Project 1 – Information Diffusion and Influence in Twitter

Aim:

Implement algorithms to simulate information flow via diffusion and influence in a network based on a real-world Twitter dataset.

Tasks:

1- Dataset

- Load the following dataset: https://snap.stanford.edu/data/higgs-twitter.html.
- Examine the following files: a. retweet_network.edgelist.gz b. reply_network.edgelist.gz c. mention_network.edgelist.gz
- Reverse all edges in graphs
- Sum the edge weights for reply, mention and retweet set (use this for ICM and LTM methods). Also, normalize the edge weight sums between 0 and 1.

2- Implementation

- Implement Independent Cascade Model (ICM) algorithm.
- Implement greedy algorithm to find the best set of initial nodes to be activated in order to maximize the spread with the ICM algorithm.

3- Analysis & Visualization

- Use ICM to plot the number of activated nodes over time. Analyze once with the normal ICM algorithm and another with an intervention after iterations of your choice.
- Plot the number of nodes reached (spread) as a function of different interesting values for the budget parameter *k* for the greedy algorithm.
- Compute the Pearson correlation between the nodes in the network for mention, retweet and reply edges, and compare the values. Each node should have 3 ordinal values: the sum of all outgoing 1) 'mention' edges, 2) 'retweet' edges, and 3) 'reply' edges.
- Use Linear Threshold Model (LTM) to plot the number of activated nodes over time. Examine multiple runs with different random thresholds.

Contact person: Akansha Bhardwaj (akansha.bhardwaj@unifr.ch)

Project 2 - Information Diffusion and Influence in Twitter

Aim:

Implement algorithms to simulate information flow via diffusion and influence in a network based on a real-world Twitter dataset.

Tasks:

1- Dataset

- Load the Twitter dataset from https://snap.stanford.edu/data/higgs-twitter.html
- Examine the following files: a. retweet_network.edgelist.gz b. reply_network.edgelist.gz c. mention_network.edgelist.gz
- Reverse all edges in graphs
- Sum the edge weights for reply, mention and retweet set (use this for ICM and LTM methods). Also, normalize the edge weight sums between 0 and 1.

3- Implementation

- Implement Linear Threshold Model (LTM) algorithm. Initialize each node with a randomly generated threshold.
- Implement Pearson correlation. For this, each node should have 3 ordinal values: the sum of all 1) ‘mention’ edges, 2) ‘retweet’ edges, and 3) ‘reply’ edge.

4- Analysis & Visualization

- Use Linear Threshold Model (LTM) to plot the number of activated nodes over time. Examine multiple runs with different random thresholds.
- Use ICM to plot the number of activated nodes over time.
- Use the greedy algorithm to plot the number of nodes reached (spread) as a function of different interesting values for the budget parameter k.
- Compute Pearson correlation between the nodes in the network.

Contact person: Akansha Bhardwaj (akansha.bhardwaj@unifr.ch)

Project 3 - Social Recommendation Systems (FilmTrust Dataset)

Aim:

The aim of this project is to build a tool that implements and compares different types of recommendation algorithms on a real-world dataset. The tool should be able to:

- Load the dataset
- Run different recommendation algorithms on the dataset
- Compare and discuss the results, parameter sensitivity

Tasks:

1. **Dataset**

- Load the FilmTrust dataset from https://www.librec.net/datasets.html.
- Explore the properties of the graph.

2. **Implementation**

- Implement User based Collaborative Filtering Recommender with cosine similarity.
- Implement the Probabilistic matrix factorization algorithm.

3. **Analysis**

Perform the following analytical tasks using the previous implementation:

- Split the dataset into 80%-20% training and testing data.
- Investigate the effect of using different neighbourhood sizes of your choice for the collaborative filtering algorithm and compare the difference. Compare the results using MAE and RMSE.
- Investigate the sensitivity of the parameters for matrix factorization algorithm by varying the range of latent dimension from 5-50.

Contact person: Rana Hussein (rana.hussein@unifr.ch)

Project 4 - Social Recommendation Systems (FilmTrust Dataset)

Aim:

The aim of this project is to build a tool that implements and compares different types of recommendation algorithms on a real-world dataset. The tool should be able to:

- Load the dataset
- Run different recommendation algorithms on the dataset
- Compare and discuss the results, parameter sensitivity

Tasks:

1. **Dataset**

- Load the FilmTrust dataset from https://www.librec.net/datasets.html.
- Explore the properties of the graph

2. **Implementation**

- Implement Item based Collaborative Filtering Recommender with Pearson correlation coefficient.
- Implement the Probabilistic matrix factorization algorithm.

3. **Analysis**

Perform the following analytical tasks using the previous implementation:

- Split the dataset into 80%-20% training and testing data.
- Investigate the effect of using different neighbourhood sizes of your choice for the collaborative filtering algorithm and compare the difference. Compare the results using MAE and RMSE.
- Investigate the sensitivity of the parameters for matrix factorization algorithm by varying the range of latent dimension from 5-50.

Contact person: Rana Hussein (rana.hussein@unifr.ch)

Project 5 - Social Recommendation Systems (CiaoDVD Dataset)

Aim:

The aim of this project is to build a tool that implements and compares different types of recommendation algorithms on a real-world dataset. The tool should be able to:

- Load the dataset
- Run different recommendation algorithms on the dataset
- Compare and discuss the results, parameter sensitivity

Tasks:

1. **Load the dataset**

- Load the CiaoDVD dataset from https://www.librec.net/datasets.html
- Explore the properties of the graph

2. **Implementation**

- Implement User based Collaborative Filtering Recommender with pearson correlation coefficient.
- Implement the Probabilistic matrix factorization algorithm.

3. **Analysis**

Perform the following analytical tasks using the previous implementation:

- Split the dataset into 80%-20% training and testing data.
- Investigate the effect of using different neighbourhood sizes of your choice for the collaborative filtering algorithm and compare the difference. Compare the results using MAE and RMSE.
- Investigate the sensitivity of the parameters for matrix factorization algorithm by varying the range of latent dimension from 5-50.

Contact person: Akansha Bhardwaj (akansha.bhardwaj@unifr.ch)

.

Project 6 - Social Recommendation Systems (CiaoDVD Dataset)

Aim:

The aim of this project is to build a tool that implements and compares different types of recommendation algorithms on a real-world dataset. The tool should be able to:

- Load the dataset
- Run different recommendation algorithms on the dataset
- Compare and discuss the results, parameter sensitivity

Tasks:

1. **Load the dataset**

- Load the CiaoDVD dataset https://www.librec.net/datasets.html
- Explore the properties of the graph

2. **Implementation**

- Implement the Item based Collaborative Filtering Recommender with cosine similarity.
- Implement the Probabilistic matrix factorization algorithm.

3. **Analysis**

Perform the following analytical tasks using the previous implementation:

- Split the dataset into 80%-20% training and testing data.
- Investigate the effect of using different neighbourhood sizes of your choice for the collaborative filtering algorithm and compare the difference. Compare the results using MAE and RMSE.
- Investigate the sensitivity of the parameters for matrix factorization algorithm by varying the range of latent dimension from 5-50.

Contact person: Akansha Bhardwaj (akansha.bhardwaj@unifr.ch)

Project 7 - Human Computation and Crowdsourcing on Amazon Data

Aim:

The aim of this project is to build a tool that implements and compares different types of output aggregation algorithms on real-world datasets. The tool should be able to:

- Load the dataset
- Run different output aggregation algorithms on the dataset
- Compare and discuss the results

Tasks:

1. Load the dataset

Consider the Product dataset obtained by asking workers whether two products identified each by an ID are matching or not:
https://drive.google.com/drive/folders/1FTvV-h9rLavtk1OZSCNZOhCFDxKoW0Ol

The dataset contains three files:

- truth.csv: contains the true label (column truth) for each question (column question)
- data_product: contains the id, the name, the description and the price of each product.
- answer.csv: represents the worker answer matrix. For each question, three workers provide a label whether two products (identified by an id) are matching or not.

- Load the Product dataset.
- Explore the properties of the graph.

2. Implementation

- Implement the majority voting algorithm.
- Implement the Dawid and Skene (DS) algorithm using random initialization.

3. Analysis & Visualization

Perform the following analytical tasks using the previous implementation:

- Plot the distribution of the products labelled per worker and the distribution of worker accuracy for the dataset.
- Split the labelled products into training (60%), validation (20%) and test set (20%)
- Train the algorithms on the worker answers using DS in the training set and evaluate the algorithms against the gold labels in the test set using the accuracy and F1 metrics.
- Infer the true labels using majority voting.

Contact person: Rana Hussein (rana.hussein@unifr.ch)

Project 8 - Human Computation and Crowdsourcing on Fashion Data

Aim:

- The aim of this project is to build a tool that implements and compares different types of output aggregation algorithms on real-world datasets. The tool should be able to:
- Load the dataset
- Run different output aggregation algorithms on the dataset
- Compare and discuss the results

Tasks:

1. **Dataset**

Consider the Fashion dataset obtained by asking workers to name fashion influencers. Each worker provides at least three twitter usernames of candidate fashion influencers: https://drive.google.com/drive/folders/1X9E34IkWhRCAnWHhB3bF-KLX4-Vnr9hD
The dataset contains three files:

- social features fashion: contains the features of each candidate influencer including the number of followers, followings, tweets and the bag of words of their tweet.
- aij_fashion.csv: represents the worker answer matrix. The matrix consists of three columns. The first one is the worker id, the second column represents the influencer id and the third column is the naming relation (1 if the worker named the candidate influencer and 0 otherwise).
- labels_fashion.csv: for each candidate influencer identified by an id (first column), we provide a label (1 if the candidate is a rela influencer and 0 otherwise)

- Load the Product dataset.
- Explore the properties of the graph.

2. **Implementation**

- Implement the majority voting algorithm.
- Implement the Dawid and Skene (DS) algorithm. Initialize using Majority Voting.

3. **Analysis & Visualization**

Perform the following analytical tasks using the previous implementation:

- Plot the distribution of the products labelled per worker and the distribution of worker accuracy for the dataset.
- Split the labelled products into training (60%), validation (20%) and test set (20%)
- Train the algorithms on the worker answers using DS in the training set and evaluate the algorithms against the gold labels in the test set using the accuracy and F1 metrics.
- Compare the true labels computed using DS with random initialization and majority voting initialization.

Contact person: Rana Hussein (rana.hussein@unifr.ch)

Helpful libraries:

- You can optionally use the following libraries to load your data:
  - NetworkX: https://networkx.github.io/
  - SNAP: https://snap.stanford.edu/snappy
  - iGraph http://igraph.org/python/
- For visualization: You can use the following libraries:
  - matplotlib http://matplotlib.org/
  - vis.js http://visjs.org/
  - d3.js https://d3js.org/