Progress Evaluation Report

ContentGCN: Enhancing LightGCN with Content-Based Filtering for Music Recommendation

Advanced Machine Learning (CS4681)

Kahapola K.V., Student ID: 210266G

2025/08/24

Contents

1 Introduction

This report outlines the initial progress for the ContentGCN project, which aims to enhance the state-of-the-art LightGCN model for music recommendation systems by incorporating content-based filtering alongside collaborative filtering. The project addresses a fundamental limitation in current graph-based recommendation systems: their reliance solely on user-item interaction data, which leads to challenges with cold-start problems and limited diversity in recommendations.

LightGCN, developed by He et al. (2020) [5], represents the current state-of-the-art in graph convolution networks for collaborative filtering. It simplifies traditional Graph Convolutional Networks (GCNs) by removing feature transformation and nonlinear activation, focusing purely on neighborhood aggregation for learning user and item embeddings. While LightGCN achieves superior performance on collaborative filtering tasks, it cannot leverage rich content information such as music genres, artist information, audio features, and user demographics.

ContentGCN proposes to bridge this gap by integrating content-based features into the LightGCN architecture, creating a hybrid recommendation system that combines the strengths of collaborative and content-based filtering. This enhancement aims to improve recommendation quality, particularly for new users and items with limited interaction history, while maintaining the computational efficiency that makes LightGCN attractive for large-scale applications.

The project will be evaluated on the Last.fm music recommendation dataset, comparing against the baseline LightGCN model and other state-of-the-art recommendation systems. The goal is to achieve measurable improvements in recommendation accuracy, diversity, and cold-start performance while producing a conference-ready research paper.

2 Literature Review

2.1 Theoretical Background on Graph Convolution Networks for Recommendation

Graph Convolution Networks have emerged as powerful tools for recommendation systems by modeling user-item interactions as bipartite graphs. The fundamental idea is to learn user and item representations through neighborhood aggregation, where each node's representation is refined by aggregating information from its connected neighbors.

Traditional GCNs for recommendation, such as NGCF [11], apply feature transformation and nonlinear activation at each layer:

$$\mathbf{e}_u^{(l+1)} = \sigma\left(\mathbf{W}_1^{(l)}\mathbf{e}_u^{(l)} + \sum_{i\in\mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_i|}}\mathbf{W}_2^{(l)}\mathbf{e}_i^{(l)}\right)$$

LightGCN simplifies this formulation by removing the feature transformation matrices $\mathbf{W}_1$ and $\mathbf{W}_2$, and the nonlinear activation σ:

$$\mathbf{e}_u^{(l+1)} = \sum_{i\in\mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_i|}}\mathbf{e}_i^{(l)}$$

This simplification reduces overfitting and improves computational efficiency while maintaining or improving recommendation performance [5].

2.2 Taxonomy of Recommendation System Approaches

Modern recommendation systems can be categorized into several paradigms:

- **Collaborative Filtering**: Matrix factorization [7] and neural collaborative filtering [4] methods that rely solely on user-item interaction patterns.
- **Content-Based Filtering**: Systems that recommend items based on item features and user preferences [8]. Effective for cold-start scenarios but limited by feature engineering requirements.
- **Hybrid Systems**: Combinations of collaborative and content-based approaches [1]. These systems aim to leverage the strengths of both paradigms while mitigating their individual weaknesses.
- **Deep Learning Approaches**: Neural networks for recommendation including autoencoders [10], recurrent networks [6], and attention mechanisms [2].
- **Graph-Based Methods**: Recent advances including GraphSAGE [3], NGCF [11], and LightGCN [5] that model recommendations as graph learning problems.

This project focuses on enhancing graph-based collaborative filtering with content-based features, representing a hybrid approach within the graph neural network paradigm.

2.3 Comparative Analysis of Key Papers

- **He et al. (2020)** [5]: Introduces LightGCN, the baseline model for this project. Demonstrates that simplifying GCN architecture can improve recommendation performance and computational efficiency. Their method achieves state-of-the-art results on multiple datasets but lacks content-based capabilities.

- **Wang et al. (2019)** [11]: Proposes Neural Graph Collaborative Filtering (NGCF), which applies message passing on user-item bipartite graphs. While innovative, NGCF suffers from over-smoothing and computational complexity issues that LightGCN addresses.

- **Koren et al. (2009)** [7]: Classical matrix factorization approach that forms the foundation of collaborative filtering. Limited by its inability to capture complex user-item relationships and lack of content integration.

- **He et al. (2017)** [4]: Neural Collaborative Filtering (NCF) demonstrates the effectiveness of deep learning for recommendation systems. Provides insights into neural architectures that can be adapted for graph-based systems.

- **Hamilton et al. (2017)** [3]: GraphSAGE introduces inductive learning on graphs, enabling generalization to unseen nodes. Relevant for handling new users and items in recommendation systems.

- **Burke (2002)** [1]: Comprehensive survey of hybrid recommendation approaches, providing theoretical foundation for combining collaborative and content-based filtering.

- **Lops et al. (2011)** [8]: Overview of content-based recommendation techniques, informing the feature integration strategy for this project.

2.4 Research Gap and Contribution

While LightGCN achieves excellent performance in collaborative filtering scenarios, it cannot leverage content information that could improve recommendations, especially for cold-start users and items. Existing hybrid approaches either sacrifice the efficiency of LightGCN or fail to fully integrate content and collaborative signals in a principled manner.

This project aims to bridge this gap by developing ContentGCN, which extends LightGCN to incorporate content-based features while maintaining its computational efficiency and simplic-

ity. The contribution includes novel feature integration mechanisms, handling of heterogeneous content types in music recommendation, and comprehensive evaluation on cold-start scenarios.

3 Proposed Methodology

3.1 Content Feature Integration Framework

The core innovation involves extending LightGCN's graph convolution to incorporate content-based features for both users and items. The proposed ContentGCN modifies the neighborhood aggregation to include content-aware weighting:

$$\mathbf{e}_u^{(l+1)} = \sum_{i \in \mathcal{N}_u} \alpha_{ui} \cdot \frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_i|}} \left(\mathbf{e}_i^{(l)} + \beta \cdot \mathbf{f}_i^{content} \right)$$

where α_{ui} represents content-based attention weights computed from user-item content similarity, $\mathbf{f}_i^{content}$ denotes the content feature vector for item i, and β is a learnable parameter controlling the contribution of content features.

3.2 Multi-Modal Content Feature Engineering

For music recommendation on Last.fm, the following content features will be integrated:

Item (Music Track) Features:

- **Genre Information**: Multi-hot encoding of music genres
- **Artist Features**: Artist popularity, activity period, and genre affiliation
- **Audio Features**: When available, features like tempo, key, danceability from audio analysis
- **Temporal Features**: Release year, trending patterns

User Features:

- **Demographic Information**: Age group, country (when available)
- **Listening Patterns**: Activity level, diversity of preferences, temporal listening patterns
- **Genre Preferences**: Inferred genre preferences from listening history

3.3 Hybrid Loss Function

A composite loss function balances collaborative filtering objectives with content-based regularization:

$$\mathcal{L} = \mathcal{L}_{BPR} + \lambda_1 \mathcal{L}_{content} + \lambda_2 \mathcal{L}_{reg}$$

where:

- $\mathcal{L}_{BPR}$ is the Bayesian Personalized Ranking loss used in LightGCN [9]
- $\mathcal{L}_{content}$ encourages similar content features to have similar embeddings
- $\mathcal{L}_{reg}$ provides L2 regularization to prevent overfitting

3.4 Implementation Strategy

- **Base Architecture**: Extend the existing LightGCN implementation from the official repository
- **Content Integration**: Design feature fusion mechanisms that preserve LightGCN's simplicity
- **Attention Mechanism**: Implement content-based attention for weighted neighborhood aggregation
- **Scalable Design**: Ensure computational efficiency for large-scale datasets

3.5 Dataset and Evaluation Framework

Dataset:

- **Last.fm Dataset**: User-artist listening data with timestamps and play counts
- **Content Augmentation**: Artist genre information, user demographics, and temporal features
- **Data Preprocessing**: Handle implicit feedback, filter inactive users/artists, and normalize features

Evaluation Metrics:

- **Ranking Metrics**: NDCG@K, Recall@K for different values of K (10, 20, 50)

- **Diversity Metrics**: Intra-list diversity, catalog coverage to assess recommendation diversity
- **Cold-Start Performance**: Separate evaluation on users/items with limited interaction history
- **Computational Efficiency**: Training time, inference speed, and memory usage

Baseline Comparisons:

- Original LightGCN implementation
- Neural Collaborative Filtering (NCF)
- Neural Graph Collaborative Filtering (NGCF)
- Content-based baselines using feature-based similarity
- Hybrid approaches combining matrix factorization with content features

4 Project Timeline

The project timeline is illustrated in Figure 1, highlighting milestones for literature review, implementation, evaluation, and documentation over a 12-week period.

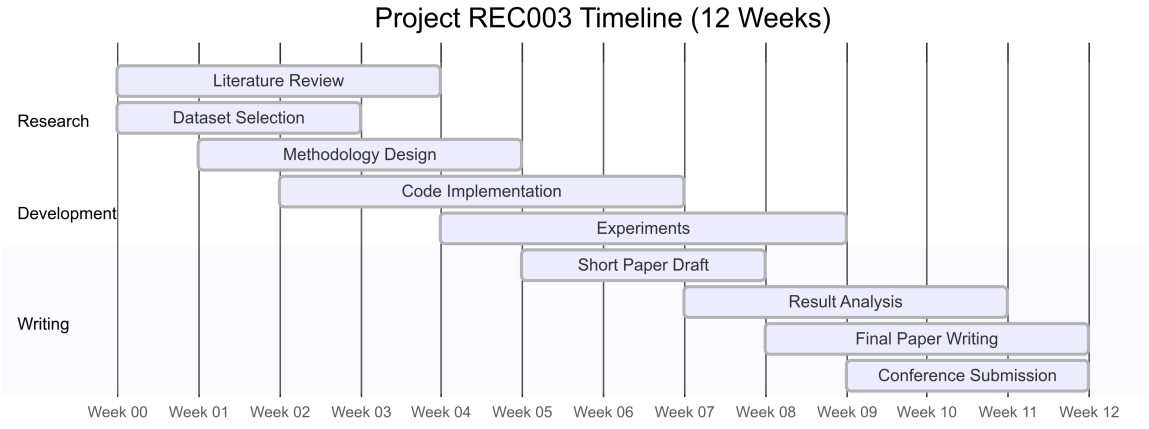

Figure 1: 12-week project timeline outlining key milestones: literature review, implementation, evaluation, and documentation.

Weeks 1-2: Foundation and Setup

- Literature review completion (In Progress)

- LightGCN baseline implementation and understanding
- Last.fm dataset acquisition and preprocessing
- Development environment setup

Weeks 3-4: Content Feature Engineering

- Music content feature extraction and preprocessing
- User profile feature engineering
- Feature integration pipeline development
- Initial content-based baseline implementation

Weeks 5-6: ContentGCN Development

- Core ContentGCN architecture implementation
- Content-aware attention mechanism development
- Hybrid loss function integration
- Initial model training and validation

Weeks 7-8: Comprehensive Evaluation

- Full experimental suite execution
- Baseline comparison studies
- Cold-start performance evaluation
- Ablation studies and hyperparameter optimization

Weeks 9-10: Analysis and Refinement

- Performance analysis and interpretation
- Model refinement based on evaluation results
- Diversity and fairness analysis
- Computational efficiency optimization

Weeks 11-12: Documentation and Submission

- Research paper writing and revision
- Code documentation and repository organization
- Conference submission preparation
- Final presentation materials

5 Resources

Software and Frameworks:

- Python, PyTorch for deep learning implementation
- PyTorch Geometric for graph neural network components
- Scikit-learn for content feature processing and baseline methods
- Pandas, NumPy for data manipulation and analysis
- Matplotlib, Seaborn for visualization and result analysis

Computational Resources:

- University GPU cluster for training large-scale models
- Local development environment for prototyping and testing
- Cloud computing resources if additional computational power needed

Datasets and Code:

- Last.fm dataset from official sources or research repositories
- LightGCN official implementation: kuandeng/LightGCN repository
- Additional music metadata from MusicBrainz or similar sources
- Evaluation metrics implementations from RecBole or similar frameworks

6 Challenges and Mitigation

Technical Challenges:

- **Feature Integration Complexity**: Combining heterogeneous content features with graph embeddings without losing LightGCN's simplicity. *Mitigation*: Design modular feature integration components, extensive ablation studies to identify optimal integration strategies.

- **Scalability Concerns**: Maintaining computational efficiency when adding content features to large-scale graphs. *Mitigation*: Implement efficient attention mechanisms, feature dimensionality reduction, and batch processing optimizations.

- **Cold-Start Evaluation**: Properly evaluating cold-start performance requires careful experimental design. *Mitigation*: Design specific cold-start evaluation protocols, use temporal splits for realistic evaluation scenarios.

Data Challenges:

- **Content Feature Quality**: Last.fm content features may be sparse or noisy. *Mitigation*: Implement robust feature preprocessing, explore external content sources, design handling mechanisms for missing features.

- **Dataset Preprocessing**: Handling implicit feedback, data sparsity, and temporal dynamics. *Mitigation*: Follow established preprocessing protocols, implement multiple data filtering strategies, validate preprocessing choices.

Evaluation Challenges:

- **Fair Baseline Comparison**: Ensuring fair comparison across different methodological approaches. *Mitigation*: Implement multiple baseline methods, use standardized evaluation frameworks, report confidence intervals and statistical significance.

- **Hyperparameter Sensitivity**: ContentGCN introduces additional hyperparameters that may affect performance. *Mitigation*: Systematic hyperparameter search, sensitivity analysis, robust default parameter selection.

7 Current Progress and Expected Outcomes

Current Progress:

- Project selection and baseline model identification completed
- Initial literature review of graph-based recommendation systems
- In Progress: LightGCN paper analysis and implementation understanding
- In Progress: Last.fm dataset exploration and preprocessing pipeline design

Expected Outcomes:

- **Technical Contribution**: Novel architecture combining LightGCN efficiency with content-based filtering capabilities
- **Empirical Results**: Demonstrated improvements in recommendation accuracy, especially for cold-start scenarios
- **Practical Impact**: Scalable hybrid recommendation system applicable to music and other domains
- **Academic Output**: Conference-ready paper with comprehensive experimental evaluation

Success Criteria:

- Achieve statistically significant improvements over LightGCN baseline in overall recommendation metrics
- Demonstrate substantial improvements in cold-start user/item scenarios
- Maintain computational efficiency comparable to original LightGCN
- Produce reproducible implementation with comprehensive documentation

8 Conclusion

This progress report establishes a clear research direction for enhancing LightGCN with content-based filtering capabilities to create ContentGCN, a hybrid recommendation system specifically designed for music recommendation tasks. The project addresses a significant limitation of

current graph-based collaborative filtering methods while building on the proven efficiency and effectiveness of the LightGCN architecture.

The proposed methodology combines theoretical soundness with practical applicability, incorporating multi-modal content features through attention-based mechanisms while preserving the computational advantages that make LightGCN attractive for large-scale applications. The comprehensive evaluation framework will enable rigorous validation against established baselines and specific assessment of cold-start performance improvements.

The 12-week timeline provides adequate scope for implementation, experimentation, and documentation while maintaining realistic expectations for the level of contribution appropriate for this course assignment. The project is well-positioned to achieve measurable improvements over the LightGCN baseline and produce a conference-ready research paper documenting this novel hybrid approach to graph-based recommendation systems.

The next phase will focus on LightGCN baseline implementation, content feature engineering, and initial ContentGCN prototype development, with the goal of demonstrating the viability and effectiveness of content-enhanced graph convolution for music recommendation tasks.

References

[1] Burke, R. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12(4):331–370, 2002.

[2] Chen, J., Zhang, H., He, X., Nie, L., Liu, W., and Chua, T.-S. Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 335–344, 2017.

[3] Hamilton, W., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1024–1034, 2017.

[4] He, X., Liao, L., Zhang, H., Nie, L., Hu, X., and Chua, T.-S. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, pages 173–182, 2017.

[5] He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., and Wang, M. Lightgcn: Simplifying and

powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 639–648, 2020.

[6] Hidasi, B., Karatzoglou, A., Baltrunas, L., and Tikk, D. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*, 2015.

[7] Koren, Y., Bell, R., and Volinsky, C. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[8] Lops, P., De Gemmis, M., and Semeraro, G. Content-based recommender systems: State of the art and trends. In *Recommender systems handbook*, pages 73–105. Springer, 2011.

[9] Rendle, S., Freudenthaler, C., Gantner, Z., and Schmidt-Thieme, L. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 452–461, 2009.

[10] Sedhain, S., Menon, A. K., Sanner, S., and Xie, L. Autorec: Autoencoders meet collaborative filtering. In *Proceedings of the 24th International Conference on World Wide Web*, pages 111–112, 2015.

[11] Wang, X., He, X., Wang, M., Feng, F., and Chua, T.-S. Neural graph collaborative filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 165–174, 2019.