Recommender Systems Lecture —2

Evolution of Recommender Systems

- **Pre 2007:**
  - Focused on similarity-based, content-based, and collaborative filtering algorithms.
- **2007 - 2015:**
  - Netflix Prize in 2007 catalyzed advancements; Matrix Factorization became popular.
- **Post 2015:**
  - Shift towards deep learning algorithms for enhanced recommendations.
- **Formulating Recommender System Problems:**
  - Unlike classification or regression, the goal is to suggest a ranked list of items to users based on historical data, with a vast scale of users and items.
- **Dataset Representation:**
  - Represented as a matrix A, where rows are users, columns are items, and ,A i,j indicates the interaction between user i and item j, ranging from binary values to real-valued interactions.
- **Handling Non-interaction:**
  - Inevitable sparsity due to the impossibility of universal interaction; many matrix cells remain empty, reflecting unexplored user-item interactions.
- **Sparsity of Matrix A:**
  - Given the enormity of potential user-item combinations, the matrix is inherently sparse, with a minuscule fraction of interactions compared to possible connections.
- **Addressing Sparsity:**
  - The core of recommender systems is to predict user preferences for uninteracted items based on sparse data, hence "fixing" sparsity is not the goal but the premise for building effective recommendation algorithms.
- **Calculating Sparsity:**
  - Sparsity quantifies the ratio of non-empty cells to the total possible interactions, illustrating the challenge and necessity of predictive modeling in recommender systems.

Recommender systems have evolved significantly, adapting to technological advancements and increasing data scales, moving from basic filtering techniques to sophisticated models that navigate the complexities of user preferences and interactions.

Collaborative Filtering System

Introduction:
Collaborative Filtering (CF) enhances recommendations by considering user interactions and preferences. Unlike content-based filtering, CF focuses on finding user or item similarities to predict future interests.

Key Aspects:
1. **Data Sources:** Applies to various data types, including e-commerce, financial, and social media data.
2. **Workflow:** Rates items, identifies similar users, and recommends items those users liked but the current user hasn't seen.

Challenges:
- **Cold Start:** Difficulty in recommending for new users or items without historical data.
- **Sparsity:** The vast number of possible user-item interactions leads to sparse matrices, making it hard to find correlations.
- **Popularity Bias**: Tendency to recommend popular items, neglecting niche interests.
- **Shilling Attacks:** Users manipulating ratings to bias recommendations.
- **Scalability:** Computational costs grow with the number of users and items.

Collaborative Filtering Types:
1. **Item-Item Based:** Recommends items similar to those the user liked, using item similarity metrics.
2. **User-User Based:** Finds users with similar preferences to the target user and recommends items those similar users liked.
3. **User-Item Based:** Utilizes both user and item similarities, often employing matrix factorization to handle sparsity and scalability.

Methods:
- **Hamming Distance:** Measures the difference between binary vectors, useful for item-item comparisons.
- **Euclidean Distance/Cosine Similarity**: Measures similarities between users or items, aiding in user-user or item-item recommendations.

Advantages:
- Offers personalized recommendations based on user behavior.
- Can uncover hidden preferences and niche items.

Drawbacks:
- Struggles with new users/items due to the lack of data (cold start).
- Recommendations can be biased towards popular items.

- Requires handling for data sparsity and scalability.

Evolution:

- **Matrix Factorization:** Introduced as a solution to the Netflix Prize, it addresses scalability and sparsity effectively.
- **Deep Learning:** Recent approaches leverage neural networks to capture complex user-item interactions.

Collaborative Filtering remains a foundational technique in recommender systems, constantly evolving to address its inherent challenges and adapt to new data environments.