

A Case Study of Exploiting Data Mining Techniques for an Industrial Recommender System

Iván Cantador, Desmond Elliott, Joemon M. Jose

Department of Computing Science

University of Glasgow

Lilybank Gardens, Glasgow, G12 8QQ, UK

{cantador, delliot, jj}@dcs.gla.ac.uk

ABSTRACT

We describe a case study of the exploitation of Data Mining techniques for creating an industrial recommender system. The aim of this system is to recommend items of a fashion retail store chain in Spain, producing leaflets for loyal customers announcing new products that they are likely to want to purchase.

Motivated by the fact of having little information about the customers, we propose to relate demographic attributes of the users with content attributes of the items. We hypothesise that the description of users and items in a common content-based feature space facilitates the identification of those products that should be recommended to a particular customer.

We present a recommendation framework that builds Decision Trees for the available demographic attributes. Instead of using these trees for classification, we use them to extract those content-based item attributes that are most widespread among the purchases of users who share the demographic attribute values of the active user.

We test our recommendation framework on a dataset with one-year purchase transaction history. Preliminary evaluations show that better item recommendations are obtained when using demographic attributes in a combined way rather than using them independently.

Categories and Subject Descriptors

H.2.8 [Database Management] Database Applications – *Data Mining*. H.3.3 [Information Storage and Retrieval] Information Search and Retrieval – *information filtering, retrieval models*.

General terms

Algorithms, Measurements, Human Factors.

Keywords

Recommender Systems, Data Mining, Decision Trees.

1. INTRODUCTION

The application of recommender systems is achieving pervasive success, with general commerce¹, movie², music³, and joke⁴ recommendation services available. In this paper, we present a case study on how to exploit the combination of customer, purchase transaction, and item attributes for providing item

recommendations in the fashion retail business. Our goal is to implement a system that recommends products in a Spanish fashion retail store chain, to produce leaflets for loyal customers announcing new products that they are likely to want to purchase.

Typical recommendation techniques can be classed as either content-based or collaborative filtering approaches [1]. Content-based techniques suggest an item to a user based on a description of the item, and a profile of the user's interests [14], whilst collaborative filtering techniques filter or evaluate items for a user through the opinions of other people [19]. Combinations of both techniques are commonly grouped under the name of hybrid recommender systems [8].

Content-based recommenders rely on the fact that a user is interested in items similar to those he liked (purchased, searched, browsed, etc.) in the past [20]. They entail the description of items that may be recommended, the creation of a profile describing the types of items the user likes, and a strategy that compares item and user profiles to determine what to recommend. A user profile is often defined in terms of content features of the items, and is built and updated in response to feedback on the desirability of the items presented to the user.

Collaborative filtering systems, on the other hand, are built under the assumption that those users who liked similar interests in the past tend to agree again about new items. In general, interests for items are explicitly expressed by means of numeric ratings [17]. Thus, these techniques usually attempt to identify users who share the same rating patterns with the active user, and use the ratings from the identified like-minded users to calculate a prediction for him.

In our case study, both content-based and collaborative filtering recommendation approaches are difficult to be applied. First, we do not have detailed descriptions of customers and products. The available information about users basically consists of their age, gender and address (postal code, city/town or province), and descriptions of items are merely composed by designer, composition materials, price, and release season. Second, the existing database has a few transactions per user. As explained in following sections, the average number of items purchased by a user is 3.38. Finally, there is no explicit feedback (ratings, judgements, etc.) from the users about previous item purchases and/or recommendations.

Because of all these limitations, we propose an alternative recommendation strategy that exploits Data Mining techniques to describe and relate both users and items in a unique content-based feature space, allowing addressing the sparsity and lack of information. More specifically, we present an approach that builds Decision Trees from the entire dataset and for all the

¹ Amazon, <http://www.amazon.com>

² Netflix, <http://www.netflix.com>

³ Last.fm, <http://www.last.fm>

⁴ Jester, <http://shadow.ieor.berkeley.edu/humor/>

available demographic user attributes. Instead of using these trees for classification, we use them to extract those content-based item attributes that are most widespread among the purchases of users who share the input demographic attribute values of the active user.

The remainder of this paper is organised as follows. Section 2 briefly describes state-of-the-art recommender systems that utilise Data Mining techniques. Section 3 presents our case study in detail, explaining how the available dataset, with one-year transaction history, is processed. Section 4 explains the proposed recommendation framework. Preliminary evaluations of that framework are given in Section 5. Finally, Section 6 provides some conclusions and future work lines.

2. RELATED WORK

In content-based recommender systems, items are suggested according to a comparison between their content and user profiles, which contain information about the users' tastes, interests and needs. Data structures for both of these components are created using features extracted from the content of the items. The roots of content-based recommendations spring from the field of Information Retrieval [3], and thus many content-based recommender systems are focused on recommending items containing textual information [14].

Unlike content-based methods, collaborative filtering systems aim to predict the utility of items for a particular user according to the items previously evaluated by other users [19]. In general, users express their preferences by rating items. The ratings submitted by a user are taken as an approximate representation of his tastes, interests and needs in the application domain. These ratings are matched against ratings submitted by all other users, thereby finding the user's set of "nearest neighbours". Upon this, the items that were rated highly by the user's nearest neighbours are finally recommended [17].

In this context, Data Mining techniques allow inferring recommendation rules or building recommendation models from large datasets [18]. In commercial applications, Machine Learning algorithms are used to analyse the demographics and past buying history of customers, and find patterns to predict future buying behaviour. These algorithms include clustering strategies, classification techniques, and Association Rules production.

Clustering methods identify groups (clusters) of costumers who appear to have similar preferences. Once the clusters are created, the active user is suggested items based on average opinions of the customers in the cluster to which the user belongs. These strategies usually produce less personal recommendation than other methods, and have worse accuracy than collaborative filtering approaches [7]. An example of application of clustering in recommender systems can be found in [21].

Classifiers are general computational models for assigning a category (class) to an input instance (pattern). Input patterns are usually vectors of attributes or relationships among the products being recommended, and attributes of the customer to whom the recommendations are being made. Output classes may represent how strongly to recommend the input product to the active user. These learning methods first build and then apply models, so they are less suitable for applications where knowledge

of preferences changes rapidly. Examples of this approach are [2], [7], [12] and [13], where the application of Bayesian and Neural Network based classifiers for recommendation purposes is investigated.

Association rules express the relationship that one product is often purchased along with other products. Thus, they are more commonly used for larger populations rather than for individual customers. The weakness of this approach is its lack of scalability, since the number of possible Association Rules grows exponentially with the number of products in a rule. In the recommender systems research field, examples of works that exploit Association Rules are [4], [10], [11] and [16].

In this work, we propose to use Decision Trees, a particular divide-and-conquer strategy for producing classifiers [6]. They offer the following benefits [5]:

- They are interpretable. Unlike to other classifiers, which have to be seen as a black box that provides a category to a given input instance, Decision Trees can be visualised as tree graphs where nodes and branches represent the classification rules learnt, and leaves denote the final categorisations.
- They enable an easy attachment of prior knowledge from human expertise.
- They tend to select the most informative attributes measuring their entropy, boosting them to the top levels of the categorisation hierarchy.
- They are useful for non-metric data. The represented queries do not require any notion of metric, as they can be asked in a "yes/no", "true/false" or other discrete value set representations.

However, despite these advantages, Decision Trees are usually over-fitted, and do not generalise well to independent test sets. Two possible solutions are applicable: stopped splitting and pruning. C4.5 is one of the most common algorithms to build Decision Trees, and utilises heuristics for pruning based on statistical significance of splits [15]. We shall use its well-known revision J4.8 in our recommendation framework.

Differently to pervious approaches, in this work, we do not use Decision Trees as classifiers. Instead, we use them as mechanisms to map demographic user attributes to content-based item attributes. We shall build a Decision Tree for each demographic attribute in order to identify which content attribute values correlate most frequently with the input demographic attribute values of the active user's profile.

3. CASE STUDY

In this case study, we aim to providing recommendations to the loyal customers of a chain of fashion retail stores based in Spain.

In particular, the retail stores would like to be able to generate targeted product recommendations to loyal customers based on either customer demographics, customer transaction history, or item properties. A comprehensive description of the available dataset with the above information is provided in the next subsection. The transformation of this dataset into a format that can be exploited by Data Mining and Machine Learning techniques is described in Subsection 3.2.

3.1 Dataset

The dataset used for this case study contained data on customer demographics, transactions performed, and item properties. The entire dataset covers the period of 01/01/2007 – 31/12/2007. There were 1,794,664 purchase transactions by both loyal and non-loyal customers. The average value of a purchased item was €35.69.

We removed the transactions performed by non-loyal customers, which reduced the number of purchase transactions to 387,903 by potentially 357,724 customers. We refer to this dataset as **Loyal**. The average price of a purchased item was €37.81.

We then proceeded to remove all purchased items with a value of less than €0 because these represent refunds. This reduced the number of purchase transactions to 208,481 by potentially 289,027 customers. We refer to this dataset as **Loyal-100**.

3.2 Dataset Processing

Before we applied the Data Mining techniques described in Section 4, we processed the **Loyal** dataset to remove incomplete data for the demographic, item, and purchase transaction attributes.

3.2.1 Demographic Attributes

Table 1 shows the four demographic attributes we used for this case study. The average item price attribute was not contained in the database; it was derived from the data.

Attribute	Original Format	Processed Codification
<i>Date of birth</i>	String	Numeric age
<i>Address</i>	String	Province category
<i>Gender</i>	String	Gender category
<i>Avg. item price</i>	N/A	Derived numeric value

Table 1. Demographic attributes

The date of birth attribute was provided in seven different valid formats, alongside several invalid formats. The invalid formats results in 17,125 users being removed from the **Loyal** dataset. The date of birth was further processed to produce the age of the user in years. We considered an age of less than 18 to be invalid because of the requirement for a loyal customer to be 18 years old to join the scheme; we also considered an age of more than 80 to be unusually old based on the life expectancy of a Spanish person⁵. Customers with an age out with the 18 – 80 range were removed from the dataset.

Customers without a gender, or a Not Applicable gender were removed from the **Loyal-100** dataset. Finally, users who did not perform at least one transaction between 01/01/2007 and 31/12/2007 were removed from the dataset. An overview of the number of customers removed from the **Loyal-100** dataset can be seen in Table 2.

Customer Data Issue	Number removed
<i>Invalid birth date</i>	17,125
<i>Too young</i>	3,926
<i>Too old</i>	243
<i>Invalid province</i>	44,215
<i>Invalid gender</i>	3,188
<i>No transactions performed</i>	227,297
<i>Total users removed</i>	295,994
<i>Customers</i>	61,730

Table 2. Customer attributes issues in Loyal dataset

3.2.2 Item Attributes

Table 3 presents the four item attributes we used for this case study.

Attribute	Original Format	Processed Codification
<i>Designer</i>	String	Designer category
<i>Composition</i>	String	Composition category
<i>Price</i>	Decimal	Numeric value
<i>Release season</i>	String	Release season category

Table 3. Item attributes

The item designer, composition, and release season identifiers were translated to nominal categories. The price was kept in the original format and binned using the Weka toolkit⁶.

Items lacking complete data on any of the attributes were not including in the final dataset due to the problem of incomplete data.

Attribute issue	No. of items removed
<i>Invalid season</i>	9,344
<i>No designer</i>	10,497
<i>No composition</i>	2,788
<i>Total items removed</i>	22,629
<i>Items</i>	6,414

Table 4. Item attributes issues in Loyal dataset

3.2.3 Purchase Transaction Attributes

Table 5 presents the two transaction attributes we used for this case study.

Attribute	Original Format	Processed Codification
<i>Date</i>	String	Calendar season category
<i>Individual item price</i>	Decimal	Numeric value

Table 5. Transaction attributes

⁵ Central Intelligence Agency – The World FactBook, <https://www.cia.gov/library/publications/the-world-factbook/geos/sp.html>

⁶ Weka Machine Learning tool, <http://www.cs.waikato.ac.nz/ml/weka/>

The transaction date field was provided in one valid format and presented no parsing problems. The date of a transaction was codified into a binary representation of the calendar season(s) according to the scheme shown in Table 6. This codification scheme results in the “distance” between January and April being equivalent to the “distance” between September and December, which is intuitive.

	Spring	Summer	Autumn	Winter
January	0	0	0	1
February	1	0	0	1
March	1	0	0	0
April	1	0	0	0
May	1	1	0	0
June	0	1	0	0
July	0	1	0	0
August	0	1	1	0
September	0	0	1	0
October	0	0	1	0
November	0	0	1	1
December	0	0	0	1

Table 6. Codifying transaction date to calendar season

The price of each item was kept in the original decimal format and binned using the Weka toolkit. We chose not to remove discounted items from the dataset. Items with no corresponding user were encountered when the user had been removed from the dataset due to an aspect of the user demographic attribute causing a problem. An overview of the number of item transactions removed from the **Loyal** dataset based on the processing and codification step can be seen in Table 7.

Issue	No. of transactions removed
Refund	2,300
Too expensive	6,591
No item record	74,089
No user record	96,442
Total item purchases removed	179,422
Remaining item purchases	208,481

Table 7. Transaction attributes issued in Loyal dataset

As a result of performing these data processing and cleaning steps, we are left with a dataset we refer to as **Loyal-Clean**. An overview of the **All**, **Loyal**, and the processed and codified dataset, **Loyal-Clean**, is shown in Table 8.

	All	Loyal	Loyal-Clean
Item transactions	1,794,664	387,903	208,481
Customers	N/A	357,724	61,730
Total Items	29,043	29,043	6,414
Avg. items per customer	N/A	1.08	3.38
Avg. item value	€35.69	€37.81	€36.35

Table 8. Dataset observations

4. RECOMMENDATION FRAMEWORK

The recommendation framework we propose for this case study recommends items to a user by a) transforming the demographic attributes of the user’s profile into a set of weighted content-based item attributes, and 2) comparing the obtained content-based user profile with item descriptions in the same feature space. Figure 1 depicts the profile transformation and item recommendation processes, which are conducted in four stages:

1. Decision Trees are generated on different user attributes (UA) in terms of the item attributes (IA). The nodes of a tree are associated to the item attributes IA , which may appear at several levels of the tree. The output branches of a node IA_j correspond to the possible values of such attribute: $v_1(IA_j), v_2(IA_j), \dots, v_k(IA_j)$. The leaves of the tree are certain values $v_1(UA_i), v_2(UA_i), \dots, v_l(UA_i)$ of user attributes UA_i . The demographic attributes in the profile of the active user u_m are matched with the tree leaves that contain the specific values $v_m(UA_i)$ of the user profile. More details are given in Section 4.1.
2. The item attribute values comprising a user attribute value occurrence in these Decision Trees are re-weighted based on depth and classification accuracy, as explained in Section 4.2.
3. The re-weighted item attributes are linearly combined to produce a user profile in terms of weighted item attributes. This combination is described in Section 4.3.
4. The generated user profile is finally compared against the attributes of different items in the dataset to predict the probability of an item being relevant to a user. Section 4.4 presents the comparison strategy followed in this work.

4.1 Demographic Decision Trees

In this stage, the entire dataset containing information about users, purchase transactions, and items (see Section 3) is used to build Decision Trees on the demographic attributes. For that purpose, the database records are converted into a format processable by Machine Learning models.

For each item purchased in the **Loyal-Clean** dataset, the following data are output to a file in Attribute-Relation File Format⁷ (ARFF) for classification:

- User province
- User gender
- User age
- User average item price
- Designer category
- Composition category
- Spring purchase
- Summer purchase
- Autumn purchase
- Winter purchase
- Item purchase price

where the attributes *User average item price* and *Item purchase price* are both transformed into ten discrete bins.

⁷ Attribute-Relation File Format, <http://www.cs.waikato.ac.nz/~ml/weka/arff.html>

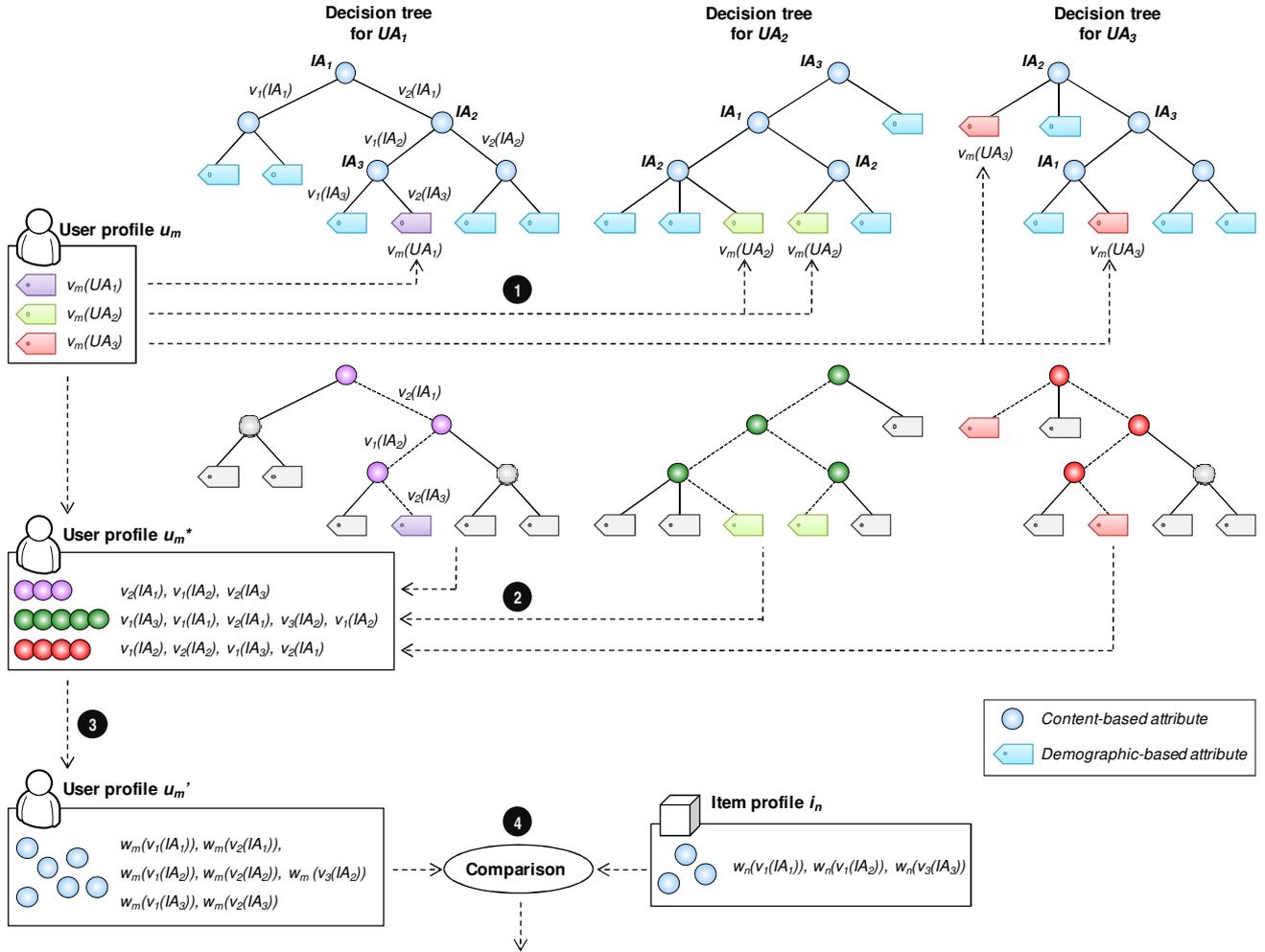


Figure 1. Recommendation framework that transforms demographic user attributes (UA) into content item attributes (IA) by using Decision Trees, and recommends items to users comparing their profiles in a common content-based feature space

ARFF, which is a format generally accepted by the Machine Learning research community⁸, establishes the way to describe the problem attributes (type, valid values), and to provide data patterns (instances) from which building the models.

We create individual Decision Trees using the C4.5/J4.8 algorithm for the *User province*, the *User age*, and the *User average item price* attributes.

The general structure of these Decision Trees can be seen in Figure 1, label 1. The nodes of the Decision Trees represent the item (content) attributes, whereas the leaves represent the user (demographic) based attribute values. The produced demographic Decision Trees are then used to determine the item attributes that best define the user in terms of the item attributes to which they are most closely related.

4.2 Item Attribute Weighting

Given the Decision Trees created for the three user attributes, the next step in the framework is to attach a weight to the

contribution of those content nodes IA_j leading from the root of the tree to the relevant leaf $v_l(UA_i)$ as specified in the user profile \mathbf{u}_m . The weight w_m that each value of content node contributes to the user profile is given by the following equation:

$$w_m(v_k(IA_j)) = \text{depth}(v_k(IA_j)) \cdot \left(1 - \frac{\text{neg}(v_l(UA_i))}{\text{pos}(v_l(UA_i))}\right) \cdot \text{pos}(v_l(UA_i))$$

where v_k is the value of the content node, for example: “Designer 1” or “Designer 2”; $\text{depth}(v_k(IA_j))$ represents the depth of node IA_j with value v_k on this path through the Decision Tree; $\text{neg}(v_l(UA_i))$ represents the number of incorrectly classified purchase transaction instances from the **Loyal-Clean** dataset at this value of the content node, and $\text{pos}(v_l(UA_i))$ represents the number of correctly classified purchase transaction instances from the **Loyal-Clean** dataset at this value of the content node. This stage is illustrated in Figure 1 at label 2, where the new version of the user profile is represented as \mathbf{u}_m^* .

⁸ UCI Machine Learning Repository, <http://archive.ics.uci.edu/ml/>

This method of calculating the weight of the value of a content node gives higher significance to content nodes occurring closer to the demographic leaves.

4.3 Item Attribute Combination

The weighted values $w_m(v_k(IA_j))$ of the content attributes IA_j in the user profile \mathbf{u}_m^* are linearly combined to produce a final description of the user profile \mathbf{u}_m in the item attribute space (see Figure 1, label 3).

This produces a representation of the user that can be directly compared to the representation of items for the purpose of recommending items.

4.4 Item Recommendation

In this final stage (Figure 1 at label 4) of the recommendation framework, the user profile \mathbf{u}_m is compared to item profiles \mathbf{i}_n in the item feature space. We use the cosine similarity between both profiles to create a ranked list of the similarity of items to the user.

$$score(u_m, i_n) = \cos(\mathbf{u}_m, \mathbf{i}_n) = \frac{\mathbf{u}_m \cdot \mathbf{i}_n}{\|\mathbf{u}_m\| \times \|\mathbf{i}_n\|}$$

The top scored items are the ones recommended to the user. Note that alternative similarity measures can be computed [9]. This forms part of our future work tasks. See Section 6 for a discussion on this issue.

5. EVALUATION

In this Section, we present preliminary evaluations of the proposed recommendation framework. These evaluations are focused on the comparison of the item recommendations obtained when exploiting a single Decision Tree associated to a demographic attribute (age, province, average customer item price), against the item recommendations obtained when exploiting the three Decision Trees.

From our **Loyal-Clean** dataset, described in Section 3.1, we build three Decision Trees, each of them associated to one of the above demographic attributes, as explained in Section 4. Afterwards, we run our recommender using the information of each tree in an isolated and combined way, and evaluate its outputs for all users and items in the dataset.

The goal of this study is twofold. First, we aim to identify which of the used demographic attributes contributes more significantly to assigning higher personal scores to items purchased by a particular user. Second, we are interested in determining whether or not the combination of several demographic attributes enhances the items recommendations.

The specific nature of our domain of application, the sparsity of the available data, and the lack of explicit feedback from the customers about their transactions, make very difficult to properly evaluate the presented recommendation framework. To our knowledge, experimentation on a similar scenario has not been published yet in the recommender system research community. Thus, we are not able to compare our approach with state of the art recommendation strategies.

Moreover, precision-based metrics, such as Mean Average Error (MAE) or F -measure, are not applicable in our problem because we do not have user judgements (e.g., ratings) for item recommendations. For this reason, we consider the use of coverage-based metrics well known in the Information

Retrieval field, such as Recall and Mean Reciprocal Rank (MRR) metrics. Instead of evaluating how accurate our item recommendations are, for a given user, we compute the percentage of his purchased items that appear in his list of top recommended items.

Let $J_m \subset J$ be the set of items purchased by user $u_m \in \mathcal{U}$, and let $rank(u_m, i_n) \in \mathbb{N}$ be the rank (i.e., ranking position) of item $i_n \in J_m$ in the list of item recommendations provided to user u_m .

We define recall at Q as follows:

$$recall_Q = \frac{1}{|\mathcal{U}|} \sum_{u_m \in \mathcal{U}} \frac{|i_n \in J_m \wedge rank(u_m, i_n) \leq Q|}{|i_n \in J_m|} \in [0,1]$$

A recall value of 1 means that 100% of the purchased items is retrieved within the top Q recommendations. A recall value of 0 means that no purchased items are retrieved within the top Q recommendations.

Figure 1 shows the average recall values for the top $Q = 10, \dots, 1000$ recommended items for users who bought at least 10 items. These values were obtained with recommenders using the different Decision Trees.

The obtained small recall values are due to the fact that we are attempting to retrieve the small set of items purchased by a user (around 10 items) from the whole set of 6414 items (see Table 1). With such a difference, it is practically impossible to recommend the purchased items in the top positions. Exploiting the information of all the trees, we are able to retrieve the 10% of those items within the top 150 recommendations, whilst, at position 1000, we retrieve around 65% of the purchased items.

Comparing the recall curves for the three Decision Trees, it can be seen that the combination of information inferred from all the trees improves the recommendations obtained from the exploitation of information from a single tree. The demographic attribute that best recall values achieves is *User age*, followed by *User province* and *User average item price* attributes. These results are reasonable. People with comparable ages or living in close provinces buy related products (i.e., similar clothes), and thus are more effectively clustered for recommendation purposes. In this context, we believe that a better categorisation of customers based on geographic locations may improve our results. Instead of considering individual provinces, we could group them in different regions, according to weather and physical conditions (e.g., by differentiating mountain and coast regions). The average price, on the other hand, is applied to any type of products, and thus it is not suitable to discriminate clusters of related users and purchase transactions.

Complementary to recall metric, we compare the different rankings measuring the Mean Reciprocal Rank [22], which we define for our problem as follows:

$$MRR_Q = \frac{1}{|\mathcal{U}|} \sum_{u_m \in \mathcal{U}} \left(\sum_{\substack{i_n \in J_m \\ rank(u_m, i_n) \leq Q}} \frac{1}{rank(u_m, i_n)} \right) \in [0,1]$$

Values of MRR_Q close to 1 indicate that the relevant (i.e., purchased) items appear in the first positions of the rankings, and values of MRR_Q close to 0 reveal that the purchased items are retrieved in the last ranking positions.

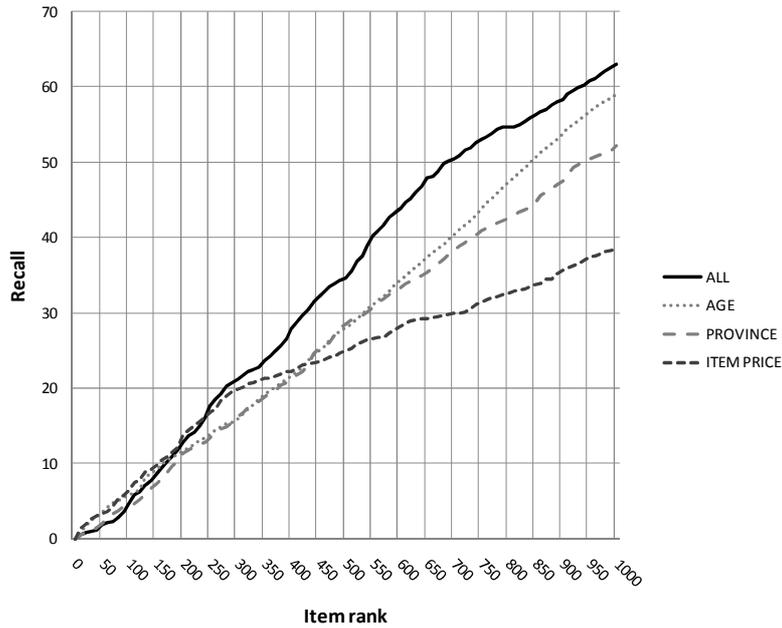


Figure 2. Average recall values

Table 9 shows the MRR_Q values of our recommender exploiting each Decision Tree. Analogously to recall metric, because of the large number of available items, very low MRR_Q values are obtained. We also measure the *Average Score* of the purchased items. As shown in the table, the average score when using all the trees is 0.866, which is very close to 1. Despite these high score values, MRR_Q values are low. Analysing the scores of the items in the top positions, we find out that many of the items were assigned the same score. As future work, we shall investigate alternative matching functions between users and items profiles in order to better differentiate them. The establishment of thresholds in the weights of the attributes in the profiles is another possibility to be tested.

In any case, these results show again that the exploitation of all attributes seems to improve the recommendations obtained when only one attribute is used.

	Age	Province	Item price	All
MRR_Q	0.014	0.012	0.011	0.017
<i>Avg. item score</i>	0.855	0.644	0.531	0.866

Table 9. Mean reciprocal rank and Avg. item score values

6. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a recommendation framework that applies Data Mining techniques to relate user demographic attributes with item content attributes. Based on the description of customers and products on a common content-based feature space, the framework provides item recommendations in a collaborative way, identifying purchasing trends in groups of customers that share similar demographic characteristics.

We have implemented and tested the recommendation framework in a real case study, where a Spanish fashion retail store chain needs to produce leaflets for loyal customers announcing new products that they are likely to want to purchase.

Exploiting Decision Trees built from a dataset with one-year transaction history to transform user demographic user profiles, Preliminary evaluations have shown that better item recommendations are obtained when using demographic attributes in a combined way rather than using them independently.

Our recommender is based on computing and exploiting global statistics on the purchase transactions made by all the customers. This allows providing a particular type of content-based collaborative recommendations. As future work, we want to study the impact of strength personalisation within the recommendation process giving higher priority to items similar to those purchased by the user in the past. In this work, we were limited to a database of transactions made in one year, having an average of 3.38 items purchased per user. We expect to increase our dataset with information of further years in order to enhance our recommendations, and improve the evaluations.

In our approach, user profiles, represented in terms of item attributes, can become populated with fractionally weighted item attribute values. This means that the framework will calculate similarity between a user and an item when there is no possible chance of the item being similar enough to the user profile to be a worthwhile recommendation. A possible approach to resolving this issue is to remove values of item attributes in a user profile that fall below a certain threshold.

In addition to Decision Trees, we plan to investigate alternative Machine Learning strategies to relate demographic and content-based attributes. Of special interests are Association Rules. Although the number of possible rules grows exponentially with the number of items in a rule, approaches that constrain the confidence and support levels of the rules, and thus reduce the number of generated rules, have been proposed in the literature [11].

We also expect to collaborate with the client in order to gather explicit judgements or implicit feedback (via purchase log analysis) from customers about the provided item recommendations. Then, we could compare the accuracy results obtained with our recommendation framework against those obtained with for example approaches based on recommending most popular (purchased) items.

7. ACKNOWLEDGEMENTS

This research was supported by the European Commission under contracts FP6-027122-SALERO, FP6-033715-MIAUCE and FP6-045032 SEMEDIA. The expressed content is the view of the authors but not necessarily the view of SALERO, MIAUCE and SEMEDIA projects as a whole.

8. REFERENCES

- [1] Adomavicius, G., Tuzhilin, A. 2005. *Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-art and Possible Extensions*. In: IEEE Transactions on Knowledge and Data Engineering, 17(6) pp. 734–749.
- [2] Ansari, A., Essegaier, S., Kohli, R. 2000. *Internet Recommendation Systems*. In: Journal of Marketing Research, 37(3), pp. 363-375.
- [3] Baeza-Yates, R., Ribeiro-Neto, B. 1999. *Modern Information Retrieval*. ACM Press, Addison-Wesley.
- [4] Basu, C., Hirsh, H., Cohen, W. 1998. *Recommendation as Classification: Using Social and Content-based Information in Recommendation*. In: Proceedings of the AAAI 1998 Workshop on Recommender Systems.
- [5] Bellogin, A., Cantador, I., Castells, I., Ortigosa, A. 2008. *Discovering Relevant Preferences in a Personalised Recommender System using Machine Learning Techniques*. In: Proceedings of the ECML-PKDD 2008 Workshop on Preference Learning.
- [6] Bishop, C. M. 2006. *Pattern Recognition and Machine Learning*. Jordan, M., Kleinberg, J., Schoelkopf, B. (Eds.). Springer.
- [7] Breese, J. S., Heckerman, D., Kadie, C. 1998. *Empirical Analysis of Predictive Algorithms for Collaborative Filtering*. In: Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI 1998), pp. 43-52.
- [8] Burke, R. 2002. *Hybrid Recommender Systems: Survey and Experiments*. In: User Modeling and User-Adapted Interaction, 12(4), pp. 331-370.
- [9] Cantador, I., Bellogin, A., Castells, P. 2008. *A Multilayer Ontology-based Hybrid Recommendation Model*. In: AI Communications, 21(2-3), pp. 203-210.
- [10] Haruechaiyasak, C., Shyu, M. L., Chen, S. C. 2004. *A Data Mining Framework for Building a Web-Page Recommender System*. In: Proceedings of the 2004 IEEE International Conference on Information Reuse and Integration (IRI 2004), pp. 357-362.
- [11] Lin, W., Álvarez, S. A., Ruiz, C. 2002. *Efficient Adaptive-Support Association Rule Mining for Recommender Systems*. In: Data Mining and Knowledge Discovery, 6(1), pp. 83-105.
- [12] Mooney, R. J., Roy, L. 2000. *Content-based Book Recommending Using Learning for Text Categorization*. In: Proceedings of the 5th ACM Conference on Digital Libraries, pp. 195-240.
- [13] Pazzani, M. J., Billsus, D. 1997. *Learning and Revising User Profiles: The Identification of Interesting Websites*. In: Machine Learning, 27(3), pp. 313-331.
- [14] Pazzani, M. J., Billsus, D. 2007. *Content-Based Recommendation Systems*. In: The Adaptive Web, Brusilovsky, P., Kobsa, A., Nejdl, W. (Ed.). Springer-Verlag, Lecture Notes in Computer Science, 4321, pp. 325-341.
- [15] Quinlan, J. R. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publisher.
- [16] Sarwar, B. M., Karypis, G., Konstan, J. A., Riedl, J. 2000. *Analysis of Recommendation Algorithms for E-Commerce*. In: Proceedings of the 2nd Annual ACM Conference on Electronic Commerce (EC 2000), pp. 158-167.
- [17] Sarwar, B. M., Konstan, J. A., Borchers, A., Herlocker, J., Miller, B., Riedl, J. 1998. *Using Filtering Agents to Improve Prediction Quality in the GroupLens Research Collaborative Filtering System*. In: Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work (CSCW 1998), pp. 345-354.
- [18] Schafer, J. B. 2005. *The Application of Data-Mining to Recommender Systems*. In: Encyclopaedia of Data Warehousing and Mining, Wang, J. (Ed.). Information Science Publishing.
- [19] Schafer, J. B., Frankowski, D., Herlocker, J., Sen, S. 2007. *Collaborative Filtering Systems*. In: The Adaptive Web, Brusilovsky, P., Kobsa, A., Nejdl, W. (Ed.). Springer-Verlag, Lecture Notes in Computer Science, 4321, pp. 291-324.
- [20] Terveen, L., Hill, W. 2001. *Beyond Recommender Systems: Helping People Help Each Other*. In: Human-Computer Interaction in the New Millennium, Carroll, J. M. (Ed.), pp. 487-509. Addison-Wesley.
- [21] Ungar, L. H., Foster, D. P. 1998. *Clustering Methods for Collaborative Filtering*. In: Proceedings of the AAAI 1998 Workshop on Recommender Systems.
- [22] Voorhees, E. M. 1999. *Proceedings of the 8th Text Retrieval Conference*. In: TREC-8 Question Answering Track Report, pp. 77–82.