International Journal of Computational Intelligence Systems, Vol. 7, No. 1 (February 2014), 80-89

Weighting of Features in Content-Based Filtering with Entropy and Dependence Measures

Jorge Castro [1] **Rosa M. Rodriguez** [1] **Manuel J. Barranco** [1]

[1] *Department of Computer Science, University of Jaén,*
Campus Las lagunillas,
Jaén, 23071, Spain

E-mail: jcastro,rmrodrig,barranco@ujaen.es

Received 28 May 2012

Accepted 31 May 2013

Abstract

Content-based recommender systems (CBRS) are tools that help users to choose items when they face a huge amount of options, recommending items that better fit the user's profile. In such a process, it is very interesting to know which features of the items are more important for each user, thus the CBRS provides them higher weight. The Term Frequency-Inverse Document Frequency (TF-IDF) method is one of the most used for weighting of features, however, it does not provide the best results when the features are multi-valued. In this contribution, it is proposed a new method for obtaining the weights of the features by means of entropy and coefficients of dependency.

Keywords: content-based filtering, recommender systems, weighting of features, entropy

1. Introduction

Recommender systems are used in different scenarios (web pages, e-commerce, tourism applications, etc) where users face a vast amount of options that can overwhelm them. Particularly, content-based recommender systems, CBRS [1,5,13,14] is a type of such systems, that uses the available information about the choices that the user made in the past. This information is used to build a user profile that represents the user's preferences or necessities. Besides, a database of descriptive information about the items, in which each item is described by a set of features, is required. The basic functions of a CBRS consists of (i) updating the profile of each user (ii) filtering the available products with the user's profile and (iii) recommending the products that better fit the profile.

Belkin and Croft [3] proposed one of the first CBRS by using technology related to information retrieval [16], such as Term Frequency-Inverse Document Frequency (TF-IDF) and Rocchio's method. This system deals with the users' profiles and item descriptions by using textual analysis, so that the features are words or terms that describe the items. In this way, each item is represented by a vector compounds of ones and zeros that indicate whether a term appears or not in the text description of that item. Nevertheless, in a more general case, the features can be assessed by multi-valued variables or other domains, such as, numeric, linguistic.

The filtering process should consider that not all features are equally important. Obviously, when a user selects an item, he/she is watching some features that are important and ignoring others that are worthless to him/her. This consideration represents an implicit weighting of features which is subjective and different for each user.

The aim of this paper is to introduce a new method to obtain such weights in CBRS, where features can be assessed by multi-valued variables or in multiple domains,

by using the implicit ratings obtained from the users in the past. Thus, assigning weights to features, according to the weighting that the user has implicitly provided, the profile will be more useful in the recommendation process. Our proposal computes two measures for weighting each feature. First it is taken into account the entropy or amount of information for each feature, the more entropy the more weighting. Afterwards, it is considered the correlation (for quantitative features) and contingency (for qualitative features), between the user ratings and the values of the feature on the rated items, for each feature. The greater the relationship, the higher the weight for the feature.

This paper is structured as follows. Section 2 reviews necessary concepts for our proposal. Section 3 describes in further detail our proposal for weighting multi-valued features which is evaluated by a case study in Section 4. Finally, Section 5 points out some conclusions.

2. Previous works

This section reviews briefly the CBRS and the TF-IDF method for weighting of features.

2.1. Content-based Recommender Systems

CBRSs are based on item features, recommending items that are similar to those that a user liked in the past [1,11,14]. Those systems use a database with a set of items $A = \{a_i, i = 1,\ldots,n\}$ described by a set of features $C = \{c_j, j = 1,\ldots,m\}$ defined each one in a domain D_j, so that each item a_i is described by a vector $V_i = \{v_j^i \in D_j, j = 1,\ldots,m\}$ (see Table 1).

Table 1. Data for a CBRS

	c_1	$\ldots$	c_j	$\ldots$	c_m
a_1	v_1^1	$\ldots$	v_j^1	$\ldots$	v_m^1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
a_n	v_1^n	$\ldots$	v_j^n	$\ldots$	v_m^n

For each user u, there is a set $A_u = \{a_i^u \in A,\ i = 1,\ldots,n_u\}$, where a_i^u are the items chosen by the user u, and each item a_i^u is assessed, $r_i^u \in D_u$ (implicit or explicit), by the user in an expression domain D_u (see Table 2).

By means of the user's information, the CBRS computes a user profile P_u, that represents the user preferences, and a weighting vector W_u, that includes the weights of each feature according to their relevance in the user's needs:

- $P_u = \{p_j^u \in D_j, j = 1,\ldots,m\}$ are the values for each feature that better fix the user's preferences. They can be obtained from different ways [1,11,14].
- $W_u = \{w_j^u, j = 1,\ldots,m,\ 0 \leqslant w_j^u \leqslant 1\}$ are the weights that show the relevance of each feature, according to user's needs.

Table 2. User data for a CBRS

	c_1	$\ldots$	c_m	R_u
a_1^u	v_{11}^u	$\ldots$	v_{1m}^u	r_1^u
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$a_{n_u}^u$	$v_{n_u1}^u$	$\ldots$	$v_{n_um}^u$	$r_{n_u}^u$
P_u	p_1^u	$\ldots$	p_m^u	
W_u	w_1^u	$\ldots$	w_m^u	

The performance of a CBRS consists of the following phases:

1. *Acquisition of the features of the items and users' profiles*. The system calculates the user profiles by analysing the user ratings and the item description of the rated items, obtaining the implicit preference of the user over the item features. This task must be done periodically to keep the user profiles up to date.

2. *Filtering process*. For each item and feature, the system calculates the similarity with the user profile. The values obtained are then aggregated to obtain the similarity of the user profile with each item.

3. *Recommendations*. The system selects the most similar items to user's necessities.

2.2. Weighting of Features in CBRS

In most of the cases, the users take into account different item features with different strength. To incorporate this into the recommendation process, feature weighting is considered. In the literature, there are different methods of weighting of features in CBRS which deal with textual descriptions, i.e., the item features are keywords that describe the items [14,20]. These systems perform the weighting of features by using the TF-IDF (Term Frequency - Inverse Document Frequency) method [2]. This

method is one of the weighting schemes of features more commonly used in information retrieval [8] and decision making problems [21]. However, it has been also used regularly in the CBRS [20].

The CBRS with weighting of features [20] builds the user profile by using implicit ratings inferred from past items used by the user. The assessments for the features will depend on the existence or absence of certain terms in the item description. Therefore, given a feature and an item, the profile can take two values: 1 if the feature exists, 0 otherwise.

The filtering process looks for the most suitable items for a user by matching the user profile and the items description. The more relevant is a feature for the user, the higher the weight applied to it in the matching process. The TF-IDF has been used to compute the relevance of each feature f for a user u by means of the following equation:

$$W(u,f) = FF(u,f) * IUF(f) \quad (1)$$

The relevance of the feature f for the user u is obtained as the product between two factors:

1. A quantification of the intra-user similarity FF (feature frequency), which indicates the characteristic frequency of f for the user u.

2. A quantification of the inter-user dissimilarity IUF (inverse user frequency), which provides a higher value to the distinctive characteristics, i.e. the least repeated in the set of users.

Commonly, the factor $FF(u,f)$, is computed by using the number of times that the feature f appears in the items that the user u has rated positively. The second factor, according to the TF-IDF scheme [2], is computed as $IUF(f) = log\frac{|U|}{UF(f)}$ being $UF(f)$ the number of users that have rated positively any item that has the feature f, and $|U|$ the total number of users registered in the system.

This method for feature weighting is useful in CBRS that deals with binary features based on text descriptions, but in those systems that manage more complex item descriptions, such as multivalued features, the previous approach can be improved.

The problem of weighting multivalued features has been managed in other areas such as, information retrieval and machine learning [9,10]. Nevertheless, this problem has been poorly performed in recommender systems. Therefore, the aim of this proposal is to deal with the weighting of multivalued features in CBRS.

3. Weighting of Features based on Entropy and Dependency Measures

When items are described by means of multivalued features, classical methods for feature weighting such as, TF-IDF do not provide successful results. TF-IDF deals with boolean features, therefore, when multivalued features appear, they must be transformed into boolean features. Let us suppose a set of items $A = \{a_1, a_2, a_3, a_4, a_5, a_6\}$ assessed by using multivalued features as shown in Table 3. To apply TF-IDF method, the multivalues features are transformed into boolean features as shown in Table 4. This transformation has two problems:

- Oversize of user profiles. As can be seen, comparing Tables 3 and 4, the dimensionality of data used in boolean models can be very high when features take a large number of different values. This problem implies low efficiency and high storage needs.
- Loss of information. In boolean models, the information provided by features defined in ordered domains is lost. For instance, let "publishing year" be a feature of a song. If the user has only rated songs published in 1962 and 1964, the similarity between a song published in 1963 and the user profile will be zero. This way, the system is wasting valuable information: the numeric order that provides the feature "publishing year".

Table 3. Multivalued items

	Style	Language	Year
a_1	Jazz	English	1962
a_2	Pop	Spanish	1960
a_3	Pop	French	1962
a_4	Jazz	English	1962
a_5	Jazz	English	1960
a_6	Pop	Spanish	1962

Table 4. Boolean items

	Jazz	Pop	English	Spanish	French	1960	1962
a_1	1	0	1	0	0	0	1
a_2	0	1	0	1	0	1	0
a_3	0	1	0	0	1	0	1
a_4	1	0	1	0	0	0	1
a_3	1	0	1	0	0	1	0
a_3	0	1	0	1	0	0	1

To overcome previous limitations, we propose a new method for feature weighting in CBRS that deals with items described by multivalued features, numeric or nominal. The proposed method works directly with multivalued features without any transformation by using the information of the features defined in ordered domains and reducing the dimensionality of user profiles. It is based on the TF-IDF scheme, since the weight of a feature is obtained as the product between two factors, the inter-user dissimilarity and intra-user similarity of such a feature. To obtain these factors, we propose the following measures:

- *The amount of information provided by each feature to assess the inter-user dissimilarity*: Multi-valued features can provide different amount of information. The greater the domain that a feature is defined on, the higher the relevance of such a feature. We propose the use of entropy to compute the amount of information that a feature can provide.
- *The correlation between user ratings and feature values to assess the intra-user similarity*: Given a feature and a set of items experienced by a user, a high correlation between user ratings and feature values, for this set of items, means a high relevance of such a feature.

The proposal uses the data structure showed in Tables 1 and 2. So, the new system will work with two sets of vectors: item descriptions $V^u_{i.} = \{v^u_{ij}, j = 1,\ldots,m\}$ (rows in Table 2) and feature descriptions $V^u_{.j} = \{v^u_{ij}, i = 1,\ldots,n_u\}$ (columns in Table 2).

To assess the correlation between user ratings and feature values, two classes of features must be distinguished: numeric and nominal. Obviously, the measures must be different because there is a clear difference between them: usually, numeric domains are ordered and nominal ones are not. Therefore, in this proposal, a correlation measure is used for numeric features and a contingency measure for nominal features [4].

The proposed method for weighting of features consists of the following phases (see Figure 1):

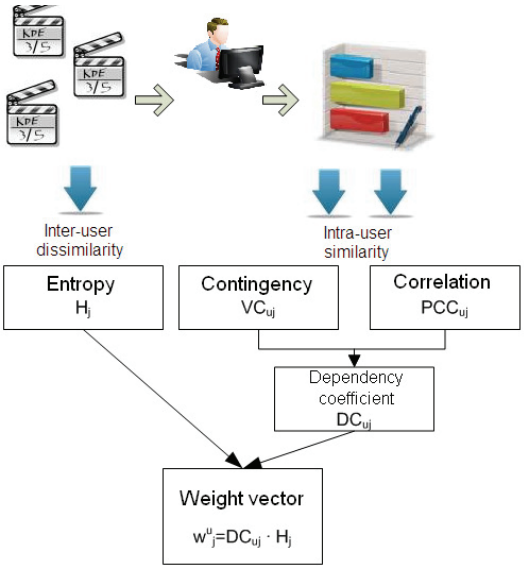

Fig. 1. Feature weighting method

1. *Calculation of inter-user dissimilarity*: it shows which features are more relevant to the user. It computes the entropy H_j for each feature c_j, the greater the value of the entropy, the greater the relevance of such a feature.

2. *Calculation of intra-user similarity*: given a set of items that the user has experienced in the past, the system computes the correlation between the ratings that the user provides to these items, and the feature values on this set of items. Let c_j be a feature and u be a user. A coefficient of dependency DC_{uj} is computed between the user's ratings, $R_u = \{r^u_i, i = 1,\ldots,n_u\}$, and the assessments of the feature c_j in the items rated, $V^u_{.j} = \{v^u_{ij}, i = 1,\ldots,n_u\}$. The way of computing this value depends on the nature of the features:

   - Correlation coefficient for numeric features.
   - Contingency coefficient for nominal features.

3. *Calculation of weights*. Finally, the weight for each feature is obtained combining its entropy H_j and its coefficient of dependency DC_{uj}.

These phases are explained in further detail in the following sections.

3.1. Inter-user dissimilarity

Its aim is to look for features that allow finding out the tastes or necessities of the user to recommend items that better suit his/her preferences. Therefore, diversity measures are good to assess the discriminating capability of features, i.e. , features with a higher number of different values provide higher discrimination than features with few values. There are several diversity measures in the literature. The most common ones are the Simpson index [19] and the Shannon diversity index, so called Shannon Entropy [18]. The Simpson index provides more importance to features with a high number of values, so it is focused on have big groups of values, giving less by importance to infrequent values. The Shannon Entropy provides importance to features with great amount of different values but also takes care of the distribution of these values, having the maximum value when the distribution of the values is uniform. These reasons lead us to use the Shannon Entropy to assess the inter-user dissimilarity.

The Entropy is defined as follows.

Definition 1 [7,18] *It is defined as the average amount of information, measured in bits, which contains a random variable. Let x be a random variable, its entropy is:*

$$H(x) = -\sum_i p(x_i)\, log_2\,(p(x_i)) \tag{2}$$

being $p(x_i)$ the probability of the value x_i.

In the search process of items similar to a given user profile, the features with higher entropy are most interesting and have a higher weight. Therefore, when a user assesses an item, positively or negatively, the information provided to the system depends on the entropy or amount of information provided by each feature. The higher entropy, the higher weight. For example, let c_1 and c_2 be two features (see Table 5). In this case the weight for the feature c_2 should be higher than the weight for the feature c_1, because it provides more information to the system due to the number of different values it can take.

Table 5. Two features with different entropy

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
c_1	*A*	*B*	*B*	*A*	*B*	*A*	*A*	*B*
c_2	1	3	2	4	4	5	6	5

This phase is offline, because it is carried out without the interaction of the user.

For each feature c_j, the system computes the entropy H_j, and the normalized entropy $H_j^* \in [0,1]$ as follows:

$$\begin{aligned} H_j &= -\textstyle\sum_{k_j}(f_{k_j}/n) log_2(f_{k_j}/n) \\ H_j^* &= \frac{H_j}{\sum_i H_i} \end{aligned} \tag{3}$$

being $\{k_j\}$ the set of values that takes the feature c_j, f_{k_j} the frequency of the value k_j in the whole set of items A, and n the total number of items. This calculation considers $log(0) = 0$, hence the values whose frequency is 0 do not affect the result.

The value H_j^* indicates the normalised amount of information that the attribute c_j provides to the system. For example, for an attribute that only takes two values and is equally distributed, its entropy H will be 1, i.e., provides information about one bit. While, another attribute that takes 16 different values gives about 4 bits of information. The entropies are normalized in [0,1] dividing by the sum of the entropies, hence the system gives a higher weighting to features with a higher amount of information.

3.2. Intra-user similarity

In this phase the system measures the contingency or correlation between the user's ratings on a set of experienced items and the values of a feature c_j on this set of items. If there is a dependency between these variables, it suggests that such a feature is important for the user. There are several measures to obtain this factor. We distinguish between numeric and nominal features.

For numeric features a correlation index can be used to measure its dependency. The most common ones are Pearson and Spearman correlation coefficient [4]. We will use the Pearson coefficient because of its ability to detect inverse relationships between variables, which the Spearman one cannot reveal.

For nominal features, it is computed a dependency measure by means of contingency indexes. To do so, there are different coefficients such as, Karl Pearson Contingency Coefficient [15] and Cramer V Coefficient [4]. The Karl Pearson Coefficient presents a problem since, although the relation between two variables can be perfect, the index could not ever raise to the value 1. Nevertheless, the Cramer V coefficient does not have this problem and it is adequate for the proposed method.

Definition 2 [4] *The Pearson correlation coefficient is a statistical index that measures the linear relationship between two variables. Unlike the covariance, the Pearson correlation is independent of the scale of measurement of variables. The computation of the coefficient of linear correlation is obtained by dividing the covariance by the product between standard deviations of both variables.*

$$r = \frac{\sigma_{XY}}{\sigma_X \sigma_Y} \tag{4}$$

$$s.t. \begin{cases} \sigma_{XY} = \frac{1}{n}\sum_{i=1}^{n} x_i y_i + \overline{xy} \\ \sigma_X = \sqrt{\sum_{i=1}^{n} \frac{x_i^2}{n} - \bar{x}^2} \\ \sigma_Y = \sqrt{\sum_{i=1}^{n} \frac{y_i^2}{n} - \bar{y}^2} \end{cases}$$

where σ_X and σ_Y are the standard deviation of variable X and Y; σ_{XY} is the covariance of X and Y values; x_i and y_i are the i-esimo values of variable X and Y; $\bar{x}$ and $\bar{y}$ are the mean values of the variable X and Y; and n is the number of values of the variables.

An interesting property of the Pearson Correlation Coefficient is that is unaffected by linear transformations. This property tackles directly the problem of different rating scales for different users, e.g., a user rates products consistently higher than the rest.

Definition 3 [4] *Cramer V coefficient is one of the most commonly used contingency ratios to measure the dependence between two random variables, X and Y, where at least one of them is qualitative.*

$$V = \sqrt{\frac{\chi^2}{n \cdot min\,(I-1, J-1)}} \tag{5}$$

$$s.t. \begin{cases} \chi^2 = \sum_{i=1}^{I}\sum_{j=1}^{J} \frac{\left(p_{ij} - q_{ij}\right)^2}{q_{ij}} \\ n\ total\ number\ of\ occurrences \\ I\ number\ of\ distinct\ values\ of\ variable\ X \\ J\ number\ of\ distinct\ values\ of\ variable\ Y \\ p_{ij}\ frequency\ of\ pair\ (i,j) \\ q_{ij} = \frac{p_{X=i} \cdot p_{Y=j}}{n}\ theor.\ frequency\ of\ pair\ (i,j) \\ p_{X=i}\ frecuency\ of\ X = i \\ p_{Y=j}\ frecuency\ of\ Y = j \end{cases}$$

An important aspect of Cramer V coefficient is the scale invariance, i.e., the number of values of the sample does not affect the behavior of the coefficient. This means that the Cramer V has the same behavior for users with different number of ratings.

Therefore, the dependence coefficient *DC*, between the ratings provided by the user u over a set of items and the values of the feature j for each product is given by the following expression,

$$DC_{uj} = \begin{cases} \left|PCC_{uj}\right| & if\ c_j\ is\ quantitative \\ VC_{uj} & if\ c_j\ is\ qualitative \end{cases}$$

being PCC_{uj} the Pearson correlation coefficient according to the variables R_u and $V^u_{\cdot j}$.

$$PCC_{uj} = \frac{\sum_i r_i^u v_{ij}^u - \frac{\sum_i r_i^u \sum_i v_{ij}^u}{n_u}}{\sqrt{\left(\sum_i \left(r_i^u\right)^2 - \frac{\left(\sum_i r_i^u\right)^2}{n_u}\right)} \sqrt{\left(\sum_i \left(v_{ij}^u\right)^2 - \frac{\left(\sum_i v_{ij}^u\right)^2}{n_u}\right)}} \tag{6}$$

and VC_{uj} is the Cramer V contingency coefficient according to the same variables for qualitative features,

$$VC_{uj} = \sqrt{\frac{\sum_{k_u}\sum_{k_j} \frac{\left(f_{k_u,k_j} - \frac{f_{k_u} f_{k_j}}{n_u}\right)^2}{\frac{f_{k_u} f_{k_j}}{n_u}}}{n_u\ \ min\left(\left|D_u\right|, \left|D_j\right|\right)}} \tag{7}$$

being k_u and k_j indexes for the set of different values in R_u and $V^u_{\cdot j}$ respectively, f_{k_u}, f_{k_j} are the frequencies of values indexed by k_u and k_j respectively and f_{k_u,k_j} is the frequency of simultaneous occurrences of the two values indexed by k_u and k_j.

The Pearson coefficient is bounded on the interval [-1,1] providing information on the degree of dependence and the type of dependence, direct or inverse. Due to the type of dependence is not important for the intra-user similarity, we take the absolute value, thus the result is in the interval [0,1]. The Cramer V is also bounded in [0,1], therefore, the dependence coefficient *DC*, will be bounded in that interval, being the value 1 the maximum dependence degree.

3.3. Calculation of features weights

Once the normalized entropy H_j^* and dependence coefficient DC_{uj}, have been obtained, it is computed the weight for each feature c_j by means of the product between both factors according to the formula (1).

$$w_j^u = DC_{uj} \cdot H_j^* \tag{8}$$

To normalize the weights vector $\{w_i\}$, it is necessary to satisfy the property $\sum w_i = 1$, hence the final weights vector W_u^* is obtained as follows,

$$W_u^* = \left\{ w_j^{*u} \mid j = 1, \ldots, m,\ w_j^{*u} = \frac{w_j^u}{\sum_i w_i^u} \right\} \qquad (9)$$

4. Evaluation: a case study

This section presents a case study to validate the proposed method by using the MovieLens* dataset, and carries out a comparative analysis among different basic methods for CBRS.

The aim of this task is to study whether the proposed method improves the other popular algorithms [1,14] by analyzing their effectiveness and efficiency. To perform the study case, two type of evaluations can be carried out: online and offline. We have performed an offline evaluation, because its cost is lower and there are many public datasets that can be used.

In the experiment performed, the proposed method is compared with the following ones:

- Boolean model for CBRS without feature weighting, using as similarity measure the cosine coefficient [12], (Boolean-Cos).
- Boolean model for CBRS with feature weighting using a scheme based on TF-IDF[1,14], (Boolean-TF_IDF).

4.1. Description of the dataset

The algorithms of recommender systems need a dataset that gathers the user interaction with the system. This interaction is, in the most of the cases, a set of products and the ratings that users provide to each product.

There are many available datasets that can be used for recommender systems. The dataset used in this experiment is pulled out of the system MovieLens, developed by the research group *GroupLens Research*, of the University of Minnesota. This system offer the possibility of rate movies by users and afterwards it can make suggestions to them by using collaborative filtering algorithms.

The dataset consists of tuples <user, movie, rating>, where a rating is an integer between 1 and 5, being 1 the worst rating for a movie and 5 the best one. Taking into account that the algorithms considered for this case study are CBRS, it is necessary to complete the information provided by the dataset with information that describes the products' content. The descriptive information of the movies has been obtained from IMDB† considering the following features:

- *Gender*: Categorical feature with 25 distinct values.
- *Director*: Categorical feature with 3.999 distinct values.
- *Year*: Numerical feature with values between 1915 and 2008.
- *Country*: Categorical feature with 70 distinct values.

We have selected users that have rated at least 15 movies. Therefore, the dataset contains 9.773 movies, 69.878 users and 9.464.734 ratings. This way, the dataset has a sparsity of 98,6%.

4.2. Experiment

To perform the experiment, the protocol K-fold cross validation [6] is applied. In this experiment, a Cross Fold Validation with $k = 5$ is performed. To ensure that the results are meaningful, this validation protocol has been executed 50 times.

Different evaluation measures can be applied to evaluate how the algorithms are performing. In this experiment, the measures used are Precision, Recall and $F_\beta - score$ [17], which are defined below:

$$Precision = \frac{tp}{tp + fp} \qquad (10)$$

$$Recall = \frac{tp}{tp + fn} \qquad (11)$$

$$F_\beta - score = (1 + \beta^2) * \frac{Precision * Recall}{\beta^2 * Precision + Recall} \qquad (12)$$

being tp the number recommended items that are relevant, fp the number of recommended items that are not relevant and fn the number of recommended items that were left out by the system but are relevant.

Before applying these measures, it is necessary to fix a relevance criteria to decide whether a product is relevant for a user. Taking into account that the ratings are integer values between 1 and 5, the relevance criteria chosen

*http://www.movielens.org
†Internet Movie Database http://www.imdb.es/

for this experimentation considers a product rated with a value greater than or equal to 4 means the user liked the product, therefore, is relevant.

To evaluate the efficiency of the algorithms, it has been analyzed the time the algorithms take to build the model and obtain recommendations for a user.

The algorithms have been implemented by using the programming language JAVA, and the execution of the algorithms has been performed in a computer with CPU Intel Core i3-2600 CPU @ 3.40 GHz with RAM size of 8GB.

4.3. Results

Once the experiments has been described, its results are shown. A comparative analysis among the three models pointed out previously has been carried out by using the precision, recall and F_1-score, according to the number of recommendations that the recommender system provides to the user.

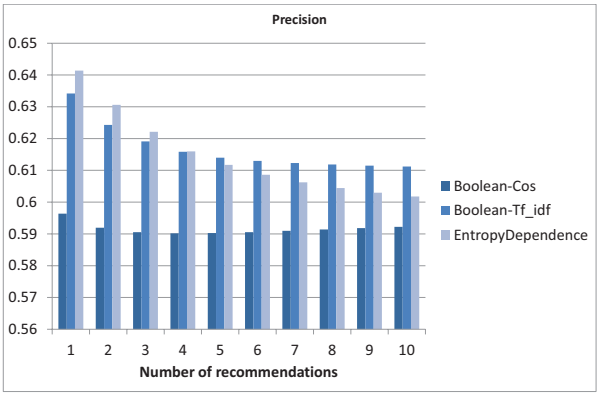

Fig. 2. Precision of the algorithms according to number of recommendations

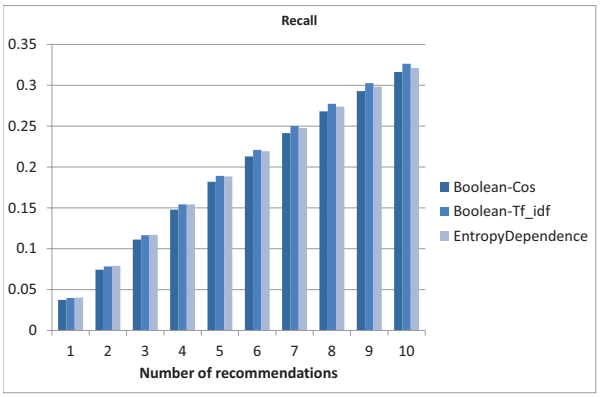

Fig. 3. Recall of the algorithms according to number of recommendations

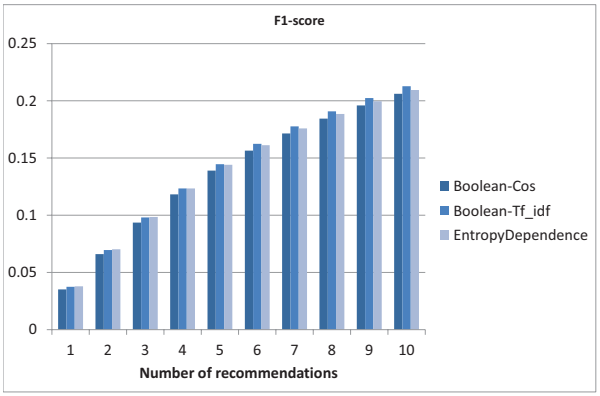

Fig. 4. $F_1 - score$ of the algorithms according to number of recommendations

Figure 2, shows the precision (Eq. 10) for different number of recommendations. As can be seen, the weighting of features, introduced in TF-IDF and EntropyDependence methods, improves the results of Boolean-Cos, which do not perform weighting of features. If we compare the approaches TF-IDF and EntropyDependence, it can be seen that EntropyDependence obtains better results when the number of recommendations provided by the system is lower than or equal to 4. This indicates that the proposed method is suitable when the user receives a short list of recommendations.

Figure 3, shows the recall (Eq. 11) for different number of recommendations and similarly to the results obtained for precision, our proposal improves Boolean-Cos approach for all values of number of recommendations

and TF-IDF when the number of recommendatinos is lower than or equal to 4.

In Figure 4, is shown the $F_1 - score$ measure (Eq. 12) for different number of recommendations. The results obtained by using the proposed method outperforms Boolean-Cos for all number of recommendations and TF-IDF when the number of recommendations is lower than or equal to 4.

It has been also analyzed the time for building the model of the CBRS and the time for computing recommendations to a user.

Figure 5 shows the model building time. The TF-IDF method takes four times more than our method and Boolean-Cos method, because the computation of the Inverse Document Frequency takes it $O(n \cdot m)$, where n is the number of boolean features, and m is the number of users in the system. The Boolean-Cos method is the fastest one, because it does not compute feature weights.

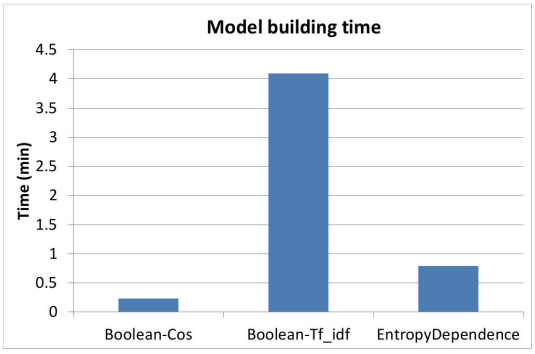

Fig. 5. Time spent in computing all user models

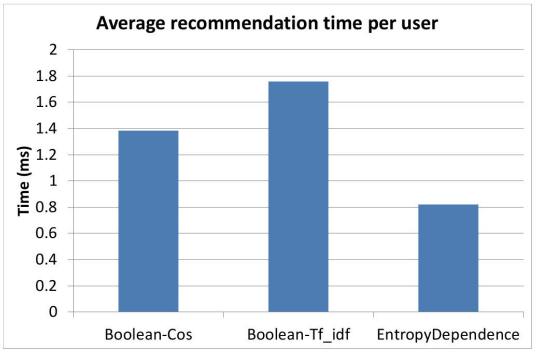

Fig. 6. Time spent in computing all user models

Figure 6 shows the average recommendation time for a user. This measure is the response time of the algorithm. In this case, the proposed method improves to Boolean-Cos and TF-IDF methods due the high dimensionality of the profiles, i.e., the Boolean-Cos and TF-IDF need to compute the similarity with larger profiles. This is caused by the transformation of the multivalued features into boolean features, e.g., in the dataset used in the experimentation, the items have 4 features that are transformed into 4000 boolean ones, because each boolean feature corresponds with a different value of the original feature.

This evaluation shows that the proposed model improves smoothly the effectiveness of the classic algorithms for CBRS with multivalued features. In addition, the efficiency obtained with the proposed model is better than the TF-IDF model both in the model building and in the delivering of recommendations.

5. Conclusions

The methods for weighting of features improve the results in content-based recommender systems. TF-IDF method is the most used method and it obtains successful results with boolean features. Nevertheless, it presents some drawbacks when the recommender system deals with multi-valued features or different information domains. In this contribution we have proposed a new method for computing feature weights in content-based recommendation systems, where features can be quantitative or qualitative. This method is based on two factors: intra-user similarity and inter-user dissimilarity. The former is computed by using Pearson correlation for quantitative features and Cramer V for qualitative ones. The latter is computed by means of entropy that measures the amount of information of each feature. Finally, we have evaluated the proposed method by using a dataset of movies and we have obtained successful results both in effectiveness and efficiency.

Acknowledgments

This work is partially supported by the Research Project TIN2012-31263, P08-TIC-03598, P10-AGR-6581, UJA2011/12/23 (supported by Caja Rural de Jaén) and ERDF funds.

References

1. G. Adomavicius and A. Tuzhilin: Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Trans. on Knowledge and Data Engineering, vol 17, No. 6, June, pp.734-749, (2005).
2. A. Aizawa: An information-theoretic perspective of TF-IDF measures. Information Processing and Management, 39:45-65 (2003).
3. N. Belkin and B. Croft: Information Filtering and Information Retrieval: Two Sides of the Same Coin?. Communications of the ACM 35(12):29-38, (1992).
4. Y.M.M.Bishop, S.E. Fienberg, P.W. Holland: Discrete Multivariate Analysis: Theory and Practice. The MIT Press, England (1995).
5. T. Bogers and A. Bosch: Comparing an evaluating information retrieval algorithms for news recommendation. Proc. of the 2007 ACM Conference on Recommender Systems, Minneapolis, USA, 141-144 (2007).
6. P. Burman: A comparative study of ordinary cross-validation, v-fold cross-validation and the repeated learning-testing methods. Biometrika, 76(3), 503-514 (1989).
7. T.M. Cover and J.A. Thomas: Elements of Information Theory. John Wiley & Sons, Inc. (1991).
8. H. Fang and T. Tao and C. Zhai: A formal study of information retrieval heuristics. Proc. of the 27th annual int. ACM SIGIR conf. on Research and development in information retrieval, 49-56 (2004).
9. T.P. Hong and J.B. Chen: Finding relevant attributes and membership functions. Fuzzy Sets and Systems 103: 389-404 (1999).
10. G.H. John and R. Kohavi and K. Pfleger: Irrelevant features and the subset selection problem. Machine Learning: Proc. of the 11th international conference, 121-129, Morgan Kaufmann Publishers, San Francisco, CA (1994).
11. L. Martínez and L.G. Pérez and M. Barranco: A Multi-granular Linguistic Content-Based Recommendation Model. International Journal of Intelligent Systems, 22(5):419-434 (2007).
12. R. Meteren and M. Someren Using content-based filtering for recommendation. Proc. of MLnet/ECML2000 Workshop, Barcelona, Spain (2000)
13. R.J. Mooney and L. Roy: Content-based book recommending using learning for text categorization. Proc. of the 15th ACM conf. on Digital libraries, Texas, USA, 195-204 (2000).
14. M.J. Pazzani and D. Billsus: Content-Based Recommendation Systems, The Adaptive Web. Lecture Notes in Computer Science, Springer-Verlag, 4321:325-341 (2007).
15. K. Pearson and J.A. Harris and A.E. Treloar and M. Wilder: On the Theory of Contingency. Journal of the American Statistical Association, 25(171):320-327 (1930).
16. G. Salton and D. Harman: Information retrieval. Encyclopedia of Computer Science, John Wiley and Sons Ltd., 858-863 (2003).
17. B. Sarwar and G. Karypis and J. Konstan and J. Riedl: Application of dimensionality reduction in recommender systems-a case study. In ACM WebKDD Workshop (2000).
18. C.E. Shannon: A mathematical theory of communication. The Bell System Technical Journal, 27:379-423,623-656 (1948).
19. E. H. Simpson: Measurement of Diversity. Nature, 163:688, doi:10.1038/163688a0 (1949).
20. P. Symeonidis and A. Nanopoulos and Y. Manolopoulos: Feature-weighted user model form recommender systems. Lecture Notes in Computer Science, Springer-Verlag, 4511:97-106 (2007).
21. Ho Chung Wu and R. W. Pong Luk: Interpreting TF-IDF term weights as making relevance decisions. ACM Trans. on Information Systems, 26(3):13,1-37 (2008).