Received September 24, 2015, accepted November 2, 2015, date of publication December 29, 2015, date of current version March 1, 2016.

Digital Object Identifier 10.1109/ACCESS.2015.2513000

Neha M

Reg. No: 14CO127

8095299474

[nehamohan1996@gmail.com](mailto:nehamohan1996@gmail.com)

Improving Scalability of Personalized Recommendation Systems for Enterprise Knowledge Workers

Authors

Chetan Verma, Michael Hart, Sandeep Bhatkar, Aleatha Parker-Wood, Sujit Dey

* Enterprise Data is huge in size and is spread across different systems.
* To manually fetch user specific data from such huge amount of data by enterprise knowledge is impossible.
* The authors had previously proposed a recommendation system which analyses user file accesses in order to train personalized models, and recommend relevant content with a high degree of accuracy.
* In this paper, they propose a technique called Active Features-based Model Selection (AFMS), which eliminates user models which are most likely to generate negative predictions and hence optimises the recommendation system.
* The system also studies collaborative features between users using normalized triangle count and produces a CF model.

Work Described

* Basic metadata models based on the metadata features are built.
* Using these models, an innovative approach to leverage collaboration among users for building collaborative filtering aware models that provide better performance over the basic metadata models, are built.

Why is this system important?

We often hear complaints from knowledge workers about overwhelming data that they need to work on in enterprises. Earlier, there were attempts to create recommendation models that could recommend data to the user based on their previous activities. But these did not work for new content, i.e., they had cold start problem. The system proposed in this paper takes care of it. Also, comparing every new file with every model takes a huge amount of time. Hence. An algorithm has been proposed to optimise the system which reduces time complexity by 23 times in best case.

**Methodology**

Features are extracted for each of the files in the training phase. And then, using these features the models are built.

Metadata features: The metadata features include features extracted from different attributes of a file.

* Folder: A folder feature corresponding to every folder and its ancestor folders is created in training phase. For a file f, folder features as a binary vector are captured with the value 1 in the locations corresponding to the folder or the ancestor folders of f , else 0.
* Token: The file path is tokenized. While extracting feature, the tokens are extracted as bag-of-words representation on the vocabulary of tokens.
* Extension: An extension vocabulary is constructed for all the popular extensions. This vocabulary is used to represent the new file’s extension.

Modelling Techniques:

1. METADATA-BASED MODELING :

* Using the metadata features extracted during training phase, a model is created for each user based on his file accesses.
* Each of the features are given weights using feature weights capability of SVM classifier.

1. COLLABORATIVE FILTERING (CF)-BASED MODELING:

* Normalised triangle count is used to capture the degree of collaboration of users.
* In this technique, a model for each user is created using metadata-based modelling.
* For each file in the validation set, the metadata models of all the users are applied, and a vector of the resulting predicted labels is generated
* Then, CF model is trained for the user u using the files in the validation set.

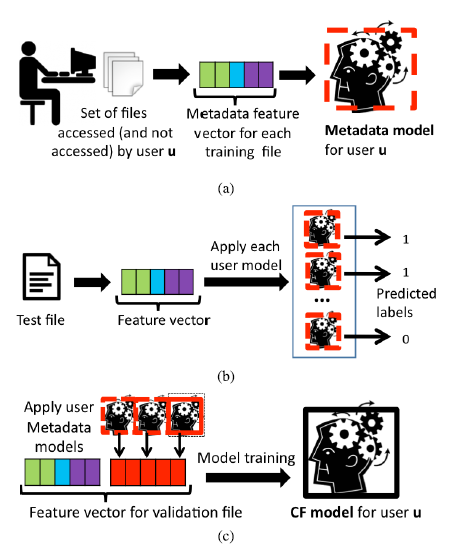


Fig 1: (a) **Training**: metadata models for each user. (b) **Testing**: apply the trained models for all users in share.

(c) **Training**: CF models for each user.

**Results**

METADATA MODEL

The numbers in column Max AR@75P are reasonable performance estimate of a well tuned system. The average of these numbers across all the shares, 49.4%, shows that the metadata models can capture nearly half of users' access for new files, while having under 25% wrong file recommendations.

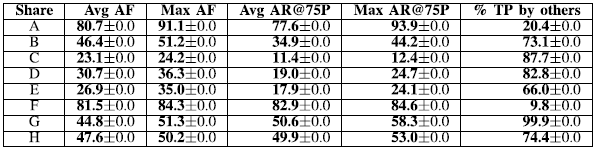


Fig 2: Performance of metadata model

CF BASED MODEL

The results in Table 3 clearly highlight the significant gains of CF models over metadata models. The average of Max AR@75P across all shares is 70.2%, which is an improvement of a whopping 20% over the performance of metadata models.

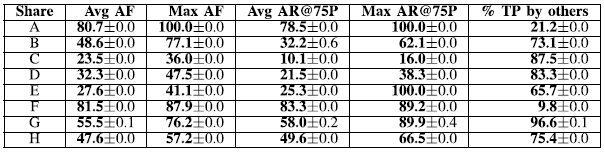


Fig 3: Performance of CF model

**Pros**

* The proposed system can predict recommendations for new files as well, unlike other systems.
* The AFMS algorithm proposed decreases execution time by a considerable amount.

**Future work**

The current system uses only the file metadata information to provide recommendations. In future, the content of files could also be utilized to improve the effectiveness of the system.

**Implementation as a part of mini project**

* Language : Python
* Scikit-learn library in python will be used for model selection.
* Linear kernel SVM has been suggested by the authors for better performance. Hence, the same will be used.
* Building the system using metadata based model completely will be of first preference. The CF model will be added later.