meet or interact. Struggle on your own, or try to cheat with AI. Discover in the last week that member X didn't do anything. Panic. Email lecturer.
- **Examples on what IS the intention:**
  meet often (also possible during instructions) and have lots of fun. Think of nice practical examples, problem sets, and solutions. Enjoy and appreciate the nice math. Brainstorm together, and enjoy the learning process. [Acquiring skills like these exactly makes you so valuable for future employers.] Realize that math may be (even) more fun than you thought. Explain something you understand to another member, and vice versa. Make long-term friendships. Do a pizza bet with another group who will have the highest grade. Go for the 2 bonus points.

ON THE USE OF AI

- This group work has 2 main goals:
  (1) to have **fun** together with your group mates and
  (2) to **learn** a lot of useful math together, by doing effort and struggling in a good way with the class material. The questions are often on exam level, so that they prepare you for taking the exam.
  The use of AI is not in line with both of these goals.
  We therefore advice and request you not to use AI for the group work.
  It is really for your own good to do it together and learn from this.
- AI sometimes gives an incorrect answer to many math questions. It will regularly state an answer with much certainty, while it is in fact wrong. The lecturer had some recent linear algebra experiences with this.

  If graders know that AI tends to give a certain type of incorrect answer, with a certain flaw in it, and your group has this answer, the graders will generally deduct all points for the question.

Complete the following assignments:

(1) on recommender systems (guess your fellow student's taste!)
(2) on Google PageRank / Markov chains (ranking)
(3) on eigenvector-based clustering
(4) on data mining
(5) **EITHER (5a)** on graph centrality
OR (5b) on programming language competition
(6) **EITHER (6a)** on image compression using the TSVD
OR (6b) on population dynamics.

Most importantly, have lots of fun together, and enjoy the practical applications!

(1) Vector angles and movie ratings (recommender systems); cf. Classes 1, 14

NB 1: For this and other assignments, use linear algebra notions such as norms, angles, and low-rank matrices as seen in class, and develop your own methods and code. It is **not the intention that you use techniques found elsewhere**, such as advanced non-LA methods.
NB 2: This question is posed in terms of your taste of movies. However, as a group you may decide on another type of items; for instance, pets, food, drinks, board games, music, or anything else your group likes. Originality is appreciated!

(a) In your group, agree on ca 8 concrete movies (or TV series, movie types, etc) and let all group members rate them $-2, \dots, 2$. **HOWEVER, 1 group member only rates half of the items, and keeps his/her remaining ratings as a secret. The aim of the assignment is to try to predict those!**
For instance, you can think of: Nature movie, Comedy series, Romantic comedy, Science Fiction, latest James Bond (or action), Detective, Candid camera (Just for laughs / Bananensplit / Verstehen Sie Spass), Documentary of historic events, Travel program, Cartoon, Sport match PSV–Feyenoord (or similar, such as a CL match ☺), ...

(b) What could be a reason to scale $-2, \dots, 2$ instead of the usual $1, \dots, 5$?
Hint: suppose there are only 2 movies to rate, what are the possible angles in each case?

(c) Who has the "average" taste, and who has the most extreme taste?

(d) **Now the main part of the assignment: try to predict the missing ratings of your group's member, based on the other ratings of your group!**
There may be various options. For this part, think e.g. of norms (Class 2) or angles (Class 1). How well did you predict the true scores?

(e) **Now do the same as (d)**, but use the more advanced concept of low-rank matrices (Class 14), via a TSVD (for instance, with $k = 1$ or $k = 2$).
(One option might be to replace the missing entries by zeros at first.)
Does this give better results than in **(d)** ?
(NB: if you really would have great difficulty doing part **(e)**, you are still eligible for a max of 3 points of 4 for the other parts.)

Background info: see `en.wikipedia.org/wiki/Netflix_Prize` for a famous interesting competition on predicting user ratings. In the literature this problem is also known as matrix completion. Although the problem is simple to state, solutions approaches may be difficult, and it may be very important for many companies.

(2) Google PageRank type ranking; cf. Class 3 (and 8)

Create a Google PageRank type of ranking of items you as group members may select yourself. There have to be at least ≈ 10 items to sort (more is allowed; for instance, you can take 18 Dutch soccer clubs). To create a **directed graph**, there has to be a 'directed' relation, such as "X has beaten Y in a game", "X would like to date Y" (and possibly but not necessarily vice versa), "X thinks he/she is better at math than Y", etc. For your inspiration, examples are:

- The Dutch eredivisie voetbalclubs (Ajax, Feyenoord, PSV, . . .): create a link from X to Y if Y has beaten X. You can pick any year of your liking ☺! You may for instance take this year, and see if Ajax is really that bad . . . There are many options: for instance, you can add a link with weight 1 if Y has beaten X, but also with weight 3 (as a victory gives 3 points in soccer). Or you can take the number of goals into account as well. In conclusion: the matrix elements are not restricted to 0s or 1s only. A nice aspect of the Google PageRank type rating is that a win against a strong opponent weighs relatively more, so that the final ranking may be different. For instance, if PSV ends up second in the competition but did well against strong opponents, they may still appear first in the Google PageRank type order.
- Results from any other sports competition you are interested in.
- Internet pages of a small number of sites you know. They should have enough mutual links to be interesting. (Obtaining such as example might not be easy in practice.)
- Friendships on social media are generally not suitable since they are undirected (friendships or connections are both ways). However, you can take some people and create a link from X to Y if Y thinks he/she is better at math than X (some bluffing/boasting is allowed!).
- You can also create a random internet of (say) 10 pages, with random links (for instance, 40% probability that there is a link) from site j to site i. (In the case you study an internet, do not include self-links.)

If your group agrees on the test case, perform these steps:

(a) Create a matrix G representing the graph, with $G_{ij} = 1$ (or perhaps some appropriate nonzero weight) if there is a link from j to i, and $G_{ij} = 0$ if there is no link from j to i. (If your problem models an internet, G_{ii} should always be 0.) Then create $\widehat{G}$ from G by dividing each column by the 1-norm of that column, so that all columns have 1-norm equal to 1. If a column contains only zeros, take all entries equal (to $\frac{1}{n}$). Finally, create $A = p \cdot \widehat{G} + (1-p) \cdot \frac{1}{n} \cdot \text{ones}(n)$, where ones is the matrix of appropriate size with all ones. First take the original Google value $p = 0.85$. Double-check that the column-sum of all columns equals 1 (so A should be a column-stochastic matrix).

(b) Compute (using the computer) the eigenvector $\mathbf{x}$ corresponding to eigenvalue 1, that is, $A\mathbf{x} = \mathbf{x}$. Normalize $\mathbf{x}$ to have $\|\mathbf{x}\|_1 = 1$, and all entries ≥ 0. Round the entries to 2 digits after the comma (e.g., 0.12).

(c) List the ordered items. Can you explain the ranking? Is it logical?

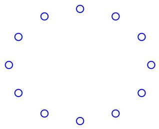
(d) Take another p value, for instance $p = 0.99$ or $p = 0.50$. What is the influence on the PageRank?

(e) Now assume you are the owner of the first node (i.e., the first site, or club, or . . .). You may change **1 element** in the graph to maximize your PageRank: adding a link or, removing a link to your liking. What can you try to do this? Try some ideas and see how they turn out.

(3) Eigenvector-based clustering; cf. Class 9 (and 8)

Create at least ca 10 elements and a corresponding **undirected graph** to cluster. You can think of the following:

- Friendships on social media. If there is missing data, try to guess it as well as possible.
- Some other real-life data.
- A challenging data set such as one presented in the slides of Class 9. You can choose the number of points yourself.
- Nodes in a circle, with random edges.



Important! The graph must be **connected**, that is, not consist of 2 or more disconnected parts, without edges between them. (In case the graph is not connected, the Laplacian matrix will have more than one eigenvalue equal to zero, so that the Fiedler vector is not well defined.)
On the other hand, if you have a graph with a lot of edges between nodes with (almost) equal weights, then this may be a very hard problem to cluster.

If your group agrees on the test case, perform these steps:

(a) Briefly explain your items and graph.

(b) Plot the graph, by computer or by hand (in case this is easily possible; in the Class 9 slides, only edges between different groups are plotted for clarity).

(c) Set up the symmetric Laplacian matrix L ($L_{ij} = -1$ when there is an edge connecting node i with node j, and $L_{ii} =$ the number of edges of vertex i). You do not have to give the matrix in the report.

(d) Compute (using the computer) the Fiedler vector, the eigenvector corresponding to the second smallest eigenvalue. The smallest eigenvalue λ_1 should be 0 (or, on your computer, due to rounding errors, very close to zero, such as 10^{-15} or -10^{-16}), but the next-to-smallest eigenvalue λ_2 should not be very small. If λ_2 equals (say) 10^{-14}, it means that the graph is probably disconnected. On the other hand, if λ_2 is a large number (such as 25, depending on the size of L), you can just continue, but this means that the Fiedler clustering thinks this is a tricky problem to cluster.

(e) Color the nodes corresponding to the sign of the Fiedler vector, and display the result. Is it as expected? Can you explain the result?

(f) For this last part, take another graph, with more, or fewer edges:
If your first graph had only few edges, choose a second with more edges.
If your first graph had many edges, choose a second with fewer edges.
Give the Fiedler value (second smallest eigenvalue) in both cases, and compare the result of the clustering in both cases.
Which Fiedler value is smaller: that for the graph with many or few edges?
Which graph seems easier to partition: the graph with many or few edges?
Do you see a relation?

(4) Data mining: term–document matrix; cf. Class 11 (and 10, 13)

Collect data in one of the following ways:

- Create a term–document matrix A of ca 10 terms (rows) and ca 10 documents (columns) of your own choosing. Assign $a_{ij} = 1$ if document j contains term i, and 0 otherwise. You may also modify the value of 1, if you think that the term is important in the document. Choose your terms and documents wisely so that the matrix contains not too few, and not too many nonzeros (ca 20% to 50% nonzeros). Also try to choose some related and some unrelated terms. Here, the goal is to discover potential relations between terms.
- Or choose your own data in a similar way, for instance a keyword–tweet matrix.
- But you can also take a supermarket case such as shown in the SVD class.

(a) Briefly explain your terms and documents, or customers and products.

(b) Give the matrix (in very small font).

(c) Compute (using your computer) the SVD $A = U\Sigma V^T$, ordered in decreasing singular values as usual. Take $U_{1,2}$ = the first two columns of U, and $V_{1,2}$ the first two columns of V. **Take care that the elements of $\mathbf{u}_1$ and $\mathbf{v}_1$ are all nonnegative.** (In case they are negative, then multiply both $\mathbf{u}_1$ and $\mathbf{v}_1$ by -1; then all elements should be nonnegative.)
Now project the terms onto $U_{1,2}$: for every term $j = 1, \dots, 10$, let $\mathbf{c}_j = U_{1,2}^T \mathbf{e}_j \in \mathbb{R}^2$, where $\mathbf{e}_j$ denotes the jth standard basis vector. (The norms of the $\mathbf{c}_j$ should be ≤ 1.)
Likewise, let $\mathbf{d}_j = V_{1,2}^T \mathbf{e}_j \in \mathbb{R}^2$, $j = 1, \dots, 10$, be the projected documents (or customers).

(d) The intention of the previous part is that we now can visualize the $\mathbf{c}_j$. Plot the $\mathbf{c}_j$ in $\mathbb{R}^2$ as points (more precisely, they should be in the rectangle $[0, 1] \times [-1, 1]$), with the terms as labels.
Likewise, in a **separate** figure, plot the $\mathbf{d}_j$, with the documents as labels.
See the slides of Class 11 for inspiration.

(e) Are the results as expected? Can you explain it? Are related terms (or products) more or less in the same direction? Are related documents (or customers) more or less in the same direction? Do you see nice unexpected connections?

DO EITHER 5a OR 5b

(5a) Graph centrality and (social) networks; cf. Class 9, 11

(a) Select an undirected graph to study with ca 10–20 nodes.
Briefly describe your chosen data and print your graph (small font).

(b) Let A be the associated adjacency matrix with eigenvalues λ_i, and let $\alpha > 0$. Explain: why are A's eigenvalues real? Why do we need $\|\alpha\, A\| < 1$ for Katz centrality? Why does this imply $\alpha < (\max|\lambda_i|)^{-1}$?

(c) Compute the largest eigenvalue of your matrix and use this to compute Katz centrality for 3 or more appropriate values of α. Do you see any differences? (You can also plot the ranking as function of α.)

(d) Compute the exponential centrality. Compare the resulting vector with those you obtained in **(c)**. What do you see, and can you interpret any differences?

(5b) "Nerd alarm"! ☺ Programming language competition; cf. Class 5

The internet is full of discussions which is the best programming language.
See, e.g., `www.tiobe.com/tiobe-index/`.
Suppose you are the CTO (chief technology/technical officer) of your company, and have to decide on the language of choice. Forget about your past experiences; you are going to do a factual and impartial test along these lines:

- Choose (at least) 3 languages, for instance Matlab, Julia, Python; but you may also consider C++, R, Java, Octave, ... (Julia, Python, and R may for instance be called using Jupyter; Rstudio is also convenient for R. You may install Matlab via TU/e, using a minimal number of toolboxes.)
- Choose (at least) 3 tasks that take at least ca 5–10 seconds. You may for instance think of:
  - solving a linear system $A\mathbf{x} = \mathbf{b}$ with a rather large (random) $n \times n$ matrix A
  - computing the eigenvalues/vectors of a rather large (random) $n \times n$ matrix A (if you choose a symmetric matrix, your eigenvalues and eigenvectors will be real, but this is not necessary)
  - solving a least-squares system $A\mathbf{x} \approx \mathbf{b}$ involving a rather large (random) $m \times n$ matrix A (where $m > n$)
  - perfoming a loop over ca 10^6 runs of some rather cheap operation (e.g. inner product)
  - computing the singular values/vectors of a rather large (random) $n \times n$ or $m \times n$ matrix A
  - ... or any other linear algebra task.
- Try to code as efficiently as possible, and time your programs.
  Time only the relevant action, and not tasks as printing output. Try to use timing commands such as `tic` and `toc`, depending on the language.

Present the results, and **give an expert recommendation, also useful at this moment for your fellow students and teachers**:

- Which do you consider to be the winner in running time?
- Which do you consider to be the winner in programming convenience?
- Any other points? (You may think about a "Consumers Union guide" type of table with several aspects.)
- What is your final recommendation? Are there any "DOs" or "DON'Ts" ?

(Now you can also boast about your programming experience during your future job interview ☺)

DO EITHER 6a OR 6b

(6a) Image compression by TSVD; cf. Classes 11 and 14

(a) Download the slides of Class 11 (about the SVD) and of Class 14 (about the TSVD and image compression).
Select a black-white image; for the SVD, your computer should probably be able to deal easily with (say) 2000×2000 pixels (which means $4 \cdot 10^6$ bytes if 1 byte per pixel were used, so this would be 4 MB). You can also take a color picture and convert it to black-white.
(NB: b/w pictures are standard in image analysis research papers.)

(b) Read it in into a computer program and display it properly. (For instance, Matlab has an Image Processing Toolbox with commands image, imshow, imagesc, colormap, and rgb2gray.)

(c) Compute the SVD of the matrix of pixels (the matrix may be rectangular, so non-square). If possible, compute the "reduced" SVD for efficiency reasons, this does not mean any loss of information yet (in contrast to the truncated SVD of the next item).

(d) Display the TSVD $A_k = U_k \Sigma_k V_k^T$ of A for ca 3 values of k as images, and decide what quality is still acceptable to you.

(e) Compute the savings that you can reach in this way (you may assume that each number takes 1 byte).

(6b) Population dynamics; cf. Class 8, 3

In this assignment we study a population of rabbits. We assume that the maximum age is 9.9. Initially, there are 100 rabbits: 15 of 0 year old (between 0 and 1), 14 of 1 old, and then: 13, 12, 11, 9, 8, 7, 6, 5. We assume that if a rabbit is in the youngest category, it has probability 0.99 to reach age 1. Here is a table with all the rates:

Age	Survival rate
0	0.99
1	0.98
2	0.97
3	0.95
4	0.90
5	0.85
6	0.70
7	0.50
8	0.20

Age	Reproduction rate
1	$0.10\,\alpha$
2	$0.90\,\alpha$
3	$0.80\,\alpha$
4	$0.20\,\alpha$

For instance, a rabbit of age 2 gets on average $0.9\,\alpha$ young rabbits, where α is a number that we are going to study.

(a) Argue why $\alpha = 0.5$ is a reasonable first estimate for a stable population, but also why the population will steadily decrease with this value.

Now set up a 10×10 matrix A, with the reproduction rates on positions (1,2)–(1,5), and the survival rates on (2,1), ..., (10,9). If $\mathbf{x}_k$ is a vector with the numbers of rabbits, then we consider the iteration $\mathbf{x}_{k+1} = A\,\mathbf{x}_k$.

(b) Briefly explain why this iteration, using this A, describes the dynamics of the population.

(c) Explain why this matrix does not represent a Markov chain, and why this implies that the total number of rabbits may increase or decrease.

(d) Plot the 10 different age groups over 50 years, and also what happens to the total population size.

Let $A = X\Lambda X^{-1}$ be the eigenvalue decomposition of A.

(e) Show that $\mathbf{x}_k = A^k\,\mathbf{x}_0$, and find an elegant formula for A^k in terms of X and Λ.

(f) Let $\mathbf{c} = X^{-1}\mathbf{x}_0$, and let λ be the eigenvalue (of the 10 in total) with the maximal absolute value. Explain (informally, using **(e)**) why the population will decrease when $|\lambda| < 1$, increase when $|\lambda| > 1$, and remain stable when $|\lambda| = 1$ (although not exactly at 100).

(g) Now find (using trial-and-error, or a smarter method) a value of α (3–4 decimal places) such that the total number remains stable. Check both the population and the maximum eigenvalue λ.