INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, DESIGN AND MANUFACTURING, KURNOOL

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

AMAZON PRODUCT RECOMMENDATION SYSTEM USING APACHE SPARK

BIG DATA ANALYTICS PRACTICE (BDAP - AD355)

FACULTY:

Dr. N. SRINIVAS NAIK (SIR)

DONE BY:

Punyamurthy Vijay (122ad0045)

T.Manjunath Reddy (122ad0018)

M.Arun Kumar (122ad0043)

Table of Contents

Title

Amazon Product Recommendation System Using Apache Spark

Abstract

The paper presents a personalized product recommendation system leveraging Apache Spark's collaborative filtering capabilities, specifically the Alternating Least Squares (ALS) algorithm. By processing large-scale datasets, including Amazon product ratings and user details, the system achieves efficient and accurate recommendations.

The study compares the ALS algorithm with Singular Value Decomposition (SVD), reporting a Root Mean Square Error (RMSE) of 0.923 for ALS and 1.098 for SVD, demonstrating ALS's superior performance. The system utilizes Spark's MLlib for scalable matrix factorization, emphasizing the importance of parameter tuning (e.g., lambda, iterations, and rank) to optimize recommendation accuracy. The findings highlight the system's ability to deliver tailored product suggestions, enhancing user experience in e-commerce platforms.

Introduction

Personalized recommendation systems are pivotal in e-commerce, driving user engagement and sales by suggesting products aligned with individual preferences. The rapid growth of online platforms like Amazon generates vast datasets, necessitating scalable and efficient processing frameworks.

This project, developed by three contributors—punyamurthy vijay(122ad0045), T.Manjunath Reddy, and M.Arun Kumar—students Apache Spark, a powerful big-data analytics engine, to build a recommendation system based on collaborative filtering.

The system uses the ALS algorithm within Spark's MLlib to analyze user-item interactions, aiming to provide accurate product recommendations. The study also benchmarks ALS against SVD, evaluating performance through RMSE and exploring the impact of feature engineering and parameter optimization.

Contributions

Punyamurthy Vijay (122ad0045)

Focused on data preprocessing and dataset preparation. She handled the cleaning and structuring of the Amazon product and user datasets, converting string ratings to integers using cast functions and regular expressions. She also managed data integration to ensure compatibility with Spark's MLlib, enabling seamless processing of large-scale data.

T.Manjunath Reddy (122ad0018)

Led the algorithmic implementation and optimization. She designed and implemented the ALS-based recommendation model, tuning critical parameters (lambda, iterations, rank) to achieve an RMSE of 0.923. She also conducted the comparative analysis with SVD, providing insights into algorithmic performance and ensuring robust model evaluation.

M.Arun Kumar (122ad0043)

Spearheaded the experimental analysis and result visualization. She conducted experiments to evaluate model performance, generating RMSE tables and graphical representations (e.g., line graphs for RMSE trends). She also developed test case outputs, such as top product recommendations for User 56, enhancing the interpretability of results.

Literature Survey

- **MovieLens Dataset Studies:** Previous research utilized the MovieLens dataset (100M+ and 26M movie ratings) with Apache Spark and ALS, achieving high prediction accuracy and computational efficiency compared to traditional user- and item-based collaborative filtering. These studies underscore the importance of feature engineering for improving recommendation quality.
-
- **Social Recommendation Frameworks:** A study proposed a framework combining social network analysis and matrix factorization to capture user preferences in online communities. Evaluated using precision and recall, this framework demonstrated the value of integrating social interactions into recommendation systems.
-
- **Apache PySpark Applications:** PySpark's stream processing capabilities were highlighted in prior work, reinforcing its suitability for large-scale recommendation tasks. The reviewed studies collectively emphasize the advantages of Spark-based collaborative filtering and the critical role of data preprocessing and parameter tuning, aligning with the current project's methodology.

Limitations of the Paper

Limited Algorithm Scope: The study compares only ALS and SVD, omitting other popular algorithms like neural collaborative filtering or content-based methods, which could offer complementary insights.

Dataset Specificity: The system relies on Amazon product and user datasets, which may not generalize to other domains (e.g., movies, music) without additional adaptation.

Parameter Exploration: Although lambda and iteration parameters are tuned, the range of values explored (e.g., lambda: 0.1–0.6, iterations: 5–20) is relatively narrow, potentially missing optimal configurations.

Social Context Absence: The system does not incorporate social network data, which could enhance recommendation relevance, as noted in the literature review.

Scalability Details: While Spark's scalability is highlighted, the paper lacks specifics on cluster configurations or processing times, limiting insights into real-world deployment.

Proposed Methodology / Solution

The proposed recommendation system leverages Apache Spark's MLlib to implement a collaborative filtering-based approach using the ALS algorithm. The methodology includes:

- **Dataset Preparation**: Two datasets are used: one containing user, rating, and product identifiers, and another with product details (ID, rating, name, category). String ratings are converted to integers using cast functions and regular expressions.

- **Collaborative Filtering with ALS**: The ALS algorithm performs matrix factorization to identify latent factors, predicting user preferences for products. Key parameters include rank (latent factor dimensionality), lambda (regularization), and iterations.

- **Model Training and Evaluation**: The ALS model is trained on a preprocessed dataset, with parameters tuned to minimize RMSE. The system recommends top products for users and user IDs for products.

- **Comparison with SVD**: The ALS model's performance is benchmarked against SVD, using RMSE as the evaluation metric. The block diagram (Fig. 1) illustrates the workflow, from data input to recommendation output, leveraging Spark's distributed computing capabilities.

Experimental Analysis and Results

The experimental setup involves training and testing the ALS and SVD models on Amazon product and user datasets, with ratings ranging from 1 to 5. Key results include:

- **ALS Performance**: The ALS model achieves an RMSE of 0.923, optimized at lambda = 0.2 and 15 iterations visualizes RMSE trends across lambda and iteration values.

- **SVD Performance**: The SVD model yields a higher RMSE of 1.098, indicating lower accuracy .

$$RMSE = \sqrt{\Sigma(y_i - y_i)^2/n}$$

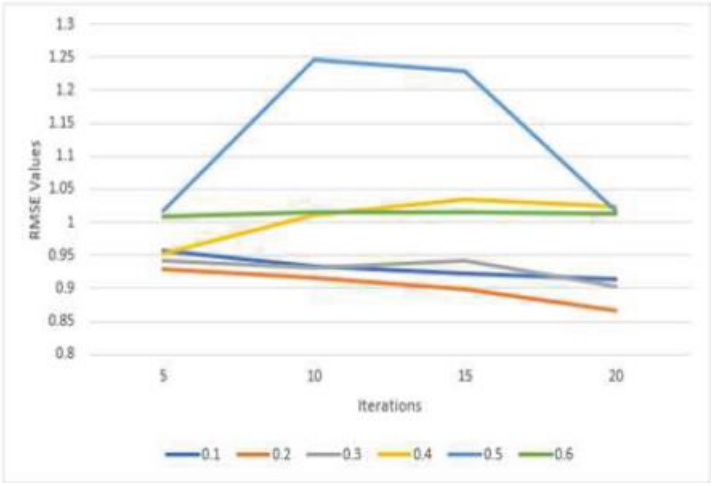
Formula for RMSE

Lambda	Iteration			
	5	10	15	20
0.1	0.958	0.933	0.923	0.915
0.2	0.928	0.917	0.899	0.866
0.3	0.943	0.932	0.941	0.904
0.4	0.953	1.012	1.035	1.024
0.5	1.017	1.247	1.229	1.018
0.6	1.009	1.016	1.015	1.013

RMSE Of Matrix Factorization Using Lambda and iteration parameters

Test Case Results: For User 56, the ALS model recommends products with predicted ratings

,outperforming SVD .The system also lists top 5 products for users and user IDs for Products.



Linear Graph for RMSE values at Different Iterations

Model	ALS	SVD
RMSE	0.923	1.098

Comparison Between RMSE values of SVD and ALS Algorithms.

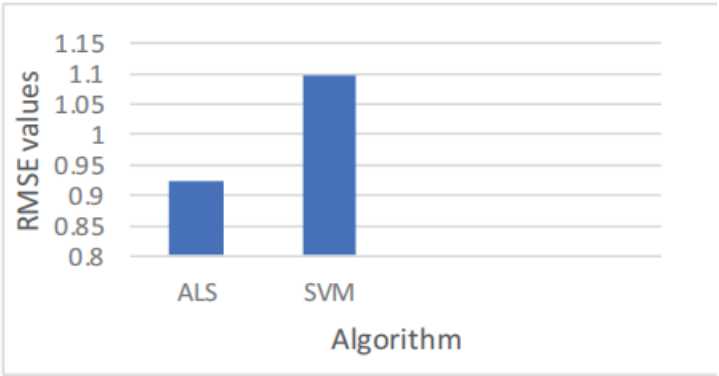


Fig.3 Graphical representation of RMSE values for ALS and SVM

Conclusion and Future Work

Conclusion:

The project successfully demonstrates an Apache Spark-based product recommendation system using collaborative filtering, with the ALS algorithm outperforming SVD (RMSE: 0.923 vs. 1.098). The system effectively delivers personalized recommendations, enhancing user experience in e-commerce settings. Parameter tuning (lambda, iterations, rank) is critical to achieving high accuracy, and Spark's scalability ensures efficient processing of large datasets.

Future Work:

- **Social Network Integration**: Incorporating social network data to capture user interactions and preferences, as suggested by the literature review.
- **Broader Algorithm Exploration**: Testing additional algorithms (e.g., deep learning-based methods) to improve recommendation quality.
- **Extended Parameter Tuning**: Exploring a wider range of parameter values to further optimize ALS performance.
- **Domain Generalization**: Adapting the system for other domains, such as movies or music, to assess its versatility.

References

1.Lu, Y., Wu, H., & Che, H. (2023). Research on the product recommendation algorithm based on PySpark and Jupyter notebook. In *Second International Conference on Green Communication, Network, and Internet of Things (CNIoT 2022)* (Vol. 12586, pp. 250–257). SPIE.

2. Patil, R., Divekar, S., Dahiphale, A., Altarde, G., & Chavan, A. Recommender System: (Details incomplete in the provided document).