



**HAL**  
open science

## A RBN-based recommender system architecture

Mouna Ben Ishak, Nahla Ben Amor, Philippe Leray

► **To cite this version:**

Mouna Ben Ishak, Nahla Ben Amor, Philippe Leray. A RBN-based recommender system architecture. International Conference on Modeling, Simulation and Applied Optimization (ICMSAO 2013), 2013, Hammamet, Tunisia. pp.1-6, 10.1109/ICMSAO.2013.6552609 . hal-00812168

**HAL Id: hal-00812168**

**<https://hal.science/hal-00812168>**

Submitted on 17 Apr 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A RBN-based recommender system architecture

Mouna Ben Ishak  
LARODEC & LINA Laboratories,  
ISG-Tunis, Polytech’Nantes  
Tunisie, France  
mouna.benishak@gmail.com

Nahla Ben Amor  
LARODEC Laboratory,  
ISG-Tunis,  
Tunisie  
nahla.benamor@gmx.fr

Philippe Leray  
LINA Laboratory,  
Polytech’Nantes,  
France  
philippe.Leray@univ-nantes.fr

**Abstract**—With the widespread use of Internet, recommender systems are becoming increasingly adapted to resolve the problem of information overload and to deal with large amount of on line information. Several approaches and techniques have been proposed to implement recommender systems. Most of them rely on flat data representation while most real world data are stored in relational databases. This paper proposes a new recommendation approach that explores the relational nature of the data in hand using relational Bayesian networks (RBNs).

## I. INTRODUCTION

Recommender systems [17] are being increasingly adopted in a wide range of applications, mainly in e-commerce applications as they help increase on line sales and improve customer loyalty. Several approaches based on data mining techniques have been developed in order to tackle this issue. Especially, probabilistic graphical models [10] are among the most used and accurate techniques in this context [14]. Most of them commonly work with flat data representation, in contrast, a large portion of real-world data is stored in relational database systems whose transformation into flat representation leads to the loss of a deeper understanding of the designed domain and the omission of useful relationships.

This paper proposes to take into account the relational data representation by using probabilistic relational models (PRMs) as a basis of recommender systems. PRMs emerge as a new family of graphs that allow the representation of a joint probability distribution over the attributes of a relational database. We are in particular interested in relational Bayesian networks (RBNs) [16], [5]. which allow to represent uncertainty over objects and relations while using the entire rich structure of relational databases. We describe a new RBN-based recommender system architecture that applies probabilistic inference over objects to provide personalized recommendations.

The remainder of this paper is structured as follows. In sections 2 and 3 we provide a brief representation of recommender systems and relational Bayesian networks respectively. In section 4, we introduce our new approach. In the final section, we summarize conclusions reached and we outline directions for future research.

## II. RECOMMENDER SYSTEMS

Recommender systems [17] emerged in the mid-1990s as a new research area whose interest has increased recently with the intension of reducing part of the information overload

problem produced on the Net. They are invoked in many Internet sites such as Amazon, YouTube, Yahoo, Netflix, etc. In 2009, Netflix awarded a million dollar prize to the team that first succeeded in improving substantially the accuracy of predictions of its recommender system<sup>1</sup>. The ultimate goal of a recommender system is to deliver a list of personalized recommended items to a particular user within a specific domain.

Several recommender systems have been developed. Nonetheless, collaborative filtering and content-based approaches stay the most familiar and mature ones [14]. The former attempts to identify groups of users with similar tests as the active user and recommends items that they have liked. The latter learns to recommend items that are similar to those the user has liked in the past. Data mining techniques have been largely applied for the first as for the second approach.

Content-based recommendation approaches analyze features of items previously rated by a user in order to build a profile of user interests. Then, the recommendation process consists in matching up the attributes of the user profile against the attributes of a content item with the intension of providing the user’s level of interest in that item.

While content-based recommender systems need only ratings provided by the active user to build her own profile, collaborative filtering methods need ratings from other users in order to find users that have similar tastes since they rated the same items similarly. Then, only the items that are most liked by this group will be recommended.

Each of these approaches presents some deficiencies: Collaborative filtering suffers essentially from the sparsity of the user-item rating matrix and the cold start problem which occur when recommendations must be made on the basis of few recorded ratings. While content-based approaches have also a start-up problem in that they must build a reliable classifier about user with very few ratings. These deficiencies are among the most challenging issues when conceiving a recommendation approach. Thus, some research established trade-offs between these two approaches in order to provide hybrid systems that overcome the shortcomings of each [1].

On the other hand, several data and knowledge sources can be available for a recommender system, however, their exploitation depends on the used recommendation technique.

<sup>1</sup><http://www.netflixprize.com/>

Various recommendation approaches are derived from machine learning techniques that rely on simple data representation of the user-item rating matrix. Some avenues of research attempt to use additional domain knowledge to the classical user-item interaction with the intension to achieve better performance [3], [20], [2].

Relational databases representation dominates computer industry mainly for storing and retrieving data. So, we propose a new hybrid recommendation approach, based on relational Bayesian networks. This latter rely on the entire rich structure of relational databases instead of flat data representation. Before describing our approach, we give a brief representation of RBNs.

### III. RELATIONAL BAYESIAN NETWORKS

Probabilistic graphical models (PGMs) [10] offer a framework including famous probabilistic graphical methods such as Bayesian networks that efficiently encode and manipulate probability distributions over high-dimensional spaces. Despite their enhanced use in various real world applications (e.g. web search, medical and fault diagnosis, speech recognition, natural language processing), input data used to construct such models is in the form of a flat set of instances without taking into account the structure of data. This eliminates a great resource of information that can be extracted. Recently, there has been growing interest in extracting interesting statistical patterns from relational models which are the most common representation of structured data. Namely, Probabilistic Relational Models (PRMs) emerge as a new family of PGMs to represent the uncertainty over the properties of an entity, capturing its probabilistic dependence both on other properties of that entity and on properties of related entities [12]. PRMs are classified into three main groups depending on their graphical representation: Relational Bayesian Networks (RBNs) representing directed acyclic graphs [16], [5], Relational Markov Networks (RMNs) representing undirected graphs [19] and Relational Dependency Networks containing both directed and undirected components [12].

In this paper we focus on RBNs which are an extension of Bayesian networks (BNs) [15] in the context of relational data, where the probability model specification concerns classes of objects rather than simple attributes.

A RBN  $\Pi$  for a relational schema  $\mathcal{R}$  (i.e., set of entities and relations) is defined through a qualitative dependency structure  $\mathcal{S}$  and a set of parameters associated with it  $\theta_{\mathcal{S}}$ . The relational schema  $\mathcal{R}$  describes a set of classes  $\mathcal{X} = \{X_1, \dots, X_n\}$ , each of which has a set of descriptive attributes denoted by  $\mathcal{A}(X)$ , which take on a range of values  $\mathcal{V}(X.A)$  and a set of reference slots denoted by  $\mathcal{R}(X) = \{\rho_1 \dots \rho_k\}$ . Each  $X.\rho$  has  $X$  as domain type and  $Y$  as a range type, where  $Y \in \mathcal{X}$ . A sequence of slots  $\rho_1 \dots \rho_k$ , where  $\forall i, Range[\rho_i] = Dom[\rho_{i+1}]$  defines a slot chain  $K$ . The notion of aggregation is also adopted from the database theory: An aggregate  $\gamma$  takes a multiset of values of some ground type, and returns a summary of it.

Formally,  $\Pi$  is defined as follows. For each class  $X \in \mathcal{X}$  and each descriptive attribute  $A \in \mathcal{A}(X)$ , we have:

- A set of parents  $Pa(X.A) = \{U_1, \dots, U_l\}$ , where each  $U_i$  has the form  $X.B$  if it is a simple attribute in the same relation or  $\gamma(X.K.B)$ , where  $K$  is a slot chain and  $\gamma$  is an aggregate of  $X.K.B$ .
- A legal conditional probability distribution (CPD),  $P(X.A|Pa(X.A))$ .

A skeleton structure  $\sigma_r$  of a relational schema specifies a set of objects and relations that hold between them in a specific domain, without specifying values of probabilistic attributes. Thus, given a relational skeleton  $\sigma_r$ , the RBN  $\Pi$  defines a distribution over the possible worlds consistent with  $\sigma_r$ .

On other words, A RBN  $\Pi$  together with a relational skeleton  $\sigma_r$  define an instance dependency structure  $\mathcal{I}$  known as the ground Bayesian network  $GBN = \langle \mathcal{G}_{\sigma_r}, \theta_{\mathcal{G}_{\sigma_r}} \rangle$ :

- $\mathcal{G}_{\sigma_r}$  represents its qualitative structure, whose nodes correspond to the set of descriptive attributes of entities in  $\sigma_r$ , they are the random variables of the model. There are a directed edge from  $y.B$  to  $x.A$  if  $y.B$  is an actual parent of  $x.A$  as defined previously.
- $\theta_{\mathcal{G}_{\sigma_r}}$  represents the quantitative parameters of the network. They are defined by the CPDs in the RBN, with the same CPD used multiple times in  $\mathcal{G}_{\sigma_r}$ .

*Example 1:* Figure 1(a) presents a relational schema for a simple movie domain, having three relations: Movie, Item and Rating. each of these entities has a set of descriptive attributes. *Rating.MovieID* and *Rating.UserID* are reference slots that reference respectively objects from the Movie and User relations. Figure 1(b) we present the RBN dependency structure: The *User.occupation* depends on his *age* and the *Rating.timeStamp* depends on *Rating.User.Age*

Figure 2(a) presents a relational skeleton for the movie domain, containing 3 users, 5 movies and 9 rating instances.

Figure 2(b) gives the ground Bayesian network of the skeleton in figure 2(a).

As with standard Bayesian networks, the joint distribution over the instantiations compatible with our particular skeleton  $\sigma_r$  is factored, which leads to the following chain rule:

$$P(\mathcal{I}|\sigma_r, \mathcal{S}, \theta_{\mathcal{S}}) = \prod_{X \in \mathcal{X}} \prod_{x \in \sigma_r(X)} \prod_{A \in \mathcal{A}(X)} P(x.A|Pa(x.A)) \quad (1)$$

### IV. RELATIONAL RECOMMENDER SYSTEMS

Only few works used RBNs to model recommender systems. [7] describe how to apply RBNs in the context of collaborative filtering, and emphasize their ability to deal with much more relational information available than the simple user-item relationships. [13] use RBN with class hierarchies [5] to the movie recommendation task and show that their approach achieves state-of-the-art results. [4] use RBNs in the context of collaborative filtering in order to improve recommendation quality for low grade users. This combination improves the recommendation quality but sacrifices the efficiency. [9] treat the recommendation problem as a special type of the relational learning problem. The idea is to estimate a RBN model and to

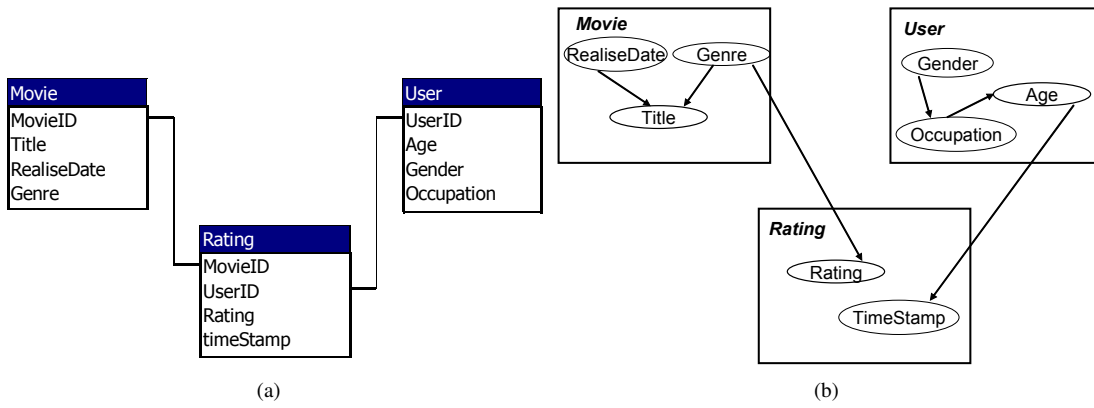


Fig. 1. 1(a) A Relational schema for a simple movie domain. 1(b) The RBN dependency structure for the movie example.

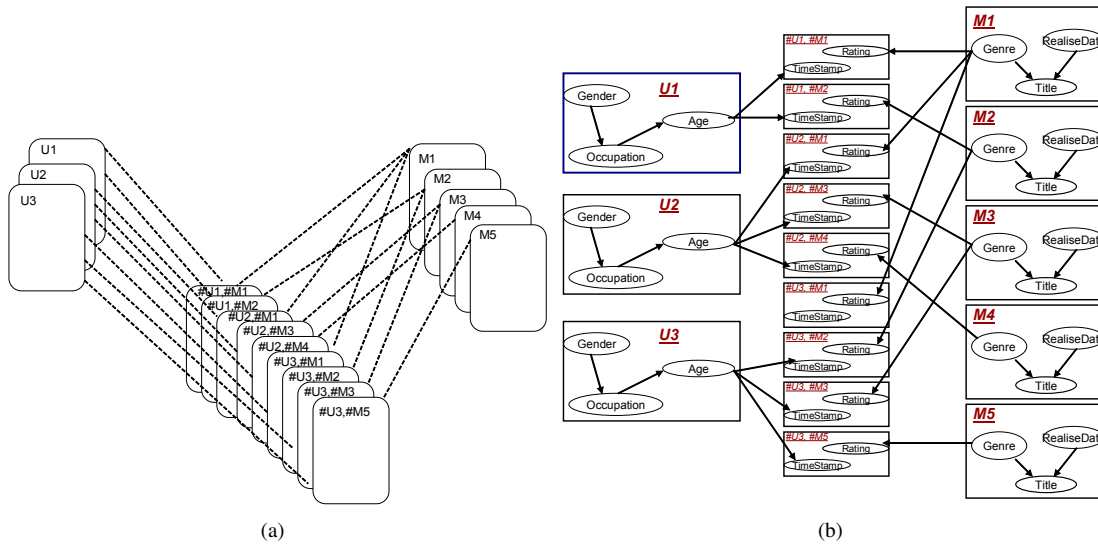


Fig. 2. 2(a) An example of a relational skeleton for the movie domain. 2(b) The underlying instance dependency graph.

specify a set of relevant attributes to perform recommendation. Using all the relevant attributes the model is considered as a unified recommender system that combine several recommendation approaches at once. Otherwise, the model can be similar to one recommendation technique: user-based collaborative filtering, content-based or demographic filtering, depending on the selected relevant attributes used to perform prediction.

Most of these methods tackle the collaborative filtering issue. In what follows, we will describe a new RBN-based recommendation approach. Our approach uses RBNs to model the recommender system domain. Then, resorts to probabilistic inference to provide recommendation.

#### A. A new RBN-based recommender system architecture

Relational data base representation is a natural data structure representation to storage and easily and efficiently handle large data sets. In each recommender system, a set of data about users as well as items is available: A user can rate many items. Mutually, an item can be chosen by many users. Besides, to each user we can storage a set of its demographic information such as age, gender, location, etc. On the other hand, a set of

items characteristics is also provided. For instance, figure 1(a) can be considered as a simplified relational representation of the movie recommendation domain.

In this work, we suppose that the recommendation domain data is stored in a relational data base, and a RBN over its relational schema (i.e., the class dependency graph and the CPD of each variable) is constructed either by an expert or by a RBN learning algorithm [8].

The RBN qualitative dependency graph  $\mathcal{S}$  represents two types of links:

- **Intra-class links:** connecting attributes of the same class. This type of connections allows to find dependencies among features of the same class.
- **Inter-class links:** connecting attributes of different classes. This type of connections allows to find dependencies among features of different classes.

Our RBN-based representation allows to benefit from different recommendation approaches at once. So the system does not rely exclusively on users or items similarities. Rather, different features are thrown together into a single recom-

mendation process while optimizing the selection of the most correlated and most relevant attributes for the recommendation through the graph dependency structure. It is a feature combination hybrid method [1]. In the following we present the different steps of our recommendation process.

1) *Modification of the initial RBN*: The initial qualitative dependency graph  $\mathcal{S}$  given as input from an expert or using a learning process, contains a *Rating* class. Each instance dependency structure  $\mathcal{I}$  defined on  $\mathcal{S}$  will contains instances of the *Rating* class. These objects represent the rating values of items that are really rated by users. However, we are in need of representing some fictitious objects of this class. These latter represent rating values of items that are not yet rated by the users and we aim to predict their values. The probability that a user will rate a particular item depends not only on the item and user attributes but also on the previous ratings of this item by the user's neighbors. Such a dependency is not allowed in class dependency graph having only one *Rating* class as it causes a cycle. Consequently, we propose to change the initial representation of  $\mathcal{S}$  by duplicating the *Rating* class:

- **Sound-Votes**: This class represents objects that really exist.
- **Forecast-Votes**: This class represents objects that we suppose their presence and we want to find those that are more likely to exist. Forecast-Votes is just a copy of the Sound-Votes class. It contains exactly the same inter and intra dependencies to which we only add a new dependency from *Sound - Votes.Rating* to *Forecast - Votes.Rating*.

*Example 2*: The relational representation of the movie recommendation domain given in figure 1(b) is modified as represented in figure 3.

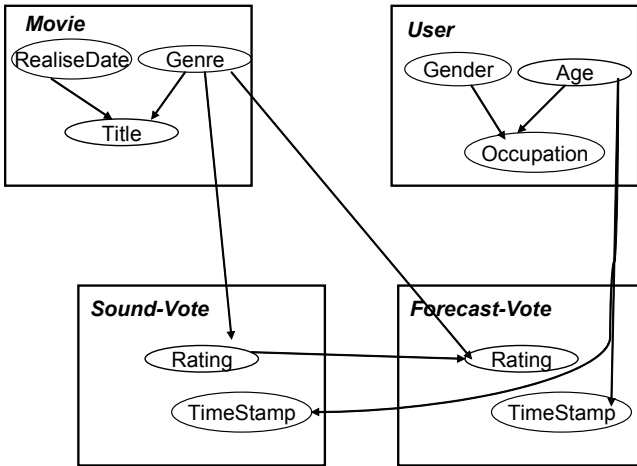


Fig. 3. The new qualitative dependency graph of the movie domain.

The new qualitative dependency graph  $\mathcal{S}'$  presents a pattern to use whenever we need to make a personalized recommendation: As it is independent of the skeleton, regardless the user to which we will make recommendation, dependencies among attributes of the objects forming its relational skeleton

are those present in  $\mathcal{S}'$ . Consequently, having the graph dependency structure and our active user, all that remain is to define the set of objects that should be invoked in the recommendation process and dependencies among them.

2) *RBN instantiation*: For a recommender system, the number of existing items and users may dramatically increase, especially when it is dedicated for on line use. Thus, the construction of a RBN instance of all users and items of the system database is a daunting task. In addition, it has to be redone for each time where a new user or item integrates the system. Moreover, inference in such networks is a time consuming task. Therefore, for each active user we will build an instantiation and this is by following a set of rules to find the elements of an RBN instantiation appropriate to a particular user.

The active user profile is not yet based on items previously rated by the user, or by her similarities with a group of users. Rather, it is based on both of them. In addition it is refined by digging dependencies intra and inter-classes.

Looking for the similar users to recommend items that they have previously tasted does not necessarily mean they will appeal to the active user. So having an active user, its neighborhood and all the items they recommended, the idea is to search through these items those which seem most pertinent.

More precisely, given an active user  $u_a$ , his profile is presented via a relational skeleton  $\sigma_{u_a}$  that defines the possible instantiations we consider. It specifies the set of objects for each class and the relations that hold between the objects. We will build his own relational skeleton, in order to use it later to predict his preferences. Based on basic SQL queries, the skeleton of the active user  $u_a$  contains as objects:

- $I_{u_a}$  = the set of items rated by  $u_a$  and all items having similar attributes values.
- $U_{u_a}$  = the set of users who rated at least an item of  $I_{u_a}$ , which we refer to as  $u_a$  neighborhood.
- $I_{others}$  = All other items rated by a user  $u \in U_{u_a}$ .
- $V_{u_a} = \{\{u_r\} \times I_{u_a}\}$  the set of Vote objects of  $u_r$ .
- $V_{U_{u_a}} = \{U_{u_a} \times \{I_{u_a} \cup I_{others}\}\}$  the set of Vote objects of  $U_{u_r}$ .  
 $\Rightarrow V_{u_a}$  and  $V_{U_{u_a}}$  belong to the Sound-Votes class.
- $FV_{u_a} = \{\{u_a\} \times I_{others}\}$  the set of Forecast-Votes objects representing vote of item objects which are not yet rated by the active user.
- In addition, we should specify for each one of these new objects the set of Sound-Votes objects on which they depend probabilistically. Our aim is to compute  $P(FV.rating)$  for each object FV of the Forecast-Votes class. So we have to define the set of parents of  $P(FV.rating)$  denoted by  $Pa(FV.rating) = Pa_{u_a} \cup Pa_i \cup Pa_{sv}$ , where:
  - $Pa_{u_a}$  is the set of descriptive attributes from the active user object on which  $FV.rating$  depends.
  - $Pa_i$  is the set of descriptive attributes from the considered item object on which  $FV.rating$  depends.
  - $Pa_{sv}$  is the set of all  $sv.rating$  attributes of Sound-Votes objects on which  $FV.rating$  depends. Each

$FV$  object related to the active user  $u_a$  and an item  $i$  has its  $FV.rating$  depends on  $sv.rating$  attribute of each  $sv$  object on which either  $u_a$  or  $i$  or an item  $j \in I_{u_a}$  are invoked.

*Example 3:* Suppose that we consider the qualitative dependency graph of figure 3 and the qualitative dependency graph of figure 1(b). We note the active user by  $u_1$ ,  $I_{u_1} = \{i_1, i_2\}$ ,  $U_{u_1} = \{u_2, u_3\}$ ,  $I_{u_2} = \{i_1, i_3, i_4\}$  and  $I_{u_3} = \{i_1, i_2, i_3, i_5\}$ .  $V_{u_a}$  and  $V_{U_{u_a}}$  represents all rating objects from the *Sound – Votes* class. Then, this skeleton is enriched by as many rating objects, from the *Forecast-Votes* class, as items that are not currently connected to the active user. These latter are dashed in figure 4.

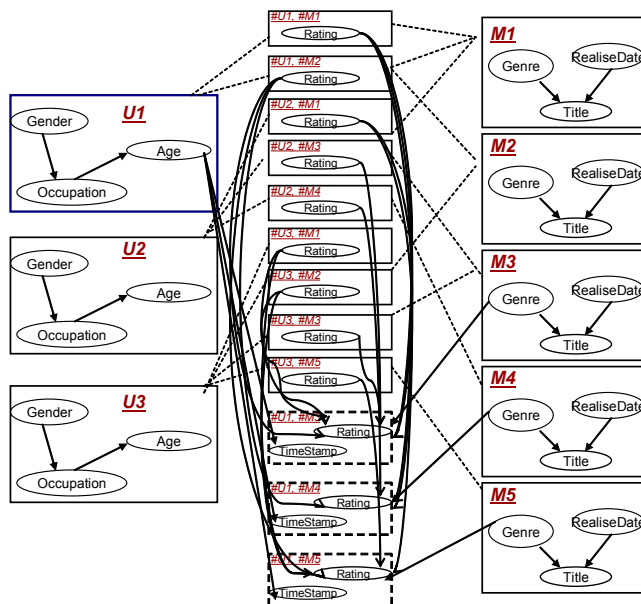


Fig. 4. The relational skeleton of the active user  $u_1$ .

3) *Performing recommendation:* The relational skeleton of an active user, together with the modified RBN, define the instance dependency structure  $\mathcal{I}_{u_a}$  known as the ground Bayesian network of  $u_a$ . All that remains is to perform probabilistic inference on all the objects of the *Forecast – Votes* class. We should predict for the active user, the items that belong to his relational skeleton and which are the most appropriate to his tastes. This reverts to calculate  $P(Rating.rating|Pa(Rating.rating))$  for each of the *Forecast – Votes* objects. Relevant objects are those having the highest probability value. A probability threshold can be defined to determine the list of elements to be delivered to the active user.

## B. Discussion

To summarize, our approach consists of three components:

- First, from an initial relational schema and a RBN  $\Pi$  associated with this schema, we construct a new RBN  $\Pi'$  by dividing the class associated to the ratings into two classes, namely, *Sound-Votes* and *Forecast-Votes*.

- Then, we create a relational skeleton  $\sigma_{u_a}$  to each active user based on a set of rules.  $\sigma_{u_a}$  and the set of conditional probability distributions of  $\Pi'$  form the ground Bayesian network of  $u_a$ .
- Finally, using an inference process we compute  $P(Rating.rating|Pa(Rating.rating))$ , relevant objects are those having the highest probability value.

The overall process is represented by figure 5.

On one hand, what we promote is different from these earlier proposed methods. [7], [4], [13] are interested in collaborative filtering technique, however our approach is an hybrid one. The hybrid approach proposed in [9] may collapse to one of the standard techniques used in recommendation. The ours implies usually features from different recommendation data sources as this combination allows to reach better accuracy.

On the other hand, each recommendation approach needs to enhance its performance by addressing key challenges in the field of recommender systems. Here we describe the most known ones [18], [11], and we position our RBN-based modeling approach with respect to these issues.

1) *Cold start problem:* When a new item or a new user join the system, it is hard to find its group as there is not enough information about it. A new item needs to be rated to be classified and a new user has not yet a purchase history to provide appropriate recommendations. Our approach allows to define a neighborhood to a new user based on the items that he voted for. Also a new item can be integrated as not only the set of items rated by  $u_a$  is invoked but also and all items having similar attributes values.

2) *Scalability:* Often we need high scalability of a recommender system especially when it is used in real-time application. here trade offs between scalability and prediction performance have to be established. Dealing with recommendation as an inference task in a probabilistic graph can make it a time consuming process. That's why we resorted to a set of rules to generate a ground Bayesian network of a reduced size. Also, if needed, we can improve our selection approach using similarity measures.

3) *Neighbor transitivity:* Where users of similar tastes are not identified by the system as they voted for different items: e.g., user A votes for two action movies and user B votes for two other action movies, so they are not identified similar whereas both like action movies. According to our skeleton definition, all the items of action genre will be invoked, and consequently user A and user B will be considered as similar users.

## V. CONCLUSION

In this work, we have emphasized the relational nature of recommender systems by the use of RBNs. Our approach allows the integration of relational data into the recommendation process. So, features related to users and items are exploited to provide useful information during recommendation. We have designed the recommendation task using a RBN. Then, we resorted to probabilistic inference to predict user's preferences.

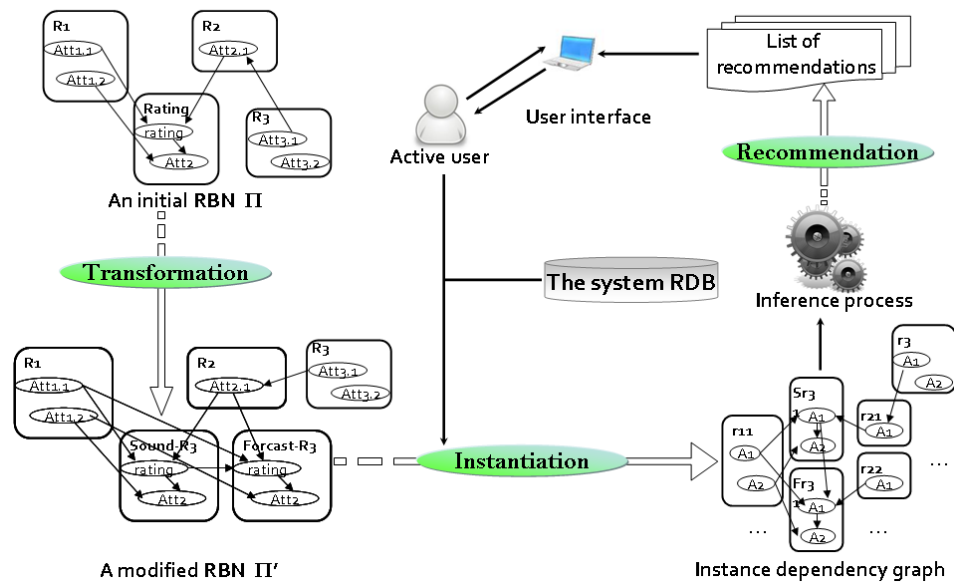


Fig. 5. The overall architecture of the RBN-based recommender system

In our ongoing work, we focus on the implementation of the RBN framework in order to apply the proposed architecture on real recommendation tasks and compare it to state-of-art methods.

#### REFERENCES

- [1] R. Burke, *Hybrid Recommender Systems: Survey and Experiments*, User Modeling and User-Adapted Interaction, 12(4), 331–370, 2002.
- [2] W. Carrer-Neto, M.L. Hernandez-Alcaraz, R. Valencia-Garca, F. Garca-Snchez, *Social knowledge-based recommender system. Application to the movies domain*, To appear in Expert Systems with applications, 2012.
- [3] Y. Deng, Z. Wu, H. Si, H. Xiong, Z. Chen, *A Collaborative Filtering Approach to Making Recommendations Based on Ontology in the Movie Domain*, Energy Procedia. 13, 228–236, 2011.
- [4] Y. Gao, H. Hong, J. Liu, D. Liu, *A Recommendation Algorithm Combining User Grade-Based Collaborative Filtering and Probabilistic Relational Models*, Proceedings of the Fourth International Conference on Fuzzy Systems and Knowledge Discovery, pp. 67–71. Washington, DC, USA, IEEE Computer Society, 2007.
- [5] L. Getoor, *Learning statistical models from relational data*, PhD thesis. Stanford University, 2002.
- [6] L. Getoor, N. Friedman, D. Koller, B. Taskar, *Learning Probabilistic Models of Link Structure*, Journal of Machine Learning Research. 3, 679–707, 2002.
- [7] L. Getoor, M. Sahami, *Using Probabilistic Relational Models for Collaborative Filtering*, Working Notes of the KDD Workshop on Web Usage Analysis and User Profiling, 1999.
- [8] N. Friedman, L. Getoor, D. Koller, A. Pfeffer, *Learning probabilistic relational models*, Proceedings of the International Joint Conference on Artificial Intelligence, 1300–1309, 1999.
- [9] Z. Huang, D. D. Zeng, H. Chen, *A unified recommendation framework based on Probabilistic Relational Models*, Proceedings of the fourteenth Annual Workshop on Information Technologies and Systems. pp. 8–13, 2004.
- [10] D. Koller, N. Friedman, *Probabilistic Graphical Models*. Cambridge, MIT Press, 2009.
- [11] L. Lü, M. Medo, C. H. Yeung, Y-C. Zhang, Z-K. Zhang, T. Zhou, *Recommender systems*, Physics Reports 519(1),1–50, Elsevier 2012.
- [12] J. Neville, D. Jensen, *Relational dependency networks*, Journal of Machine Learning Research. 8, 653–692, 2007.
- [13] J. Newton, R. Greiner, *Hierarchical probabilistic relational models for collaborative filtering*, Proceedings of the 21st International Conference on Machine Learning, Workshop on Statistical Relational Learning. pp. 249–263, 2004.
- [14] D. H. Park, H. K. Kim, I. Y. Choi, J. K. Kim, *A literature review and classification of recommender systems research*, Expert Syst. Appl. 39(11), 10059–10072, 2012.
- [15] J. Pearl, *Probabilistic reasoning in intelligent systems*. San Francisco, Morgan Kaufmann, 1988.
- [16] A. J. Pfeffer, *Probabilistic Reasoning for Complex Systems*, PhD thesis. Stanford University, 2000.
- [17] F. Ricci, L. Rokach, B. Shapira, P. Kantor, *Recommender Systems Handbook*. New York, USA, Springer, 2010.
- [18] X. Su, T. M. Khoshgoftaar, *A survey of collaborative filtering techniques*, Adv. in Artif. Intell. 2009. Hindawi Publishing Corp. 2009.
- [19] B. Taskar, P. Abbeel, D. Koller, *Discriminative Probabilistic Models for Relational Data*, Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence. pp. 485–492, 2002.
- [20] N. Zheng, Q. Li, *A recommender system based on tag and time information for social tagging systems*, Expert Syst. Appl. 38(4), 4575–4587, 2011.