Towards an Effective Recommendation Algorithm for E-Commerce

Final Year Project Presentation

Student: Tan Choon Wee

Supervisor: Prof. Zhang Jie

Examiner: A/P Long Cheng

Project ID: PSCSE23-0069

Agenda

1. Background
2. Recommendation Algorithms
3. Methodology
4. Experimental Approach
5. Experimental Result
6. Conclusion

Background

Motivation

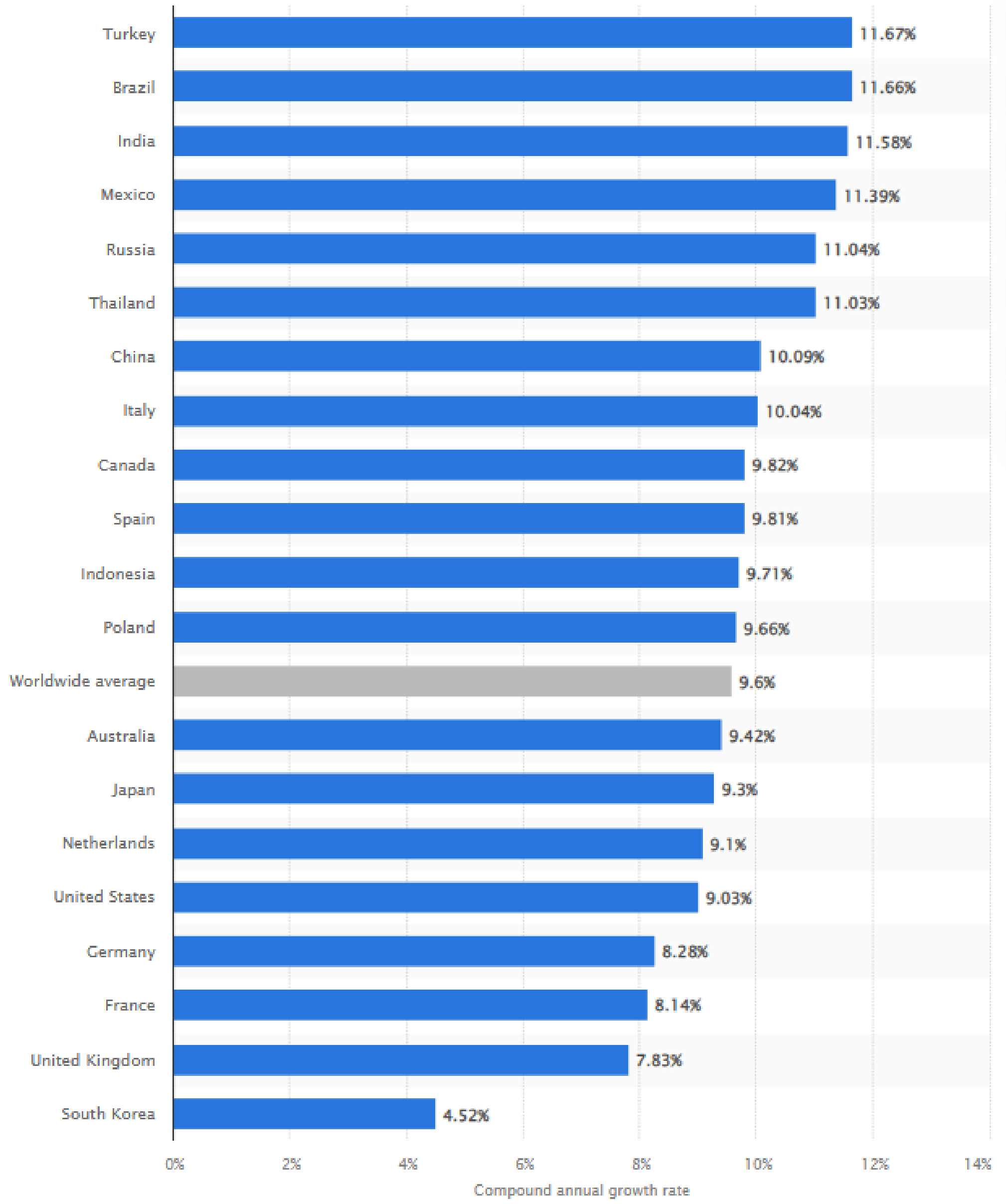
- E-commerce has transformed the global retail landscape, offering consumers convenience and access to an extensive selection of products

- Online platforms feature millions of items, making it challenging for users to find products that align with their preferences

- E-commerce relies heavily on recommender systems

- Increasing competition makes personalized recommendations vital

Retail e-commerce sales compound annual growth rate (CAGR) from 2025 to 2029, by country



Reference: https://www.statista.com/forecasts/220177/b2c-e-commerce-sales-cagr-forecast-for-selected-countries

Objectives

- Evaluate and compare 4 recommendation algorithms:
  1. Matrix Factorization (MF)
  2. Bayesian Personalized Ranking (BPR)
  3. Neural Collaborative Filtering (NCF)
  4. Neural Graph Collaborative Filtering (NGCF)

Goal

- Identify the most effective recommendation algorithm for e-commerce platforms

Recommendation Algorithms

Recommendation Algorithms

Algorithms	Key Idea	Strengths	Limitations
Matrix Factorization (MF)	Decomposes user-item matrix into latent factors.	Simple, scalable, interpretable.	Struggles with sparsity and cold-start; assumes linear relationships.
Bayesian Personalized Ranking (BPR)	Optimizes personalized rankings using implicit feedback.	Effective for implicit feedback; ranking-based.	Requires careful tuning; limited in capturing complex interactions.
Neural Collaborative Filtering (NCF)	Uses neural networks to model non-linear user-item interactions.	Captures complex interactions using MLP.	Computationally expensive; requires hyperparameter tuning.
Neural Graph Collaborative Filtering (NGCF)	Extends NCF with graph neural networks for high-order interactions.	Captures high-order relationships; contextually aware.	Computationally expensive; complex to implement.

Methodology

Code Implementation and Experimentation Setup

- Utilized open-source implementations from Cornac and Recommenders
- Experiments were conducted using Jupyter Notebook on VSCode

PreferredAI/**cornac**

A Comparative Framework for Multimodal Recommender Systems

https://github.com/PreferredAI/cornac

recommenders-team/ **recommenders**

Best Practices on Recommendation Systems

https://github.com/recommenders-team/recommenders

Dataset Utilization

- Employed the MovieLens 100K dataset, a widely used benchmark for recommendation systems

- Dataset contains 100,000 ratings from 943 users on 1,682 movies

	ML-100K
Number of users	943
Number of movies	1,682
Number of ratings	100,000
Number of all genres	19
Average number of genres	1.7
Rating scales	1–5

https://grouplens.org/datasets/movielens/100k/

Hyperparameters

- Initial Parameter Settings: Uniform settings across models for fair comparison.

Parameter	Description	Value
Num_Factors	Dimension of the latent space	200
Num_Epochs	Number of iterations of Stochastic Gradient Descent (SGD)	100
Learning_Rate	Step size (α) in gradient update rules	0.01
Lambda_Reg	L2-Regularization (γ) in the objective function	0.001
Batch Size	Size of training batch	256

Experimental Approach

1. Baseline Performance Evaluation

- Objective: Establish a performance baseline for each algorithm
- Default parameters: Epochs: 100, Learning rate: 0.01
- Evaluation metrics: (at K = 5, 10, 15, 20)

Evaluation Merics

1. MAP@K (Mean Average Precision):
   Evaluates ranking accuracy

2. NDCG@K (Normalized Discounted Cumulative Gain):
   Measures ranking quality considering item positions

3. Precision@K:
   Assesses the ratio of relevant items in top-K recommendations

4. Recall@K:
   Calculates the proportion of retrieved relevant items out of all relevant items

2. Hyperparameter Optimization

- Objective: Determine each model optimal configurations by adjusting key hyperparameters
- Tuned Parameters: **Epochs:** 20, 40, 60, 80, 100

  Learning rate: 0.005, 0.01, 0.05, 0.1

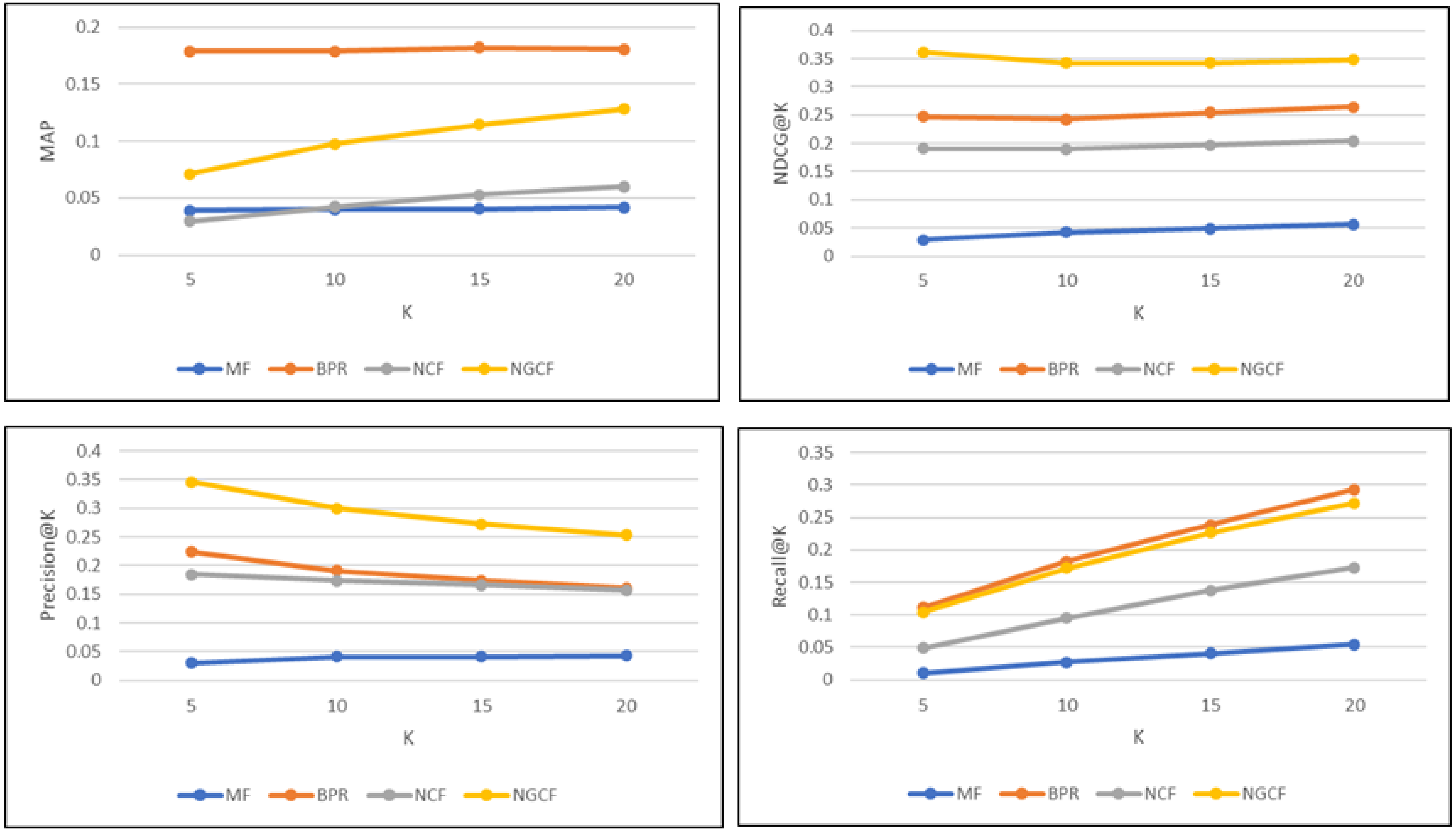
3. Comparative Analysis

- Objective: Rank model effectiveness and identify the best algorithm for e-commerce application

Experimental Result

Baseline Performance Analysis

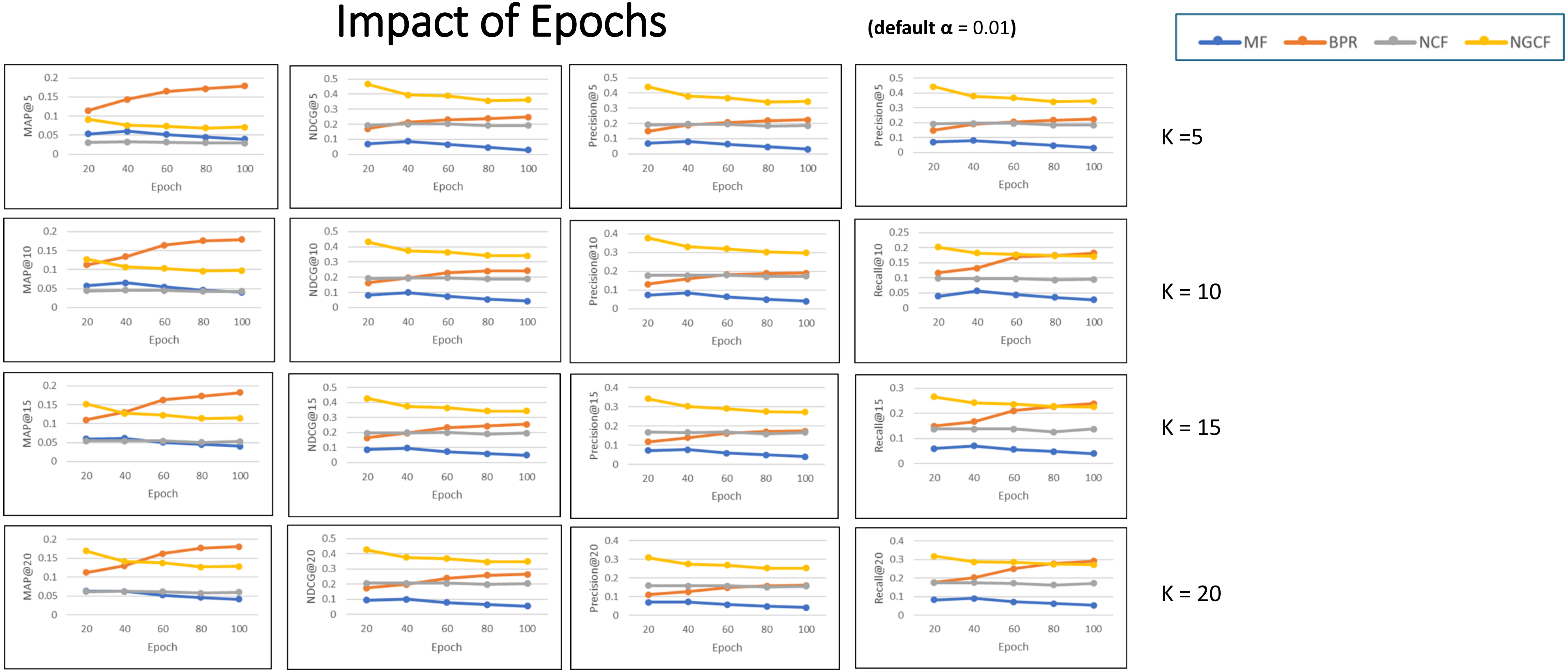
(K = 5, 10, 10, 20)



- BPR consistently outperforms other models in MAP@K.
- NGCF outperforms all models in NDCG@K and Precision@K.
- MF performs poorly across all metrics, NCF slightly better than MF.

Impact of Epochs

(default α = 0.01**)**



K =5

K = 10

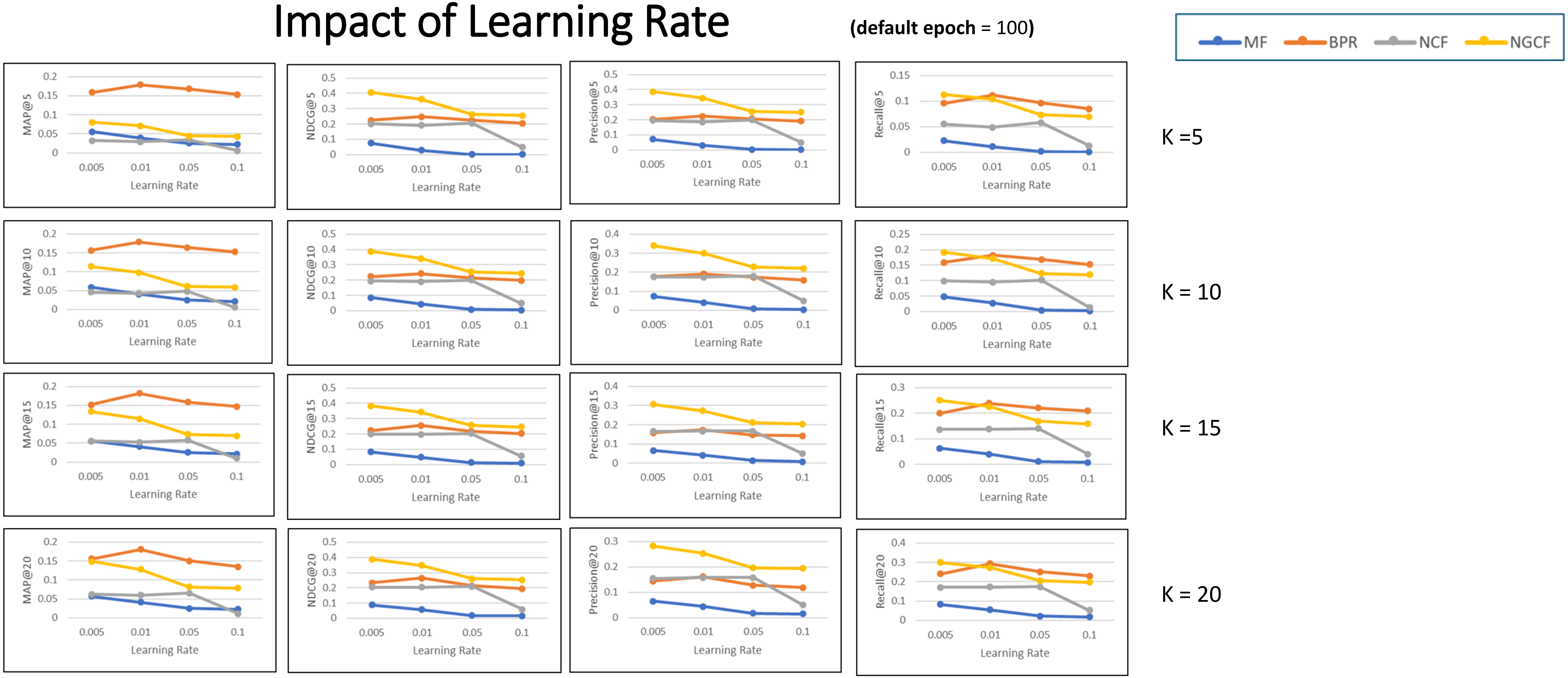
K = 15

K = 20

- BPR improves consistently with increasing epochs
- NGCF performs best at lower epochs (20-40) and then degrades
- NCF remains relatively stable
- MF peaks around 40 epochs and then declines

Impact of Learning Rate

(default epoch = 100)



- NGCF performs best at lower learning rates (0.005-0.01)
- BPR is stable across learning rates
- NCF is sensitive to high learning rate 0.05
- MF performs poorly at higher learning rates

Conclusion

Recommendations for E-Commerce

Scenario	Recommended Algorithm	Optimal Hyperparameters	Reason
Ranking Quality & Relevance	NGCF	Learning rate: 0.005 Epochs: 100	Best at ranking and precision due to modelling high-order interactions
Retrieval of Relevant Items	BPR	Learning rate: 0.01 Epochs: 100	High MAP and Recall performance, great for top-K item retrieval
Computational Efficiency	NCF	Learning rate: 0.005 - 0.01 Epochs: 20 - 40	Stable and lightweight, good for limited training resources
Poor Performance (Avoid)	MF	-	Performs poorly and is unstable in most settings

Lessons Learnt

Model Selection Matters:
Choose models based on application context, deep-learning is not always better.

Hyperparameters are Critical:
Epochs and learning rate greatly affect performance; overfitting is a significant concern.

Know the Algorithms:
NGCF and BPR outperform MF, but each has unique strengths and limitations.

Efficiency vs Accuracy:
There's always a trade-off between computational cost and performance, crucial for real-world applications

Systematic Approach:
Rigorous experimentation and data-driven decisions are crucial for effective recommendation systems.

Acknowledgements

- Supervisor: Prof. Zhang Jie
- Advisors: Dr. Du Yingpeng, Sun Zhu
- Family & Friends

Final Thoughts

> *"Selecting the right model and hyperparameters is key to building an effective and scalable recommendation system."*
>
> \- Inspired by Geoffrey Hinton

Thank you!

Any questions?