

CROSSING FRAMEWORK

A Dynamic Infrastructure to Develop Knowledge-based Recommenders in Cross Domains

Mustafa Azak and Aysenur Birturk

Dept. of Computer Engineering, Middle East Technical University, Inonu Bulvarı, Ankara, Turkey

Keywords: Web Personalization, Recommender Systems, Cross Domain Recommendation, Recommendation Frameworks, Recommender Engines.

Abstract: We propose a dynamic framework that differs from the previous works as it focuses on the easy development of knowledge-based recommenders and it proposes an intensive cross domain capability with the help of domain knowledge. The framework has a generic and flexible structure that data models and user interfaces are generated based on ontologies. New recommendation domains can be integrated to the framework easily in order to improve recommendation diversity. We accomplish the cross-domain recommendation via an abstraction in domain features if the direct matching of the domain features is not possible when the domains are not very close to each other.

1 INTRODUCTION

Over the last decade, several researches have been made to improve the recommendation methods and user modeling techniques in order to extend quality of recommendations and provide maximum user satisfaction (Anand & Mobasher, 2003).

Therefore, exploiting the user profiles of Web 2.0 style sharing platforms (Szomszor et al, 2008) and applying advanced recommendation methods, recommender systems have achieved remarkable practical and commercial success. Amazon.com, Last.fm, Netflix and MovieLens are the most successful and popular examples of the recommender systems. Although web-based social and commercial networks provide variety of interests for users in various domains such as books, movies, music and games; most recommendation systems currently provide recommendations within a single domain based on the domain specific knowledge. The current infrastructures of the recommender systems cannot provide the complete mechanisms to meet user needs in different domains and recommender systems show poor performance in cross-domain item recommendations. Generally, they only consider the statistical analysis on the cross-selling markets and try to make use of relations between popular items without personalized recommendations. Therefore, there is a lack of cross-domain recommenders which can

make successful personalized recommendations in cross-domains. Moreover, there is no framework to develop cross-domain recommender systems.

In this paper, we propose a dynamic framework, which enables addition of new recommendation domains and provides the development of knowledge-based cross-domain recommenders as well as single domain recommenders. The addition of a new recommendation domain is crucial because it enables to improve the recommender system's capability and evolve the system according to new user needs. In order to provide dynamic domain additions to framework, our system considers each domain as a pluggable component that provides the required domain data via well defined interfaces.

Providing accurate predictions and useful recommendations among these integrated domains are the other important capabilities of the proposed framework. The framework provides knowledge-based recommender development with a hybrid approach for generating recommendations. Our recommendation engine has different types of recommendation strategies which are applied depending on the available data about the user and the items in domains. Collaborative and content-based strategies use feature-weighted models in similarity calculations. In addition, we have also high level abstractions and relations between domains. The relational information about domains can be dynamically added to the framework by

defining the rule sets which provide inter-domain knowledge between two specific domains. The inter-domain knowledge is used for feature mapping between domains and it affects the weights of features in target domain while generating a cross domain recommendation.

The remainder of the paper is organized as follows. In the next section, we overview the related work, then we present the overview of our proposed framework in the section 3. Then, the implementation process is described in detail in the section 4. Evaluation process is explained in the section 5. Finally, the future work and concluding remarks are presented in the last section.

2 RELATED WORK

There has been much research done on the possible extensions of current recommenders systems. In this section, we briefly mention some related work in recommender systems architectures, cross-domain and multi-content recommendation approaches. (Loizou, 2007) introduces a framework outline for achieving multi domain recommendation by providing detailed user representation with a universal vocabulary and community experts' data during the addition of new domains. (Chung et al, 2007) discusses another framework for individual functioning in multiple domains. (Gonzalez et al, 2006) analyzed the cross-disciplinary trends from human perspective to develop ambient recommender systems by introducing smart user models. (Berkovsky et al, 2005) represents mechanisms to develop cross-domain user modeling by mediating and integrating partial user models stored in different resources and knowledge bases.

There are few framework approaches to meet users' needs in multiple domains. Our framework approach differs from the previous works as it focuses on the easy development of knowledge-based recommenders and proposes an intensive cross domain capability with the help of domain knowledge. Our aim is to develop reasonable recommenders and generating useful and personalized cross-domain recommendations. We accomplish the cross-domain recommendation via an abstraction in domain features if the direct matching of the domain features is not possible when the domains are not very close to each other.

3 OVERVIEW OF THE FRAMEWORK

The objective of this study is to provide a framework for recommender systems that is capable of integrating new domains dynamically, creating semantic relationships between existing domains and extending traditional recommendation approaches to provide more accurate and useful recommendations in cross domains. The capability of adapting new domains and creating useful cross domain recommendations result in offering a bundle of related items from different domains and thus improve the possibility of user satisfaction. The adaptive and flexible infrastructure also provides the capability to integrate variety of new and existing systems to framework easily. In addition, the cold start problems of new domains can be minimized by providing domain knowledge and assigning features' weights as accurate as possible in user models. The dynamically provided inter-domain knowledge helps us to determine the domains' relationships and features mapping in different domains. In order to achieve the above goals, we designed our framework in a modular way, shown in Figure 1. The details of the framework and its components are explained in the rest of the section.

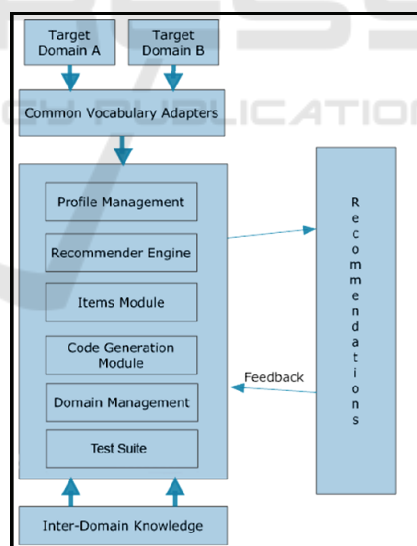


Figure 1: Overview of the Framework.

3.1 Profile Management

Creating a comprehensive and detailed user model is very important to analyze user interests and needs correctly. Our profile management component is responsible for creating and maintaining the user profiles. It has two main parts to support the

recommenders in the system. “Data” part provides required knowledge about user preferences while “Actions” part enables users to edit their profiles.

3.2 Recommender Engine

Our framework provides knowledge-based recommender development and provides a hybrid approach for generating recommendations. The recommendation engine has different types of recommendation strategies which are applied depending on the available data about the user and the items in domains. The system starts with the simple recommendation techniques and move to more complex algorithms with user’s experience.

3.2.1 Recommendation Strategies

Most Popular Items strategy is the simplest technique in the system but it helps us to generate useful prediction when no data available about users. The algorithm finds the most popular items in integrated domains by sorting the item scores. This strategy also used with other strategies to find the popular items for a specific set of users.

Demographic Filtering is similar to collaborative filtering but it uses the demographic information of user in order to find the similar user profiles. User’s demographic information is represented as feature vector and each feature has weights for each domain.

Content Based Filtering recommends similar items to the ones that user preferred in the past. The candidate items compared with the previously rated items and best matching items are recommended as described in (Adomavicius & Tuzhilin, 2005).

Collaborative Filtering recommends the items that people sharing the similar preferences with the user prefer. It assumes that user likes the items that are highly rated by similar users. The similarity between users is calculated based on commonly rated items. Our system uses Pearson correlation coefficient to measure the user similarity (Adomavicius & Tuzhilin, 2005).

Surprise strategy is used to overcome “Overspecialization Problem” and “Sparsity Problem in Rating”. Our framework generates one surprising recommendation randomly among the newly added items, popular items or few rated items

Knowledge-Based recommendation is not used standalone but it helps other recommendation strategies. The knowledge base is used by our system to retrieve rules about the users, features, domain relations and feature mappings. These rules help us to create user groups in domains and define the relations between user groups and item features. The rules for these relations are defined as 6-tuples

$GroupRule < d, uf, ufv, if, r_i, status >$ where d represents the domain, uf defines the user feature name and ufv is the value of the given feature in uf , if indicates the item feature name, ifv is the value of that item feature and status determines whether this rule affects rating prediction positively or negatively.

For instance, we can define a group rule such as “Observer people like Animation movies”. It is based on “personality” feature of user and it uses “genre” feature of items in “movie” domain. It also states that if this rule holds for a user, it affects the rating positively.

$GroupRules < Movie, Personality, Observer, Genre, Animation, Like >$

The group rules directly affect the predicted ratings of a user on items by using the formula below:

$$\bar{r}_{i,k} = r_{i,k} \pm \begin{cases} \text{knowledge effect (if conditions holds)} \\ 0 \text{ (otherwise)} \end{cases}$$

$\bar{r}_{i,k}$ is the final rating calculated where $r_{i,k}$ is the rating by user i on item k determined by Collaborative Filtering or Content-Based filtering algorithm. The value of knowledge effect is between 0 and 1. It is calculated experimentally and can be configured.

3.2.2 Recommendation Generation

When a new user logs in to system and there is no information available about user, recommendations are generated using the “most popular item” strategy. If the demographic information is available, “demographic filtering” is also applied. Recommendation generation evolves with the user experience in the system. In the case of having enough rating history for a domain but not having required number of co-rated items with other users in target domain, “content-based filtering” is used. When the user has enough number of co-rated items with other users in target domains, “collaborative based filtering” is used to generate recommendations. “Knowledge-Based” recommendation is used with Collaborative Filtering or Content-Based filtering algorithm. “Surprise” Strategy is always used but it only generates one random recommendation at a time.

3.3 Domain Management

The domain management component mainly deals with the integrated domains and their relationships. It includes the “Domain Knowledge” of all domains which is required by knowledge-based

recommendation. Domain management component also contains the general ontology interface with XML schema of the framework. General ontologies of framework have the specifications of features and relationships between features of different domains. They provide the uniformity and knowledge exchange between different domains.

3.4 Code Generation Module

Code generation module generates the code for all dynamic graphical user interfaces based on the data models defined in ontology files. When new domains are integrated to framework or data models are updated, the changes can easily be applied to system by this module.

3.5 Items Module

Items are the elements of domains such as books, movies, or songs that are recommended to user according to the user preferences. Our items module is responsible for retrieving and maintaining items from integrated domains.

3.6 Test Suite

It is one of the most important components of the framework. Test Suite provides an environment to test, evaluate and verify the algorithms of recommender engine. It also allows us to observe the effects of the knowledge base on the integrated domains.

3.7 Common Vocabulary Adapters

Each domain data has its own data structures and ontology but in order to integrate a domain to framework, we have to transform its domain information to a common vocabulary and structure. Therefore, for each domain, a common vocabulary adapter is needed to be developed. Considering the general ontologies and XML schema of the framework; a common vocabulary adapter deals with ontology mapping and type conversions of the features.

3.8 Target Domains

Recommendation domains are the specific fields of interests and constitute of knowledge about users, items, concepts and relationships in a field. We divided domain information into four categories:

User data is a feature vector that represents the

user preferences. User data has also transaction history and item ratings which constitute the knowledge about user actions history that help us predict user attitude to new items and new domains. **Items** are the elements of the domains which have certain features based on the domain structures and specifications. **Domain knowledge** stores the structured knowledge about the domains which is retrieved and used to determine weights of factors which have effects on items during the recommendation process. **Domain ontology** represents the meanings of the terms in user data, community data and item features. The concepts and relationships are defined by ontological categories.

4 IMPLEMENTATION DETAILS

A prototype of the framework has been developed to evaluate the proposed structure in the Figure 2. It is an online cross domain recommender system which will be available at www.crossingframework.org. The main purposes of the prototype are to test applicability of the framework, provide interaction with real world users, observe the performance of inter-domain knowledge rules and prepare an environment to evaluate the recommendation strategies. The core of framework was developed with Java programming language. Dynamic and flexible data interaction is provided over the ontology files with XML interfaces. Common vocabulary adapters are developed for target

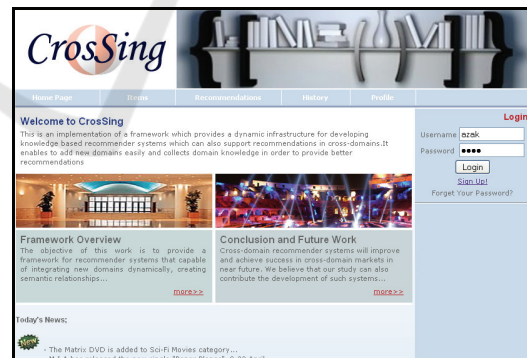


Figure 2: The Prototype Implementation of the Framework.

domains such as movie and book domains.

Ontology files are required for the generation of the data structures in the system and creation of tables in the database. They consist of features with the description and other required attributes for system. All dynamic graphical user interfaces of the framework are generated automatically based on these features.

When a new domain is needed to integrate with the framework, the corresponding data files should be created according to the following ontology files: Domain Ontology, Item Ontology, User Ontology, Rating Ontology, User Profile Similarity Ontology and Item Similarity Ontology.

The framework has different interfaces and perspectives for end users and administrators.

5 EVALUATION

Automated testing is not applicable for cross-domain testing because there is no dataset available for our target domains. Therefore, we focus on the single domain testing and defer cross-domain tests until the deployment of the prototype of the framework and beta testing with real end users. For each single target domain, recommender engine is tested with framework's test suite and we observed the effects of group rules on collaborative filtering algorithm. Besides the evaluation results, our aim is to show that our framework enables easy development of knowledge-based recommenders and it also facilitates a testing environment to test, evaluate and verify the algorithms of recommender engine and its knowledge base. For their evaluation process, we follow the steps explained below.

5.1 Knowledge Acquisition

In order to form a knowledge base about the target domains 'movies', 'music' and 'books', we prepared an online survey and 100 people from 10 different countries were participated. The purpose of the survey is to learn users' preferences and needs in target domains and their personality features. Some example questions from the survey are as follows:

- Would you sort the following MOVIE features considering the importance for you?
"Title, Actor, Actress, Producer, Director, Year, Genre, Tags, Language, Country"
- Which types of MOVIES you like?
"Action, Animation, Comedy, ..., Other"
- How do you define yourself?
"Perfectionist, Helper, Performer, Romantic, Observer, Questioner, Adventurer, Boss, Peacemaker"

For personality types we chose the nine types of the Enneagram of personality given above which are useful in classifying characters. The results of the survey are analyzed statistically with the SPSS (Statistical Package for the Social Sciences) software by the help of two professional

statisticians. The feature relations and rules about users' preferences and tendencies such as "Observers dislike Romance movies" and "Helpers like Romance movies" are obtained.

We tested each rule's effects on the collaborative and content-based recommendation algorithms.

5.2 Metrics

In order to determine the prediction quality of our knowledge-based approach which extends collaborative and content-based algorithms, Mean Absolute Error (MAE) metrics (Sarwar et al, 2001) was used. The MAE is computed by first summing the absolute errors of the N corresponding ratings-prediction pairs and then averaging the sum. A smaller value of MAE indicates a better accuracy.

5.3 Data Sets, Common Vocabulary Adapters and Data Preprocessing

In order to test our approach we developed common vocabulary adapters for the movie, music and book domains using the datasets available datasets. For this work, we present the dataset for movie domain.

We used a popular database, the MovieLens dataset by the GroupLens Research group. The data set contains 1682 movies, 943 users and 100,000 ratings (1-5 scales), where each user has rated at least 20. We matched the movie's information with the IMDb dataset to extract extra features.

To compare our approach with the state of art collaborative algorithm, we chose the cross validation technique with holdout method and performed the experiments under the different configurations.

As our knowledge base rules make use of user's personality features, some preprocessing is required in order to determine the active user's personalities in these configurations. We used Weka (Waikato Environment for Knowledge Analysis) which is a popular suite of machine learning software in order to classify users via Decision Trees.

5.4 Evaluation Results

Because of space limitation, we present two different knowledge base performances against a state-of art collaborative filtering technique (CF) (Adomavicius & Tuzhilin, 2005) on movie data set. We prepare the knowledge bases with the following rules:

Knowledge Base 1 (KB1)

- "Observers like Animation movies"
- "Observers dislike Romance movies"

Knowledge Base 2 (KB2)

- “Helpers like History movies”
- “Helpers like Romance movies”

The number of nearest neighbors in collaborative filtering is set as 35 and the knowledge effect variable is set to 0.4 in all configurations since it is best value shown in Figure 3.

Table 1: MAE comparison of methods.

Training Users	Methods	Given5	Given10	Given20
Movie 100	CF	0.8605	0.8461	0.8591
	KB1	0.8628	0.8482	0.8606
	KB2	0.8619	0.8474	0.8612
Movie 200	CF	0.8528	0.8357	0.8453
	KB1	0.8541	0.8365	0.8461
	KB2	0.8533	0.8358	0.8454
Movie 300	CF	0.8490	0.8430	0.8377
	KB1	0.8500	0.8433	0.8383
	KB2	0.8493	0.8432	0.8376

In table 1, we can observe that our prediction approach cannot improve the quality of the state-of-art collaborative filtering algorithm in any configurations. The only improvement is in the Movie300Given20 with KB2 but the difference between Collaborative filtering is not significant. Although the initial results do not seem very satisfying, we can assume that our framework infrastructure and testing suite is working and group rules have effects on the predictions. The results can be better with different rule combinations. Additionally, we had some disadvantages about determining the users' personality in the survey and the dataset. The participants might make mistakes about deciding their real personalities in the survey and there is an error rate at the decision trees used in WEKA.

In order to examine the sensitivity of the knowledge effect variable, we varied the value of the variable and computed the MAE for each variation. The Figure 3 shows that 0.4 is the optimum value for the knowledge effect variable.

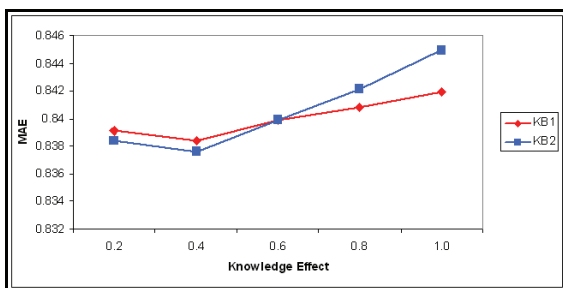


Figure 3: Impact of Knowledge Effect on MAE.

6 CONCLUSIONS

In this work, we proposed a dynamic framework for developing knowledge-based cross-domain recommender systems. The framework has a generic and flexible structure that data models and user interfaces are generated based on ontologies. New recommendation domains can be integrated with the framework easily in order to improve recommendation diversity. In addition, knowledge base helps to generate useful recommendations in cross-domains and maximize user satisfaction.

We can conclude that cross-domain recommendation approach will gain more attention in near future and our framework can be used to develop successful recommenders.

REFERENCES

- Adomavicius, G. and Tuzhilin A. 2005. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-art and Possible Extension. *IEEE Trans. Know. And Data Eng.* 17, 6: 734-749.
- Anand, S. and Mobasher, B. 2003 Intelligent Techniques for Web Personalization. *IJCAI 2003 Workshop.*
- Berkovsky, S., Kuflik, T. and Ricci, F. 2005. Entertainment Personalization Mechanism through Cross-domain User Modeling. *1st Int'l Conf. Intelligent Tech. for Interactive Ent.*, 215-219.
- Chung, R., Sundaram, D. and Srinivasan, A. 2007. Integrated Personal Recommender Systems. *ICEC'07.* Minneapolis, Minnesota, USA.
- Gonzalez, G., Lluís de la Rosa, J., Dugdale, J., Pavard, B., El Jed, M., Angulo, C. and Klann, M. 2006 Towards Ambient Recommender Systems: Results of New Cross-disciplinary Trends. In *Proc. of the EU Conf. on AI, WS on Recommender Systems.*
- Loizou, A. 2007. Unlocking the Potential of Recommender Systems: A Framework to Achieve Multiple Domain Recommendations. *Sch. of Electronics and CS, University of Southampton.*
- Sarwar, B., Karypis, G., Konstan, J., & Reidl, J. 2001. Item-based collaborative filtering recommendation algorithms. In *Proc. of the 10th Int'l conf. on WWW.*
- Sebastiani, F. 2002. *ML in Aut. TC.* ACM Comp. Surv.
- Szomszor, M., Cantador, I. and Alani, H. 2008. Correlating User Profiles from Multiple Folksonomies. *ACM Conf. on Hypertext and Hypermedia.*