Data Mining

Homework 4

Due: 8/1/2023, 23:59

Instructions

You must hand in the homework electronically and before the due date and time.

This homework has to be done by each **person individually**.

Handing in: You must hand in the homework by the due date and time by an email to Gianluca (`decarlo.1805894@studenti.uniroma1.it`) that will contain as attachment **(not links to some file-uploading server!)** a .zip file with your answers. The filename of the attachment should be
`DM_Homework_1__StudentID_StudentName_StudentLastname.zip`;
for example:
`DM_Homework_1__1235711_Robert_Anthony_De_Niro.zip`.
The email subject should be
`[Data Mining] Homework_1 StudentID StudentName StudentLastname`;
For example:
`[Data Mining] Homework_1 1235711 Robert Anthony De Niro`.
After you submit, you will receive an acknowledgement email that your project has been received and at what date and time. If you have not received an acknowledgement email within 2 days after you submit then contact Gianluca.

The solutions for the theoretical exercises must contain your answers either typed up or hand written clearly and scanned.

For information about collaboration, and about being late check the web page.

Problem 1. In this exercise we will build a simple recommender system. We will work with the dataset MovieLens. This dataset describes 5-star ratings and free-text tagging activity from MovieLens, a movie recommendation service. It contains 100836 ratings and 3683 tag applications across 9742 movies. These data were created by 610 users between March 29, 1996, and September 24, 2018. This dataset was generated on September 26, 2018. Users were selected at random for inclusion, and all selected users had rated at least 20 movies.

1. **Data preparation**

   The dataset can be downloaded from here; select the **latest** one (We suggest that you start with the **small version** to avoid time loss due to computational complexity. Direct download here). After downloading the dataset, the data are contained in the files 'links.csv', 'movies.csv', 'ratings.csv', and 'tags.csv'. For more details about the contents and structure of the files, read the **README.txt** file of the dataset.

2. **Exploratory data analysis** (EDA)

   Perform an exhaustive EDA. It is useful to understand the insights of the key characteristics of various entities of the dataset. Compute and visualize the statistics that seem important to you (*e.g.*, average rating, the average number of films rated by each user, etc.).

3. **Model design and training** Look at the book chapter on recommender systems the various approaches on colloaborative filtering. Preprocess the data and design a recommender

system. You are free to experiment with different methods (item-based, user-based, matrix-factorization, etc.).

4. **Model evaluation** Split the data into training and testing (80–20 percent). Make sure that the split is fair; it just removes random entries. Make sure that the users and movies in the test set are also represented in the training set. Use the RMSE to evaluate and compare your approaches.

Problem 2. Building a Graph-Based Recommendation System

Here we will use graphs and GNNs to build a recommender system for MovieLens. This dataset describes 5-star ratings and free-text tagging activity from MovieLens, a movie recommendation service. It contains 100836 ratings and 3683 tag applications across 9742 movies. These data were created by 610 users between March 29, 1996, and September 24, 2018. This dataset was generated on September 26, 2018. Users were selected at random for inclusion, and all selected users had rated at least 20 movies.

1. **Data preparation**

   You need to build a **bipartite graph**. In this network, users and films represent the nodes, and an edge between a user and a film exists only if the user has rated such film; you can select a threshold $t > 0$ such that an edge between a user u and a film f is added *iff* $r_{u,f} > t$, where $r_{u,f}$ is the ranking given by u to f.

2. **Model design and training**

   Build your own Graph Neural Network (GNN). Try different layers and architectures. You are free to choose between the libraries **PyTorch** and **DGL**. The task you have to perform is **edge prediction**: if the score of the predicted edge between a user u and a film f is greater than the threshold you have defined before, then you can suggest f to u. Remember to divide your dataset into train and test sets. The performances on the test set are the ones that matter! To further test your model, you can mask out some of the original edges and see how many of them are recovered by your trained model. You have to hand in the code along with a report (about 3–5 pages) in which you describe all the steps made (plots are welcome). In particular, show how you handled the data, describe the models you chose, and provide tables with the performance metrics. Plot also a confusion matrix. Feel free to add any comment/observation you think to be relevant. Compare the results with the more classical approaches from the previous problem.