Intelligent System for Automatic News Article Recommendation

Midterm Exam Work by **Giga Shubitidze**

For the course of Intelligent Systems by Artūras Serackis.

Table of Contents

Introduction

It might be difficult to find news on websites that actually fit a reader's interests because there is so much articles there. Not everyone is interested in popular subjects, and visitors may find it difficult to locate relevant information without visiting several websites. In order to save users time and improve their reading experience, this project offers an intelligent system that automatically suggests articles based on each user's particular interests.

Problem Formulation

The main goal is to create a system that chooses and suggests articles based on user preferences. However, there are unidentified variables and difficulties:

- **User Preferences**: Because preferences vary over time, it can be difficult to determine exactly what users find interesting at that time.
- **Data Availability**: Recommendations may not be accurate, particularly for new users, if there is insufficient data on user interactions.
- **Algorithm Selection**: Selecting an algorithm that can help recommend news can be challenging because different algorithms work better with different types of data and may perform differently depending on user behavior and an article data.

Dividing the Solution into Tasks

We can divide the project into multiple tasks to deal with this problem:

1. **Data Collection**: Collect article details as well as user behavior data, such as reading history, liked and/or shared articles and their metadata like tags, author, etc.
2. **Algorithm Selection**: Choose the recommendation algorithm which most closely meets the requirements of the system.
3. **Recommendation Generation**: Develop the mechanism that generates recommendations.
4. **User Feedback**: Develop a feedback loop to enable the system to gradually learn and adjust.

Tasks for Different Professionals to develop the system:

- **Project Manager**: Ensures that every phase goes without any problems by supervising project schedules and allocating resources for data gathering and system development.
- **Data Scientist**: Creates and optimizes algorithms that evaluate the similarity of articles and produce personalized suggestions according to user preferences.
- **Software Engineer**: Implements the algorithms in code and ensures the system works reliably and efficiently for all users.
- **Marketing Manager**: Uses information from recommendations to improve content strategies and match them to the preferences of particular audience categories.

Possible Solutions

I considered several methods to achieve effective article recommendations:

1. **Collaborative Filtering**: This method compares a user's behavior with the behavior of other users who are similar to them in order to recommend articles based on their reading history. The theory goes that two people may like the same content if their reading habits are similar. Because it considers the preferences of users who have similar behaviors, this method is popular in recommendation systems and works well, offering personalized recommendations based on shared interests. This approach traditionally rely on similarity measures like cosine similarity or Pearson correlation to recommend content based on the reading patterns of similar users. This is often effective but can be limited with sparse datasets. To learn more about collaborative filtering, see (Xiaoyuan Su, 2009), who offers a thorough analysis of the efficacy of this technique in recommendation systems.
2. **Content-Based Filtering**: This approach makes article recommendations based on particular aspects of the material, such as keywords, subjects, or categories that align with a user's interests. In short, it finds patterns throughout the articles the user has read and suggests related ones. Because it immediately addresses each user's distinct reading preferences and adapts to their changing interests, this method is advantageous. (Pasquale Lops, 2011) provide a more thorough explanation of content-based filtering and its applicability across a range of fields.
3. **Machine Learning Algorithms (K-Nearest Neighbors or Matrix Factorization)**: The accuracy of recommendations can be further improved using machine learning. For example, K-Nearest Neighbors (KNN) uses a user's past interactions to find articles that are similar to their previous selections. By dividing the user-article interaction data into elements, Matrix Factorization adopts a more complex strategy that helps reveal hidden patterns and expedite prediction. By simplifying big datasets, this approach increases system scalability and efficiency. For additional information on Matrix Factorization in recommendation systems, a commonly used method in collaborative filtering, see (Koren, 2009).
4. **Hybrid Approach**: To maximize the advantages of all three, the hybrid approach combines machine learning with collaborative and content-based filtering. It increases suggestion accuracy and relevancy by combining user data with article characteristics. This strategy is advantageous because it finds a balance between content features and user history, allowing both new and regular readers and adapting to a variety of reading preferences. In recommendation applications, hybrid systems are becoming more and more common. (Burke, 2007) provides further information on hybrid approaches and how they are used.

Justification and Best Solution

Although every recommendation method has advantages, the Hybrid approach appears to be the best choice since it combines the best features of each method, making it perhaps the most effective for a news recommendation system. I will try to explain why:

- **Versatility**: The Hybrid Approach is flexible to both new and returning users. The content-based filtering feature can recommend articles for new users with little reading experience based on the current topic of interest, while collaborative filtering, which takes use of similar reader behaviors, is effective for regular users. Relevant recommendations are given to users at any level of interaction thanks to this dual approach.
- **Enhanced Accuracy**: The model can take into consideration both user interaction history and particular article characteristics by combining collaborative filtering with content-based filtering. The system gains a better grasp of user preferences by utilizing data from both viewpoints, which leads to recommendations that are more appropriate and personalized.

- **Increased Robustness**: The disadvantages of any stand-alone strategy are reduced by the hybrid model. Content-based filtering fills the void left by collaborative filtering, which may have trouble with new members. On the other hand, collaborative filtering makes up for the potential lack of contextual variety in content-based filtering. When combined, they produce a well-rounded solution that more accurately represents the diverse interests of users.
- **Adaptability and Scalability with Machine Learning**: Large datasets can be processed efficiently by integrating machine learning methods like Matrix Factorization and K-Nearest Neighbors (KNN). The system may easily develop as user and article data increases thanks to techniques like matrix factorization, which not only decreases data dimensionality but also reveals hidden patterns in user preferences. This flexibility guarantees that even with growing data, the recommendation system will continue to deliver quick, accurate results.

Implementation Steps

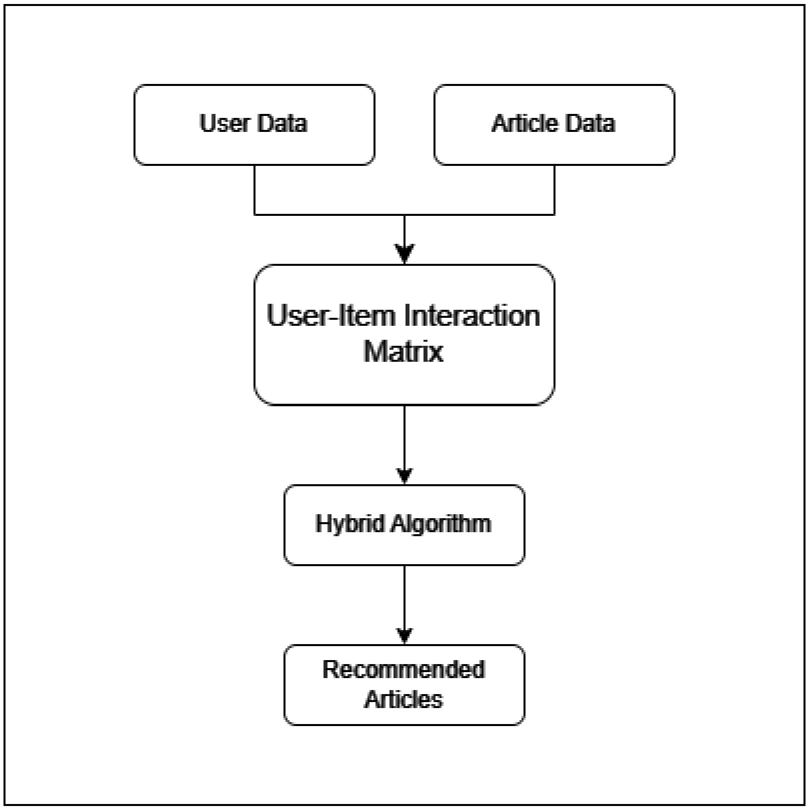
Inputs and Outputs:

- **Inputs**: User data (e.g., reading history, preferences) and article data (e.g., keywords, topics).
- **Outputs**: A ranking of recommended articles according to the user's interests.

Algorithm Flow:

1. Collect user and article data**.**
2. Combine data into a user-item interaction matrix**.**
3. Apply the hybrid algorithm to recommend articles based on user preferences.
4. Display recommendations and collect feedback for further improvements.

Solution Algorithm Diagram



Key Formulas

Cosine Similarity: For collaborative filtering within the hybrid approach, we use the cosine similarity formula to measure similarity between user interaction vectors:

$$Similarity(A, B) = \frac{A * B}{\| A \| * \| B \|}$$

Where:

- A and B are vectors representing articles or user interactions.
- ‖A‖ and ‖B‖ are the Euclidean norms of vectors A and B, respectively.

K-Nearest Neighbors (KNN): The algorithm identifies the k closest neighbors to a target article based on the distance metric (commonly Euclidean distance):

$$D(A,B) = \sqrt{\sum_{i=1}^{n} (A_i - B_i)^2}$$

Where:

- D(A, B) is the distance between articles A and B.
- A_i and B_i are the feature values of articles A and B.

Matrix Factorization: This technique factorizes the user-item interaction matrix R into two lower-dimensional matrices, U (user features) and V (item features), to predict user preferences:

$$R \approx U * V^T$$

Where:

- R is the user-item interaction matrix.
- U is the user feature matrix.
- V is the item feature matrix.

Conclusion

In conclusion, by automatically choosing articles according to users' interests, an intelligent news recommendation system can offer a unique experience. The system can handle limited user data and adjust to a variety of user preferences by utilizing a hybrid method that combines collaborative and content-based filtering. This system is useful, scalable, and able to provide relevant recommendations, all of which will eventually increase customer satisfaction.