Presentation

Web Recommender Systems

Kasper Nicolaj Schiller, 04-04-2025

KØBENHAVNS UNIVERSITET

Agenda

- **Data**
- **Evaluation of models**
- **Discussion & Future Work**

Data

Musical Instruments dataset from Amazon Review 2023 (5-core)

Metric	**Metric**
Reviews	9913
Users	800
Items	509
Sparsity	97.6%
Rating distribution	$\mathcal{N}(4.54, 0.69)$

Table: Summary of the training split after cleaning.

Collaborative Filtering

- Compute predicted ratings of unseen user item pairs based on reviews in the training set
- KNN Baseline
- Singular Value Decomposition (SVD)

Fine-tuning

- 5-fold cross validation with MAE
- KNN Baseline: item-based approach with $k = 10$ and the MSE measure
- SVD: 50 latent factors and 20 epochs
- Search space table is shown in the paper.

Evaluation

- Performed on the test set
- Regression-based
- Rank-based

Regression-based

- KNN-Baseline RMSE: 1.068
- SVD RMSE: 0.992

Assume test split $\sim \mathcal{N}(4.54, \mathbf{0.69})$ as in the training split:

$$1.068 > 0.992 > \sqrt{\mathbf{0.69}} = 0.83$$

Better off guessing the mean according to RMSE. But then item ranks would be arbitrary.

Rank-based

Order items from the training set by predicted rating.
Binary ground truth vector: $r_{ui} \geq 3$, obtained from the test split

	Mean HR@10	Mean P@10	MAP@10	MRR@10	Coverage
TopPop	0.254	0.032	0.034	0.116	1.93%
KNN Baseline	0.092	0.010	0.010	0.035	63.9%
SVD	0.092	0.010	0.009	0.027	28.4%

- **SVD Problems**:
  - $\textit{epochs} \in \{10, 20, 30, 40, \mathbf{50}\, 60\} \Rightarrow$ risk of overfitting
  - $\textit{latent factors} \in \{5, 10, \mathbf{20}, 30, 40\} \Rightarrow$ too many categories
- **KNN Problems:**
  - Sparsity & Missing Not At Random Property
  - Coincidental Rating Commonality: 67.6% of users have $\hat{r}_{u,1} = \hat{r}_{u,20}$

A Content-Based Recommender System based on Word2Vec

- **metadata**: 23984 musical products with associated title & description
- Utilize users own ratings & product features to predict ratings
- **Word2Vec**: Pre-trained embeddings (300-dimensional from Google)
- **Preprocessing**: lowercasing $\rightarrow$ tokenizing $\rightarrow$ stopword removal

A Content-Based Recommender System based on Word2Vec

- We represent each user by a rating-weighted average of the items' Word2Vec embeddings.

$$sim(u, i) = \alpha \cdot cos(v_u^{title}, v_i^{title}) + (1 - \alpha) \cdot cos(v_u^{desc}, v_i^{desc})$$

with $\alpha = \frac{2}{3}$.

- Rank items for each user according to the similarity.

Evaluation

	Mean HR@10	Mean P@10	MAP@10	MRR@10	Coverage
TopPop	0.254	0.032	0.034	0.116	1.93%
KNN Baseline	0.092	0.010	0.010	0.035	63.9%
SVD	0.092	0.010	0.009	0.027	28.4%
Word2Vec Content-Based	0.096	0.011	0.008	0.035	40.0%

- Problems:
  - Empty description columns
  - OOV words
  - No domain information $\Rightarrow$ Consider TF-IDF?

Domain matters

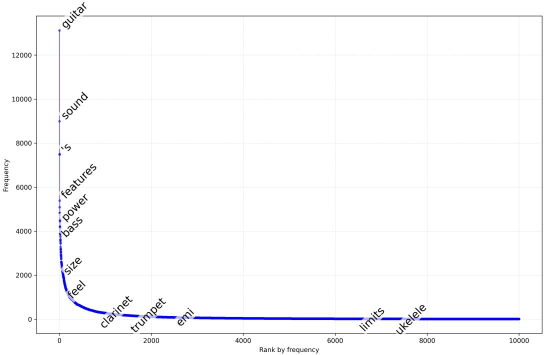

Figure: Term frequencies over descriptions of all items in the metadata.

Zipf's Law

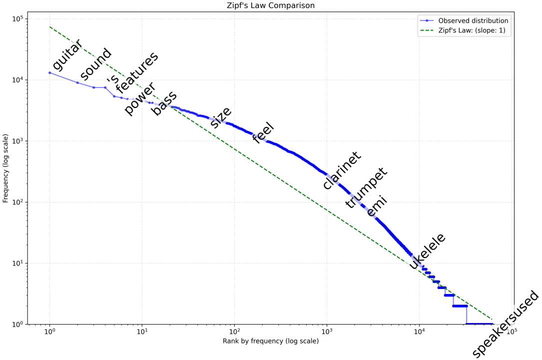

Figure: Term frequencies over descriptions of all items in the metadata.

Hybrid Recommender System

- Ensemble Hybrid Model:
  - Weighted sum of KNN Baseline and the content-based systems predicted ratings.
  - $\alpha = \frac{1}{3}$, $\beta = \frac{2}{3}$.

Evaluation

	Mean HR@10	Mean P@10	MAP@10	MRR@10	Coverage
TopPop	0.254	0.032	0.034	0.116	1.93%
KNN Baseline	0.092	0.010	0.010	0.035	63.9%
SVD	0.092	0.010	0.009	0.027	28.4%
Word2Vec Content-Based	0.096	0.011	0.008	0.035	40.0%
Parallel Hybrid	0.104	0.011	0.007	0.024	57.7%

- Finetune α and β
- Other evaluation measures

Llama-3.2-1B generated descriptions

- A model from Meta with 1.24 billion parameters, optimized for summarization tasks.
- Prompt:

Generate a detailed and accurate description for the following musical instrument: {item title}.

- Maximum amount of tokens: 50
- No fine-tuning.

Llama-3.2-1B generated descriptions

- Word2Vec embeddings: this time only with the description.
- Preprocessing: *lowercasing* $\rightarrow$ *tokenizing* $\rightarrow$ *stopword removal*
- Each user is represented by a rating-weighted average of Word2Vec embeddings.
- Compute cosine similarity between unseen user and item pairs
- Rank according to similarity

Evaluation

	Mean HR@10	Mean P@10	MAP@10	MRR@10	Coverage
TopPop	0.254	0.032	0.034	0.116	1.93%
KNN Baseline	0.092	0.010	0.010	0.035	63.9%
SVD	0.092	0.010	0.009	0.027	28.4%
Word2Vec Content-Based	0.096	0.011	0.008	0.035	40.0%
Parallel Hybrid	0.104	0.011	0.007	0.024	57.7%
Llama3.2 Content-Based	0.106	0.011	0.011	0.037	40.3%

- No empty descriptions.
- No typos in LLMs $\Rightarrow$ less OOV words
- In general:
  - Low performance
  - Data and model-specific issues
  - Binary metrics

Future Work

- SVD with less epochs and latent factors
- KNN: Assign default values to items with low amount of reviews to solve CRC
- Content-based: Consider TF-IDF
- Fine-tune models with nDCG
- Word2Vec Session-based models: Utilise user ids and time stamps $\Rightarrow$ drop assumptions
- User-item graphs: generate neighborhoods with Katz' Measure or Personalised PageRank
- Switching strategy: use a content-based model for cold-start users, else CF