Dynamic Personalized Recommendation on

Sparse Data

**ABSTRACT**

Recommendation techniques are very important in the fields of E-commerce and other Web-based services. One of the main difficulties is dynamically providing high-quality recommendation on sparse data. In this paper, a novel dynamic personalized recommendation algorithm is proposed, in which information contained in both ratings and profile contents are utilized by exploring latent relations between ratings, a set of dynamic features are designed to describe user preferences in multiple phases, and finally a recommendation is made by adaptively weighting the features. Experimental results on public datasets show that the proposed algorithm has satisfying performance.

**EXISTING SYSTEM:**

The neighborhood methods can be user-oriented or item-oriented. They try to find likeminded users or similar items on the basis of co-ratings, and predict based on ratings of the nearest neighbors. While latent factor models involve most ratings to capture the general taste of users, they still have difficulties in catching up with the drifting signal in dynamic recommendation because of sparse, and it is hard to physically explain the reason of the involving. In our experiences, the interest cycle differs from user to user, and the pattern how user preferences changes cannot be precisely described by several simple decay functions. Moreover, CF approaches usually accounted the cold-start problem which is amplified in the dynamic scenario since the rate of new users and new items would be high.

**Disadvantages:**

* Hybrid approaches which combine content based and collaborative filtering in different

ways were proposed to alleviate the sparsity problem where more information

were mined than just in each of them.

* The principle of utilization of rating data in these algorithms some approaches emphasize utilization of time information to deal with the dynamic nature.
* The involved ratings can reflect similar users’ preferences and provide useful information for recommendation.

**PROPOSED SYSTEM:**

We can use only historical data but not future data for current prediction in real

applications. In traditional RMSE evaluations training and testing data are randomly sampled and the train and test split is not based on time. This would produce current prediction based on future data. The data in different phases of interest at different training ratios. It is clear that the proposed algorithm is quite robust in the phases, and we found it is not true that the more recent ratings should have heavier weights across the whole time, which illustrates the advantages of the features – light computation, flexibility and high accuracy.

**Advantages**

* The proposed algorithm on different phases of historical data. We applied the proposed algorithm on the data in each single phase defined before, and the RMSEs are calculated separately according to the definition of users’ multiple phases of interest
* More information and may achieve better recommendation accuracies if the information mined is sufficient and the dynamic nature of data is well handled.
* The data in different phases of interest at different training ratios. It is clear that the proposed algorithm is quite robust in the phases, and we found it is not true that the more recent ratings should have heavier weights across the whole time, which illustrates the advantages of the features – light computation, flexibility and high accuracy.4

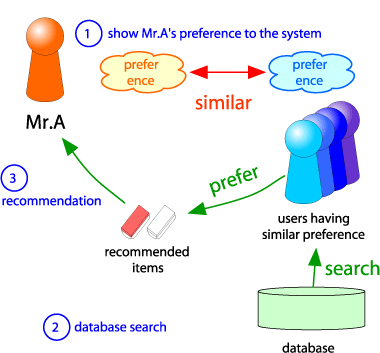
**PROBLEM STATEMENT:**

The CF approaches usually accounter the cold-start problem which is amplified in the dynamic scenario since the rate of new users and new items would be high. Some researchers have previously attempted to solve the above problems. Hybrid approaches which combine content based and collaborative filtering in different ways were proposed to alleviate the sparsity problem where more information were mined than just in each of them. et al. classified items into many categories using content information and chose recent categories to perform Item-Based Collaborative Filtering (IBCF). Kim and Li introduced group similarity by clustering and used it to modify original item-item similarity matrix.

**SCOPE:**

Dynamic recommendation in traditionally RMSE evaluations (even for the Netflix competition), training and testing data are randomly sampled and the train and test split is not based on time. This would produce current prediction based on future data. Even if it is guaranteed that testing instances of each user/item come later than its training instances, the aforementioned issue still exists in algorithms like *IBCF* and *latent factor models* due to the utilization of other users’ future ratings.

**ARCHITECTURE:**



**MODULE DESCRIPTION:**

# **Number of Modules**

After careful analysis the system has been identified to have the following modules:

1. **Dynamic Recommendation Module.**
2. **Relation Mining Of Rating Data Module.**
3. **Dynamic Feature Extraction Module.**
4. **Multiple Phase Of Interest Module.**

**1. Dynamic Recommendation Module:**

Personalized recommendation is a desirable way to improve customer satisfaction and retention. There are mainly three approaches to recommendation engines based on different data analysis methods, i.e., rule-based, content-based and collaborative filtering. Among them, collaborative filtering (CF) requires only data about past user behavior like ratings, and its two main approaches are the neighborhood methods and latent factor models. The neighborhood methods can be user-oriented or item-oriented. They try to find like-minded users or similar items on the basis of co-ratings, and predict based on ratings of the nearest neighbors. Latent factor models try to learn latent factors from the pattern of ratings using techniques like matrix factorization and use the factors to compute the usefulness of items to users.

**2. Relation Mining Of Rating Data Module:**

For the sparsity of recommendation data, the main difficulty of capturing users’ dynamic preferences is the lack of useful information, which may come from three sources – user profiles, item profiles and historical rating records. Traditional algorithms heavily rely on the co-rate relation (to the same item by different users or to different items by the same user), which is rare when the data is sparse. Useful ratings are discovered using the co-rate relation, which is simple, intuitional and physically significant when we go one or two steps along, but it strongly limits the amount of data used in each prediction.

**3. Dynamic Feature Extraction Module:**

Users’ preferences or items’ reputations are drifting, thus we have to deal with the dynamic nature of data to enhance the precision of recommendation algorithms, and recent ratings and remote ratings should have different weights in the prediction. These methods help to make progress in precision of dynamic recommendation, but they also have their weaknesses: decay functions cannot precisely describe the evolution of user preferences and only isolating transient

noise cannot catch up with the change in data.

**4. Multiple Phase Of Interest Module:**

A set of dynamic features to describe users’ multi-phase preferences in consideration of computation, flexibility and accuracy. It is impossible to learn weights of all ratings for each user, but it is possible to learn the general weights of ratings in the user’s different phases of interest if the phases include ranges of time that are long enough.

**SOFTWARE REQUIREMENTS**:

Operating System : Windows

Technology : PHP

Web Technologies : Html, JavaScript, CSS

Web Server : IIS, WAMP,XAMP

Database : My SQL

**HARDWARE REQUIREMENTS**:

Hardware : Pentium

Speed : 1.1 GHz

RAM : 1GB

Hard Disk : 20 GB

Floppy Drive : 1.44 MB

Key Board : Standard Windows Keyboard

Mouse : Two or Three Button Mouse