Explaining Recommendations

Quentin Deligny

2262655D

Proposal

Motivation

In the area of recommender systems, the focus is usually on improving the performance of various recommender systems. As such, these systems often act as a black box which provides little to no insight to the end user. However, recent studies have shown that providing clear explanations to the users may have several benefits, such as improving user satisfaction, trust and forgiveness (in cases where the system may be wrong). The explanation process tends to make users feel more involved, resulting in a higher acceptance of the recommendations. Further studies have proven that various types of explanations also had different effects on users. Businesses or entities seeking to improve the quality of their recommendations should also consider the quality of their explanation methods to ensure a high user satisfaction.

Aims

This project aims to develop a set of explanation techniques for one (potentially more for comparison purposes) type of recommender system. These explanation methods should rely on visualisations to further elicit the inner-workings of recommendation algorithms to the user. The chosen visualisations may be presented to users using an interface in order to assess their efficiency.

Progress

- Research survey and setting up of the initial environment in Python.
- Defining tools: using MovieLens to provide the dataset, SpotLight to implement models, t-SNE to produce graphical visualisations and Tkinter for the user interface (alongside other libraries).
- Establishing requirements for the projects as well as a list of potential explanation techniques.
- Implemented basic collaborative-filtering model with SpotLight and visualised output with t-SNE.
- Produced various types of graphs representing different movie genres, users and their neighbouring movies, users and their neighbouring users (in both cases showing closest and farthest neighbours).

- Produced multiple graphs to show the evolution of a specific user as the model is being trained.
- Built tables to track the evolution of RMSE scores for both the train and test sets.
- Decomposed the training process by splitting up the training data into different sets, and even single interactions (feeding each split progressively to the model in order to assess its impact on the recommendations made to a specific user). This was done to avoid overfitting the model.
- Showing current closest items to a specific user along with closest items from the previous training step to better understand how the model evolved.
- Started basic work on the GUI with Tkinter.
- Further testing with larger dataset and different parameter combinations (such as learning rate, train/test splits, number of model steps, perplexity, embedding dimension, type of visualisation) to better understand their impact.

Problems and risks

Problems

- Some initial issues with the compatibility of all the different libraries: could not use Jupyter Notebook (some versions did not work together).
- Producing interactive graphs was glitchy due to the large number of graphs. Many issues with the labelling of items in different graphs.
- How to handle some data without creating a bias in the final representation (multi-label movies, movies with no labels…).
- Finding relevant explanations methods (judged effective by previous research). Not all can be applied to this project.
- Tweaking the parameters to find an ideal configuration is exhaustive given the number of parameters and time taken to run the algorithm (cannot use cross-validation as the parameters are not just for the model).
- The algorithm takes too long to run on a larger dataset making it very difficult to test a wide combination of parameters. Furthermore, a large dataset caused too much clutter in the graphical visualisations.
- The dimension reduction operated by t-SNE was quite unstable in higher dimensions (PCA mitigated the effects) leading to inconsistent visualisations.
- There are some ‘oddballs’ in the visualisations (uncoherent shapes) and lack of consistency between each step.

Risks

- If a survey is needed, delays to get approval from Board of Ethics should be accounted for. If it is approved, it may be a challenge to find a significant number of relevant users to participate.
- There will likely be compatibility issues with the current project when trying to implement other kinds of recommender systems (evaluation techniques may not suit all systems).
- There are a lot of variations between each training step. The key is finding the right time to stop training the model (RMSE test score does not increase clearly or consistently). This may result in different outputs (recommendations) given the same input.
- Lacking clear metrics to evaluate the project's success (other than successful implementation of explanation techniques).

Plan

- Week 1-3: Polish and improve the GUI presenting various explanation techniques (as well as the techniques themselves). Potentially design small survey to gather initial feedback from a small sample of users (if deemed beneficial and compatible with the project).
- Week 4-6: Conducting survey if it was implemented. Otherwise, could implement other types of recommender systems to see how they compare with the initial ones.
- Week 7-9: Final changes to the product. Starting to work on the final dissertation.
- Week 10+: Keep working on final dissertation and send draft to supervisor 1-2 weeks before deadline.