A Method for Weighting Multi-valued Features in Content-Based Filtering

Manuel J. Barranco and Luis Martínez

University of Jaén, Campus Las Lagunillas, 23071, Jaén, Spain
barranco@ujaen.es, martin@ujaen.es
http://sinbad2.ujaen.es

Abstract. Content-based recommender systems (CBRS) and collaborative filtering are the type of recommender systems most spread in the e-commerce arena. A CBRS works with two sets of information: (i) a set of features that describe the items to be recommended and (ii) a user's profile built from past choices that the user made over a subset of items. Based on these sets and on weighting items features the CBRS is able to recommend those items that better fits the user profile. Commonly, a CBRS deals with simple item features such as key words extracted from the item description applying a simple feature weighting model, based on the TF-IDF. However, this method does not obtain good results when features are assessed in multiple values and or domains. In this contribution we propose a higher level feature weighting method based on entropy and coefficients of correlation and contingency in order to improve the content-based filtering in settings with multi-valued features.

1 Introduction

Recommendation systems are excellent tools to customize information in settings where the vast amount of information overloads users. Particularly, content-based recommender systems, CBRS [1,4,12,13], are one of the traditional types of such systems, which uses the available information about the choices that the user made in the past. This information is used to build a user profile that exposes the user's preferences or necessities. Besides, it is necessary a database of descriptive information about the items, each item is described by a set of features.

Typically, a CBRS works with textual analysis so that the features are words or terms that describe the items. In this way, given a set of features or terms, the vector describing an item is filled by ones and zeros that indicates whether or not a term appears in the text description of that item. Nevertheless, in a more general case, the features can be assessed by multi-valued variables or different domains: numeric, linguistic or nominal, boolean, etc.

Similar to other recommendation systems, the content-based ones present advantages and disadvantages. So they often are combined with other models such as collaborative or knowledge-based in order to improve its performance.

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part III, LNAI 6098, pp. 409–418, 2010.

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 fits the profile. 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 feature weighting which is subjective and different for each user.

The aim of this paper is to introduce a new method to obtain these weights, in recommendation settings where the features can be assessed with multi-valued variables or in multiple domains, by using the implicit ratings obtained from the users in the past. Thus, assigning weights to the 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 we take into account the entropy or amount of information for each feature, the more entropy the more weighting should have. And second we consider the correlation (for quantitative features) and contingency (for qualitative features), between items chose by the user in the past and the values of some features of the set of items. The greater the relationships, the higher the weight for the feature.

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

2 Preliminaries

In this section we review briefly content-based recommender systems and methods commonly used for weighting features in this setting.

2.1 Content-Based Recommender Systems

First we will review briefly content-based recommendation systems [1,11,13]. Those systems use a database with a set of items $A = \{a_i, i = 1...n\}$ described by a set of features $C = \{c_j, j = 1...m\}$ defined each one in a domain D_j, so that each item a_i is described by a vector $V_i = \{v_{ij} \in D_j, j = 1..m\}$ (see Table 1).

Table 1. Data for a CBRS

	c_1	...	c_j	...	c_m
a_1	v_{11}	...	v_{1j}	...	v_{1m}
...	...	...	...	...	...
a_n	v_{n1}	...	v_{nj}	...	v_{nm}

For each user, u, there exists a set $A_u = \{a_i^u \in A, i = 1, \ldots, n_u\}$, where a_i^u are the items experienced by the user, u. The assessments of their features

are described by, v_{ij}^u, and a user's preference assessments, $r_i^u \in D_u$ (implicit or explicit) being D_u the expression domain (see Table 2).

Table 2. User data for a CBRS

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

By using 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:

- $P_u = \left\{p_j^u \in D_j, j = 1..m\right\}$ are the user's values for item features. They can be obtained in different ways [1,11,13].
- $W_u = \left\{w_j^u, j = 1..m, 0 \leq w_j^u \leq 1\right\}$ are the weights that show the relevance of each feature, according to user's needs.

Once we know the available data for a CBRS, we will describe its working:

1. Acquisition of the items' features and users' profiles. The system updates the user profiles based on implicit information obtained in the past.
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 with each item.
3. Recommendations. The system selects the most similar items to user's necessities.

2.2 Feature Weighting in CBRS

We have aforementioned that the user profile, P_u, is used to filter the most suitable items together a weighting vector, W_u. In the literature is common a feature weighting method based on item descriptive words [13,15] based on the Term Frequency - Inverse Document Frequency (TF-IDF) [2] that has been used as a weighting scheme in Information Retrieval [7], Decision-Making problems [5], etc.

The use of TF-IDF feature weighting in CBRS [15], consists of user's profiles with zeros and ones that indicate the existence or absence of key words in the item description. The weights are computed for each user, u and each feature, c_j by using two factors: (i) A quantification of the intra-user similarity, FF (feature frequency), which indicates the characteristic frequency of c_j for user u and (ii) 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.

$$W(u, c_j) = FF(u, c_j) * IUF(c_j) \tag{1}$$

Commonly, the factor $FF(u, c_j)$, is computed by using the number of times that feature c_j appears in the items that user u has rated positively. The second factor, according to the TF-IDF scheme [2], is computed as $IUF(c_j) = Log\frac{|U|}{UF(c_j)}$ being $UF(c_j)$ the number of users that have rated positively any item that has the feature c_j, and $|U|$ the total number of users registered in the system. Such weights are used in the filtering process to match the user's profile and the items descriptions.

This feature weighting method is useful in CBRS dealing with binary features based on text descriptions, i.e., a feature is a word that can appear (value 1) or not appear (value 0) in an item's description. However, for those systems dealing with multi-valued features, the previous approach is not appropriate because a richer modelling with more than 2 values is necessary. The problem of weighting multi-valued features has been addressed in other areas like information retrieval and machine learning [9,10]. Nevertheless, this problem has been performed poorly in recommender systems settings. Our aim is to provide a proposal to address satisfactorily this issue in CBRS.

3 Feature Weighting Based on Entropy and Dependency Measures

Our aim is to propose a new feature weighting method for CBRS that can cope with multi-valued features. This proposal uses the data structure showed in Tables 1 and 2. We consider the item descriptions $V^u_{i.} = \{v^u_{ij}, j = 1 \dots m\}$ and the features descriptions $V^u_{.j} = \{v^u_{ij}, i = 1 \dots n_u\}$.

Our proposal consists of a feature weighting method that computes a weight for each feature according to (i) the amount of information provided by itself (multi-valued features can provide different amount of information), and (ii) the correlation between the items experienced by the user and the features of items. The feature weighting method follows the phases described below (see Figure 1):

1. Calculation of inter-user similarity. It is computed to know which features are more relevant to the user. For each feature c_j, we propose the use of the entropy H_j to compute the amount of information that it can offer.
2. Calculation of intra-user similarity. It is calculated the correlation between user's past items and the features values on the set of items. This calculation will depend on the nature of the features (qualitative, quantitative). For a feature c_j and a user u, it is calculated a coefficient of dependency, DC_{uj}, between the ratings obtained from the user, $R_u = \{r^u_i, i = 1, \dots n_u\}$, and the valuations of the feature in the items rated, $V^u_{.j} = \{v^u_{ij}, i = 1 \dots n_u\}$:
   - Correlation coefficient: for quantitative features.
   - Contingency coefficient: for qualitative features.

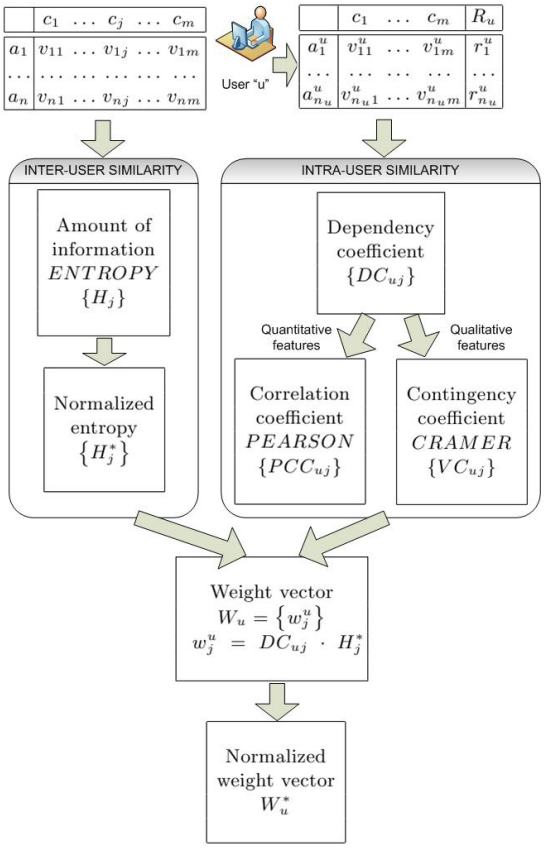

Fig. 1. Feature weighting based on entropy and dependency measures

3. Calculation of weights. Finally, it is obtained the feature weight as a result of the product of entropy and degree of dependency.

In the coming sections we show in further detail each phase.

3.1 Inter-user Similarity

To compute how informative is each feature we propose the use of the entropy.

Definition 1[6,14]. Entropy of information is the average amount of information, measured in bits, which contains a random variable. Given a random variable x, its entropy is given as:

$$H(x) = -\sum_i p(x_i)\, Log_2\,(p(x_i)) \tag{2}$$

Features with a greater entropy are most interesting and should have a greater weight. For example, given two features, c_1 and c_2 (see Table 3), if a user rates

positively the item a_1, the value $c_2 = 1$ must have more weight than $c_1 = A$ because it provides more information to the system.

Table 3. Example: 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

For each feature c_j the system computes the entropy H_j that is normalized in $H_j^* \in [0, 1]$ (see equation 3):

$$\begin{aligned} H_j &= -\textstyle\sum_{k_j}(f_{k_j}/n)Log_2(f_{k_j}/n) \\ H_j^* &= \frac{H_j}{\sum_i H_i} \end{aligned} \tag{3}$$

being $\{k_j\}$ the set of values that the feature c_j takes, 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 consideres Log 0 = 0. The value H_j^* indicates the amount of information that the feature c_j provides to the system. For example, for an attribute that only take two values, its entropy H is around 1 bit of information. While, another attribute that takes 16 different values is around 4 bits of information.

3.2 Intra-user Similarity

In this phase, the system measures the contingency or correlation between the ratings in the user's profile and the values of a feature c_j on the set of items. If there is a dependency between these variables, it suggests that the feature is important for the user. Depending on the nature of the feature we propose different measurements. For quantitative features is used a correlation coefficient. For qualitative features is used a coefficient of contingency. We propose to use two well known coefficients of dependency for measuring the intra-user similarity: Pearson's correlation coefficient for correlation, and Cramer's V coefficient for contingency [3].

Pearson's correlation coefficient should be used when data are approximately distributed according to a normal distribution. We assume this premise. In other cases we may use other coefficients, like Spearman's one [3], instead of Pearson's coefficient.

Definition 2 [3]: Pearson's correlation coefficient measures the linear relationship between two variables. Unlike the covariance, the Pearson correlation is independent of the scale of measurement of variables:

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

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

Definition 3 [3]: Cramer's V coefficient . It is a contingency ratio that measures the dependence between two random variables, X and Y, where at least one of them is qualitative.

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

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

The dependence coefficient, DC, between the ratings made by the user u and the values of items for the feature, c_j, is given by the following expression

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

being PCC_{uj} the Pearson's correlation coefficient corresponding to the variables R_u and $V_{.j}^u$.

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

and VC_{uj} is the Cramer's V, contingency coefficient corresponding to the same variables for qualitative features.

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

where k_u and k_j are indexes for the set of different values in R_u and $V_{.j}^u$ 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 also the type of dependence, direct or inverse. For our purposes it is not important the type of dependence. So, we take the absolute value, thus the result is in the interval [0,1]. In this way, and because of the Cramer V is also bounded in [0,1], DC coefficient will be bounded in that interval, being the value 1 the maximum dependence degree.

3.3 Calculation of Features' Weights

Once the factors H_j^* and DC_{uj} have been obtained, the system will compute the weight of a feature c_j as the product of both factors:

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

Since a vector of weights $\{w_i\}$ must satisfy the property $\sum w_i = 1$, the final vector of weights W_u^* is given by:

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

4 Example

Let us consider an example with a set of items $A = \{a_1, \ldots, a_{20}\}$ where each item is described by a set of features or characteristics $C = \{c_1, \ldots, c_5\}$ being $c_1, \ldots, c_4$ quantitative features assessed in the domains $\{1, \ldots, 4\}$, $\{1, \ldots, 6\}$, $\{1, \ldots, 20\}$ and $\{1, \ldots, 30\}$ respectively, and c_5 a qualitative feature assessed in the domain $\{A, B, C, D\}$. In Table 4 it is shown the description of each item according to such a set of features.

Table 4. Item-Feature matrix

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	a_{16}	a_{17}	a_{18}	a_{19}	a_{20}
c_1	1	3	2	2	4	2	1	1	4	2	2	1	1	3	4	1	2	4	1	2
c_2	2	6	2	3	4	4	2	6	5	4	3	6	2	4	5	2	2	4	2	4
c_3	2	5	12	14	8	16	4	4	15	14	18	17	10	13	17	1	1	19	9	11
c_4	5	20	10	11	25	13	5	2	28	12	9	4	6	18	27	3	10	29	1	14
c_5	A	C	B	B	D	B	A	A	D	B	B	A	A	C	D	A	B	D	A	B

On the other hand, we have a set of user's ratings in the domain $\{1, \ldots, 5\}$ that are shown in Table 5. These are the items that the user has rated in the past. In our case we assume 10 items have been already rated.

Our goal is to obtain a vector of weights for the five features given, to be used during the search of the items that best fit the user profile. So, we will apply entropy based weighting method.

Table 5. User's ratings

a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}
1	5	2	3	4	3	1	1	5	2

Calculation of entropies. We obtain: $H = \{H_j\} = \{1.86, 2.13, 3.92, 4.12, 1.86\}$. Its normalization provides a vector of relative entropies:

$$H^* = \left\{H_j^*\right\} = \{0.13, 0.15, 0.28, 0.30, 0.13\}$$

Calculation of dependence degrees. For the features $c_1, \ldots, c_4$ we calculate the correlation coefficient by using equation (6) obtaining the values:

$PCC_{u1} = 0.91$, $PCC_{u2} = 0.52$, $PCC_{u3} = 0.40$, $PCC_{u4} = 0.93$

For the qualitative feature c_5, the contingency coefficient obtains, $VC_{u5} = 0.73$. Therefore, the vector of dependence degrees will be:

$$DC_u = \{DC_{uj}\} = \{0.91, 0.52, 0.40, 0.93, 0.73\}$$

Calculation of features' weights. Finally, applying the formula (8) we obtain the vector of weights

$$W_u = \left\{w_j^u\right\} = \{0.12, 0.08, 0.11, 0.28, 0.10\}$$

We then normalize it according to (9) and obtain the final vector of weights for the features considered. That will be used by the CBRS in order to compute the recommendations.

$$W_u^* = \left\{w_j^{*u}\right\} = \{0.18, 0.12, 0.16, 0.40, 0.14\}$$

5 Conclusions and Future Works

The use of feature weighting methods in content based recommender systems has been a usual solution for their filtering processes. Additionally, the most common weighting method has been the TF-IDF, but it presents some drawbacks when the information manages by the recommender system is multi-valued or assessed in different domains.

In this contribution we have proposed a new method for calculating weights of features for content-based recommendation systems, where the features can be both quantitative and qualitative. The new method is based on two factors: intra-user similarity and inter-user dissimilarity that in our proposal the former is computed either by the Pearson correlation for quantitative features or by the Cramer's V for qualitative ones. And the latter is computed by the entropy that measures the amount of information of each feature.

Acknowledgements

This work is partially supported by the Research Project TIN2009-08286, P08-TIC-3548 and FEDER funds.

References

1. Adomavicius, G., Tuzhilin, A.: 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 17(6), 734–749 (2005)
2. Aizawa, A.: An information-theoretic perspective of TF-IDF measures. Information Processing and Managemente 39, 45–65 (2003)
3. Bishop, Y.M.M., Fienberg, S.E., Holland, P.W.: Discrete Multivariate Analysis: Theory and Practice. The MIT Press, England (1995)
4. Bogers, T., Bosch, A.: Comparing an evaluating information retrieval algorithms for news recommendation. In: Proc. of the 2007 ACM Conference on Recommender Systems, Minneapolis, USA, pp. 141–144 (2007)
5. Chung Wu, H., Pong Luk, R.W.: Interpreting tf-idf term weights as making relevance decisions. ACM Trans. on Information Systems 26(3), Article No. 13, 1–37 (2008)
6. Cover, T.M., Thomas, J.A.: Elements of Information Theory. John Wiley & Sons, Inc., Chichester (1991)
7. Fang, H., Tao, T., Zhai, C.: A formal study of information retrieval heuristics. In: Proc. of the 27th annual int. ACM SIGIR conf. on Research and depvelopment in information retrieval, pp. 49–56 (2004)
8. Hayes, C., Massa, P., Avesani, P., Cunningham, P.: An On-line Evaluation Framework for Recommender Systems. Technical Report TCD-CS-2002-19, Department of Computer Science, Trinity College Dublin (2002)
9. Hong, T.P., Chen, J.B.: Finding relevant attributes and membership functions. Fuzzy Sets and Systems 103, 389–404 (1999)
10. John, G.H., Kohavi, R., Pfleger, K.: Irrelevant features and the subset selection problem. In: Machine Learning: Proc. of the 11th int. conf., pp. 121–129. Morgan Kaufmann Publishers, San Francisco (1994)
11. Martínez, L., Pérez, L.G., Barranco, M.J.: A Multi-granular Linguistic Content-Based Recommendation Model. International Journal of Intelligent Systems 22(5), 419–434 (2007)
12. Mooney, R.J., Roy, L.: Content-based book recommending using learning for text categorization. In: Proc. of the 15th ACM conf. on Digital libraries, Texas, USA, pp. 195–204 (2000)
13. Pazzani, M.J., Billsus, D.: Content-Based Recommendation Systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 325–341. Springer, Heidelberg (2007)
14. Shannon, C.E.: A mathematical theory of communication. The Bell System Technical Journal 27, 379–423, 623–656 (1948)
15. Symeonidis, P., Nanopoulos, A., Manolopoulos, Y.: Feature-weighted user model form recommender systems. In: Conati, C., McCoy, K., Paliouras, G. (eds.) UM 2007. LNCS (LNAI), vol. 4511, pp. 97–106. Springer, Heidelberg (2007)