CSE 515 Multimedia and Web Databases

Phase #1

(Due October 22nd 2017, midnight)

Description: In this project, you will experiment with

- vector models and
- graph models.

This project phase will be performed as a group. You will be provided with sample MovieLens+IMDB data in the form of CSV files. You are free to store the data in a relational database (such as MySql) or create an in-memory data structure to store the provided network. For PCA, SVD, LDA, and CP decomposition you can use existing packages.

- **Task 1:**
  - **Task 1a:** Implement a program which, given a genre, identifies and reports the top-4 latent semantics/topics using
    - PCA in TF-IDF space of tags,
    - SVD in TF-IDF space of tags, and
    - LDA in the space of tags.
  - **Task 1b:** Implement a program which, given a genre, identifies and reports the top-4 latent semantics/topics using
    - PCA in TF-IDF space of actors,
    - SVD in TF-IDF space of actors, and
    - LDA in the space of actors.
  - **Task 1c:** Implement a program which, given an actor, finds and ranks the 10 most similar actors by comparing actors'
    - TF-IDF tag vectors,
    - top-5 latent semantics (PCA, SVD, or LDA) in the space of tags.
  - **Task 1d:** Implement a program which, given a movie, finds and ranks the 10 most related actors who have not acted in the movie, leveraging the given movie's
    - TF-IDF tag vectors,
    - top-5 latent semantics (PCA, SVD, or LDA) in the space of tags.
- **Task 2:**
  - **Task 2a:** Implement a program which
    1. creates an *actor-actor* similarity matrix (using tag vectors),
    2. performs SVD on this *actor-actor* similarity matrix,
    3. reports the top-3 latent semantics, in the actor space, underlying this *actor-actor* similarity matrix, and
    4. partitions the actors into 3 non-overlapping groups based on their degrees of memberships to these 3 semantics.
  - **Task 2b:** Implement a program which

1. creates a *coactor-coactor* matrix based on co-acting relationships (recording the number of times two actors played acted in the same movie),
2. performs SVD on this *coactor-coactor* matrix,
3. reports the top-3 latent semantics, in the actor space, underlying this *coactor-coactor* matrix, and
4. partitions the actors into 3 non-overlapping groups based on their degrees of memberships to these 3 semantics.

- **Task 2c:** Implement a program which
  1. creates an *actor-movie-year* tensor, where the tensor contains 1 for any actor-movie-year triple if the given actor played in the stated movie and the movie was released in the stated year (the tensor contains 0 for all other triples)
  2. performs CP on this *actor-movie-year* tensor with target rank set to 5,
  3. reports the top-5 latent
     - actor
     - movie
     - year

     semantics underlying this tensor, and
  4. partitions
     - actors
     - movies
     - years

     into 5 non-overlapping groups based on their degree of memberships to these 5 semantics.
- **Task 2d:** Implement a program which
  1. creates *tag-movie-rating* tensor, where the tensor contains 1 for any tag-movie-rating triple if the given tag was assigned to a movie by at least one user and the movie has received an average rating lower than or equal to the given rating value (the tensor contains 0 for all other triples)
  2. performs CP on this *actor-movie-rating* tensor with target rank set to 5,
  3. reports the top-5 latent semantics in terms of
     - tag
     - movie
     - rating

     memberships underlying this tensor, and
  4. partitions
     - tag
     - movies
     - ratings

     into 5 non-overlapping groups based on their degree of memberships to these 5 semantics.

- **Task 3:**
  - **Task 3a:** Implement a program which
    1. creates an *actor-actor* similarity matrix (using tag vectors),
    2. given a set, S, of "seed" actors (indicating the user's interest), identifies the 10 most related actors to the actors in the given seed set using Random Walk with ReStarts (RWR, or Personalized PageRank, PPR) score. See *J.-Y. Pan, H.-J. Yang, C. Faloutsos, and P. Duygulu. Au- tomatic multimedia cross-modal correlation discovery. In KDD, pages 653658, 2004.*

  - **Task 3b:** Implement a program which

    1. creates a *coactor-coactor* matrix based on the number of movies two actors acted in together,
    2. given a set, S, of "seed" actors (indicating the user's interest), identifies the 10 most related actors to the actors in the given seed set using RWR.

- **Task 4:** Implement a program which

  1. given all the information available about the set of movies a given user has watched, recommends the user 5 more movies to watch.

Every result should be presented in decreasing order of weights!

Deliverables:

- Your code (properly commented) and a README file.
- Your outputs for the provided sample inputs.
- A short report describing your work and the results.

Please place your code in a directory titled "Code", the outputs to a directory called "Outputs", and your report in a directory called "Report"; zip or tar all off them together and submit it through the digital dropbox.