GROUP ASSIGNMENT 2DRR00, SPRING 2024

© MICHIEL HOCHSTENBACH, TU EINDHOVEN, 2024

Register your group members on Canvas before the specified deadline.

READ THIS INFO WELL:
Write a report giving results of these assignments, to be carried out in groups of 4, 5, or 6. (The default is 6, but you don't need permission for 4 or 5.) Submit the report via Canvas before the submission deadline. The format has to be PDF. If you have submitted a version, you can submit an update up to the deadline. The report should be well readable and printable. Answer the questions completely, but rather concisely. Do not overdo the number of pages: try to have approximately 10 (or fewer) pages, but it is OK if you have a few more. You may use Latex to have beautiful math, but Word or another program is also fine. Scanned handwriting is also allowed, but please take care that the file size remains modest ($<$ 5 MB if possible). You may plot matrices in small script $\left[\begin{smallmatrix}1 & 2\\ 3 & 4\end{smallmatrix}\right]$ 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).

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

Occasional contact and discussions with other groups are 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.

It is important that all group members do their share. If you have the feeling that this is not the case, please contact the lecturer as soon as possible. If there are unwilling group members, you may get permission to split the group; however, this is of course not the first option.

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 at least a bit into every question. Didactical studies suggest that learning as a group has benefits. So really 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.

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, explain well what you do, give the results, and analyze and discuss them.

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

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 data fitting.

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

Each of the 6 questions will be scored between 0 to 4 according to the rubric on the next page. An additional 2 bonus points per report may be obtained for original ideas:

- 1 point for an **original problem**
- 1 point for an **original solution**

The final score for the assignment will be given by

number of points / 2.4, rounded to 1 digit after the comma

Category	A	B	C
Approach (1 mark)	Wrong method or inappropriate test problem. Some parts of the question missing.	Appropriate methods applied to suitable test problem. All parts of the question attempted.	–
Mathematical reasoning (3 marks)	Some correct mathematics Missing or imprecise explanations	Mostly correct. Explanation is clear but some details may be missing or explained in too much detail	Mathematically correct, with concise explanation of each step. Sufficient details given
Interpretation/ analysis (3 marks)	Errors in mathematical analysis and little or no interpretation	Interpretation mostly correct but analysis is surface-level	Interpretation of mathematical results is correct and analysis is in suitable depth
Presentation (1 mark)	Layout of solution is confusing The writing is mostly clear and may be missing details or too long. Results are presented in text instead of figure/table. Figures are unclear (e.g. too small, missing labels)	Solution clearly laid out with e.g. subheadings or sections. The writing is clear and an appropriate level of detail. Results are presented in a suitable form (e.g. figure/table). Any figures are large enough to be read easily with legends/labels in a legible font.	

Table 0.1

*(A, B, C) correspond to (0–0.5,0.5–1, 1–1.5) for Approach and Interpretation and (A, B) correspond to (0–0.25, 0.25–0.5) for Approach and Presentation. An additional 2 bonus marks are available **per assignment** for an original problem and an original solution.*

(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 of the question that you use techniques found elsewhere** (e.g. some standard software).

NB 2: If you have 4 group members, it is recommended to ask an arbitrary friend for his/her opinion, to have sufficient input. For this first question only, you may also work together with 1 other group, to have more opinions. However, this is not necessary, it is up to you to decide which you like best.

NB 3: 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!

NB 4: In case you use Latex: norm bars should be done with `\|` and not `||`.

(a) In your 1–2 groups, agree on ca 8 concrete movies (or TV series, movie types, etc) and let all group members rate them $-2, \ldots, 2$. **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: Action, Romantic comedy, Bollywood, Musical, Sci-Fi, Superhero, Comedy, Indie . . .

(b) What could be a reason to scale $-2, \ldots, 2$ instead of the usual $1, \ldots, 5$?

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?

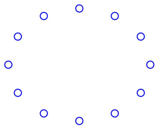
(d) Take another p value, for instance $p = 0.99$ or $p = 0.50$. What is the influence on the PageRank?

(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 $j = 1, \ldots, 10$, 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 $\mathbf{d}_j = V_{1,2}^T \mathbf{e}_j \in \mathbb{R}^2$, $j = 1, \ldots, 10$, 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, eigenvectors, page rank, inverse

(a) Select a graph to study with 10 - 20 nodes. Briefly describe your chosen data and print the adjacency matrix for your graph.

(b) Compute the largest eigenvalue of your matrix and use this to compute Katz centrality for 3 or more appropriate values of α.

(c) Compute at least one more measure of centrality, specifying which measure you are using and with a brief description of how you computed it. You can use the measures from Class 9 or alternative measures. Compare the resulting vector with those you obtained in part (b). What do you see, and can you interpret any differences?

(d) Suppose I want to perform a centrality analysis for railway stations in Europe (over 65,000). Which method would you choose, and what would that method tell us? Discuss the pros and cons of the methods you have used here in terms of computational cost and what they can tell us about our network. Would your answer be different if I wanted to analyse monthly active twitter users (368 million nodes)?

(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. If Matlab is not yet installed on your laptop, you may install it 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. (If you would find this very hard, you can also time by hand if really necessary.)

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 could even think about a “consumentenbond” type of scheme 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 next/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`.)

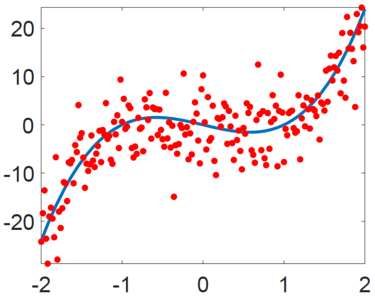
(c) Compute the SVD of the matrix of pixels (the matrix may be rectangular, so non-square). If possible, compute the "reduced" SVD for efficiency reasons, this does not mean any loss of information yet (in contrast to the truncated SVD of the next item).

(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) Linear data fitting; cf. Class 12

In this assignment (on Class 12) we study the polynomial $p(t) = 4t^3 - 4t$ on the interval $[-2, 2]$, in the presence of "noise" (data errors).
Generate the 101 "equidistant" points $t_1, \dots, t_{101}$ as $t_1 = -2$, $t_2 = -1.96$, $t_3 = -1.92$, $\dots$, $t_{99} = 1.92$, $t_{100} = 1.96$, $t_{101} = 2$.
Next, generate $y_i = 4t_i^3 - 4t_i + \eta_i$, for $i = 1, \dots, 101$, where η_i are normally distributed random variables with mean 0 and standard deviation 5.



- Now fit with a cubic polynomial: generate the 101×4 "Vandermonde matrix" A with as elements: 101 ones, and the three sequences of 101 values t_i, t_i^2 and t_i^3.
- Solve the least squares problem $A\mathbf{x} \approx \mathbf{y}$, for $\mathbf{x} \in \mathbb{R}^4$ (on a computer of course!).
- Discuss: to what extent have you been able to reconstruct the polynomial p from the noisy data?
- Also try to fit with a quadratic and linear polynomial. Do you think the results are logical, and as expected?