CS5228 – Tutorial 8

Recommender Systems

1. **Recommendation Systems – Basic Challenges.** In the lecture, we came across 2 basic problems when building recommendation systems or engines: *popularity bias* and *cold-start problem*. Briefly describe both problems in your own words.

2. **Explicit vs implicit feedback.** Amazon's 5-star rating scheme or Reddit's upvote/downvote scheme are considered explicit feedback. In contrast, implicit feedback may refer to users' playlist/purchase/clickthrough/etc. history. What are the main limitations of implicit feedback compared to explicit feedback?

3. **Normalization.** Why do we typically normalize the ratings by mean-centering them, i.e., by subtracting the mean, either for a user or an item?

4. **Content-Based Recommender Systems**. Content-based recommender systems require to represent items as some form of features vector (item profiles) to calculate distances/similarities between them.

   (a) For the following 5 types of items, what are arguably useful information to create a item profile to allow for meaningful recommendations

   - Electronic devices (e.g., phone, cameras, laptops)
   - News articles
   - Hotel (rooms)
   - Books
   - Property/Housing

   (b) Based on your answers in (a), how would you classify items into 2 basic categories when it comes to building a content-based recommendation system? This is a very open question, and there are probably many good answers.

5. **KNN-Based Recommender System**. We saw that many data mining algorithms can be used to make recommendations, such as Clustering, Association Rule Mining, or Classification/Regression models. Let's consider the K-Nearest Neighbor Algorithm here.

   (a) Sketch a KNN algorithm to recommend items based on user similarity derived using only the rating matrix R!

   (b) How does the choice of K in KNN is likely to affect the quality of recommendations?