COMP9727: Recommender Systems

Project Design Report

Name: Jianyu Huang
Student ID (zID): z5474273

---

1. Scope

Domain and Target Users

This project will focus on the **tourism domain**. It is aiming to build a personalized recommendation system. And assist travelers in discovering destinations, attractions, and accommodations that align with their individual preferences. The targeted user group includes travelers aged 18–45, who actively seek customized and tailored recommendations. This system can enhance their travel planning experiences.

Recommendation Presentation and User Interaction

- **Platform:** Web-based.
- **Items Displayed:** 5–10 recommendations per interaction, including tourist attractions, destinations, and hotels.
- **User Interaction:**
  - Explicit Feedback**:** Users rate recommended items (1–5 stars).
  - Implicit Feedback**:** Click behavior, browsing duration, saving favorites.
  - Users can also define interest tags to refine personalized recommendations.

User Interface (UI) Mockup

- **Homepage:** Personalized recommendation lists, highlighting travel destinations, attractions, and accommodations.
- **Detail Page:** Comprehensive information on each recommendation, including images, descriptions, ratings, and user reviews.
- **User Profile Page:** Allows users to view and update their preference tags for more precise recommendations.

Cold-Start and Dynamic Updates

- **Cold-start:** Mitigated by an initial user-preference questionnaire combined with popularity-based recommendations.
- **Dynamic Updates:** Continuous collection of user feedback with periodic model retraining ensures recommendations remain relevant and responsive.

Business Model

- Revenue generated through partnerships with online travel agencies (OTAs), earning commissions from bookings made through recommendations.
- Premium membership options offering enhanced personalization and tailored recommendation services.

---

2. Datasets

To support this tourism-focused recommendation system, three high-quality, publicly available datasets are selected:

1. **Travel Recommendation Dataset (Kaggle)**
    - **Source:** Kaggle
    - **Description:** Contains user ratings and reviews for travel destinations worldwide.
    - **Application:** Used to train and evaluate collaborative filtering and graph-based recommendation models.
    - **Limitations:** Relatively small dataset size; may require integration with other datasets for improved coverage.
2. **HotelRec Dataset**
    - **Source:** TripAdvisor
    - **Description:** Over 50 million hotel reviews, suitable for comprehensive user-preference modeling and sentiment analysis.
    - **Application:** Enables hotel-specific recommendations leveraging detailed user feedback.
    - **Limitations:** Primarily hotel-focused; additional datasets needed to expand into broader tourism recommendations.
3. **Foursquare Check-in Dataset**
    - **Source:** Foursquare
    - **Description:** Global user check-in data, providing geographic and temporal insights into user travel behaviors.
    - **Application:** Facilitates location-based recommendations, capturing geographical preferences.
    - **Limitations:** Lacks explicit rating information, necessitating integration with other datasets to enrich user preference insights.

---

3. Methods

The recommendation system will employ two primary methods—**Collaborative Filtering (CF)** and **Graph Convolutional Networks (LightGCN)**—combined through a hybrid strategy.

Method 1: Collaborative Filtering (CF)

Collaborative filtering utilizes historical user-item interaction data to recommend new items to users. The detailed implementation steps are:

- **Data Preparation:**
  - Construct a user-item rating matrix from the Travel Recommendation and HotelRec datasets.
  - Represent sparse rating data efficiently.
- **Similarity Computation:**
  - Calculate similarity scores using Cosine Similarity or Pearson Correlation.
  - Determine neighbor sets for recommendation.
- **Rating Prediction and Recommendation Generation:**
  - Predict unseen user ratings using neighbor-based methods.
  - Provide top-N recommendations based on predicted ratings.
- **Strengths:** Efficient, intuitive, suitable for dense rating data.
- **Limitations:** Reduced effectiveness with sparse data or new users (cold-start).

Method 2: Graph Convolutional Network (LightGCN)

LightGCN leverages graph-based deep learning to capture complex user-item interactions. Key steps are:

- **Data Preparation and Graph Construction:**
  - Build a User-Item graph using check-in and review data.
  - Graph nodes represent users/items; edges represent interactions.
- **Model Training:**
  - Initialize embeddings for users and items.
  - Apply graph convolution (Neighborhood Aggregation):

$$e_u^{(k+1)} = AGGREGATE(\{ei_i^{(k)} : i \in N(u)\})$$

  - Employ multiple aggregation layers to capture high-order interactions.
- **Prediction and Optimization:**
  - Compute user-item preference scores from learned embeddings.
  - Tune hyperparameters via Bayesian optimization.
- **Strengths:** Robust handling of sparse and complex interactions.
- **Limitations:** Higher computational cost; training time can be significant.

Hybrid Recommendation Strategy

- Combine CF and LightGCN using a weighted-average fusion method.
- Adjust weights dynamically based on validation results and user feedback.
- Goal: Maximize recommendation accuracy and diversity.

4. Evaluation

Recommendation Model Performance

- **Accuracy Metrics:**
  - RMSE (Root Mean Square Error)
  - MAE (Mean Absolute Error)
  - Precision@N, Recall@N (N=5, 10)
- **Diversity and Novelty Metrics:**
  - Coverage (breadth of recommended items)
  - Novelty (frequency of recommending less-popular items)

User Satisfaction Evaluation (User Study)

- Recruit 20 users from the target demographic.
- Users interact with a simulated recommendation UI prototype.
- Collect feedback through surveys evaluating:
  - Recommendation quality (accuracy and relevance)
  - Diversity and novelty of suggestions
  - Ease of use and overall user experience

System Efficiency Evaluation

- Test system responsiveness under varying loads.
- Evaluate scalability with increasing data volume.

Metrics Trade-offs

- Prioritize accuracy and user satisfaction metrics.
- Balance recommendation, novelty/diversity, and computational efficiency.