Extra Question of Exam - Eremin Vladimir - Social Media Computing (Fall 2024)

What are the other algorithms that should have been used that would perfome better: name, explain why better than KNN for our project in theory.

For our case, several algorithms could potentially perform better than KNN. Here are some of the alternatives, along with explanations of why they might be better suited for our project:

1. Matrix Factorization

Description:
Matrix Factorization, such as Singular Value Decomposition or Alternating Least Squares, focuses on user to item interactions without necessarily considering the item features directly.

Advantages:
- *Scalability:* Such methods can handle large-scale datasets efficiently, which is often necessary for music recommendation systems with large piles of data.
- *Adaptability:* These methods can adapt over time as new users and items are added, improving the recommendation quality as more interaction data becomes available.

Disadvantages:
- *Dynamic Adaptation:* These models can be slow to adapt to new trends or sudden changes in user preferences since they typically require periodic retraining with new data.
- *Sparse Data:* In cases where the user-item interaction matrix is extremely sparse, matrix factorization may have difficulty finding meaningful patterns.

Why might be better than KNN:
KNN relies on explicit similarity measures in a high-dimensional space, which can be computationally expensive and less effective as the dataset grows. Matrix factorization, on the other hand, reduces the dimensionality and captures complex user-item relationships that are not immediately apparent in the raw feature space.

2. Reinforcement Learning

Description:
Reinforcement learning (RL) approaches, such as Deep Q-Learning, optimize recommendations based on user feedback and long-term engagement.

Advantages:

- *Adaptation:* RL models can continuously learn and adapt to changing user preferences and interaction patterns.
- *Engagement:* These models aim to maximize long-term user engagement and satisfaction, rather than just immediate similarity.
- *Exploration vs. Exploitation:* RL naturally balances exploring new items and exploiting known preferences, leading to a more diversified recommendation set.

Disadvantages:

- *Convergence:* Ensuring the convergence of RL algorithms can be difficult in dynamic environments with frequently changing user preferences.

Why Better than KNN:
KNN does not adapt over time and lacks mechanisms to balance exploration and exploitation effectively. RL models, however, can dynamically learn from user interactions and make recommendations that not only match current preferences but also discover new interests.

3. Graph-Based Systems
Description:
Graph-based models use graphs to model relationships between users and items, like graph Convolutional Networks (GCNs)

Advantages:

- *Capturing Relationships:* Graph-based models can capture interconnections between users and items that might influence recommendations.
- *Scalability:* Modern graph-based algorithms can efficiently handle large, sparse datasets common in recommendation scenarios.

Disadvantages:

- *Graph Manipulation:* It may be difficult to manipulate graphs and relationships in it in a dynamic environment with always-changing trends and user preferences.

Why Better than KNN:
Graph-based models can reveal deeper insights into the user-item interaction network than KNN, which only considers direct similarities. By leveraging the structure of the graph, these models can provide more accurate and contextually relevant recommendations.