

# Hierarchical Naive Bayes Models for Representing User Profiles

Juan F. Huete, Luis M. de Campos, J.M. Fernández-Luna, M.A. Rueda-Morales  
Dpt. of Computer Science and Artificial Intelligence  
University of Granada  
18071 Granada, Spain  
(jhg,lci,jmfluna,mrueda)@decsai.ugr.es

## ABSTRACT

In this paper, we show how a user profile can be enhanced when a more detailed description of the products is included. Two main assumptions have been considered: the first implies that the set of features used to describe an item can be organized into a well-defined set of components or categories, and the second is that the user's rating for a given item is obtained by combining user opinions of the relevance of each component.

## Categories and Subject Descriptors

H.3 [Information Storage and Retrieval]

## General Terms

Algorithms

## Keywords

Probabilistic Reasoning, Recommender System, Recommender Systems, Learning User Profiles

## 1. INTRODUCTION

Recommender systems (RS) [4] attempt to automatically link users with the products that they might potentially 'consume' (buy, look at, listen to, read, etc.). This paper focuses on content-based RSs which analyze item descriptions to identify those that are of particular interest to the user. In particular, using the information in the database of ratings (obtained either explicitly by means of ratings or implicitly by using visited links, etc.) and the description of the item's content, a user profile is built. The predictions for unobserved items are then obtained by considering their similarity with the user profile.

The items are traditionally described by means of a set of attributes or features and represented as (weighted) vectors which are equal in size to the total number of attributes

[4]. In this paper, we shall attempt to improve the description of the user's profile by means of a more thorough description of an item's content. In particular, we shall assume that the set of features used to describe an item can be organized into a well-defined set of  $k$  structural components or categories  $\mathcal{H} = \{H_1, \dots, H_k\}$ . For instance, the categories used to describe a movie might be **title**, **genre**, **actors**, **keywords**, **etc..** In order to describe each component,  $H_i$ , a subset of a large number  $l_i$  of features  $\mathcal{F}_i = \{F_{i,1}, F_{i,2}, \dots, F_{i,l_i}\}$  can also be used. For instance, in order to describe the component **genre** of a movie, we can use a subset of **{action, comedy, sci-fi, etc.}**.

Although we might have a better product description, it is quite common for a user only to rate the observed products with an overall rating. We may assume, however, that this rating is a mixture of the user's opinions about the relevance of those components describing the item. In this article, we will study how this kind of information may be included in an enhanced user profile. This profile could have two different parts: one which determines how the user might rate each content component, and the other representing how this information is combined in an overall rating. By using this profile, recommendation explanations are better since we might inspect the rules governing prediction of a particular instance<sup>1</sup>.

In this paper, we propose to model the profile using Bayesian Networks. From a recommending perspective, our final goal is to predict how an individual would rate an as yet unseen item. In this case, the prediction process can be seen as a classification problem: an item  $I$  must be assigned to one of a set of  $r$  pre-defined ratings,  $v_j$ , (categories). More specifically, the predicted rating is obtained by selecting  $vote = \arg \max_j Pr(v_j|I, \theta)$ .

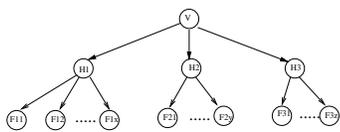
## 2. LEARNING THE USER'S BEHAVIOR

We will learn the user profile by considering a training database where each entry,  $d_i$ , contains an overall rating given to the observed item and the set of features used to describe the item, i.e.  $d_m = \{v_i^m, f_{1,1}^m, \dots, f_{k,l_k}^m\}$ . In this paper, three different alternatives will be considered:

- *NB*: Firstly, we consider a flat description of an item (no structural information is used). In this case, assuming that the rating (class) is known and the features are independent, a Naive Bayesian classifier (NB) might be used. It should

<sup>1</sup>This is important because if users trust the system, they are more likely to buy the items offered; if they don't, they will stop using it.

be noted that the NB classifier has proved to perform well [1] when the dimensionality of the input space is high.



U	N	{H1,H2,H3,H4}
A	438	{246, 87, 33, 22}
B	393	{54, 21, 26, 23}
C	626	{92, 41, 40, 24}
M	500	{84, 19, 26, 24}
D	500	{84, 19, 26, 24}

Figure 1: HNB as profile Table 1: User Data

- **HNB**: The second alternative attempts to learn a model by incorporating structural knowledge. The proposed model might be considered as a Hierarchical Naive Bayes model (HNB) [3], i.e. tree-shaped Bayesian Networks where the variables are partitioned into three disjoint sets:  $\{C\}$  is the class variable (vote variable, in our case),  $\mathcal{F}$  is the set of features, and  $\mathcal{H}$  is the set of hidden or latent variables (see Figure 1). Although latent variables are usually introduced to relax the independence statements between attributes, in our model, these variables are given an explicit semantics since they will represent the structural components or categories such as **genre**, **actors**, etc. These components will have the same number of states as the rating variable. Since the topology is fixed, the problem is to find the most likely parameters for the given data set. For this purpose, we propose the use of the Expectation-maximization (EM) algorithm which converges to a local maximum of the likelihood function.

- **INB**: The third alternative tries to reduce the computational cost involved with the EM procedure. We therefore propose the use of  $k + 1$  independent NB models learned in a three-step process:

1. For each variable  $H_i$  learn an NB model,  $NB_{H_i}$ , from  $\mathcal{D}$
2. Create the structural training data set,  $\mathcal{D}_{ST}$ .

- 2.1 For each data  $d_m \in \mathcal{D}$ ,  $d_m = [v_m, f_{1,1}^m, \dots, f_{k,l_k}^m]$ 
  - For each  $H_i$ , let  $h_i^m = \arg \max_j Pr(h_j|F_i, NB_{H_i})$
  - Insert a new entry  $[v_m, h_1^m, \dots, h_k^m]$  in  $\mathcal{D}_{ST}$

3. Let  $NB_{VH}$  be the NB model learned from the  $\mathcal{D}_{ST}$  set.

The recommended rate will then be obtained by, firstly, predicting the rate for each structural component  $H_i$ , i.e.  $h_i = \arg \max_j Pr(h_j|F_i, NB_{H_i})$ , and then, by combining these ratings as  $rate = \arg \max_j Pr(v_j|h_1, \dots, h_k, NB_{VH})$ .

## 2.1 Experimentation

A movie recommender framework is used to illustrate the ability of the models to learn the user profile. In this article, we show the results obtained when using information about {actors  $H1$ , actresses  $H2$ , directors  $H3$ , genres  $H4$ } for five different users: three of these ( $UA, UB$ , and  $UC$ ) represent real users obtained by combining MovieLens (containing user ratings) and IMDB (used to obtain the content descriptions)<sup>2</sup>. On the other hand, and in order to check the system’s ability to learn the user’s behaviour, we have used two artificial individuals,  $UM$  and  $UD$ : the first where the movie rating is defined as the majority of the ratings given to each component, and the other where the rate is the same as the one used to rate the *actors* ( $H1$ ) component. Table 1 shows the number of records,  $N$ , and the number of features in each structural component. The data sets have been split at 80% for training and 20% for testing purposes.

In order to measure the accuracy of the RS, we have considered two metrics [2]: *MAE*, which measures the average

<sup>2</sup>Since IMDB provides a cast overview “in credits order”, actors, actresses and directors have been filtered using their average gross at the box office: [www.the-numbers.com](http://www.the-numbers.com).

absolute deviation between a predicted rate and a user’s true rate, and *F1*, which balances recall and precision when trying to find good items<sup>3</sup>. The table below shows the results obtained for each candidate model (we also show the results obtained by considering each structural component separately). It is possible to draw various conclusions from this table. The first is that by using a structural profile (INB) we might improve system accuracy. This profile is particularly beneficial when considering synthetic data since all the structural components which affect the overall rating may have been considered. In such situations, HNB obtains very good results. In addition, good results are obtained by only using genres,  $H4$ , since we have more data for learning: on average, those features describing genres appear approximately 15 times more frequently than actors, actresses or directors.

	UA		UB		UC		UM		UD	
	mae	F1								
NB	1.10	0.70	0.99	0.73	1.34	0.42	1.04	0.54	0.56	0.80
HNB	1.72	0.42	1.04	0.71	1.35	0.46	0.68	0.70	0.32	0.89
INB	1.07	0.65	0.95	0.74	1.40	0.48	0.96	0.60	0.40	0.85
H1	1.33	0.67	1.19	0.76	1.55	0.30	1.41	0.31	0.40	0.85
H2	1.72	0.68	1.09	0.77	1.32	0.19	1.23	0.52	1.65	0.21
H3	1.34	0.69	1.12	0.76	1.28	0.28	1.11	0.59	1.79	0.16
H4	1.30	0.69	1.00	0.73	1.39	0.48	1.15	0.49	1.53	0.18

Although we have focused on system accuracy, one important fact is that by means of this kind of profile better explanations could be obtained for the recommendations, for instance by considering the values of those structural components which better justify the evidence (implying an abduction process). For example, the system might indicate “you might like this movie because of its genre and its cast”.

## 3. CONCLUSIONS

In this paper, we have studied the problem of learning a profile that captures the mechanisms which a user uses to rate an item. This new user profile, represented by means of Bayesian Networks, includes information about a product’s structural content. The problem of explaining the predicted rate and the selection of the best features for recommending will be tackled in future work.

## 4. REFERENCES

- [1] P. Domingos and M. Pazzani *On the optimality of the Simple Bayesian Classifier under 0-1 Loss*. Mach. Learn. 29:103-130, 1997
- [2] J. L. Herlocker, J. A. Konstan, L. G. Terveen, J. T. Riedl. *Evaluating collaborative filtering recommender systems*. ACM Transactions on Information Systems, 22(2):5-53, 2004.
- [3] H. Langseth and T.D. Nielsen. *Classification using Hierarchical Naive Bayes Models*. Mach. Learn. 63:135-159, 2006.
- [4] P. Resnick, H.R. Varian. *Recommender systems*. Communications of the ACM, 40(3):56-58, 1997.

<sup>3</sup>We consider a movie rated with 4 or 5 as “likeable”, and a score of 1 to 3 to represent it being “irrelevant” to the user.