Recommender systems via Covariance Neural Networks

Using precision matrices as Graph Collaborative Filter

Vic Vansteelant

Supervisors: Dr. Elvin Isufi, Andrea Cavallo, Chengen Liu

June 23, 2025

1 Introduction

- **Recommender systems** are everywhere and personalize content by predicting user preferences for unseen items.
- This research focuses on **collaborative filtering**, which uses past interactions from users to predict unknown ratings.
- **Graph Neural Networks (GNNs)** learn node embeddings by aggregating neighborhood features, making them effective for capturing relational patterns between users.
- **coVariance Neural Networks (VNNs)** use covariance matrices as graph structures to encode statistical dependencies [3].
- We adapt on this by using the **precision matrix**, which is the inverse of the covariance matrix, instead. This tries to capture **conditional independence** between users.
- Limited research into using these techniques for movie recommendation. This research addresses this gap.

2 Methodology

- **Model input**:
  - A leave-one-out strategy is used for the training procedure.
  - Mini-batching is used to improve efficiency while retaining learning precision.
  - The entire rating matrix is fed into the model, which predicts all ratings and is then compared to the test ratings.
- **Model Architecture**:
  - Built on the GNN framework from Sihag et al. [3].
  - Composed of multiple GNN layers followed by a multilayer perceptron (MLP), which reduces the data to a scalar rating.
  - Each GNN layer aggregates multi-hop neighborhood features using the precision matrix.
- **Graph Construction**:
  - A regularized covariance matrix is computed from the normalized training data. This matrix is then inverted to get the **precision matrix**.
  - This precision matrix is used as the **Graph Shift Operator (GSO)** in the GNN.

$$\mathrm{H}^{(l+1)} = \sigma\left(\sum_{k=0}^{K-1} \tilde{\mathbf{A}}^k \mathbf{H}^{(l)} \mathbf{W}_k^{(l)}\right)$$

TU Delft Delft University of Technology

3 Experimental Setup

Evaluation Metric

- **Mean Absolute Error (MAE):** Easier for humans to interpret, so used for early experimenting and plotting losses.
- **Root Mean Squared Error (RMSE):** Standard benchmark metric which will be used for reporting results.

Dataset & Preprocessing

- **Dataset:** MovieLens-100k, which has ratings for unique user-movie pairs ranging between 1 and 5.
- **Split:** 80% train, 10% val, 10% test (fixed seeding).
- **Normalization:** User-wise z-score normalization.
- **Precision Matrix:** $P = \left(\frac{ZZ^\top}{C-1} + \epsilon I\right)^{-1}, \quad C = \max(MM^\top, 2)$

Hyperparameters

Tuned by grid search via `ParameterGrid` (scikit-learn).

Learning rate: 0.002	`dimNodeSignals`: [1, 32, 64]
Batch size: 256	`nFilterTaps`: [4, 4]
Loss function: MSE	`dimLayersMLP`: [64, 32, 16, 1]

4 Results and Discussion

Evaluated on MovieLens-100k with standard 80/20 train-test split.

Model	RMSE	Ref
MC (2009)	0.973	[1]
GMC (2014)	0.996	[2]
GHRS (2022)	0.887	[4]
MoRGH (2022)	0.881	[5]
VNN (this work)	**0.9496**	–

Table: Benchmark results on MovieLens-100k (lower RMSE is better)

- The RMSE of 0.95 shows the ability to learn meaningful patterns from user-item pairs and ratings.
- Baselines: mean prediction ~1.2 RMSE; user-mean ~1.1 RMSE.
- Although not outperforming state-of-the-art(GHRS, MoRGH), VNN beats classical benchmarks (MC, GMC).

Prediction Deviation Analysis

- Errors analyzed by rounding predictions to nearest integer.
- Distribution of deviations (Δ = predicted rating - true rating):

Exact match ($\Delta = 0$): 46.5%	Off by 2 ratings: 9.7%
Off by 1 rating: 41.8%	Off by 3 or more: 2.0%

- Heatmap (Fig. 1) shows systematic biases:
  - Bias towards mean rating (around 3/4).
  - Almost never predicts low ratings (1/2).
  - Detects trends but struggles with extreme values.

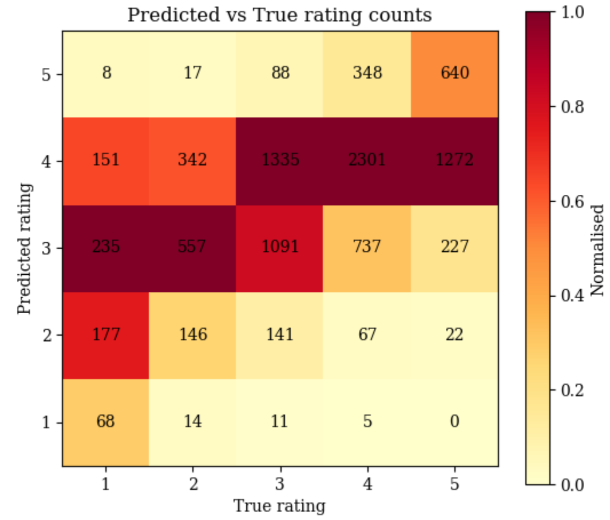

Figure: Heatmap of true vs. predicted ratings

5 Future Work

- **Output calibration:** Post-processing to calibrate the output distribution to better predict extreme ratings.
- **Larger datasets:** Test on datasets like MovieLens 1M or 10M for scalability, generalization, and robustness.
- **Sparsification:** Explore matrix sparsification to reduce computation while preserving structure.

References

[1] E. J. Candes and B. Recht. "Exact Matrix Completion via Convex Optimization". In: (2008).
[2] V. Kalofolias, X. Bresson, M. Bronstein, and P. Vandergheynst. "Matrix Completion on Graphs". In: (2014).
[3] S. Sihag, G. Mateos, C. McMillan, and A. Ribeiro. "Covariance Neural Networks". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 17003–17016.
[4] Z. Zamanzadeh Darban and M. H. Valipour. "GHRS: Graph-based hybrid recommendation system with application to movie recommendation". In: *Expert Systems with Applications* (2022).
[5] S. Ziaee, H. Rahmani, and M. Nazari. "MoRGH: Movie Recommender System using GNNs on Heterogeneous Graphs". In: (2024). DOI: 10.21203/rs.3.rs-3860094/v1.