Adaptive Recommendation System using Continual Learning

Revati Rajesh Pawar
rpawar@usc.edu

Samarth Saxena
saxenasa@usc.edu

Snigdha Srikanth Chenjeri
chenjeri@usc.edu

Abstract

This paper presents an adaptive recommendation system designed to address the challenges of concept drift and covariate shift in dynamic e-commerce environments. The system leverages continual learning techniques to predict the next item a user will interact with during a session, ensuring that recommendations remain relevant as user behavior and product trends evolve. Two primary approaches are explored: a sliding window with replay buffer method and a transfer learning-based approach. Both methods utilize a Long Short-Term Memory (LSTM) based architecture to model sequential user-item interactions. Empirical evaluations on a large-scale e-commerce dataset demonstrate the effectiveness of the proposed approaches in maintaining recommendation accuracy over time, outperforming a baseline LSTM model that does not account for temporal dynamics. The results highlight the importance of continual learning in sustaining the performance of recommendation systems in the face of evolving user preferences and data distributions.

1 Introduction

In recent years, recommendation systems have become a cornerstone of e-commerce, enhancing user experience and driving sales by providing personalized suggestions. However, these systems face significant challenges in maintaining accuracy and relevance over time. Traditional models are typically trained on static datasets, which renders them ineffective as user behavior patterns shift—a phenomenon known as concept drift and covariate shift. Addressing these challenges requires models that adapt to new data trends while retaining older learned patterns, a balance that is difficult to achieve in practice.

This paper focuses on the problem of predicting the next item a user will interact with within a given session, a task crucial for enhancing user engagement and driving sales in e-commerce, by exploring the development of an adaptive recommendation system using continual learning techniques. Leveraging sales data from an e-commerce platform, we aim to build a system capable of predicting the next likely interaction even as data distributions evolve, focusing on addressing key challenges inherent in this domain. Specifically, the problem involves sequence modeling: given a user's current session and the items they have interacted with, the system must predict the next likely interaction, all while contending with factors such as a large and growing action space, the computational cost of retraining, the need to balance adaptation to new patterns with the retention of previously learned knowledge (catastrophic forgetting), efficient parallelization, and the presence of cold-start scenarios for new items.

The study presents two approaches: the first utilizes continual learning with a replay buffer to incrementally train the model while retaining historical data, and the second employs transfer learning to balance historical and recent data through model weight freezing and fine-tuning. Both approaches are evaluated against a top-k accuracy metric, reflecting the system's performance in handling a large action space with thousands of product IDs.

Preprint. Under review.

Through this work, we demonstrate the feasibility and effectiveness of continual learning in real-world recommendation systems, highlighting how such systems can mitigate catastrophic forgetting, adapt to new patterns, and maintain scalability.

2 Background and Related Work

The problem of adaptive recommendation has been addressed through various techniques, including reinforcement learning, generative adversarial networks (GANs), and continual learning. Deep reinforcement learning (DRL) methods, such as list-wise recommendations and DQN frameworks (1), aim to improve long-term user engagement by learning from high-dimensional inputs. However, these methods often struggle with evolving user preferences and lack scalability in real-time, large-scale environments. Generative Adversarial Networks (GANs) (2) have also been applied to simulate user behavior for reinforcement learning-based recommendations, capturing long-term patterns but facing high computational costs. Conventional methods evaluated on benchmark datasets such as MovieLens and Taobao have demonstrated incremental improvements in user engagement but often fall short in handling real-time adaptability and the large-scale demands of e-commerce platforms.

Continual learning techniques like ADER (5) introduce exemplar replay to mitigate catastrophic forgetting, enabling the model to adapt to new user behavior while retaining past knowledge. Scalable architectures, like Wolpertinger (4), address large action spaces using k-nearest neighbor (K-NN) methods for efficient action evaluation. Models such as AGILE (6) employ graph attention networks to understand action interdependence, supporting decision optimization in recommendation systems where user-item interactions vary significantly.

Our approach builds upon these techniques by combining continual learning with an LSTM-based architecture to provide a flexible and scalable system that overcomes limitations related to computational intensity, real-time adaptability, and user preference evolution.

3 Methodology

3.1 Data Preprocessing

The dataset used in this study comprises user-item interactions from an e-commerce platform, where each row represents a user session and includes features such as ‘event_time‘, ‘product_id‘, ‘user_id‘, and ‘user_session‘. The data is preprocessed using PySpark to handle its large scale efficiently.

Categorical features (‘product_id‘, ‘user_id‘, ‘user_session‘) are encoded into numerical indices using StringIndexer. User interactions within each session are sorted by ‘event_time‘ and grouped to form sequences. The last item in each sequence is treated as the target variable, while the preceding items form the input sequence. Sequences are padded to a fixed length (e.g., 10) to ensure uniform input size for the model.

3.2 Model Architecture

Both approaches utilize an LSTM-based neural network architecture. The model consists of the following layers:

1. **Embedding Layer**: Maps item indices to dense vector representations of a fixed dimension (e.g., 128 for transfer learning, 32 for replay buffer).
2. **LSTM Layers**: Two LSTM layers with dropout (e.g., 0.1) to capture temporal dependencies in the input sequences.
3. **Dense Layers**: Two fully connected layers with ReLU activation, followed by a final dense layer with softmax activation to predict the probability distribution over all possible items.

The model is trained using the Adam optimizer with a learning rate of 0.0045 and sparse categorical cross-entropy loss.

Table 1: Baseline LSTM Architecture

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 10, 32)	755,136
lstm_2 (LSTM)	(None, 10, 256)	295,936
dropout_3 (Dropout)	(None, 10, 256)	0
lstm_3 (LSTM)	(None, 128)	197,120
dropout_4 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 512)	66,048
dropout_5 (Dropout)	(None, 512)	0
dense_4 (Dense)	(None, 256)	131,328
dense_5 (Dense)	(None, 23598)	6,064,686

3.3 Approach 1: Sliding Window with Replay Buffer

The model is initially trained on a subset of the data (e.g., 45k rows). As shown in Figure 1., The training window is shifted forward by a fixed number of rows (e.g., 10k), and the test set is updated accordingly. A portion of the previous training data is randomly sampled and stored in a replay memory buffer (e.g., 15k rows). The model is retrained incrementally on a combination of the new training data and a sample from the replay buffer. The model's performance is evaluated on the updated test set using top-k accuracy. The size of the replay buffer and the sampling strategy influence the model's ability to retain past knowledge and adapt to new patterns.

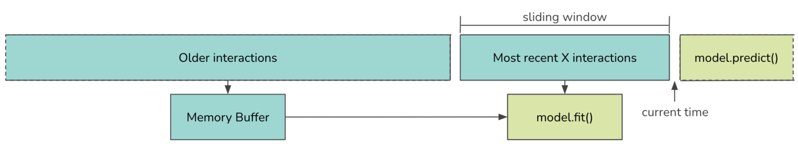

Figure 1: Sliding Window with Replay Buffer

3.4 Approach 2: Transfer Learning

A base model (M1) is trained on an initial subset of the data. The weights of the lower layers of M1 are frozen and transferred to a new model (M2) as shown in Figure 2. The upper layers of M2 are replaced, and the model is fine-tuned on new data. This is repeated for subsequent 15-day time periods, creating a sequence of models (M3, M4, etc.). During weight transfer, special care is taken for the embedding layer. Only the weights corresponding to items present in both the old and new vocabularies are transferred. The output layer is adapted to the new vocabulary size.

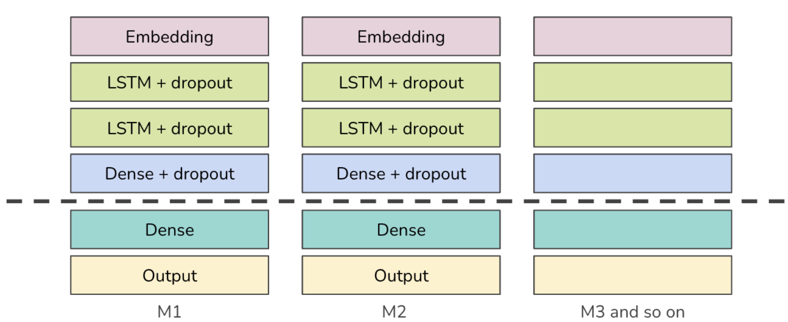

Figure 2: Transfer Learning

4 Results

4.1 Sliding Window with Replay Buffer

Table 2 shows the top-k accuracy of the sliding window with replay buffer approach compared to a baseline LSTM model without continual learning. The results demonstrate that the replay buffer method consistently outperforms the baseline, especially after new data ingestion.

Table 2: Performance Comparison of Sliding Window with Replay Buffer

Data Size	Testing Phase	Baseline LSTM	Sliding & Replay Buffer
Smaller (10%)	Initial Set	11.72%	11.91%
	After New Data Ingestion	8.11%	12.43%
Huge (1M interactions)	Initial Set	21.56%	21.15%
	After New Data Ingestion	16.79%	22.52%

4.2 Transfer Learning

Table 3 shows the top-k accuracy of the transfer learning approach compared to a baseline LSTM model. The results indicate that transfer learning maintains a higher accuracy than the baseline, particularly after a shift of 15 days.

Table 3: Performance Comparison of Transfer Learning

Data Size	Testing Phase	Baseline LSTM	Transfer Learning
Smaller (10%)	Initial Set	11.58%	11.58%
	After 15-day Shift	7.45%	12.81%
Huge (1M interactions)	Initial Set	23.56%	23.56%
	After 15-day Shift	17.09%	24.86%

5 Discussion

The results demonstrate the effectiveness of both continual learning approaches in maintaining the performance of the recommendation system over time. The sliding window with replay buffer method shows consistent improvement in top-k accuracy as the window slides forward, indicating its ability to adapt to new patterns while retaining knowledge from past interactions. The replay buffer size and sampling strategy play a crucial role in balancing the retention of old patterns and adaptation to new trends.

The transfer learning approach also exhibits superior performance compared to the baseline, particularly in scenarios with significant shifts in the data distribution. By leveraging the knowledge learned from the initial dataset and fine-tuning on new data, the model effectively adapts to evolving user preferences while mitigating the need for complete retraining.

Both approaches outperform the baseline LSTM model, which suffers from performance degradation due to its inability to adapt to concept drift and covariate shift. This highlights the importance of incorporating continual learning techniques in recommendation systems operating in dynamic environments.

6 Conclusion

This paper presented two continual learning approaches for building an adaptive recommendation system that addresses the challenges of concept drift and covariate shift in e-commerce. The sliding window with replay buffer and transfer learning methods, both based on an LSTM architecture, demonstrated superior performance compared to a baseline LSTM model. The results underscore the importance of continual learning in maintaining the relevance and accuracy of recommendations as user behavior and product trends evolve.

Future work can explore the impact of different hyperparameter settings, such as buffer size, sampling strategies, and the number of frozen layers in transfer learning. Additionally, investigating the use of more sophisticated memory mechanisms and incorporating external knowledge could further enhance the adaptability and robustness of the system.

Acknowledgements

This research was conducted as part of the coursework for CSCI 566 at the University of Southern California. We thank Professor Yan Liu and TA Jesse Zhang for their guidance and support throughout the project. We also acknowledge the use of the NeurIPS 2024 LaTeX template for the preparation of this report. The authors declare no competing interests or specific funding for this research.

References

[1] Zhao, X., Zhang, L., Xia, L., Ding, Z., Yin, D., & Tang, J. (2018). Deep Reinforcement Learning for List-wise Recommendations. *arXiv:1801.00209*. [Online]. Available: `https://arxiv.org/abs/1801.00209`

[2] Chen, X., Li, S., Li, H., Jiang, S., Qi, Y., & Song, L. (2019). Generative Adversarial User Model for Reinforcement Learning Based Recommendation System. In *Proceedings of the 36th International Conference on Machine Learning*, PMLR 97:1052-1061. [Online]. Available: `https://proceedings.mlr.press/v97/chen19f.html`

[3] Mnih, V., Kavukcuoglu, K., Silver, D., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518, 529–533. [Online]. Available: `https://doi.org/10.1038/nature14236`

[4] Dulac-Arnold, G., Evans, R., van Hasselt, H., Sunehag, P., Lillicrap, T., Hunt, J., Mann, T., Weber, T., Degris, T., & Coppin, B. (2016). Deep reinforcement learning in large discrete action spaces. *arXiv preprint*, arXiv:1512.07679. [Online]. Available: `https://arxiv.org/abs/1512.07679`

[5] Liu, T., Zhan, X., & Chang, M. (2022). ADER: Adaptively Distilled Exemplar Replay Towards Continual Learning for Session-based Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 1923–1927. [Online]. Available: `https://arxiv.org/pdf/2007.12000`

[6] Jain, A., Kosaka, N., Kim, K.-M., & Lim, J. J. (2022). Know Your Action Set: Learning Action Relations for Reinforcement Learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, Apr. 2022. [Online]. Available: `https://openreview.net/pdf?id=MljXVdp4A3N`