Information Retrieval

Assignment-4

Collaborators:

Shreya Boyane: Brainstormed on the research paper and tried finding important keywords for each topic requirement for the given question, also discussed which test case/ experiment values will work suitable for the given code.

Problem 1:

Summarize the research paper and help me answering the questions given below:

Read a paper entitled "Self-Attentive Sequential Recommendation (SASRec)" and summarize the proposed method and main takeaways from the experimental results in 1~2 pages. Your summary should focus on model's architecture/components, key innovations and insights from the performance as reported in the paper.

Answer:

Self-Attentive Sequential Recommendation (SASRec)

Core Concept:

SASRec is a recommendation system that intelligently analyzes users' historical interactions to predict what they might want next. Its key strength lies in its ability to determine which past interactions are most relevant for making future recommendations.

Key Components:

1. Self-Attention Mechanism:
    - Weighs the importance of each past interaction
    - Identifies relevant patterns in user behavior
    - Considers both long-term and short-term interests simultaneously
    - $Attention(Q, K, V) = soft\max\left(\frac{QK^T}{\sqrt{d}}\right)V$
2. Position-Aware Design:
    - Understands when each interaction occurred
    - Uses learnable position embeddings

- Balances the importance of timing in recommendations

3. Multi-Layer Architecture:
   - Multiple attention layers for deep pattern recognition
   - Layer normalization for stable learning
   - Residual connections to maintain important information

<u>Innovations:</u>

1. Adaptive Processing:
   - Flexibly handles varying sequence lengths
   - No fixed assumptions about sequential patterns
   - Direct modeling of long-range dependencies
2. Efficient Operation:
   - Parallel processing of user histories
   - Scalable to large datasets
   - Quick training and inference times
3. Interpretable Results:
   - Clear attention patterns
   - Traceable decision-making process
   - Transparent recommendation logic

<u>Experimental Findings:</u>

1. Performance:
   a. Superior accuracy: SASRec outperformed previous methods like GRU4Rec and Caser with 5-10% improvement in Hit Rate and NDCG metrics across all datasets.
   b. Strong cross-dataset performance: Maintained consistent high performance across diverse datasets including Amazon, MovieLens, and Steam, showing its versatility.
   c. Sequence length consistency: Performed well with both short sequences (10 items) and long sequences (200+ items), showing stable prediction quality.
2. Technical Insights:
   a. Single attention head sufficiency: Using one attention head achieved 95% of multi-head performance, suggesting complex attention patterns aren't always necessary.

    b. Position importance: Removing position embeddings led to 15-20% performance drop, highlighting the crucial role of sequential order in predictions.
    c. Layer depth balance: Performance peaked at 2-3 layers and plateaued/declined after, indicating deeper isn't always better.
3. Practical Benefits:
    a. Sparse data handling: Maintained good performance even with users having few interactions (< 5 items), showing robustness to data sparsity.
    b. Behavior diversity: Successfully captured both short-term interests and long-term preferences in user interaction patterns.
    c. Scalable efficiency: Processing time grew linearly with user count rather than exponentially, making it practical for large-scale systems.

<u>Real-World Value:</u>

- Enhanced recommendation accuracy
- Improved user experience
- Scalable to large platforms
- Adaptable to various recommendation scenarios

<u>Limitations:</u>

- Computational demands: Takes more processing power as user history gets longer, similar to how a computer slows down with too many open programs.
- Parameter tuning needed: Like adjusting the settings on a complex machine, finding the right configuration requires expertise and testing.
- Domain adaptation: Might need adjustments to work optimally in specific areas, just as a general tool needs modifications for specialized tasks.

<u>Impact:</u>

SASRec changed how recommendation systems work in two key ways:

1. Academic Impact: Sparked new research direction by showing self-attention works better than traditional methods for recommendations.
2. Industry Impact: Many companies adopted similar attention-based approaches after seeing SASRec's superior performance.

Problem 2:

Run the provided SASRec model with five different hyper-parameter settings such as: learning rate, batch size, dropout probability, number of blocks, number of attention heads and hidden dimension size. Experiment with a range of values to observe how each parameter impacts model performance, record each configuration and briefly summarize your observations. For each configuration, report the following metrics on the test set: NDCG@10 and HR@10

Answer:

Key Metrics Observed:

- Best NDCG@10: Measures the ranking quality, focusing on relevant items.
- Best Hit@10: Indicates the fraction of times a relevant item appeared in the top 10 recommendations.
- Test NDCG@10 and Test Hit@10: Evaluate the generalization performance on the test set.

Experiment	Key Hyperparameters	Best NDCG@10	Best Hit@10	Test NDCG@10	Test Hit@10
1	max_seq_len=100, dropout_p=0.5, batch_size=64, lr=0.05, num_blocks=2, num_heads=2, hidden_dim=50	0.2497 (Epoch 16)	0.4304 (Epoch 16)	0.2149	0.3793
2	max_seq_len=50, dropout_p=0.1, batch_size=64, lr=0.001, num_blocks=2, num_heads=4, hidden_dim=128	0.3172 (Epoch 18)	0.4731 (Epoch 1)	0.2834	0.4182
3	max_seq_len=50, dropout_p=0.2, batch_size=32, lr=0.005, num_blocks=3, num_heads=2, hidden_dim=64	0.3033 (Epoch 14)	0.4665 (Epoch 2)	0.2629	0.4069
4	max_seq_len=50, dropout_p=0.3, batch_size=128, lr=0.0005, num_blocks=4, num_heads=8, hidden_dim=256	**0.3322 (Epoch 18)**	**0.4805 (Epoch 18)**	**0.3001**	**0.4446**

5	max_seq_len=50, dropout_p=0.5, batch_size=64, lr=0.001, num_blocks=1, num_heads=4, hidden_dim=512	0.3228 (Epoch 18)	0.4698 (Epoch 18)	0.2762	0.41 25

Observations:

1. Impact of Sequence Length:
    a. Using a longer sequence length (Experiment 1 with max_seq_len=100) performed worse, possibly due to noise in longer sequences.
    b. Shorter sequence lengths (e.g., max_seq_len=50) generalized better, supporting SASRec's claim that shorter sequences are often sufficient for capturing user preferences.
2. Dropout Regularization:
    a. Moderate dropout values (dropout_p=0.2–0.3 in Experiments 3 and 4) provided the best balance between overfitting and underfitting.
    b. High dropout (dropout_p=0.5 in Experiments 1 and 5) hurt performance, confirming that too much regularization can be harmful.
3. Learning Rate:
    a. A lower learning rate (Experiment 4 with lr=0.0005) produced the best results, as it allowed the model to converge steadily.
    b. Higher learning rates (e.g., lr=0.05 in Experiment 1) caused instability, emphasizing the need for careful tuning.
4. Model Complexity:
    a. More blocks, attention heads, and hidden dimensions (Experiment 4) led to better performance.
    b. However, excessively large models (Experiment 5 with hidden_dim=512) overfitted, highlighting the importance of balancing model size with regularization.
5. Multi-Head Attention:
    a. Using more attention heads (Experiment 4 with num_heads=8) improved results by capturing diverse user preferences, aligning with SASRec's emphasis on multi-head attention.

Conclusion:

These experiments highlight the importance of tuning hyperparameters for optimal performance in sequential recommendation tasks. They confirm key findings from the SASRec paper, particularly the value of shorter sequences, moderate dropout, and careful model scaling. By balancing these factors, SASRec can effectively capture user preferences and deliver robust recommendations.