Hybrid Movie Recommendation System

EE359, 2021 Spring, SJTU

Ziliang Guo 518030910273
Aofan Jiang 518030910275

June, 2021

Outline

A Introduction

B Movie-Based

C User-Based

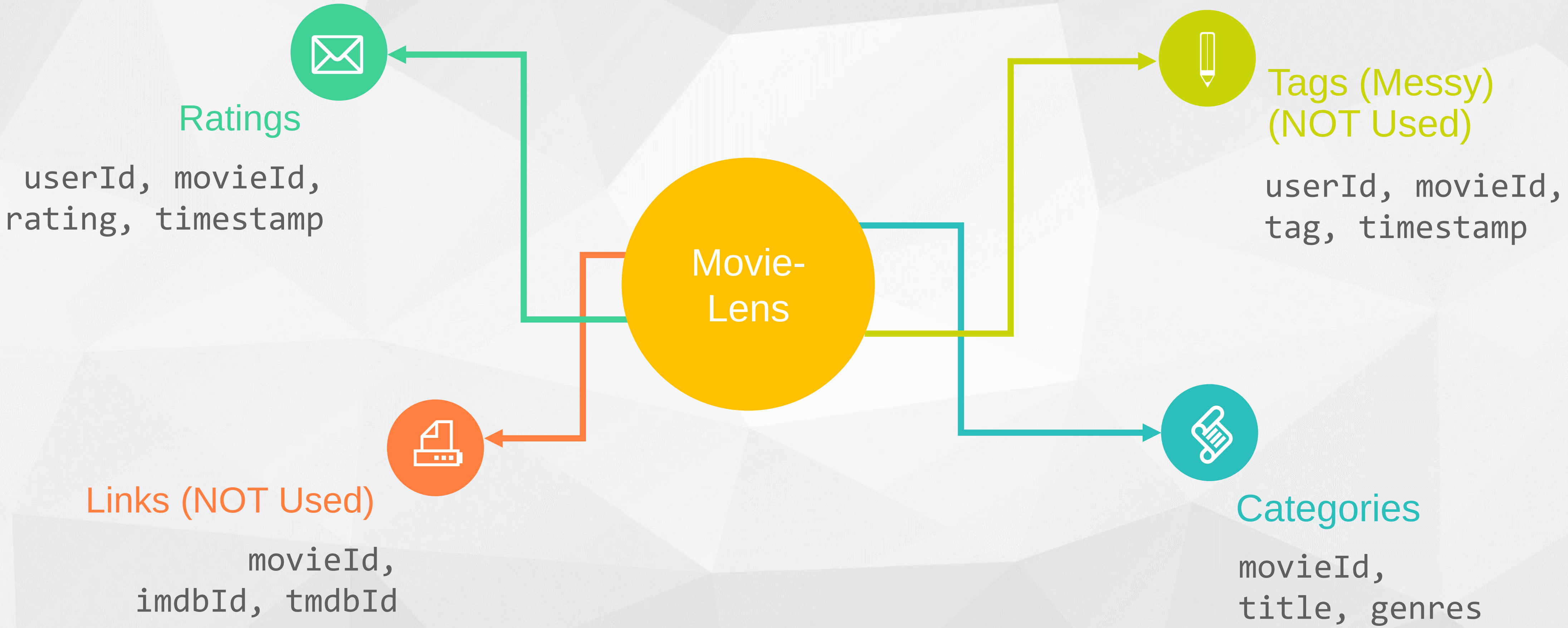
D Moive-User-Based

E Experiments

A

Introduction

Dataset - Overview



Dataset - Extraction

MovieLens (available)
- Ratings: `userId, movieId, rating, timestamp`
  [e.g.] `1, 1193, 5, 978300760)`
- Categories: `movieId, title, genres`
  [e.g.] `1, Toy Story (1995), Animation|Children's|Comedy`
- Not Used
  - tags data (messy): `userId, movieId, tag, timestamp`
  - links data: `movieId, imdbId, tmdbId`

1. Users set $\mathcal{U} = \{1,2,\cdots,u_i\}$, Movies Set $\mathcal{M} = \{1,2,\cdots,m_j\}$, Movies Categories Set $\mathcal{C} = \{1,2,\cdots,c_k\}$
2. Users' ratings logs $\boldsymbol{R} = (r_{u,m})$: user u's rating of movie m, nan if non-existent
3. Users' weighted normed movie ratings $\bar{r}_m$: rating of movie m, 0 if non-existent

$$\bar{r}_m = \frac{\sum_{u\in\mathcal{U}} r_{u,m} \cdot \left(\frac{\sum_{j\in\mathcal{M}}(r_{u,j} \neq nan)}{|r_{u,j} \neq nan|}\right)^{-1}}{\sum_{u\in\mathcal{U}} \left(\frac{\sum_{j\in\mathcal{M}}(r_{u,j} \neq nan)}{|r_{u,j} \neq nan|}\right)^{-1}}$$

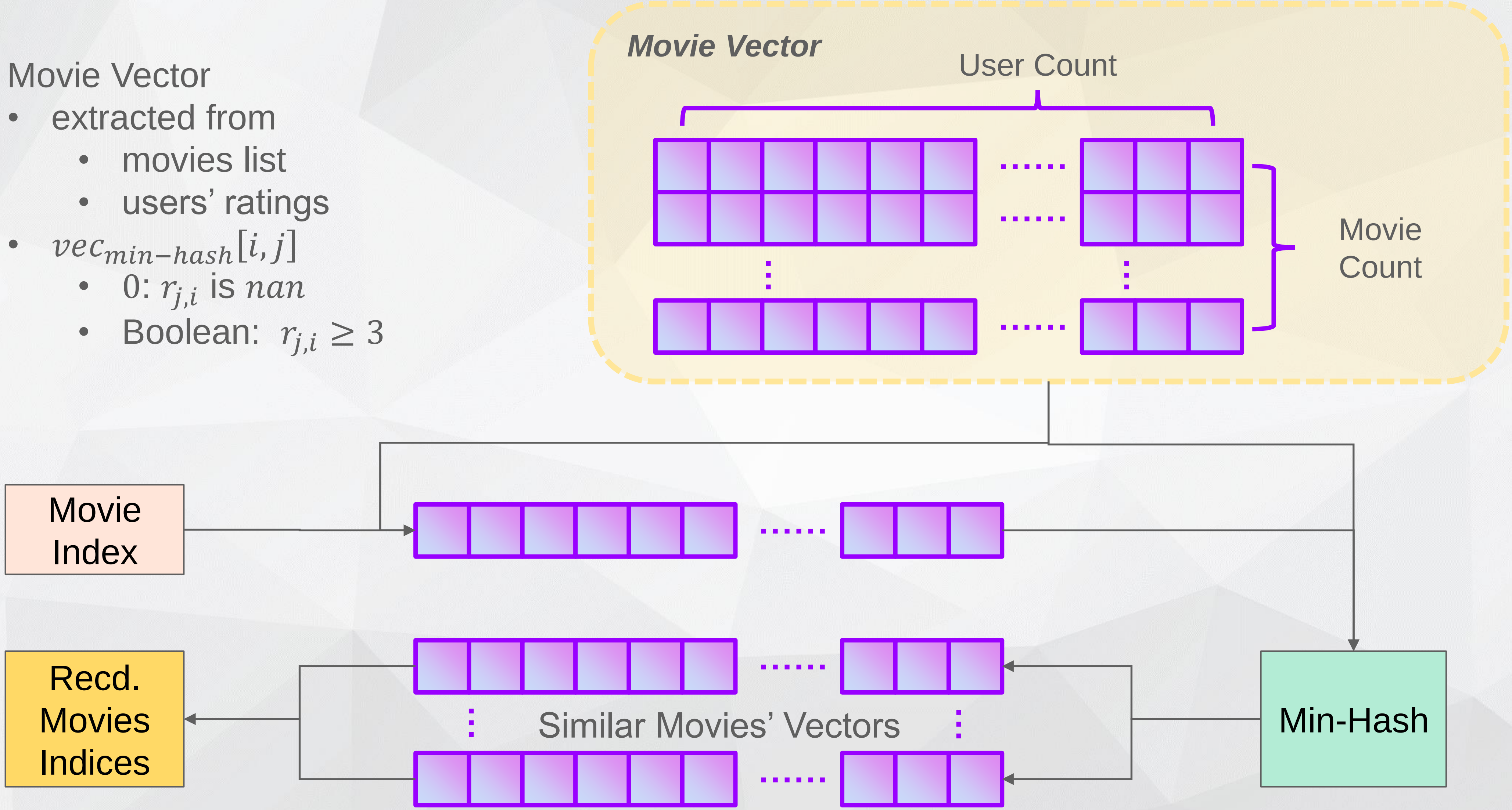
B

Movie-Based Recommendation

Min-Hash

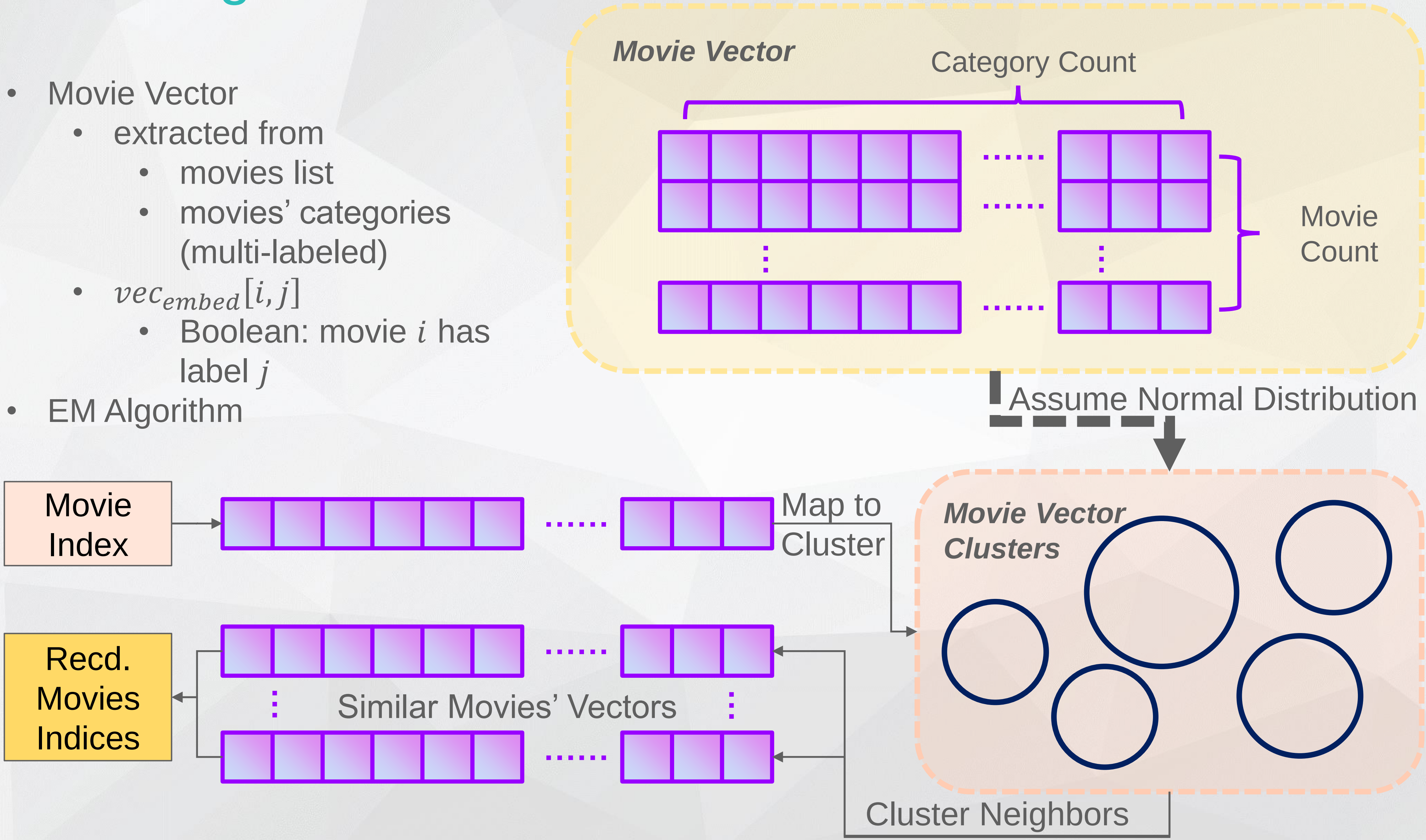
Movie Vector

- extracted from
  - movies list
  - users' ratings
- $vec_{min-hash}[i, j]$
  - 0: $r_{j,i}$ is nan
  - Boolean: $r_{j,i} \geq 3$



Clustering - Intuition

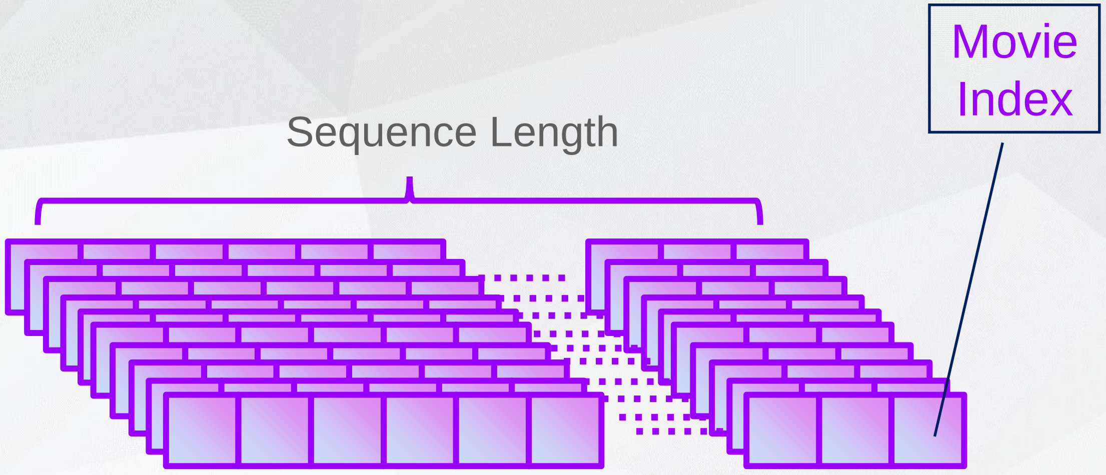
- Movie Vector
  - extracted from
    - movies list
    - movies' categories (multi-labeled)
  - $vec_{embed}[i, j]$
    - Boolean: movie i has label j
- EM Algorithm

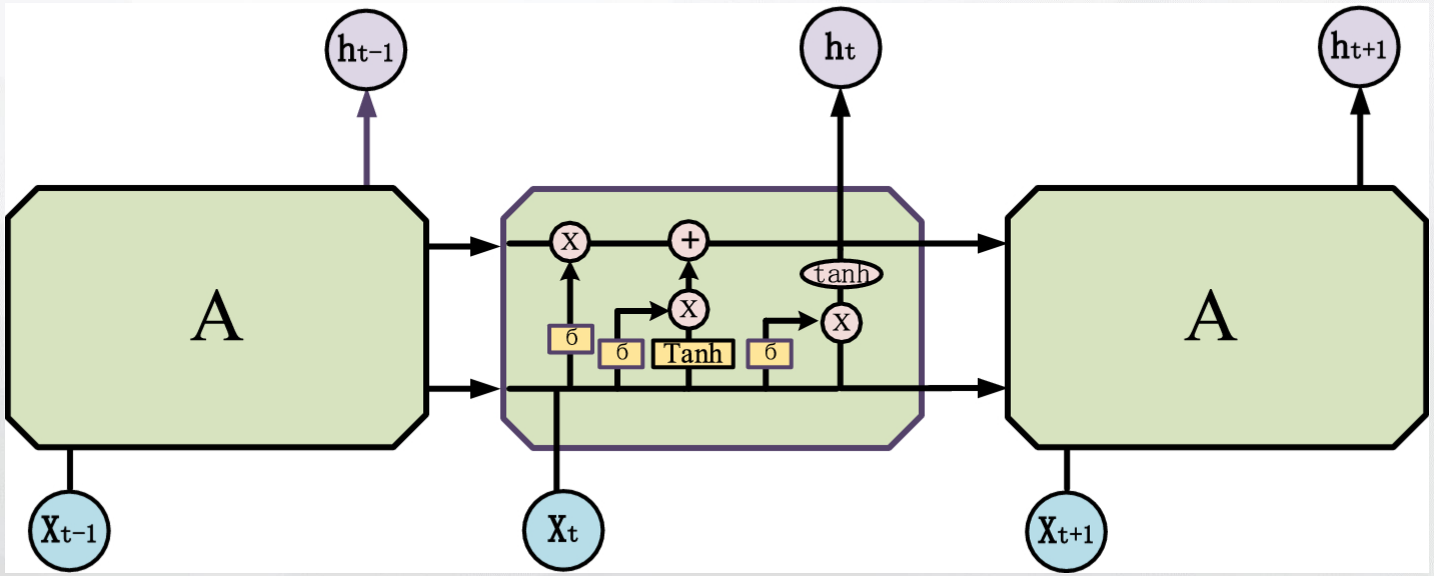


Time Sequence Analysis - Intuition

- Movie Sequence
  - extracted from
    users' ratings (as history)
  - each item of each sequence
    movie index

- LSTM
  - predict future based on the past history
  - analogy: e-shopping recommendation

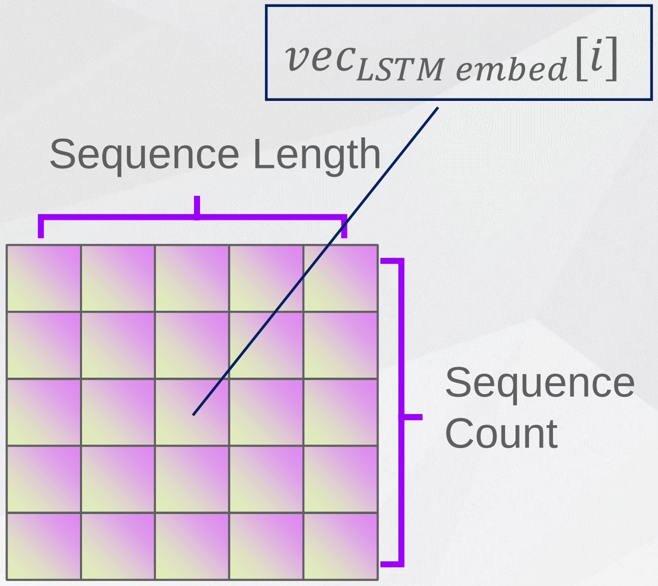


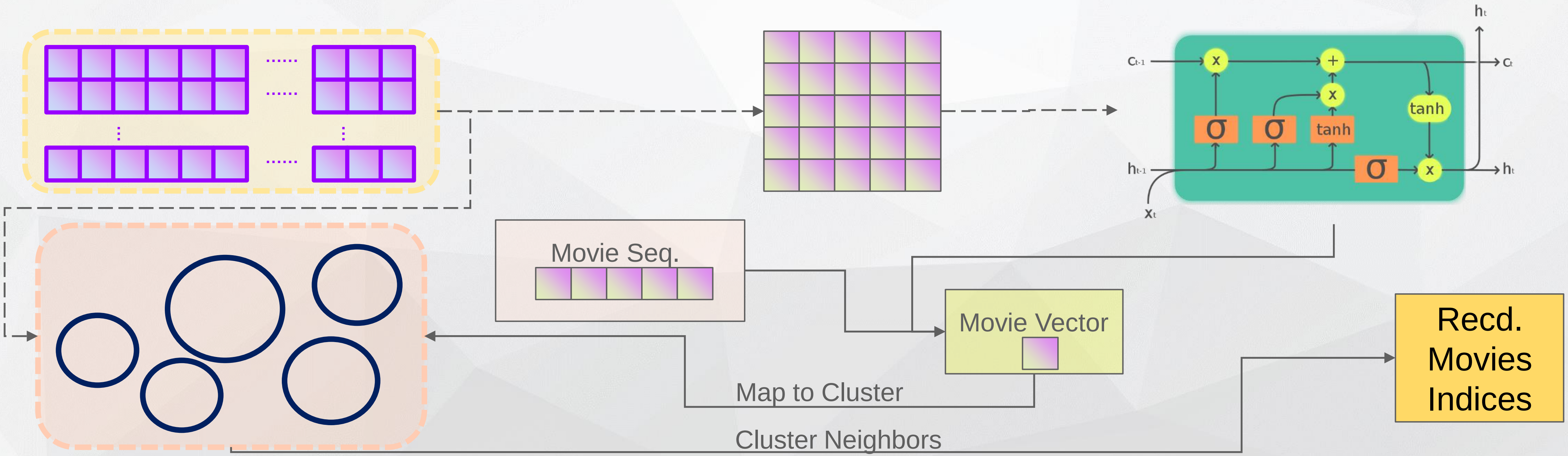


Hu, J., Wang, X., Zhang, Y. et al. Time Series Prediction Method Based on Variant LSTM Recurrent Neural Network. Neural Process Lett 52, 1485–1500 (2020).

Time Sequence Analysis + Clustering

- Movie Vector $vec_{embed}[i, j]$: considers categories only
- Improved Movie Vector
  - extracted from
    - movies list, movies' categories (multi-labeled)
    - **users' ratings** (weighted normed)
  - $vec_{LSTM\ embed}[i, j] = a \cdot \bar{r}_i$, shape $(movie\_cnt, category\_cnt)$
    - a (Boolean): movie i has label j
    - $\bar{r}_i$: users' rating of movie i



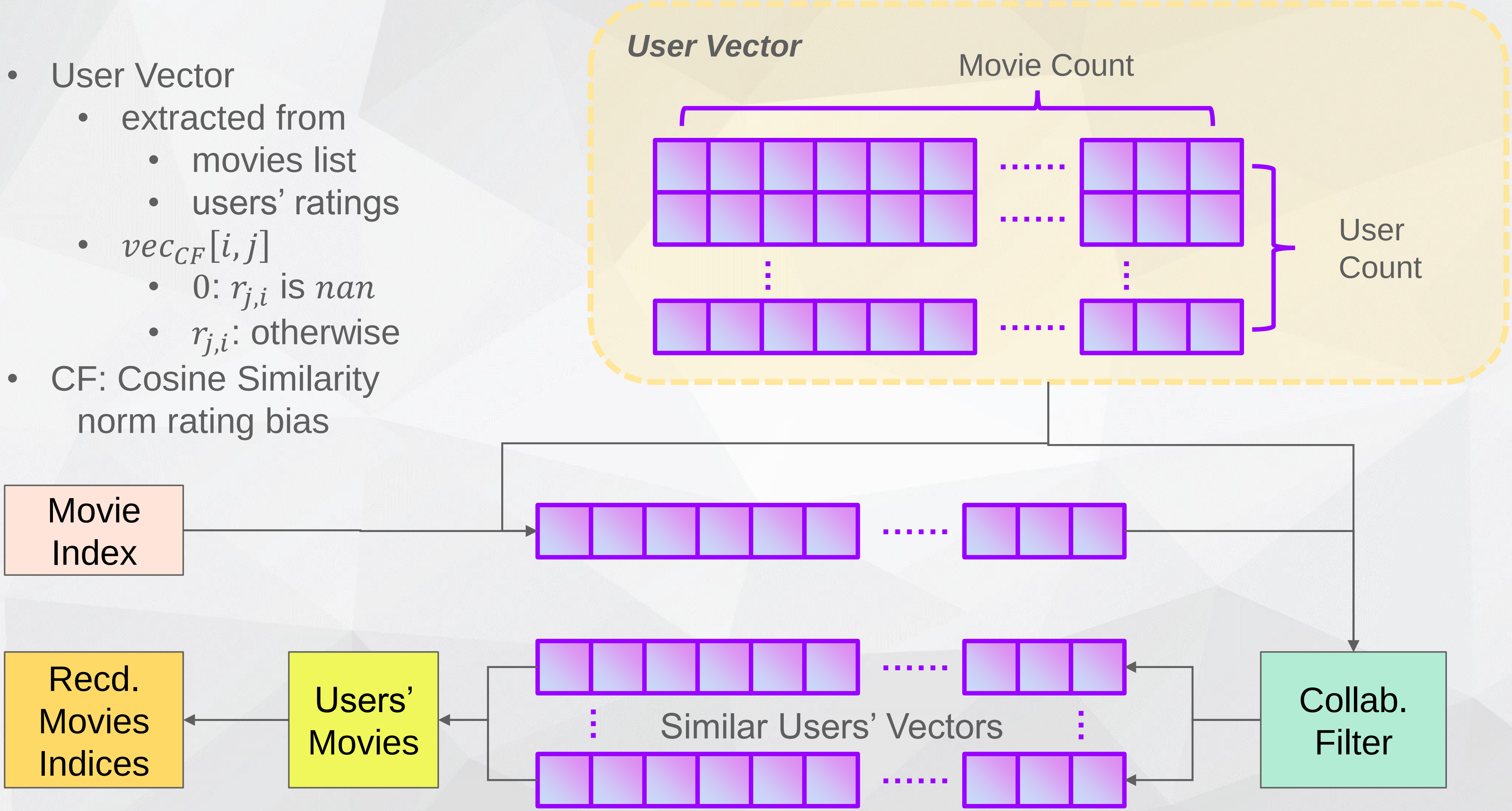


User-Based

Recommendation

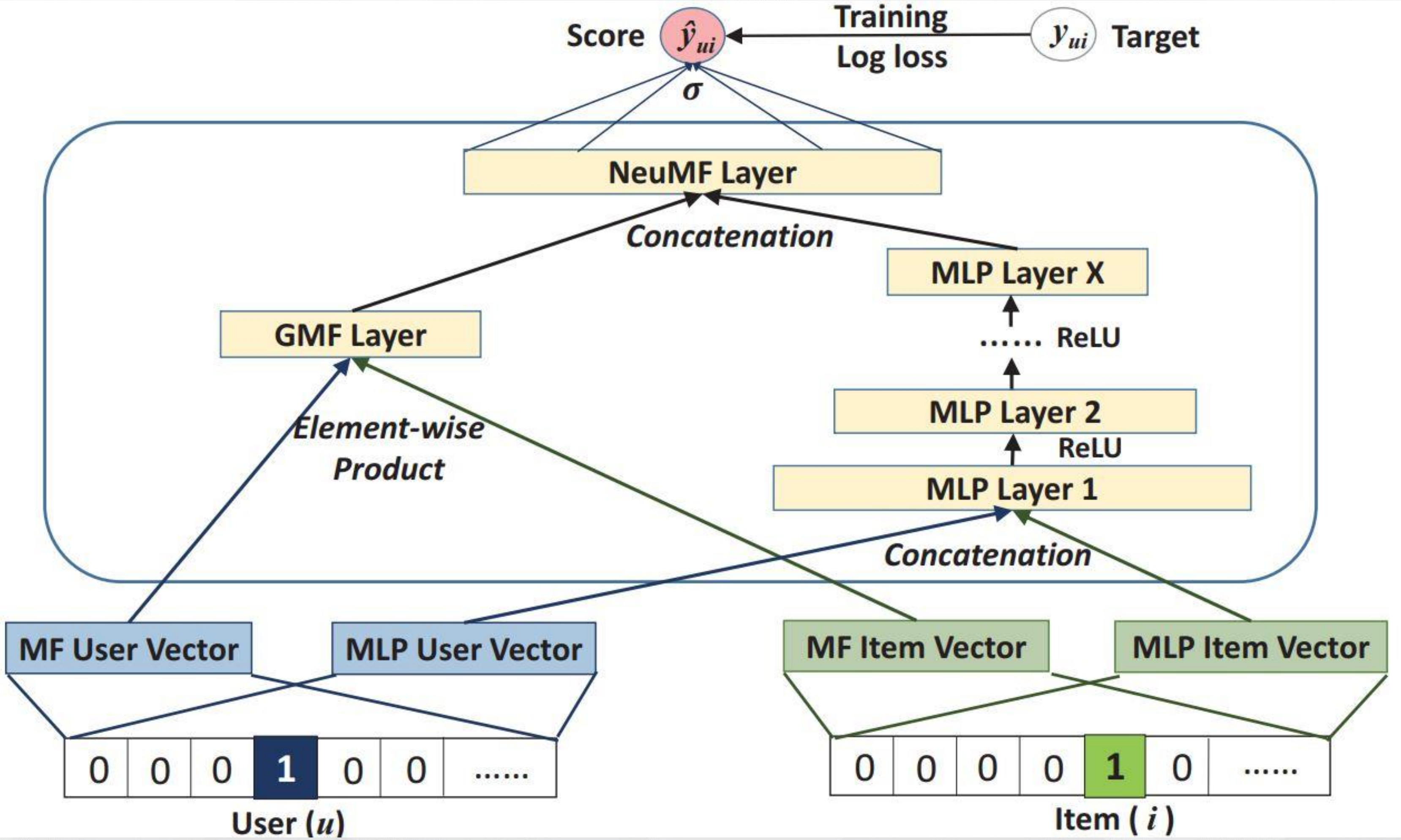
Collaborative Filtering

- User Vector
  - extracted from
    - movies list
    - users' ratings
  - $vec_{CF}[i,j]$
    - 0: $r_{j,i}$ is nan
    - $r_{j,i}$: otherwise
- CF: Cosine Similarity
  norm rating bias



Movie-User-Based Recommendation

Neural Collaborative Filtering (NCF)



Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In WWW '17. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 173–182.

Experiments

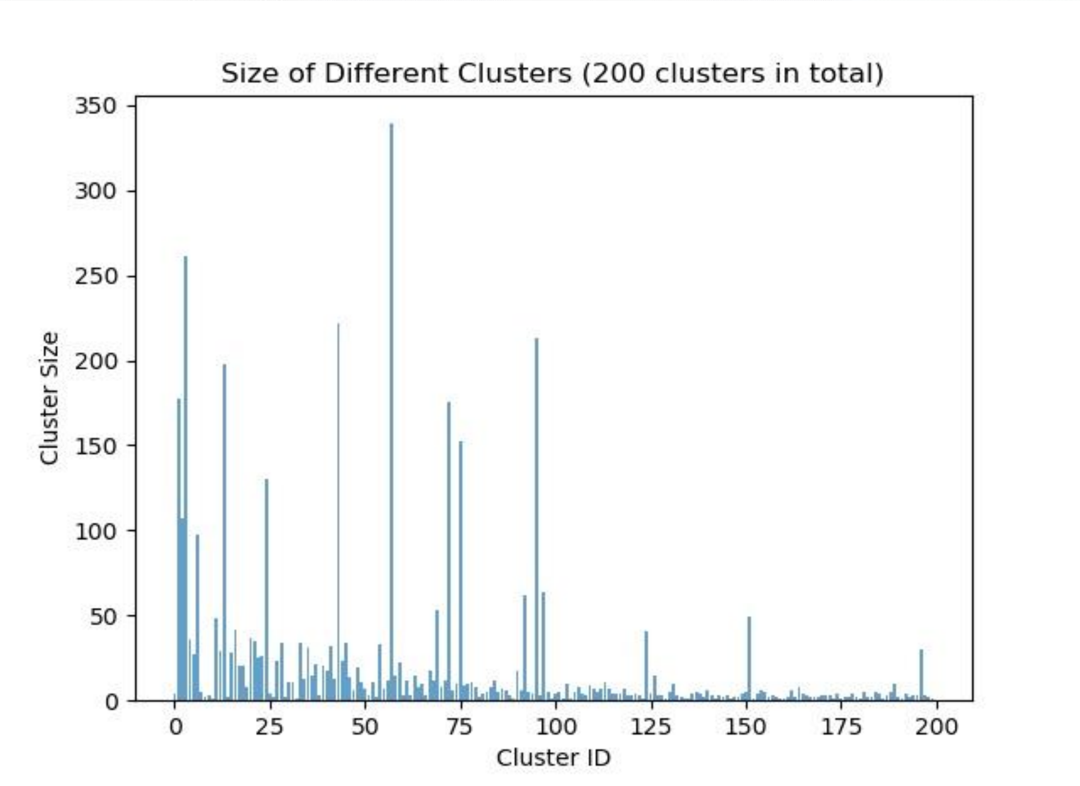
Experiment Results

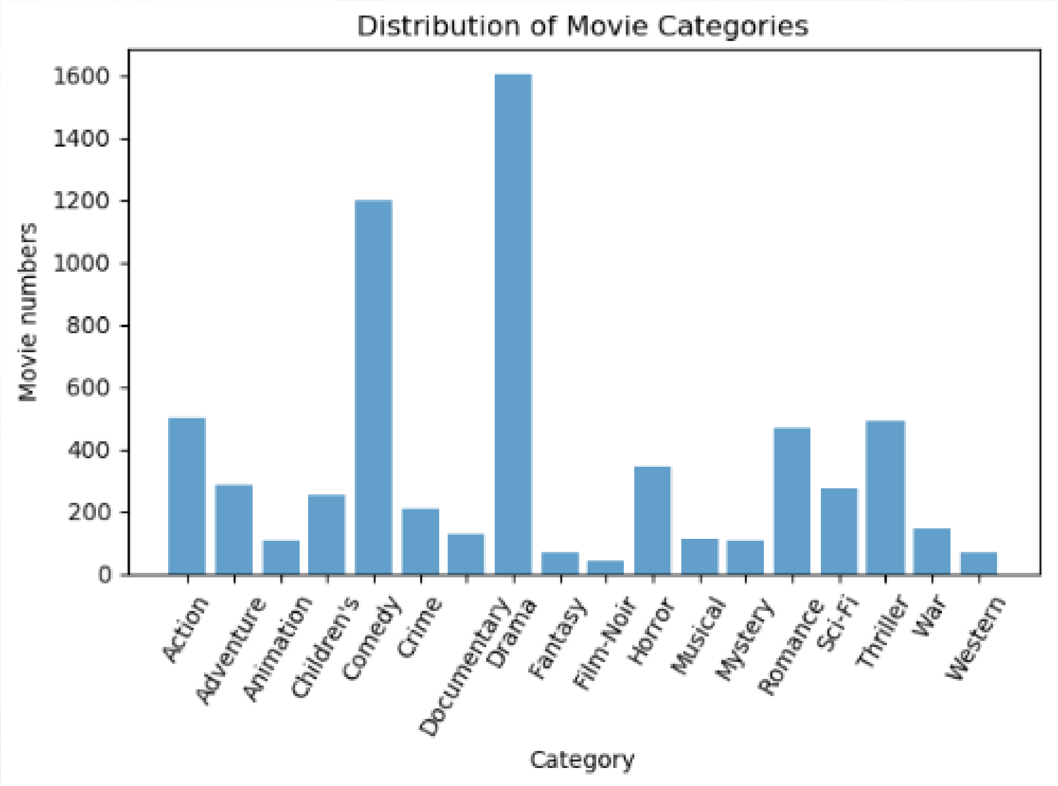
Approach	Accuracy@10 (%)	Recall@10 (%)
Random	0.2649	0.8775
Min-Hash	4.1225	8.0298
LSTM + Clustering	0.5629	1.5728
Collaborative Filtering	6.4901	16.6887
NCF	2.0695	0.5629

Analysis

- Clustering-Related
  - Biased cluster size

- NCF
  - Explicit feedback degrades to implicit feedback





THANKS