NUS | Computing
National University of Singapore

CS5228: Knowledge Discovery and Data Mining

Tutorial 8 — Recommender Systems

Question 1

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.

Question 1

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.

Solution

- Popularity bias: only a small subset of items gets regularly recommended ("few get richer", "rich get richer")

- Cold start problem:
  - Particularly when a new user joins the platform, no or not much information about that user is available to provide proper personalized information.
  - In principle, the same is true for a new item but at least some kind of item-item similarity can be used to make (halfway decent) recommendations.

Question 2

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?

Question 2

2. **Explicit vs implicit feedback.** Amazon's 5-star rating scheme or Reddit's up-vote/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?

Solution

- Not necessarily a clear indicator for a user's preference
  (e.g., just clicking on some content does not imply that the user did indeed like the content)
- Lack of a clear notion of negative feedback
  (absence of interaction does not imply lack of interest, liking, etc.)
- Likely to contain more noise
  (e.g., a user might accidentally click on a well disguised ad)

Question 3

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?

Question 3

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?

Solution

- Most numeric rating schemes use only positive values; low positive values reflect dislike
  (for most computations, representing dislike by negative values yield more expressive results)

- Normalizing s more ”generous” users and more ”grumpy” users to the same scale.

- Algorithms that use **all** entries of rating matrix R assume a meaningful interpretation of 0
  (without normalization 0 would represent a stronger dislike than the lowest value of 1 star)

Question 4

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) specs
   - News articles region, keywords, category
   - Hotel (rooms) #beds, size, amenities
   - Books author, genre
   - Property/Housing

Question 4

"Solution"

- Electronic devices (e.g., phone, cameras, laptops)
  - basically all technically features

- News articles
  - Good: source (newspaper and/or author), text features (but not trivial to extract)
  - Questionable: article length, number of images

- Hotel (rooms)
  - basic information such as size, amenities, category (star rating), location
  - The problem is that these information typically provide a very incomplete picture.

- Books
  - author, genre, publication year, (length?)
  - Again, these information will often be not sufficient

- Property/Housing
  - area size, floor height, age, location

Question 4

(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.

Question 4

(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.

Solution

- Main difference (arguably): subjective ratings/opinions vs. objective ratings/opinions
- Typically easier to find good features for "objective" items
- Ratings for "objective" items typically more "stable"

Question 5

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!

Question 5

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!

Solution

- Optional: Extract user vectors from data matrix and normalize values
- Calculate distances between user vectors using a suitable distance metric
  (e.g., Cosine Similarity, Pearson Correlation Coefficient)
- For each user *u*, find the K-nearest neighbors *N* (i.e., the most similar users)
- Aggregate the interactions of the *N* users (weighted by their similarity scores) to calculate the predicted ratings for *u* (only for items with sufficient ratings!).
- Rank the items that the *u* has not yet rated by the predicted ratings to derive meaningful recommendations (e.g., top-ranked items but with some diversity)

Question 5

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

Question 5

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

Solution

- K too small
  - recommendations rely on a very limited number of neighbors, which might lead to biased or highly personalized recommendations; this may cause overfitting
  - recommendations that are likely to be very narrow or idiosyncratic
  - recommendations may be highly accurate for some users but less diverse or useful for others, as the model may miss out on exploring broader patterns across multiple users

- K too large:
  - recommendations start to rely on too many neighbors that may not be very similar to *u*
  - too many neighbors are likely to ”dilute” the influence of the most similar users and increases the risk of introducing noise from less relevant users
  - recommendations become more generalized and less tailored to individual preferences; however, the diversity of recommendations may improve