Implementation of a New Recommendation System Based on Decision Tree Using Implicit Relevance Feedback

Anıl Utku*, Hacer Karacan, Oktay Yıldız, M. Ali Akcayol
Gazi University Computer Engineering Department, Maltepe, Ankara, Turkey.

* Corresponding author. Tel.: +90(312) (5823130); email: anilutku@gazi.edu.tr
Manuscript submitted June 5, 2015; accepted August 30, 2015.
doi: 10.17706/jsw.10.12.1367-1374

Abstract: Recommendation Systems (RSs) are used to provide users useful and effective suggestions. Effectiveness of RSs is depend on the quality of the suggestions. In this study, a new RS based on decision tree (DT) using implicit relevance feedback have been developed for movies. User behavior as implied relevance feedback is modeled by clickstreams. The DT constructed by Gini algorithm. The experimental results show that the developed method is successful for effective and useful suggestions.

Key words: Recommendation systems, user behavior analysis, collaborative filtering, relevance feedback.

1. Introduction

RSs deal with usable suggestions to users that may match with their interests. Accurate recommendations enable users to find the desired items without being overwhelmed by irrelevant information. Electronic commerce systems and social networks present new opportunities as to further improve the accuracy of RSs. In these online platforms, user-item ratings can be used for advising other users. RSs have been grouped as collaborative-filtering and content-based filtering [1], [2]. A RS with collaborative filtering predicts the relevant items of users according to the similar users [3]-[9]. These systems either use a model or neighborhood information in order to obtain the correlation [2], [3]. Model based approaches gather user-item ratings to model the user interactions. Neighborhood-based approaches use user-item ratings to predict ratings for new items [4].

Content-based RSs are based on users' tend in time [10], [11]. These systems often use machine-learning techniques to create a profile of a user [5]. In order to predict whether a new item is a good recommendation or not, content-based RSs rely on similarity between the new item and the rated items stored as part of the model [5], [10]. There are also some hybrid RSs that combine collaborative filtering with content information [12]-[17]. These hybrid RSs create a model for each user by monitoring the behavior or by analyzing declared interests or feedback of the user [18].

In this study, a hybrid RS has been developed that uses DT with the Classification and Regression Tree (CART) algorithm using implicit relevance feedback. The DT is used as a prediction model which maps the input to a predicted value. Similar to the collaborative-filtering, the information gathered from user has been used to classify the movies into two groups, like or dislike. The attributes are constructed using the rating history and content of the items. The constructed DT is used by all users as to the collaborative approach [19]. In order to apply CART algorithm, Gini impurity has been used as a measure of how often a randomly chosen element would be incorrectly labeled if it were randomly labeled according to the

distribution of class in the subset.

2. Recommendation System Based on Decision Tree Using Implicit Relevance Feedback

RSs have been developed to meet the needs of users automatically [20]. Lately, prominent RSs can deal with the pile of unnecessary or irrelevant information [21]. RSs focus on the new users with no experience or history on the items. In the RSs, user requirements, existing contents, various information about users and the past operations of the users are stored in the customized databases. Feedback from the users are gathered either by asking or by analyzing their behavior while using the system [20].

DTs are widely used for classification problems to provide accurate models [22]. DTs are composed of a root node, child nodes and leaf nodes. In DTs, nodes represent the areas of inquiry, branches show the query results and leaves represent the class tags. The most commonly used DT algorithms are CHi-squared Automatic Interaction Detection (CHAID) [23], Iterative Dichotomiser 3 (ID3) [24], CART [25], C4.5 [26] and Quest [27].

Constructing a DT consists of two major stages; creating a tree and pruning the tree. Each node in the DT refers to an attribute and each leaf represents a class tag as shown in Fig. 1.

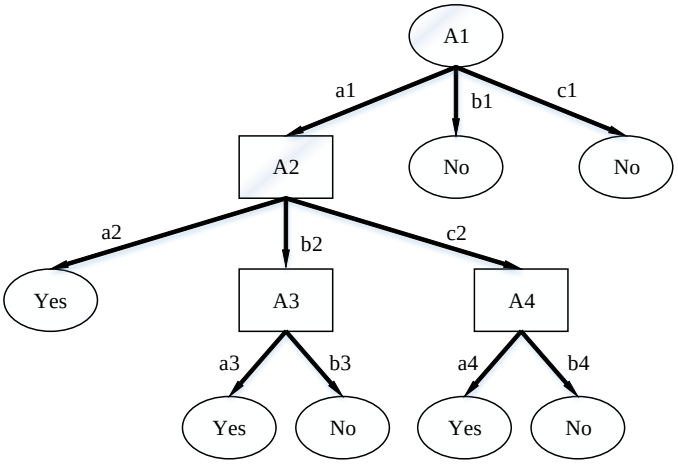

Fig. 1. Decision tree.

To create a DT, a training set have to be used in learning phase. The DT showed in Fig. 1 was created using Table 1.

Table 1. Training Set

A1	A2	A3	A4	Class
a1	a2	a3	a4	Yes
a1	a2	a3	b4	Yes
a1	b2	a3	a4	Yes
a1	b2	b3	b4	No
a1	c2	a3	a4	Yes
a1	c2	a3	b4	No
b1	b2	b3	b4	No
c1	b2	b3	b4	No

where, A1, A2, A3 and A4 are attributes, a, b, c are values and Yes/No are class tags.

Gini algorithm uses impurity measure and Gini index instead of entropy and average entropy respectively. To calculate Gini index value for an attribute, the equations, listed below, have been used.

$$Gini\,(S) = 1 - \sum_{j=1} p_j^2$$

$$Gini\,(S, A) = \sum_{i=1} (|\,S_i\,| / |\,S\,|)\; Gini\,(S_i) \qquad (1)$$

where, P_j shows the relative frequency of class *j* in target class. |*S*| the size of D, |S_i| is the size of split (left or right) of an attribute.

3. Experimental Results

In this study, the user data obtained from http://www.tavsiyemotoru.com implicitly and has been classified using the Gini algorithm. The Web site has 10 different categories for movies such as romantic (C1), action (C2), animation (C3), sci-fi (C4), drama (C5), thriller (C6), comedy (C7), horror (C8), adventure (C9) and crime (C10). The user data is composed of clickstreams on movies. The users are 200 undergraduate and graduate students from Gazi University Department of Computer Engineering between 20 to 30 years old. Table 2 shows number of users' clicks for the first ten users as an example.

Table 2. The Number of Users' Clicks for the First Ten Users

No	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
1	6	6	2	4	2	2	2	1	2	3
2	3	7	0	10	3	8	6	0	5	1
3	0	1	1	0	0	0	0	3	0	0
4	1	2	1	2	1	1	1	5	0	1
5	1	0	1	1	0	1	0	1	0	5
6	0	0	1	1	4	1	0	1	0	0
7	2	1	3	0	2	4	0	0	0	2
8	0	0	1	1	0	3	0	0	0	0
9	5	2	2	2	2	1	1	1	2	0
10	0	1	2	0	0	2	0	3	0	0

Class labels have been assigned for transactions according to clickstreams as shown in Table 3.

Table 3. Assigned Class Labels

No	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	Class
1	5	6	2	4	2	2	2	1	2	3	S2
2	3	7	0	10	3	8	6	0	5	1	S4
3	0	6	1	0	0	0	0	3	0	0	S2
4	1	2	1	2	1	1	1	5	0	1	S8
5	1	0	1	1	0	1	0	1	0	5	S10
6	0	0	1	1	4	1	0	1	0	0	S5
7	2	1	3	0	2	4	0	0	0	2	S6
8	0	0	1	1	0	3	0	0	0	0	S6
9	5	2	2	2	2	1	1	1	2	0	S1
10	0	1	2	0	0	2	0	3	0	0	S8

The proposed method consists of pre-processing, conversion of numerical data to categorical data and application of Gini algorithm. At the pre-processing stage, class labels have been assigned for each transactions using clickstreams data. To convert numerical data to categorical data, different threshold values have been used for each attribute. In order to determine the threshold value, all users' click

information are ranked and divided into two categories: bigger (b) than the threshold value and smaller or equal (se) to the threshold value. The determined threshold vales are given in Table 4.

Categorical data for the first nine users are given in Table 5 below.

Table 4. Threshold Values

Category	Threshold value
C1	3
C2	5
C3	2
C4	4
C5	3
C6	4
C7	3
C8	3
C9	3
C10	3

Table 5. Categorical Data

No	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	Class
1	b1	b2	se3	se4	se5	se6	se7	se8	se9	se10	S2
2	se1	b2	se3	b4	se5	b6	b7	se8	b9	se10	S4
3	se1	b2	se3	se4	se5	se6	se7	se8	se9	se10	S2
4	se1	se2	se3	se4	se5	se6	se7	b8	se9	se10	S8
5	se1	se2	se3	se4	se5	se6	se7	se8	se9	b10	S10
6	se1	se2	se3	se4	b5	se6	se7	se8	se9	se10	S5
7	se1	se2	b3	se4	se5	se6	se7	se8	se9	se10	S6
8	se1	se2	se3	se4	se5	b6	se7	se8	se9	se10	S6
9	b1	se2	se3	se4	se5	se6	se7	se8	se9	se10	S1
10	se1	se2	se3	se4	se5	se6	se7	se8	se9	se10	S8

Once converted numeric data to categorical data, $Gini_{left}$ and $Gini_{right}$ values for each category and their total value of the Gini Index are calculated as shown in Table 6.

Table 6. $Gini_{left}$, $Gini_{right}$ and $Gini_{Total}$ Values

Category	**Gini $_{left}$**	**Gini $_{right}$**	**Gini $_{Total}$**
C1	0,88572	0,61111	0,87784
C2	0,88651	0,62500	0,88151
C3	0,88697	0,50000	0,88327
C4	0,88809	0	0,85835
C5	0,88741	0	0,85769
C6	0,88628	0,37500	0,87649
C7	0,88604	0,50000	0,87865
C8	0,88764	0,44444	0,88128
C9	0,88589	0, 44444	0,87956
C10	0,88555	0,37037	0,86336

The smallest Gini Index value has been used for the first branching node in the DT. Category 5, with the smallest Gini Index value, has been selected for the first branching node. After the first branching, Gini

algorithm is applied iteratively for calculating other branches. The rules of the DT have been shown in Table 7.

Table 7. Rules Obtained from the Decision tree

Rule #	Rules
1	IF (b1, b2, se3, se4, se5, se6, se7, se8, se9, se10) THEN CLASS = S2
2	IF (se1, b2, se3, b4, se5, b6, b7, se8, b9, se10) THEN CLASS = S4
3	IF (se1, se2, se3, se4, se5, se6, se7, b8, se9, se10) THEN CLASS = S8
4	IF (se1, se2, se3, se4, se5, se6, se7, se8, se9, b10) THEN CLASS = S10
5	IF (se1, se2, se3, se4, b5, se6, se7, se8, se9, se10) THEN CLASS = S5
6	IF (se1, se2, b3, se4, se5, se6, se7, se8, se9, se10) THEN CLASS = S6
7	IF (se1, se2, se3, se4, se5, b6, se7, se8, se9, se10) THEN CLASS = S6
8	IF (b1, se2, se3, se4, se5, se6, se7, se8, se9, se10) THEN CLASS = S1
9	IF (se1, se2, b3, se4, se5, se6, se7, se8, se9, se10) THEN CLASS = S4
10	IF (se1, se2, b3, se4, se5, se6, se7, se8, se9, se10) THEN CLASS = S3
11	IF (se1, se2, se3, b4, se5, se6, se7, se8, se9, se10) THEN CLASS = S4
12	IF (se1, b2, se3, se4, se5, se6, se7, se8, se9, se10) THEN CLASS = S2
13	IF (se1, se2, se3, se4, se5, se6, b7, se8, se9, se10) THEN CLASS = S7
14	IF (se1, se2, se3, se4, se5, se6, se7, se8, b9, se10) THEN CLASS = S9
15	IF (b1, se2, se3, se4, se5, se6, se7, b8, se9, b10) THEN CLASS = S1
16	IF (b1, b2, se3, se4, se5, se6, se7, se8, se9, b10) THEN CLASS = S2
17	IF (se1, b2, se3, se4, se5, se6, se7, se8, se9, se10) THEN CLASS = S3
18	IF (se1, se2, se3, b4, se5, se6, b7, se8, se9, se10) THEN CLASS = S4
19	IF (b1, se2, se3, se4, se5, se6, se7, se8, b9, se10) THEN CLASS = S9

The DT has been tested using test data in Table 8 and categorized test data in Table 9.

Table 8. Test Data Clickstreams and Class Labels

No	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	Class
1	0	0	0	0	3	0	0	0	0	7	S10
2	0	0	0	0	0	0	0	3	3	4	S10
3	0	0	0	0	0	0	0	0	4	3	S9
4	0	0	0	7	0	0	0	0	0	3	S4
5	0	3	0	0	0	0	0	0	0	7	S10
6	0	0	0	0	0	0	0	0	0	7	S10
7	0	0	0	4	0	0	0	0	3	0	S4
8	0	0	0	3	0	0	0	4	0	0	S8
9	0	0	0	0	4	0	0	0	0	0	S5
10	0	0	0	0	0	10	0	0	0	0	S6

The experimental results for true-positive (TP), false-negative (FN), precision, recall, accuracy and f-measure values for the test data are given Table 10.

As shown the Table 10, 9 of items of the recommended list are TP (%90) and only 1 of the items is FN (%10). And, precision, recall, accuracy and F-measure values suitable for real applications. In the future work, more users will be included and real time usage results will be obtain for a long period of

time.

Table 9. Categorized Test Data

No	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	Class
1	se1	se2	se3	se4	se5	se6	se7	se8	se9	b10	S10
2	se1	se2	se3	se4	se5	se6	se7	se8	se9	b10	S10
3	se1	se2	se3	se4	se5	se6	se7	se8	b9	se10	S9
4	se1	se2	se3	b4	se5	se6	se7	se8	se9	se10	S4
5	se1	se2	se3	se4	se5	se6	se7	se8	se9	b10	S10
6	se1	se2	se3	se4	se5	se6	se7	se8	se9	b10	S10
7	se1	se2	se3	se4	se5	se6	se7	se8	se9	se10	S4
8	se1	se2	se3	se4	se5	se6	se7	b8	se9	se10	S8
9	se1	se2	se3	se4	b5	se6	se7	se8	se9	se10	S5
10	se1	se2	se3	se4	se5	b6	se7	se8	se9	se10	S6

Table 10. Test Results of the Decision Tree

TP (items)	9
FN (items)	1
Precision (%)	100
Recall (%)	90
Accuracy (%)	90
F-Measure	0,94

4. Conclusion

In this study, a new recommendation system based on DT using implicit relevance feedback have been developed for movies. User behavior has been obtained using implicit relevance feedback from clickstreams. The DT constructed by Gini algorithm. The developed RS has been tested using test data obtained from 200 undergraduate and graduate students between 20 to 30 years old. The experimental results show that the proposed method is successful for effective and useful suggestions.

References

[1] Kim, J. K., Kim, H. K., Oh, H. Y., & Ryu, Y. U. (2010). A group recommendation system for online communities. *International Journal of Information Management, 30(3)*, 212–219.

[2] Ricci, F., Rokach, L., Shapira, B., Kantor, P. B. (2011). *Recommender Systems Handbook*. Springer, England.

[3] Zhang, H., & Yang, Y. (2011). An e-commerce personalized recommendation system based on customer feedback. *Proceedings of Management and Service science (MASS)* (pp. 1-3).

[4] Palanivel, K., & Sivakumar, R. (2011). A study on multi-criteria recommender system using implicit feedback and fuzzy linguistic approaches. *Proceedings of the Recent Trends in Information Technology (ICRTIT)*.

[5] Briguez, C. E., Maximiliano, C. D., Budán, C. A. D., Deagustini, A. G., Maguitman, M. C., & Guillermo, R. S. (2014). Argument-based mixed recommenders and their application to movie suggestion. *Expert Systems with Applications, 41(14)*, 6467-6482.

[6] Julashokri, M., Fathian, M., Gholamian, M. R., & Mehrbod, A. (2011). Improving recommender systems efficiency using time context and group preferences. *Advances in inf. Sciences and Service Sci., 3(4)*, 162-168.

[7] Pradel, B., Sean, S., Delporte, J., Guérif, S., Rouveirol, C., Usunier, N., Fogelman-Soulié, F., & Dufau-Joel, F. (2011). A case study in a recommender system based on purchase data. *Proceedings of the 17th ACM*

SIGKDD International Conference on Knowledge Discovery and Data Mining.

[8] Tyagi, S., & Bharadwaj, K. K. (2012). Enhanced new user recommendations based on quantitative association rule mining. *Procedia Computer Science, 10*, 102-109.

[9] Adibi, P., & Ladani, B. T. (2013). A collaborative filtering recommender system based on user's time pattern activity. *Information and Knowledge Technology (IKT),* 252-257.

[10] Melville, P., & Sindhwani, V. (2010). Recommender systems. *Encyclopedia of Machine Learning,* 829-838.

[11] Abdullah, N., Xu, Y., Geva, S., & Chen, J. (2010). Infrequent purchased product recommendation making based on user behavior and opinions in e-commerce sites. *Proceedings of Data Mining Workshops.*

[12] Kulkarnil, S., Sankpal, A. M., Mudholkar, R. R., & Kumari, K. (2013). Recommendation engine: Matching individual/group profiles for better shopping experience. *Advanced Computing Technologies.*

[13] Park, D. H., Kim, H. K., Choi, Y., & Kim, J. K. (2012). A literature review and classification of recommender systems research. *Expert Systems with Applications, 39(11)*, 10059–10072.

[14] Yang, X., Yang, G., Yong, L., & Harald, S. (2014). A survey of collaborative filtering based social recommender systems. *Computer Communications, 41*, 1-10.

[15] Choi, Y. K., & Kim, S. K. (2014). An auxiliary recommendation system for repetitively purchasing items in e-commerce. *Proceedings of Big Data and Smart Computing (BIGCOMP), 39(11)*, 10059–10072.

[16] Extrand, M. D., & Riedl, J. T. (2010). Collaborative filtering recommender systems. *Foundations and Trends in Human —Computer Interaction*.

[17] Lü, L., Medo, M., Yeung, C. H., Zhang, Y. C. , & Zhou, T. (2012). *RSs. Physics reports, 519(1)*, 1-49.

[18] Pu, P., & Chen, L. (2010). A user-centric evaluation framework of RSs. *Proceedings of ACM Recsys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI)*.

[19] Gershman, A., Amnon, M., Karl-Heinz, L., Lior, R., Alon, S., & Arnon, S. (2010). A decision tree based recommender system.

[20] Xiang, L., & Yang, Q. (2009). Time dependent models in collaborative filtering based recommender system. *Web Intelligence and Intelligent agent Technologies*, 450-457.

[21] Bobadilla, J., Ortega, F., Hernando, A., & GutiéRrez, A. (2013). Recommender systems survey. *Knowledge-Based Systems*, *46*, 109–132.

[22] Kotsiantis, S. B. (2013). Decision trees: a recent overview. *Artificial Intelligence Review*, *39(4)*, 261-283.

[23] Kass, G. V. (1980). An exploratory technique for investigating large quantities of categorical data. *Applied Statistics*, *29 (2)*, 119–127.

[24] Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, *1(81)*, 81-106.

[25] Classification and Regression Trees. (1984). *Wadsworth Statistics/Probability*. Belmont.

[26] Quinlan, J. R. (1996). Improved use of continuous attributes in C4.5. *Journal of Artificial Intelligence Res., 4*, 77-90.

[27] Loh, W. Y., & Shih, Y. S. (1997). Split selection methods for classification trees. *Statistica Sinica, 7*, 815-840.

Anıl Utku received the BS degree in computer engineering from Kocaeli University, Kocaeli, Turkey, in 2010, and the MSc degree from Graduate School of Natural and Applied Sciences at Gazi University, 2015. He is currently a research assistant at the Department of Computer Engineering at Gazi University. His research interests include artificial intelligence, data mining and Web mining.

Hacer Karacan received the BS degree in computer education from Middle East Technical University, Ankara, Turkey, in 2002, and the MSc, PhD degrees from Informatics Institute at Middle East Technical University, 2005 and in 2007, respectively. He is currently an assistant professor at the Department of Computer Engineering at Gazi University. His research interests are interactive systems design, software engineering, and database management systems.

Oktay Yıldız received the BS degree in electronics and computer education from Gazi University, Ankara, Turkey, in 1997, and the MSc, PhD degrees from Institute of Science and Technology at Gazi University, 2004 and in 2012, respectively. He is currently an instructor at the Department of Computer Engineering at Gazi University. His research interests include data mining, machine learning, and bioinformatics.

M. Ali Akcayol received the BS degree in electronics and computer education from Gazi University, Ankara, Turkey, in 1993, and the MSc, PhD degrees from Institute of Science and Technology at Gazi University, 1998 and in 2001, respectively. He is currently a professor at the Department of computer engineering at Gazi University. His research interests include artificial intelligence, mobile wireless networks, web mining.