Using Structural Content Information for Learning User Profiles

Juan Huete
Dpt. of Computer Science and Artificial Intelligence
University of Granada
18071 Granada, Spain
jhg@decsai.ugr.es

Luis M. de Campos
Dpt. of Computer Science and Artificial Intelligence
University of Granada
18071 Granada, Spain
lci@decsai.ugr.es

J.M. Fernandez-Luna
Dpt. of Computer Science and Artificial Intelligence
University of Granada
18071 Granada, Spain
jmfluna@decsai.ugr.es

M.A. Rueda-Morales
Dpt. of Computer Science and Artificial Intelligence
University of Granada
18071 Granada, Spain
mrueda@decsai.ugr.es

ABSTRACT

In this paper we show how a user profile (in the context of a content-based recommender system and represented by means of a Bayesian Network) can be enhanced when a more detailed description of an item's content is considered. 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. This new user profile therefore consists of two different parts: one represents how the user should rate each content component, and the other represents how this information might be combined in order to obtain the overall rating.

As users do not normally express their opinions about structural components, in this paper we propose that these components be considered as hidden variables. The user profile is learnt by means of the Expectation-Maximization algorithm from the set of user's ratings.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Miscellaneous

General Terms

Content-based Recommender Systems, User Profile, Hidden Variables

IRGM Information Retrieval and Graphical Models

Keywords

Probabilistic Reasoning, Recommender System

1. INTRODUCTION

Thanks to the massive growth of Internet in terms of new content, it is possible to access an enormous amount of information about any subject. The development of new tools to improve the information seeking is a very important task and one which is generating an incredible amount of research.

Recommender Systems (RS) attempt to determine user preferences in order to predict their needs. They link the users with products that they might potentially 'consume' (buy, look at, listen to, read, etc.) by automatically associating the content of the products or the opinions of other users with the opinions or actions of those users who are interacting with them. In short, the objective of these systems is to generate suggestions about new items or to predict the utility of a specific product for a particular user. Not only do they work with 'typical objects' found on Internet (e.g. web pages, videos, images, documents in different formats, etc.) but they can also deal with any type of item or service.

The required information for operation is given by how users explicitly rate the products on the one hand (usually in the form of votes), and by implicitly observing user behavior on the other, i.e. studying previous user action (purchases, visited links, browsing habits, etc.). All of this information enables the system to be personalized and adapted to each user and to their individual characteristics and features.

Regarding the content description of the products, in traditional RSs [3, 16] the items are described by means of a set of attributes or features $\mathcal{F} = \{f_1, f_2, \ldots, f_l\}$. The products may therefore be seen as vectors in a vector space where the dimension equals the total number of attributes of $\mathcal{F}$. A value of 0 will be assigned in those positions in these vectors where the corresponding features do not occur in the article description, and a value other than 0 will be assigned if they do. These values are known as weights and they usually express the importance of the i^{th} attribute in the corresponding product.

Using the information in the database of ratings and the

```
<movie>
  <movieID> 1294 </movieID>
  <title> Batman (1989)</title>
  <actors>
    Keaton M., Nicholson J., Basinger K., Cory P.
  </actors>
  <director> Burton Tim </director>
 <genres> Action Fantasy Thriller  </genres>
 <keywords>
   hostess obsessive-love dc-comics joker rivalry
 </keywords>
 <plot> Gotham City: dangerous, corrupt police, etc.
 </plot>
  ....
</movie>
```

Table 1: Example of an item's XML description

description of the items' content, the user profile is built and represented in the same way, i.e. as an attribute vector containing the corresponding weights of each attribute.

Finally, the predictions are obtained by considering a similarity measure between the user profile and an unobserved item. The cosine similarity [20] is the most commonly used function. Other alternative techniques are based on data mining [21].

As pointed out in [2] *'most of the recommendation methods produce ratings that are based on a limited understanding of users and items as captured by user and item profiles and do not take full advantage of the information in the users transactional histories and other available data'*.

In this paper we shall focus on this direction and 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, on the one hand, that the set of features used to describe an item can be organized into a well-defined set of components or categories. For instance, the categories used to describe a movie might be `title, genre, actors, keywords, etc.` and in a vacation context the components of a hotel RS might be `location`, `activities`, `services, etc.`. It should be noted that this kind of information is currently available in most cases thanks to the use of the XML standards for sharing data. For example, Table 1 shows an XML description of the movie *Batman*. The second assumption is that the user's rating for a given item is obtained by mixing the user's opinions of the relevance of the set of components which describe an item. For example, I would like the movie Batman because I like its genre and its cast.

In this paper we shall show how by taking these two assumptions into account, the system predictions may be improved and the explanations of the recommendations are better. From a commercial point of view this is important because if users trust the system, then they are more likely to buy the items which this offers, but if they don't, then they will stop using it.

For practical purposes, however, it is quite common that we only know the overall rating which the user has given to the observed item (usually explicitly by submitting the rating in an online system or simply by means of a click on a web page), and we do not, therefore, know the relevance of the structural components. In this paper, we shall also present a methodology that is based on the Expectation-Maximization algorithm and which is able to capture (learn) how a user rates each different structural component and also how this information might be combined in order to obtain the overall rating. As a result, a better representation of the user profile is obtained.

The first section of this paper briefly introduces content-based recommender systems. Section 2 describes the probabilistic framework modeling the user's behavior and how it can be used to predict the active user ratings in Section 3. Section 4 explains how the user's rating behavior can be learnt when the structural component ratings are missing. Section 5 explores the experimental results obtained. Finally, Section 6 presents our conclusions and future lines of research.

2. CONTENT-BASED RECOMMENDER SYSTEMS

There are three main types of RS depending on how the recommendations are performed:

- content-based: recommended products are similar to those which the active user [1] rated positively in the past.
- collaborative-based: products consider to be good for other users with similar tastes to the active user are recommended.
- hybrid: recommendation is achieved by combining content- and collaborative-based approaches.

In [26, 4], an alternative classification with a finer granularity is presented. For further information about the state-of-the-art of this recommending field, we would like to refer our readers to the papers [1, 2, 4, 8, 9, 13, 18, 22, 24, 26] which offer a more exhaustive overview of both classic and more modern techniques.

We shall focus on the first type of recommender system as this is the type of model presented in this paper. In content-based recommendation methods, the utility of item I for a user is estimated by taking into account the utility assigned by the user to other items. Traditional heuristics are mostly based on information retrieval methods. A typical example of a real system is *Fab* [3] which recommends web pages based on the 100 most important words in each HTML document. Other techniques for content-based recommendations have also been used (e.g. Bayesian classifiers [17, 12]) and various machine learning techniques which include clustering, decision trees, and artificial neural networks [17] (mainly for user modeling problems), specifically for acquiring models of individual users interacting with an information system. A more in-depth description of the content-based technique can be found in [26].

2.1 Content Recommender Systems based on Bayesian Networks

In this section, we will briefly mention some of the papers which (like ours) explore the application of Bayesian Networks to the content-based recommendation problem, and

[1]The active user is the person who the system is recommending for.

more specifically, the construction of a user profile by means of these types of probabilistic graphical models.

In [19], Schiaffino and Amandi develop a technique that allows reasoning based on cases and Bayesian Networks to be combined. The aim is to build user profiles incrementally. More specifically, the graphical model is used to model the user information needed. The network comprises nodes representing the attributes that the user has used to express its query as well as other nodes that are added and are related to the former. The arcs linking the nodes are previously established as they depend on the domain. Inference in the graph produces the a posteriori probabilities of the nodes that represent more general concepts.

In [25, 5], a method for building user profiles using Bayesian Networks is proposed in the context of document filtering. From the set of documents that the user has judged to be relevant or not, a network is learnt by coding user preferences. Once again, the documents are sorted in descending order of their a posteriori probability.

In [15], a Bayesian network is used to learn a model that represents a questionnaire completed by users of an educational platform. This will be used as an individual profile employed to determine the order of the educational units shown to users.

3. MODELING THE USER'S BEHAVIOR

Prior to modeling the user's behavior, we will introduce some notation. By way of input, our system will consider a database that represents the product descriptions. Typically, we have a large number m of items $\mathcal{I} = \{I_1, I_2, \ldots, I_m\}$. In order to describe an item, we assume that the information is organized into a well-defined set of k structural components $\mathcal{C}$, $\mathcal{C} = \{C_1, \ldots, C_k\}$. In order to describe each component, C_i, a subset of a large number l_i of features $\mathcal{F}_i = \{F_{i,1}, F_{i,2}, \ldots, F_{i,l_i}\}$ can also be used. For instance, in order to describe the component `genre` of a movie, we can use a subset of {`action, western, comedy, sci-fi, etc.`}. The overall set of features will be denoted by $\mathcal{F}$, i.e. $\mathcal{F} = \cup_i \mathcal{F}_i$. In this paper, we will refer to a given feature in $\mathcal{F}$, independently of the structural component it belongs to, by using only one sub-index as in F_j. Therefore, the content description can be viewed as a very sparse binary vector $\mathbf{\Delta}$, with $l = \sum_{j=1}^{k} l_j$ where $\delta_j = 1$ indicates the fact that the item can be described with feature F_j, and the entry is assumed to be zero otherwise.

3.1 Probabilistic framework

In this section, we will present the probabilistic framework by characterizing the user's rating pattern. This framework defines a probabilistic generative model for user ratings. In particular, we shall assume that the user's rating for a particular item is given by combining user opinions about the components that describe this item. For instance, in a movie-based system, the user rating will depend on the `genre`, `actors`, `director`, etc.

We shall assume that the user can rate a given item I using $\#r$ different values. For instance, in MovieLens, users can rate any of the movies they have seen with 1 to 5 stars, i.e {*1*, 2*, 3*, 4*, 5**}. In addition, without loss of generality, we shall assume that each category C_i can be rated with the same number of values in $\#r$.

In this setting, after an item I has been observed, each user's rating is generated according to a probability distribution θ in the following way: first, the user decides how probable each candidate rating is for all the components which describe the item, C, and then the final rating is selected by combining the individual components. Thus, given the probability distribution representing the model, θ, we can determine the probability of the final user's rating by means of

$$Pr(v_i|I,\theta) = \sum_{c \in \mathcal{C}} Pr(v_i c|I,\theta) \quad (1)$$

where $c = \{c_1, \ldots, c_k\}$ represents any configuration for the set of components ratings. In order to characterize the model with a reduced number of parameters, we need to impose certain assumptions:

A0 : given that we know how the user rates each different component of an item, $C_1, \ldots, C_k$, the overall vote, V, is independent of the features used to describe the item.

A1 : the components ratings are marginally independent. For example, knowing that the user rated the `genre` of a movie as *5** this gives no information about the probability of the `director` component being *3**.

A2 : given that we know the user rating for one component, the features describing this component are conditionally independent. For example, knowing that the `genre` is rated as *5**, then the fact that the movie is a `western` does not provide any information about the relevance of the movie also being a `comedy`.

By combining all these assumptions, the generative model can be represented by means of the Bayesian network in Figure 3.1. This is therefore the topology used to represent the user profile where the set of relations $C_i \longrightarrow \mathcal{F}_i$ represents how the user should rate each content component and the relation between$C_1, \ldots, C_k$ and V encodes that the user's final vote is determined by combining the relevance of the different components.

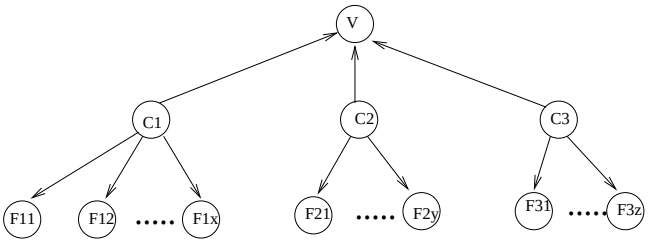

Figure 1: BN representing the generative model

3.2 Training recommender systems

Since the topology of the model is fixed, learning the model consists in estimating the parameters θ, defining the conditional probability distribution by using a set of training data $\mathcal{D} = \{d_1, \ldots, d_n\}$. Each individual data d_i represents the user's judgement for the item I_i, where $d_i = \{v_i, c_{1,i}, \ldots, c_{k,i}, f_{1,i}, \ldots f_{l,i}\}$, is a tuple where v_i is the rating of item I_i, and $c_{j,i}$, $j = 1, \ldots k$, are the ratings for each component. Finally, $f_{\bullet,i}$ will take the value 1 if the item can be described using feature $F_\bullet$ and 0 if it cannot. Our objective is to find the most probable value of θ given the

evidence of the training data and the structure of the generative model G, i.e. to find $\theta = \arg\max_\theta Pr(\theta|D, G)$. Since

$$Pr(\theta|D, G) \propto Pr(D|\theta, G)Pr(\theta|G)$$

where the quantity $Pr(D|\theta, G)$ is the likelihood function, expressing the probability of the observed data set being generated from the model represented by θ and G, and the quantity $Pr(\theta|G)$ represents the prior distribution over the parameters given the topology of the model.

Assuming that given the model, each individual's data is independent of the others, the probability $Pr(D|\theta, G)$ is obtained as the product over all the user ratings, i.e.

$$Pr(D|\theta, G) = \prod_{i=1}^{n} Pr(d_i|\theta, G)$$

Given the model topology G, the probability distribution can be factorized into a set of local distributions, one for each node X_j in the graph viewed as a function of θ_j (therefore, $\theta = (\theta_1, \ldots, \theta_s)$): $Pr(x_j|pa(x_i), \theta_j, G)$ where $pa(x_j)$ is any configuration of the parent set of X_j.

Since we are assuming that the data are drawn independently, a maximum likelihood estimate of the probabilities might be used. In certain situations, it might be that this is a combination of values that never occur together in the training set. In such cases, a frequency-based probability estimator will be zero. To overcome this problem, Laplace smoothing is usually introduced in practice and so we generally have that

$$\theta_{ijk} = Pr(x_i = k|pa(x_i) = j, \theta) = \frac{\alpha_{ijk} + N_{ijk}}{\alpha_{ij} + N_{i,j}} \qquad (2)$$

where N_{ijk} is the number of times that variable X_i takes the value k and the parent set of X_i takes the configuration j, $N_{ij} = \sum_k N_{ijk}$ and $\alpha_{ij} = \sum_k \alpha_{ijk}$ [2]. A special case of Laplace smoothing is *to add one* smoothing [11] obtained by setting $\alpha_{ijk} = 1$.

3.3 Using the recommender system

Once the parameters have been estimated from the training set, it is possible to use the model to predict how the user might rate an unobserved item based on the items previously rated by the user. In this case, the new movie plays the role of the evidence, where we consider that each feature F_s is used to describe the item as relevant and each feature which has not been used to describe the item as non−relevant and the objective is to compute $Pr(V|I, \theta)$. These evidences are introduced into the system and the a posteriori probability distribution for the user's rating will be obtained as the sum over all the possible alternatives for the variables in $\mathcal{C}$. Given the topology of the underlying BN, this probability can be expressed as

$$Pr(v_i|I, \theta) = \sum_C Pr(v_i|C_1, \ldots, C_k, \theta) \prod_{j=1}^{k} Pr(C_j) \prod_{s=1}^{l_j} Pr(f_s|C_j, \theta).$$

Since the problem is to predict how the current user might rate this unobserved item, the rate with the highest posterior probability is selected, i.e.

$$\arg\max_j Pr(v_j|I, \theta)$$

[2]The values α can be viewed as the parameters of a Dirichlet distribution over the a priori distribution $P(\theta|G)$.

3.3.1 Explaining the predicted rate

Explaining has an important role in recommender systems [23]. Among other things, good explanations inspire user trust and loyalty, increase satisfaction, enable users to know what they want faster and more easily, and persuade them to try or purchase a recommended item. These are, in fact, the same reasons as those behind the development of explanation facilities for expert systems: humans are reluctant to accept automatic advice if they are not able to understand how the system reached its conclusions.

In a Bayesian network framework, we can find different methods for generating explanations [10]. One of these is called *abduction*. The objective of an abduction process is to search for values of the non-observed variables which better justify the evidence. There are a large number of alternative explanations of the evidence and therefore the objective of abduction is to find the most probable explanation, i.e.

$$x^* = \arg\max_x Pr(x|ev)$$

It should be noted that this is not the same as finding $x = \{x_1^*, \ldots, x_k^*\}$ such that $x_i^* = \arg\max_{x_i} Pr(x_i|ev), \forall x_i \notin ev$.

In our case, the problem is slightly different: the objective is to justify to the user why the system recommends a given rating v_j for an observed item I_j. We are expressing something along the lines of *Considering your profile, you might rate this item with value v_j because ....* For explanation purposes, we must therefore consider as evidence both the proposed rate and the item description:

$$exp = \mathbf{C} = \arg\max_C Pr(C|v_j, I_j) \qquad (3)$$

Thus, an explanation is a set of values for the structural components of the item. For example we might say

> *Considering your profile, you might rate this item with value 4* because its* `genre` *is considered 5* and its* `keywords` *are considered 3*.*

4. HANDLING HIDDEN COMPONENTS

In the previous section we considered that the ratings for all the structural components describing an item are known, but this is not usually the case. In general, the user only gives the system the final rating for each observed product (or this rating is determined non-intrusively). Consequently, our training data set becomes $D^L = \{d_1^l, \ldots, d_n^l\}$ where the superscript indicates that we have not observed the values for an item's structural components, i.e. $d_i^l = \{v_i, f_{1,i}, \ldots, f_{l,i}\}$.

From a recommending perspective, our goal is the same: to predict the value of the rating that an individual would give to the as yet unseen item. In this paper, we will consider two different approaches for tackling this problem: the first, discussed in Section 4.1, where any kind of structural information when recommending is used, and the second, discussed in Section 4.2, where the objective is to learn a model that considers our knowledge about how users make their decisions. For this purpose, we propose the use of the Expected−Maximization algorithm (EM) [6] to estimate not only how the user might rate each structural components but also how this information could be combined by the user in order to obtain an item's overall predicted rating, v_i.

4.1 Naive Bayes model

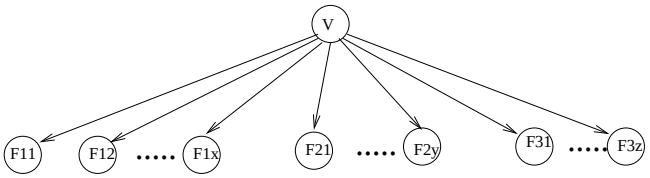

Figure 2: Naive Bayes Model.

Perhaps the most direct alternative for predicting the final user's rating is to consider only the information in D^L, i.e. the user's ratings and the set of features describing the item. These features can be considered as a flat description of an item, i.e. we do not consider any structural component. In this case, the prediction process can be seen as a classification problem: we must assign an item I to one of a set of $\#r$ pre-defined ratings (categories). A supervised learning framework is normally used to train the classifier, where a learning algorithm provides a set of N labeled training data $\{d_i(f_i, vi) : i = 1, ..., N\}$ from which it must produce a classification function mapping items to ratings.

A popular learning algorithm for classification purposes is obtained when a set of conditional independence assumptions is considered in the generative model. In particular, conditioned on the rating (class variable), the probability distributions of the observed features are independent. This is the well-known *naive Bayes* assumption which has proved to be helpful when the dimensionality D of the input space is high [14, 17] (as it is in the recommending problem) because it requires a small amount of training data to estimate the parameters necessary for recommending. A Bayesian network representing this model is illustrated in Figure 4.1.

Once the model is completed, it can be used for prediction purposes in a similar way to the one presented in Section 3.3, i.e.

$$vote = \arg\max_j Pr(v_j|I, \theta)$$

Using the Naive Bayes assumption, when the item I is described with the set of features $F_1, \ldots, F_m$, the computation of $Pr(v_j|I, \hat{\theta})$ is reduced to:

$$Pr(v_j|F_1, \ldots, F_m) = \frac{1}{S} Pr(v_j) \prod_{i=1}^{m} Pr(f_i|v_j) \qquad (4)$$

where S is a normalization constant.

Given the incomplete data set, we can use maximum likelihood estimators to fit the naive Bayes model to the training data in a similar way to the one presented in Section 3.2.

4.2 Learning hidden variables

In this section, we will discuss how the parameters related with the structural components of the products, C, can be learnt from the data set. For this purpose, we propose the use of the EM algorithm which finds maximum likelihood solutions for models with latent variables [6].

EM is essentially an iterative algorithm which performs hill-climbing in the data likelihood space, alternating between performing an expectation (E) step, which computes an expectation of the likelihood by including the latent variables as observed, and a maximization (M) step, which computes the maximum likelihood estimates of the parameters by maximizing the expected likelihood found on the E step. The parameters found on the M step are then used to begin another E step, and the process is repeated. Under certain conditions, EM converges to parameter values at a local maximum of the likelihood function.

As our approach considers that we know a more detailed description of an item, we could include the structural content information in the training data set. Thus, if we assume that features are grouped into different structural components the set of observed D^L becomes

$$D^L = \{V, [F_{1,1}, \ldots, F_{1,l_1}], \ldots, [F_{k,1} \ldots F_{k,l_k}]\}.$$

For example, considering two structural components, `genre` and `keywords` in a movie domain, an observed data is $d^l = \{4, [0, 0, 1], [1, 0, 1]\}$, where the first value is the rate and the other values represent whether a given feature in each category is used to describe the product or not.

In this framework, the hidden variables are the set of structural components $C = \{C_1, \ldots, C_k\}$ and assuming that the complete data set $D^L C$ exists, the log−likelihood function for the observed data is given by

$$\ln Pr(D^L|\theta) = \ln \left(\sum_C Pr(D^L, C|\theta) \right).$$

Let us now suppose that for each observed data we know the value of the latent variables. Thus, following on with our example, let us assume that we know the importance of the genre and keywords components for each observed movie. For example, if genre=5 and keywords=3, then completing the data d^l we obtain $d = \{4, 5, 3, [0, 0, 1], [1, 0, 1]\}$. Once we know this, the data set should be complete, and therefore by considering the generative model, the parameter θ can easily be found by maximizing the complete data set likelihood functions (see Section 3.2)

$$Pr(D^L, C|\theta) = Pr(C|D^L, \theta) Pr(D^L|\theta).$$

In practice, although we do not have the complete data set, if we know the model parameters θ, our knowledge of the values of the latent variables in C is only given by the posterior distribution $Pr(C|D^L, \theta)$. Since we are assuming the generative model for the user rating process, i.e. the probability distribution can be factorized into a set of local distributions, these probabilities can easily be computed by means of

$$Pr(C|VF, \theta) = Pr(V|C, \theta) \prod_{i=1}^{k} Pr(C_i|\theta) \prod_{j=1}^{l_i} Pr(F_j|C_i, \theta)$$

Because we do not have the complete-data log likelihood, the hidden values might be completed by alternatively using their expected value under the posterior distribution of the latent variable, which corresponds to the E step of the EM algorithm. In the subsequent M step, we can maximize this expectation by using Equation 2. In our case, we choose the initial parameters θ_0 randomly. The algorithm then iterates until it converges to a point where the estimation of θ does not change from one iteration to the next (see Table 2).

4.2.1 Computational requirements

Although, in theory, the EM will involve there being a large complete data set, in practice, the assumption that we know the topology of the generative model will be helpful for

Table 2: Learning profile from uncomplete data set.

- *Initialization step*:

  Set $t = 0$

  Set θ_0 at random.

- Iterate until convergence

E-step: Use the current model parameters θ_t to estimate the probability that the given item (data in the data set) can be generated by means of any configuration of the latent variables C, i.e.

$$Pr(c_1, \ldots, c_k | v, f_1, \ldots, f_l, \theta_t).$$

Complete the hidden values using these probabilities, i.e.

$$C^{t+1} = E[C|D^l, \theta_t].$$

M-step: From the "completed" data set, re-estimate the model parameters θ_{t+1}, i.e. calculate a new maximum a posteriori estimator for the parameters θ_{t+1}, i.e.

$$\theta_{t+1} = \arg\max_{\theta} P(D^L C_{t+1}|\theta_t) P(\theta_t).$$

computational purposes[3]. Thus, using the independences in the generative model we know that the joint probability distribution can be factorized into a set of local distributions. Moreover, if we observe Equation 2 we can find that in order to estimate the vector of parameters θ we only need to know, on the one hand, the frequency values for the set VC (involving the final vote and the structural components C), i.e. $VC = \{V, C1, \ldots, C_k\}$ and, on the other, the frequency values for l different sets $C_j F_i$, with $i = 1, \ldots, l$, where each $C_j F_i$ comprises the feature variable F_i and the only structural component C_j it belongs to.

In the E-step, therefore, having computed the probability $Pr(c_1, \ldots, c_k | v, f_1, \ldots, f_l, \theta_t)$ for each data $d^l \in D^L$, we must complete $l + 1$ different frequency sets, which implies an enormous saving in storage and time processes. Obviously, the estimation of the values θ_{t+1} in the M-step can be obtained easily from the frequency values stored in the $l + 1$ frequency sets, i.e. VC and $C_j F_i$, using Equation 2.

5. EXPERIMENTATION

In this section, we will present preliminary experimentation to study whether the proposed methodology is promising or not. In particular, we shall consider the problem of recommending a hotel for vacation purposes. In this case, and in order to describe the hotel, we use the following set of structural components:

- `Location`: this component represents the possible (non-exclusive) alternatives where the hotel might be located, taking its values in `city`, `beach` and `mountain`.

[3]Algorithms can be found in the literature (see [7]) for learning a good structure with a fixed set of hidden variables, but these algorithms have higher computational costs since they do not know the underlying structure of the model.

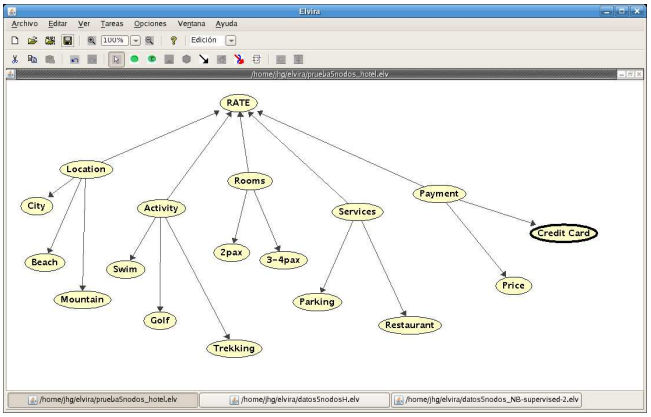

Figure 3: BN used in the experimentation

- `Activities`: this component represents the set of possible activities that can be done nearby including `swimming`, `golf` and `trekking`.

- `Services`: this component represents the services which are offered by the hotel, i.e. `parking` and `restaurant`.

- `Payment`: this component represents those concepts relating to payment, for example the `price` (cheap and expensive) and the possibility of paying by `credit card`.

- `Rooms`: this component represents the different types of rooms in the hotel. In the example, the two non-exclusive alternatives used are `2pax` and `3-4pax`.

Therefore, the generative model is the one presented in Figure 5 where we assume that the `rate` variable is bivaluated, taking its values in $\{1 = Like, 2 = Dislike\}$. It should be noted that these values will be those used for rating the structural components.

In addition, and for illustrative purposes, we shall assume that the user's behavior when rating a hotel is based on a majority strategy, i.e. the overall rating for a given hotel will depend on a simple count of the votes received for each structural component. The rating which receives the largest number of votes is then selected as the majority decision.

Definition. Majority Rate: a node V is said to represent a majority combination criterion if, given a configuration of its parents $pa(V)$, the conditional probability distributions can be represented as

$$Pr(V = s|pa(V)) = \begin{cases} \frac{1}{m} & \text{if } s = \arg\max_k count(k, pa(V)) \\ 0 & \text{otherwise} \end{cases}$$

where $count(k, pa(V)$ is a function returning the number of occurrences of the state k in the configuration $pa(V)$, and m is the number of states where $count(k, pa(G_i))$ reaches the maximum value.

For example, considering a node V with five parents and two candidate ratings, 1= *Like* and 2= *Dislike*, then $Pr(V = 1|1, 1, 2, 1, 1) = 1$ and $Pr(V = 1|1, 2, 2, 1, 2) = 0$.

In order to validate our approach, we generate a synthetic data set of 600 data split into 80% for learning and 20% for testing purposes. We use the learning data set to tune the parameters of the proposed model. In order to study

Model	%Succ.	Recall	Prec.	F1
Generative	68%	0.7179	0.5714	0.6364
Learning Parameter	64%	0.7692	0.5357	0.6316
Naive Bayes	71%	0.6668	0.5778	0.6190
Learning Profile from Hidden Components	71%	0.6668	0.6190	0.6420

Table 3: Experimental results.

our proposal's performance, we shall consider four different models: the first is the original model, i.e. the model from which the data set has been generated; the second is the model learned when considering the complete training data set, i.e. all the information is available (see Section 3.2); the third is the one learned from the incomplete data set (obtained by removing the information concerning the structural components) using Naive Bayes; and the fourth is obtained by using the algorithm in Section 4.2, which learns the user profile from the incomplete data set.

5.1 Results and Discussion

In order to measure the accuracy of each model, we will consider the frequency with which an RS makes correct or incorrect decisions about whether an item is good or not. Following [8], two different measures might be considered: *Recall*, which is defined as the ratio of relevant items selected, N_{rs}, to the total number of relevant items available, N_r, and *Precision*, which is defined as the ratio of relevant items selected, N_{rs}, to the number of items selected, N_s.

$$P = \frac{N_{rs}}{N_s} \qquad R = \frac{N_{rs}}{N_r} \tag{5}$$

It is well known that these measures are inversely correlated and depend on the number of items returned. We shall use the F1 metric [8] which is one of the most common metrics for combining P and R into a single value:

$$F1 = \frac{2PR}{P+R} \tag{6}$$

This metric takes its values in $[0, 1]$ verifying that the closer a value is to 1, the better the quality of the model when recommending good items.

The results for each model are presented in different rows in Table 3. For each model, we present the % of success, i.e. the number of times that we predict the correct rating, the *Recall*, *Precision* and *F1* metrics. From this Table, we could conclude that when learning user profiles it is better to take into account information about the item's structure. It should be noted that when this kind of information is not used, the Naive Bayes approach obtains the worst result in the F1 metric. Another promising feature is the results obtained when learning structured profiles from a set of ratings where the structural components are latent. The best result is obtained by considering the percentage of success, precision and F1 metrics. Nevertheless, in our opinion, this finding (something which should be validated with more detailed experimentation) is not conclusive if the user profile has not been captured properly.

5.1.1 What about the ability to learn the user's rating pattern?

In the previous section we discussed the accuracy of the model when recommending items to the user. Our objective in this section is to consider whether the methodology proposed in Section 4.2 is able to learn the user profile or not. With this purpose in mind, we will measure the capability of the model to learn the user's rating pattern, i.e. the relationships between *rate* variable and the set of structural components C.

	SP			
LAR	11	12	21	22
111	1.0	1.0	1.0	0.76
112	1.0	1.0	1.0	0.49
121	0.99	0.87	0.86	0.05
122	0.92	0.36	**0.61**	0.20
211	1.0	**0.22**	0.99	0.0
212	0.99	0.44	**0.55**	0.0
221	**0.33**	**0.58**	0.0	0.0
222	0.11	0.47	0.0	0.0

Table 4: $Pr(V = 1|L, A, R, S, P$

In this experiment, we have assumed that the user's rating pattern is based on a majority criterion. This pattern is encoded in the set of conditional probability distributions stored at node V, i.e. $P(V|LARSP)$ where L, A, R, S, P denotes the value of the `Location, Activity, Rooms, Services` and `Payment` variables. In Table 4 we show the probabilities learnt from the incomplete data set (we only show the values $P(V = 1|LARSP)$). Assuming that given a configuration our model will recommend the rating of greatest probability, i.e. the recommended rating is 1 if $P(V = 1|pa(V)) > 0.5$, we highlight in bold those probability values that might represent a failure in the recommendation.

From this table we can conclude that for a given configuration of the values in $Pa(V)$, we shall recommend the proper rate, i.e. following a majority criterion, in 84.37% of the times. Moreover, these failures occurs in those situations where three components were rated with a value whereas the other two were rated with the opposite. From this perspective, we can conclude that the learnt pattern of rating is close to the majority criterion and is therefore a good representation of the user profile. This will be helpful when explaining a given recommendation.

6. CONCLUSION

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 includes information about the structural content of a product: we assume that a product is described using a well-defined set of categories and that the user rates an item by combining the ratings for the different structural components. Although the second assumption might be considered restrictive in certain domains, we think that it could be valid in a recommending framework. Since the user usually only gives an overall product rating, we present a methodology for learning both how the user combines the information and how the user should rate each individual component from a set of incomplete ratings.

We consider this paper to be the first step towards a new description of the user profile that might be able to incorporate structural content information. There are, however,

three challenges which still need to be overcome: first, it is necessary for there to be more detailed experimentation to analyze how good the learnt representation of the user profile is; secondly, we must study how the system could interact with the user and also how it should be learnt from this interaction; and finally, since we have a better description of user behavior we can use this information to design collaborative-based recommender systems.

7. ACKNOWLEDGMENTS

This work has been supported by the Spanish 'Ministerio be Educación y Ciencia' and 'Consejería be Innovación, Ciencia y Empresa be la Junta be Andalucía' under Projects TIN2005-02516 and TIC-276, respectively.

8. REFERENCES

[1] S.S. Anand, B. Mobasher *Intelligent techniques for web Personalization. In Intelligent Techniques for Web Personalization*. Lecture Notes in Artificial Intelligence, 3169, 2005.

[2] G. Adomavicius, A. Tuzhilin. *Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extentions* IEEE Transactions on Knowledge and Data Engineering, 17(6):734–749, 2005.

[3] M. Balabanovic, Y. Shoham. Fab: Content-Based, Collaborative Recommendation. Communication of the ACM, 40(3):66–72, 1997.

[4] R. Burke. *Hybrid recommender systems: survey and experiments*. User modeling and User-Adapted Interaction 12:331–370, 2002.

[5] C.J. Butz. *Exploiting Contextual Independencies in Web Search and User Profiling*. In Proceedings of the World Congress on Computational Intelligence (WCCI), 1051–1056, 2002.

[6] A. Dempster, D. Laird and D. Rubin. *Maximum likelihood from incomplete data via the EM algorithm*. Journal of the Royal Statistical Society 39:1-38, 1977.

[7] N. Friedman. *The Bayesian estructural EM algorithm*. Proc. of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI-98). Morgan Kaufman, 1998.

[8] 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.

[9] S. Kangas *Collaborative filtering and recommendation systems*. VTT Information Technology. Research Report TTE4-2001-35, 2002.

[10] C. Lacave and F.J. Diez, *A review of explanation methods for Bayesian networks* The Knowledge Engineering Review 17 (2), pp. 107127, 2002.

[11] C. Manning and H. Schtze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, Massachusetts.1999

[12] R. J. Mooney, P. N. Bennett, L. Roy. *Book recommending using text categorization with extracted information*. In Recommender Systems. Papers from 1998 Workshop. Technical Report WS-98-08. AAAI Press.

[13] B. Mobasher. *Data Mining for Personalization. In The Adaptive Web: Methods and Strategies of Web Personalization*. Lecture Notes in Computer Science, 4321, 2006.

[14] K. Nigam, A.K. McCallum, S. Thurn and T. Mitchell *Text classification from labeled and unlabeled documents using EM* Machine Learning, 39, pp.103-134. 2000

[15] P. Nokelainen, H. Tirri, M. Miettinen, T. Silander. *Optimizing and profiling users on-line with Bayesian probabilistic modeling* In Proceedings of the NL 2002 Conference. 2002.

[16] M. Pazzani, *A Framework for Collaborative, Content-Based, and Demographic Filtering,* Artificial Intelligence Rev., pp. 393-408, 1999.

[17] M. Pazzani, D. Billsus. *Learning and revising user profiles: The identification of interesting web sites* Machine Learning, 27:313–331, 1997.

[18] P. Reskick, H.R. Varian. *Recommender systems*. Communications of the ACM, 40(3):56–58, 1997.

[19] S.N. Schiaffino, A. Amandi. *User profiling with Case-Based Reasoning and Bayesian Networks*. IBERAMIA-SBIA 2000 Open Discussion Track, 12–21, 2000.

[20] G. Salton y M. J. McGill. *Introduction to modern information retrieval*. McGraw-Hill, Inc., 1983.

[21] J.B. Schafer. *The application of data-mining to recommender systems*. In Encyclopedia of Data Warehousing and Mining. J. Wang (Editor). 2005.

[22] B.M. Sarwar, G. Karypis, J. Konstan, J. Riedl. *Analysis of recommender algorithms for e-commerce*. Proceedings of the ACM E-Commerce 2000 Conference, 158–167, 2000.

[23] Nava Tintarev and Judith Masthoff *A Survey of Explanations in Recommender Systems* IEEE Third Internation Workshop on Web Personalisation, Recommender Systems and Intelligent User Interfaces, pp. 801-810, 2007

[24] M. Vozalis, K.G. Margaritis. *Anlysis of recommender system's algorithms*. Proceedings of the 6th Hellenic European conference on Computer Mathematics and its Applications, 2003.

[25] S.K.M. Wong, C.J. Butz. *A Bayesian Approach to User Profiling in Information Retrieval*. Technology Letters, 4(1):50–56, 2000.

[26] C. Wei, M. J. Shaw, and R. F. Easley. *A Survey of Recommendation Systems in Electronic Commerce*. E-Service: New Directions in Theory and Practice, R. T. Rust and P. K. Kannan (Eds), M. E. Sharpe Publisher, 2002.