APPM 2360
Lab 2: Movie Ranking
Due March 23, 2017 by 11:59 pm

---

1 The Model

Sometimes watching a movie can provide an enjoyable break from the study routine; however, as hard working students in this course, viewing a less than preferable film can be a depressing waste of precious time. That's why, in this project, you will construct a model that identifies films that meet your imposed preferences based on genre and its relationship to other similar movies. The goals of this lab are to Construct a preference model in `MATLAB` based on concepts from linear algebra.

Consider the attached Excel file, `pg_movies`, on the course website (originally obtained as a subset from `http://had.co.nz/data/movies/`). You will find a list of PG-rated movies with several other measurement parameters, such as the *length* of the film, it's genre, and so forth, each of which is separated by a semicolon (";"). As we will see, these are important inputs that will ultimately determine our preferences in the `MATLAB` model we will construct.

While there are numerous ways to approach this problem, we will focus on utilizing eigenvalues and eigenvectors to rank our choices - a fundamental concept associated with the PageRank algorithm used by Google Search to sort websites based on user data, as well as other methods such as *Principal Component Analysis* in Statistics. At a basic level, perhaps you personally enjoy comedies - you might at first search a movie database simply based on films with this attribute. The number of titles returned, though, could be potentially daunting, and we can suppose that films related to other movies you enjoyed previously should be ranked higher.

Utilizing eigenvalues and eigenvectors, we can approach our problem slightly differently. In this case, films will be highlighted according to their relevance to film genre and their relationship to other films encompasing these genres. To accomplish this, we will need to compute the eigenvalues of relevant matrices and report the largest,

$$A\mathbf{x} = \lambda_1 \mathbf{x}, \quad \text{where } |\lambda_1| > |\lambda_j| \ \ \forall j \in \{2, \ldots, n\} \tag{1}$$

where $\mathbf{x}$ is the eigenvector corresponding to the largest eigenvalue λ_1.

The largest eigenvalue corresponds to the eigenvector that points in the direction of film titles that are not only most highlighted by our genre selections, but also are substantially related to other films that span these pre-defined genres as well. That is, films which are related to other films that are highly correlated to our genre selection will receive a higher rank in our algorithm, compared to movies that might be classified by many genres, but are not related to other films which are classified by many genres. Based on this, you should be able to make relevant film recommendations for yourself and friends!

2 Your Tasks

1. **Exploring the Data**: Start by importing the dataset, `pg_movies`, on the course website into `MATLAB`. You will need to save the file on a computer and open it via `MATLAB` (use the *Import Data* button). To verify that the data has been uploaded, calculate the summary statistics of each of the column vectors in our matrix by using the `summary` function. To conserve space, just report the `Year`, `Length`, and `Budget` categories. Choose two of these, and give a brief interpretation of the variable statistics.

2. **Sub-Data Problem**: Starting out, let's consider the following matrix of five movies in our dataset:

$$\begin{array}{r} \\ \text{Adventures of Robin Hood, The :} \\ \text{Star Wars: Episode V - The Empire Strikes Back :} \\ \text{Elf :} \\ \text{Ella Enchanted :} \\ \text{Shrek :} \end{array} \begin{pmatrix} \text{Act.} & \text{Ani.} & \text{Com.} & \text{Rom.} \\ \hline 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

   Notice that each film is listed by four different genre categories (*Action, Animation, Comedy*, and *Romance*), where a 0 indicates that it is not defined by the genre, and a 1 indicates that it is (for instance, *The Adventures of Robin Hood* is identified as an action and romance film, whereas The Empire Strikes Back is listed solely as an action film).

   Since this matrix is not square, computing eigenvalues is impossible. Suppose that we reflect the rows of the matrix as the columns, only this time, if a film shares a connection in genre to another, we write a 1, and if it does not, then we write 0 (so *Episode V* would be paired with *Robin Hood*, because they are both *action* films, but *Episode V* would not be paired with any other films in the list because it does not share any other common genre). In this manner, every movie would inherently be paired with itself:

$$\begin{array}{r} \\ \text{Robin Hood :} \\ \text{Episode V :} \\ \text{Elf :} \\ \text{Ella :} \\ \text{Shrek :} \end{array} \begin{pmatrix} \text{Robin Hood} & \text{Episode V} & \text{Elf} & \text{Ella} & \text{Shrek} \\ \hline 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \end{pmatrix}$$

   If the first matrix was A, then this new matrix has the same nonzero entries as AA^T. The difference is that every nonzero entry is replaced with a 1. We want to know if two movies share at least one genre.

   Consider the following questions:

   - Notice that this matrix is symmetric. Will this be the case for any film matrix we set up in this manner?
   - Is this matrix invertible?
   - Compute the eigenvalues and eigenvectors of this matrix by using the `eig` function in `MATLAB`. What is the largest eigenvalue of this matrix? What is the corresponding eigenvector? For details on the `eig` function type `help eig` in the command window.

3. **Interpretation**:

   (a) What is an eigenvalue? How does it relate to its eigenvector? Interpret the largest eigenvalue and its eigenvector pair in the problem above in the context of our movie ratings in relation to genre.

   (b) If the original matrix represents films that you enjoyed, what would our eigenvector imply about films you would like to watch in the future? (*Hint*: Consider the direction of the eigenvector - towards which movies is it pointing and why? In other words, what are the rows corresponding to the most prominent members of your eigenvector? These are the "top" movies highlighted by our algorithm.)

   (c) Why, in our analysis, is *Robin Hood* classified as a lower title than, say, *Shrek*, even though they have the same number of relationships to other films in the matrix?

4. **Bigger Data**: Now we need to adapt our original data table to a matrix similar to the one computed in problem 2. However, to avoid doing this by hand, we need to do some programming. Let's consider exactly what we did in problem 2 to create our sub-matrix:

   - We wrote a matrix with rows and columns corresponding to film titles. In our sub-case, we ended up getting a 5×5 matrix, and since we have 212 film titles in our data table, we should expect to have a 212×212 matrix.
   - Every time we identified a 1 in a category, we checked that column for other films that were classified by that genre as well. If a match existed, then we would insert a 1 in the film matrix to signify that both movies shared a connection. For example, both *Robin Hood* and *Episode V* are classified as action films, so in our film matrix we added a 1 in our entries in the matrix corresponding to their row and column (or column and row).

   Here is an algorithm for finding *Action* connections between the films in our imported table:

```
film = eye(212); % Films paired with themselves
pg_action_matrix = pg_movies.Action;

% All
for i = 1:212 % row
    if (pg_action_matrix(i, 1) == 1)
        for j = 1:212 % column
            if (pg_action_matrix(j, 1) == 1)
                film(i, j) = 1;
            end
        end
    end
end
```

   Consider the algorithm above, and change the code slightly to do this for each genre category in our film table. ALL genre categories in our film table. To find a list of all of the genres in the table, you can use the aforementioned `summary` function-there should be 7 genres in total. Also, please note that the matrix contains only 0s and 1s. (There is a 1 in the i, j and j, i entries for movies sharing one or more categories). To verify that your algorithm is correct use the `spy` function to show that the film matrices corresponding to the *action* and *comedy* genres are symmetric.

5. **Eigenstuff and its applications**: Now we must consider the film matrix from the previous problem and obtain the largest eigenvalue of the film matrix and its corresponding eigenvector. After that, we need to go back to our original table and see the movies to which these choices correspond. Finally, we need to interpret our solution.

   (a) Consider the previously calculated `film` matrix, which includes connections for ALL genres. Use the built in `MATLAB` function `eig` to compute the eigenvalue and eigenvectors of this matrix. Once you have these values, extract the largest eigenvalue and corresponding eigenvector for future use.

   (b) After obtaining your eigenvalue/eigenvector combination, we need to find out the most significant dimensions the eigenvector is pointing towards. To accomplish this, we need to use `MATLAB`'s `sort` function, specifying that we wish to find the largest (absolute) value in the eigenvector and its corresponding index.

   (c) Since our original matrix is square and symmetrical, we can access the films identified above by their row number in the original `pg_movies` table. Do this, and identify the top movie our algorithm has highlighted.

6. **Method Evaluation**: Discuss the following questions:

   - What exactly is going on here? Why do you think this movie was highlighted by our algorithm?
   - Notice that we have formed our `film` matrix by iterating through all genre categories in our data, so we should expect all of these genres to influence the films our algorithm selects. But if this is the case, why aren't all of our suggested films *Documentaries* or *Romances*, even though these genres were highlighted in our algorithm in the previous section?
   - What are some weaknesses about our algorithm? Are there any movies in our data that would never be identified?

7. **Implementation**: Now it's time to try out our algorithm! Consider John, an avid movie watcher who likes *action*, *drama*, and *romance* films. Implement the previous code to show John the top three movies our algorithm suggests for him. To do this, modify your code to generate a new film matrix (maybe call it `film2`) using only the *action*, *drama*, and *romance* genres rather than all 7 included in the data set. You can then implement your algorithm as you did in the previous part to find the top suggested movies. If several movies have the same rank (that is, their corresponding entries in the eigenvector have the same absolute value), then the algorithm can pick any of these.

3 Report Guidelines

You and your group will submit your project on D2L, in the appropriate dropbox (you can find these under the "assessments" tab in D2L) Your group must:

- You must work in a group of three (you cannot work alone). Working in larger or smaller groups will result in a significant penalty for all group members.
- Do not put off finding a group, do this early. You should have a group set up within one week of the project assignment date.

- If you cannot find a group, e-mail the instructors. For this project groups, e-mail Jeremy Thompson
- Submit your project in pdf format. When word documents are uploaded to D2L, the equations in them are commonly jumbled around.
- Submit code used for your project (.nb file(s) for Mathematica, .m file(s) for MatLab, etc).
- Have only ONE group member submit the project. Having multiple people in your group submit the project to D2L will result in multiple grades, and we will take the LOWEST one.
- Include the names of all group members working on the project.

Your report needs to accurately and consistently describe the steps you took in answering the questions asked. This report should have the look and feel of a technical paper. Presentation and clarity are very important. Here are some important items to remember:

- Absolutely make sure your recitation number is on your submitted report.
- Start with an introduction that describes what you will discuss in the body of your document. A brief summary of important concepts that you will be using in your discussion could be useful here as well.
- Summarize what you have accomplished in a conclusion. No new information or new results should appear in your conclusion. You should only review the highlights of what you wrote about in the body of the report.
- Always include units in your answers.
- Always label plots and refer to them in the text. The main body of your paper should NOT include lengthy calculations. These should be included in an appendix, and referred to in the main body.
- Labs must be typed. Including the equations in the main body (part of your learning experience is to learn how to use an equation editor). An exception can be made for lengthy calculations in the appendix, which can be hand written (as long as they are neat and clear), and minor labels on plots, arrows in the text and a few subscripts.
- Your report doesn't have to be long. You need quality, not quantity of work. Of course you cannot omit any important piece of information, but you need not add any extras.
- DO NOT include printouts of computer software screens. This will be considered as garbage. You simply need to state which software you used in each step, and what it did for you.
- You must include any plot that supports your conclusions or gives you insight in your investigations.
- Write your report in an organized and logical fashion. Section headers such as Introduction, Background, Problem Statement, Calculations, Results, Conclusion, Appendix, etc... are not mandatory, but are highly recommended. They not only help you write your report, but help the reader navigate through your paper, besides giving it a clearer look.