Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2015) 23: 900 – 912

doi:10.3906/elk-1212-145

A mobile and web application-based recommendation system using color quantization and collaborative filtering

Fidan KAYA*, **Gürel YILDIZ**, **Adnan KAVAK**
Department of Computer Engineering, Faculty of Engineering, Kocaeli University, Kocaeli, Turkey

Received: 26.12.2012 • **Accepted/Published Online:** 18.06.2013 • **Printed:** 30.04.2015

Abstract: In this paper, a recommendation system based on a mobile and web application is proposed for indoor decoration. The main contribution of this work is to apply two-stage filtering using linear matching and collaborative filtering to make recommendations. In the mobile application part, the image of the medium captured by a mobile phone is analyzed using color quantization methods, and these color analysis results along with other user-defined parameters such as height, width, and type of the product are sent to the web server. In the web application part, a large data set is first filtered via linear matching in which the color content of the medium and user-defined parameters received from the mobile application are matched to those for the products stored in the database. We then apply second-stage filtering, namely collaborative filtering, on the reduced data set. Performance evaluations of various color quantization methods and collaborative filtering methods used in the system are made. Results show the feasibility of using scalar quantization as a color quantization method and the K-nearest neighbor in the collaborative filtering method for our recommendation system. Overall evaluation of the system shows that our recommendation system provides around 90% accuracy.

Key words: Recommendation system, collaborative filtering, mobile application, web application, filtering, color analysis

1. Introduction

Due to the continuous development of technology and an increase in the number of products and services, companies need to find ways for competition [1].

In recent years, mobile communication tools and mobile applications have been widely used as emerging technologies. Due to wide variety of services being offered, such as multichannel shopping, navigation, ticket purchasing, making reservations, making payments, and establishing mobile internet connections [2], the number of mobile phone users are exponentially increasing each year.

Recommendation systems have been developed for providing effective information to users. There are many recommendation systems based on web mining. These systems are divided into two main categories. These are collaborative filtering and content-based filtering [3]. Collaborative filtering method is the most commonly used of these methods, which takes into account the similarities between users or the selections of other users [4]. Collaborative filtering techniques can also make recommendations by using specific content or by bringing together users with similar interests [5–8].

In this study, we propose a recommendation system for indoor decoration that is based on applying

*Correspondence: fidan.kaya@kocaeli.edu.tr

image processing techniques at both the mobile side and the web server, and collaborative filtering techniques at the web server side. In the collaborative filtering approach, the underlying logic is that if a user has agreed with his or her neighbors in the past, he/she will agree with the neighbors in the future [9]. We also evaluate the performance of the techniques implemented in our system. Specifically, the performance of scalar quantization for image processing and the K-nearest neighbor for collaborative filtering have been compared to their alternative methods.

The remainder of this paper is as follows: system architecture is introduced in Section 3. Section 4 explains the details of our recommendation system with a focus on the color quantization and collaborative filtering techniques used. Section 5 presents results from the performance evaluation of the techniques implemented. Finally, concluding remarks are given in Section 6.

2. Related works

The collaborative filtering method was first used in the Tapestry Project that was developed for the purpose of filtering e-mails. The purpose of the developed system is decomposing the e-mails that were labeled by users based on specific criteria [10,11]. User-based collaborative filtering was first used in the GroupLens research project [12]. This system makes personal recommendations using Usenet News. The Pearson correlation coefficient was used for the first time by GroupLens researchers [13]. Herlocker et al. applied a collaborative filtering method on different data sets and found that the Spearman similarity correlation gave better results than the Pearson correlation coefficient [14]. Goldberg et al. developed a filtering algorithm with reduced computation time, which used the Pearson correlation coefficient. [15]. Color quantization algorithms have been studied to determine the limited number of colors on the image [16–19]. Along with the proposed recommendation architecture, our study in this paper also investigates the practical use of the scalar quantization technique in the color quantization and K-nearest neighbor in the collaborative filtering stages of the system.

Collaborative filtering-based and content filtering-based recommendation systems are popularly used in web systems such as Amazon [20], Last Fm, Digg [21], Netflix [22], EBay, iTunes, and StumbleUpon [23,24]. However, the proposed recommendation system in this work differs from these systems in that our system applies two-stage filtering on the data set. Results show the feasibility of this approach with 90% accuracy, which is applied for selecting indoor decoration products.

3. System architecture

Figure 1 depicts the architecture of our recommendation system. User1 represents a user who is accessing the system with his mobile phone. User2 represents a user who is accessing the system via her personal computer. Both users aim to obtain recommendations. The mobile user captures the picture of the environment and analyzes it using color quantization algorithms. The same quantization method can be applied to previously stored pictures of the environment. The results of color content analysis along with the user-defined parameters are then sent to the server. These results are matched to those of the products stored at the web server by a linear matching algorithm. The collaborative filtering algorithm is then applied to the results of linear matching to produce recommendations.

The recommendation system consists of two main modules. The first module includes mobile application and image processing techniques implemented at the mobile device. The second module includes the collaborative filtering techniques and web application part implemented at the web server. Both modules have been developed individually and integrated into the system.

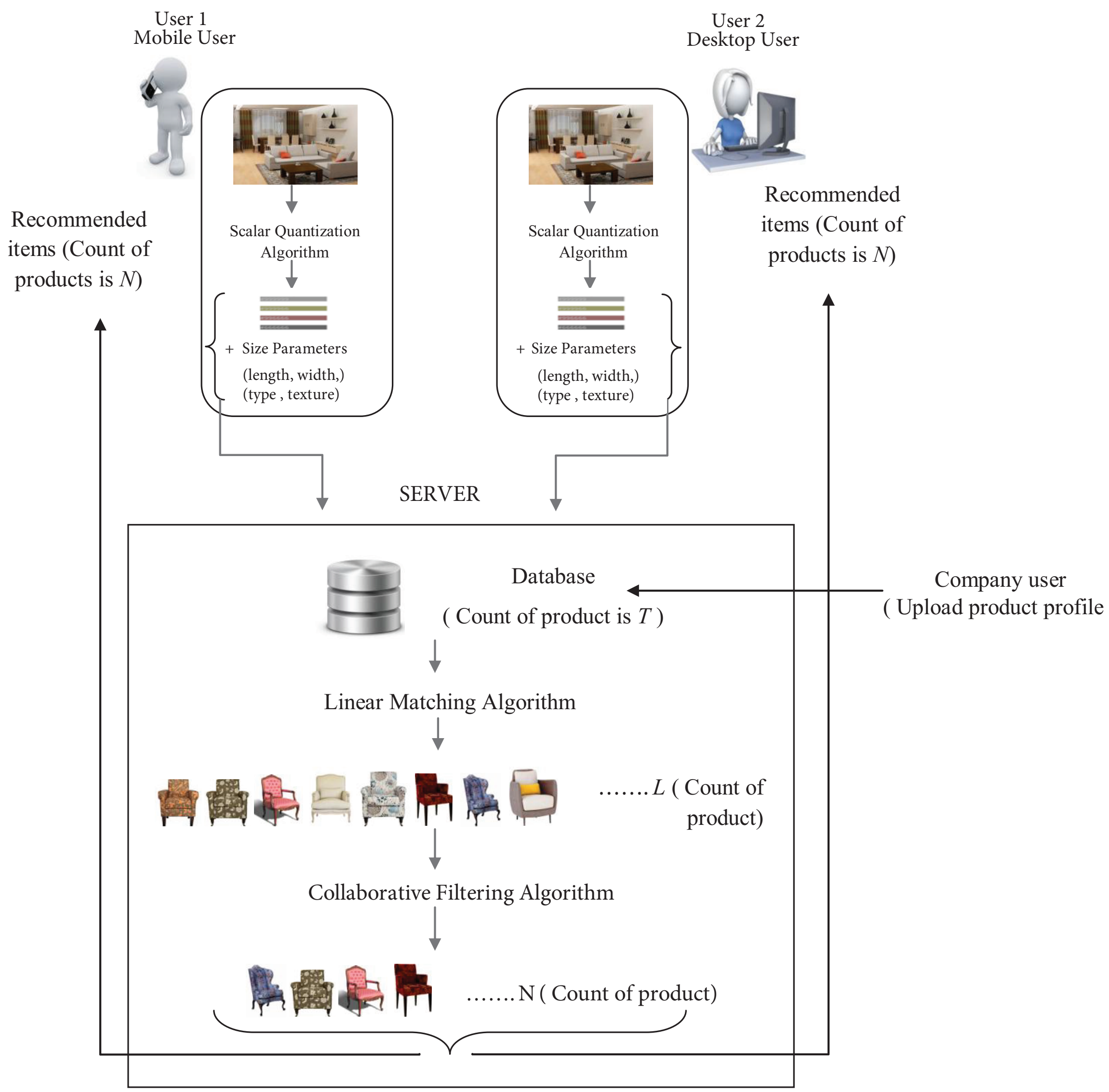

Figure 1. Overview of the recommendation system.

The mobile application module is responsible for the analysis of the captured image and forwarding this result along with other specifications (dimension, product category, etc.) determined by the user to the web part. In the mobile application, the following steps are applied sequentially to the captured image [20,21]: resizing the image, obtaining the RGB values, applying the quantization algorithm, converting the RGB color codes to hexadecimal code, and finally ordering the color codes in descending order. After these steps, color codes are classified as minimum, maximum, average, and opposite to provide the user options to choose among these color codes.

The web application module has two functionalities, namely management and recommendation functions. As the management function the system handles tasks related to users of the system, i.e. customers and companies. Companies can upload their profiles and product information. Different companies can add their products at the same time and later they can organize these products. Users of the system can perform various tasks using the product information uploaded by companies and other users' information (scores voted for a specific product, their recommendations, etc.). The recommendation part of the web application as shown in Figure 1 performs linear matching and collaborative filtering. This part runs the same image quantization algorithm as the mobile application [25].

4. The recommendation system

The flowchart of the proposed recommendation system is given in Figure 2. The system first applies a linear matching algorithm to the large data set (T products) to obtain a smaller data set (L products). The estimated vote values that belong to this smaller amount of data (L products) are then estimated. Finally, N different products that have the highest estimated vote values are presented to the user as a recommendation [26].

4.1. Linear matching stage

In the linear matching stage, the largest similarity (minimum s_j) of the candidate product with the analysis results obtained for the indoor medium to be decorated is determined by following algorithm.

for j = 1: T;
    if $D_{j,w} < D_t$
        $s_j = |E_{j,w} - E_t|$
    end
end
$s^{LM} = min\{s_j\}$

Here, E_t represents the analysis results of color quantization. D_t represents the size of the indoor medium to be decorated. $D_{j,w}$ is the dimension of the product stored in the web server database and $E_{j,w}$ is the color content of this product. Basically, the linear matching algorithm compares the results of web-based analysis to those based on the mobile device and results in the maximum similarity. The reason for the using the linear matching algorithm is the prefiltering of the products in the database before applying the collaborative filtering algorithm. Therefore, the collaborative filtering algorithm will be applied to a smaller size data set and recommendations will be produced in a shorter time.

4.1.1. Color quantization

In the proposed recommendation system, image resizing and color quantization methods are applied to the picture of the environment both on the mobile and the web platform. Color quantization methods are generally divided into two main categories: scalar quantization and vector quantization.

In the scalar quantization algorithm, RGB color values that belong to each pixel of the image are divided by a constant value. Rounding is applied on the obtained results. During this process, color values are assigned to the nearest scalar multiples of the value so that the number of colors will be reduced. For scalar value 51, RGB color values varying between 0 and 255 are converted into values of 0, 51, 102, 153, 204, and 255. Let C represent the color codes belonging to the original image, and this image has m different colors and C' represents quantized color codes with n different values obtained after the scalar quantization algorithm [27]. Assume that the original image has m different color codes denoted by $C = \{ C_1, C_2 , \ldots, C_m \}$. The purpose of color quantization is to reduce this to n different color codes given by $C' = \{ C'_1, C'_2 , \ldots, C'_n \}$, where each Ci' is related to Ci by:

$$C_i' \leftarrow \left\lfloor C_i . \frac{n}{m} \right\rfloor . \tag{1}$$

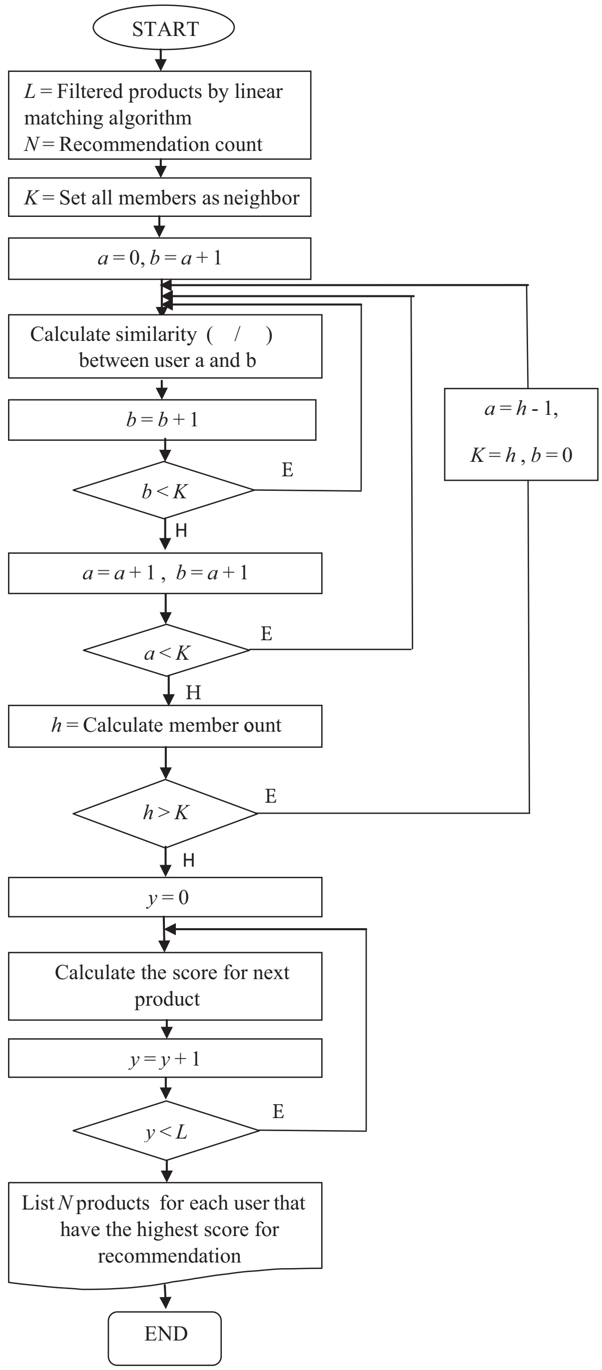

Figure 2. Flowchart of the proposed recommendation system.

Vector quantization is different from scalar quantization in that rather than the pixel's color value it acts as a vector formed by the color values. Therefore, the number of image disruptions that occur as a result of the number of color reductions is small in color quantization. Median cut and octree algorithms are commonly used as vector quantization algorithms.

In the median cut algorithm, the RGB color cube's small cell, which has all colors in the image, is selected. Along the long axis of the cell, the colors are sorted from smallest to largest. After sorting the colors, the cell is divided into two parts along the long axis of the cell. The above process is continued until the color space is reduced to the predefined number [27].

In the octree algorithm, the colors are sorted using a tree structure. Eight is used for the branching factor. In this method, a tree structure that has at most K nodes is to be established and each node has a different color. While adding new colors to tree branches, an average value is obtained by using the most similar color [17].

4.1.2. Collaborative filtering stage

Collaborative filtering algorithms are not concerned with the contents of objects while producing recommendations. These systems are referred to as content-free systems. Establishing the correct connections between users is important for these systems. Previous experiences and previous preferences are used and neighbors are determined. For example, users who love an object can be determined as a neighbor or users who do not like an object can be determined as a neighbor. To create a group, feedback taken from users can be a numerical vote, “thumbs up”, “thumbs down”, “like”, “dislike”, a user’s favorite product, a marked product, etc. For this purpose, the value of votes that are given to products by users are used to calculate the similarities between neighbors. The collaborative filtering algorithm mainly uses the score values voted for products by users. It calculates the similarities of subsets of products by using these scores. The Pearson similarity measure, which is the correlation coefficient between two different users (a and b), is given by:

$$s_{a,b}^{P} = \frac{\sum_{i\in I}(r_{a,i}-\bar{r}_a)(r_{b,i}-\bar{r}_b)}{\sum_{i\in I}(r_{a,i}-\bar{r}_a)^2(r_{b,i}-\bar{r}_b)^2}, \tag{2}$$

where I represents the number of products voted for by users a and b, $r_{a,i}$ and $r_{b,i}$ represent the vote given to product i respectively by user a and b, and $\bar{r_a}$ and $\bar{r_b}$ represent the average of votes given by user a and b, respectively.

Pearson similarity is the most common similarity measure used in collaborative filtering methods. It was first used in the GroupLens project. In [10], similarity values changed between –1 and +1. If two users' similarity is close to +1, these users' opinions are considered to be similar. If the similarity value is 0, these two users or two objects will not be similar. A value of –1 indicates that users have opposite characteristics. The cosine similarity (or vector similarity) measure is another way to calculate similarities between users, which is defined as:

$$s_{a,b}^{V} = \frac{\mathbf{R}_a^T\mathbf{R}_b}{\|\mathbf{R}_a\|\,\|\mathbf{R}_b\|}, \tag{3}$$

where $\mathbf{R_a} = [R_{a,1}R_{a,2}\ldots R_{a,I_a}]^T$ and $\mathbf{R_b} = [R_{b,1}R_{b,2}\ldots R_{b,I_b}]^T$, and $\|.\|$ denotes the Euclidian norm. The estimated vote that will be given to product i by user a is given by:

$$P_{a,i} = \bar{a} + \frac{\sum\limits_{(b\in K)\bigwedge(b\_rates\_i)} \left(b_i - \bar{b}\right) * s_{a,b}}{\sum\limits_{(b\in K)\bigwedge(b\_rates\_i)} |s_{a,b}|}, \tag{4}$$

where $\bar{a}$ represents the average of votes of user a, $\bar{b}$ is the average of votes of user b, and b_i is a vote given to product i by user b. $S_{a,b}$ is the similarity measure calculated by either method given above.

The key parameter K in Eq. (4) is the number of neighbors of user a. In order to calculate K, K-nearest neighbor (K-NN) or correlation threshold methods can be used.

The users whose similarities are larger than the threshold value will be chosen as neighbors in the correlation threshold method. However, if there is a high threshold value, a sufficient number of users may not be selected, resulting in inaccurate results.

The K-NN method is based on the selection of the largest similar K number of users. Studies performed by Herlocker et al. [14] showed that range of 20 to 50 will be sufficient for neighbor selection.

5. Results

In this section we present the performance evaluation of candidate techniques that can be used in our recommendation system and provide overall recommendation system performance.

5.1. Comparison of color quantization methods

First of all we should choose the appropriate image analysis method that captures the color content of the image. For this purpose, three different methods have been evaluated. These are scalar quantization, the median cut algorithm, and the octree algorithm.

Figure 3. A sample image uploaded by a user.

Table 1 provides information about the image in Figure 3. In Figures 4–6, scalar quantization results of the image shown in Figure 3 are given. The algorithm applies different scalar values.

Table 1. Image information of Figure 3.

Height	Width	Total number of colors	Number of single colors
172	239	41,108	25,354

Table 1 provides the height, weight, total number of colors, and number of single colors for the image. The total number of colors means the number of colors in the total pixels in the image. The number of single colors means the number of different colors in the total number of pixel colors.

The pixels' R, G, and B values are divided into 51 and rounded to an integer value. As a result of this process, the 8-bit R, G, and B values are reduced to 3-bit. There are $2^8 = 256$ states for R, G, and B, but at the end of the process the states will be reduced to 2^3 by dividing to $2^6 > 51 > 25$. Meanwhile, 4, 8, and 16 granularity values are selected for the pixel scanning process. Granularity is the value of jump at the pixels selected for the color quantization algorithm. For a granularity value of 4, more pixels will be processed compared to a granularity value of 8. After the quantization process, a histogram is created by counting the new pixel color values of the image.

Figure 4. Image analysis according to the scalar value of 51.

Figure 5. Image analysis according to the scalar value of 17.

Figure 6. Image analysis according to the scalar value of 5.

The histogram sorts the colors according to the color intensity. As shown in Figures 4–6, colors are ordered from left to right and ordered from most intense to less intense. One of the purposes of the analysis is to obtain intense colors more quickly by reducing the analysis time. For various quantization methods, mean squared error (MSE), execution time, and memory usage analysis results are summarized in Table 2.

Table 2. Comparison of quantization algorithms.

	MSE	Time (ms)	Use of memory (MB)
Scalar quantization	33,515.14	2625	1607
Octree quantization	6676.47	11,347	21,120
Median cut quantization	11,825.23	5786	30,118

According to test results, we need more processing time and more memory for the best quantization result on the mobile phone. However, using more memory and time is a failure in terms of performance for the quantization algorithm developed for mobile phones. Although the octree and median cut algorithms gives good results, in terms of color content, they require more memory and run-time. Obtaining intense colors is more important than image quality for the developed mobile application and so using the scalar quantization algorithm is more efficient in terms of time and memory usage.

In Table 3, the scalar quantization algorithm's MSE, PSNR, and time values are shown for different scalar values of 5, 17, and 51. In addition, 4, 8, and 16 granularity values are also taken into account. According to Table 3, if we compare the granularity values, the smaller scalar values correspond to larger PSNR values for the same granularity value. When the scalar value is fixed, choosing a large granularity value gives larger PSNR values. However, the processing time is reduced with increasing granularity values. Therefore, while applying the scalar quantization algorithm, we used a scalar value of 51 and a granularity value of 16 in our recommendation system.

Table 3. Scalar quantization analysis results.

Scalar value	Granularity value	MSE	PSNR (dB)	Time (s)
	4	0.189	55.353	0.010217
5	8	0.050	61.127	0.000250
	16	0.013	66.718	0.000067
	4	2.251	44.606	0.000952
17	8	0.568	50.580	0.000250
	16	0.138	56.728	0.000067
	4	14.515	36.512	0.000952
51	8	3.673	42.480	0.000249
	16	0.926	48.462	0.000067

5.2. Comparison of filtering techniques

We have used MovieLens data sets in order to evaluate the performance of the collaborative filtering methods. The MovieLens data sets were created for research in Computer Science and Engineering Department of the University of Minnesota [28]. In this study, we have used the 100K version of MovieLens. The data set is divided into two parts. Eighty percent of the data is training data and twenty percent of the data is test data. Five different data sets and training sets are created from this 100K data.

We have applied the Pearson correlation coefficient and cosine similarity (vector similarity) measures given in Eqs. (1) and (2) to select the most appropriate method for calculating similarities. In order to compare the performances of similarity measures, response times are calculated, which are given in Figure 7. The response time of the system is an important factor since our system has to be able to handle simultaneous recommendations to the users on the web server and mobile sides. As seen in Figure 7, the Pearson correlation coefficient has a lower response time than the cosine similarity measure [29].

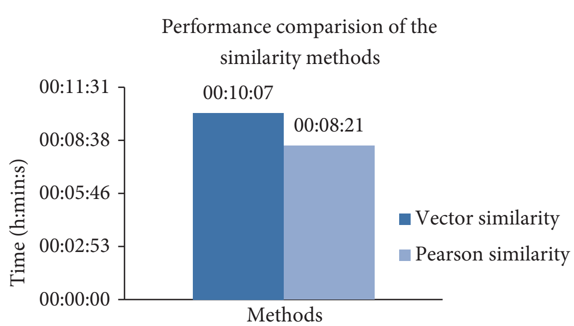

Figure 7. Performance results for similarity methods.

After choosing a suitable similarity measure, it should be decided which method will be used in the recommendation part. At this stage, two different methods have been considered, namely the correlation threshold method and the K-NN method. First the K-NN method was applied to five different data sets. The root mean squared error (RMSE) versus number of neighbors obtained using K-NN for two different similarity measures is shown in Figure 8 and results are given in Tables 4 and 5. According to the tables the minimum number of neighbors is obtained between 50 and 75 [29]. Although there is not a significant difference between the RMSE values, number of neighbors K is determined as 50. This is because the developed system is an online application and the system should make recommendations quickly. If we choose 75 neighbors it will bring processing overhead.

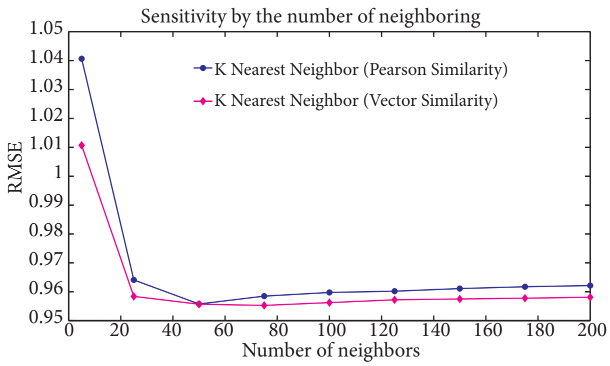

Figure 8. RMSE values for K-NN method.

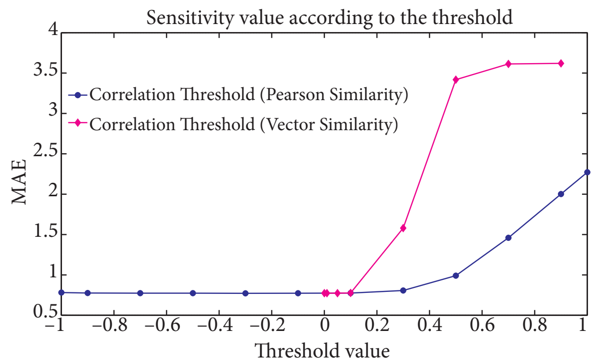

Figure 9. RMSE values for correlation threshold method.

Table 4. Number of neighbors with respect to RMSE values (Pearson correlation measure is used).

Number of neighbors (K)	RMSE
5	1.040622
25	0.964094
50	**0.955769**
75	0.958492
100	0.959755
125	0.960186
150	0.961103
175	0.961738
200	0.962139

Table 5. Number of neighbors with respect to RMSE values (cosine similarity measure is used).

Number of neighbors (K)	RMSE
5	1.010673
25	0.958405
50	**0.955693**
75	**0.955278**
100	0.956257
125	0.957220
150	0.957500
175	0.957768
200	0.958112

Table 6. Threshold value with respect to RMSE values (Pearson correlation coefficient measure is used).

Threshold value	RMSE
–1	0.964265
–0.9	0.957661
–0.7	0.955845
–0.5	0.956968
–0.3	**0.957183**
–0.1	0.960299
0.1	0.963352
0.3	1.013408
0.5	1.270321
0.7	1.813726
0.9	2.367137
1	2.633505

Table 7. Threshold value with respect to RMSE values (cosine similarity measure is used).

Threshold value	RMSE
0	0.958813
0.01	0.958807
0.05	**0.958802**
0.1	0.964412
0.3	1.847446
0.5	3.584878
0.7	3.756436
0.9	3.763345

In Figure 9, the best performance intervals obtained by the correlation threshold method are shown. The Correlation threshold method takes into account only some of the users whose similarities are larger than the threshold value. However, a sufficient number of the user's neighbors may not be available with the selected threshold value. In both the K-NN and correlation threshold methods, the cosine similarity measure is observed to have smaller RMSE compared to the Pearson similarity measure as seen in Tables 6 and 7. However, due to different date clusters, cosine similarity results in different RMSE values for different data clusters. Due to smaller response time and consistency in RMSE results, the Pearson correlation coefficient is chosen in our study. The response time of the K-NN and correlation threshold methods using the Pearson correlation coefficient is shown in Figure 10 [29].

As seen in Figure 10, there is a difference of 20 s between response times and this is not a significant difference. We therefore compared both methods in terms of uncalculated scores of the products. The correlation threshold method depends on the threshold value and uncalculated scores rose as it increased.

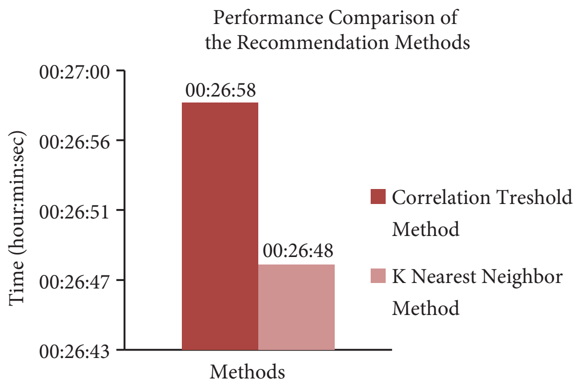

Figure 10. Timing comparison for recommendation methods.

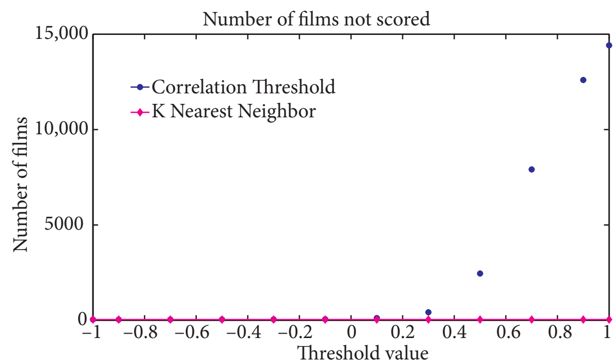

Figure 11. Score values that are not calculated.

K-NN is insensitive to the threshold value. As seen in Figure 11, the number of users with uncalculated score values obtained by the K-NN method is always 42 regardless of the threshold value. In the correlation threshold method, although there are enough neighbors of recommended users, if the similarity among neighbors remains under the threshold value, scores cannot be calculated [29].

5.3. Overall system performance

In information retrieval, a ranked list of documents is often evaluated using precision, recall, or the F1 measure. These measures have also been used by a number of researchers to evaluate recommender systems [30–32]. Precision is the ratio of relevant items recommended to the total number of items recommended. Recall is the ratio of relevant items recommended to the total number of relevant items that exist. The two measures are inversely related and are dependent on the length of the recommendation list [10]. However, if the list size increases, precision and recall values will conflict, and so the F1 measure arose. The F1 measure uses both criteria equally. Calculation of these criteria are shown in Table 8.

Table 8. Measure of evaluating recommendation [10].

Measure	Formula
Precision	$\dfrac{\lvert\{\text{ relevant documents }\}\bigcap\{\text{ retrieved documents }\}\rvert}{\lvert\{\text{ retrieved documents }\}\rvert}$
Recall	$\dfrac{\lvert\{relevantdocuments\}\bigcap\{retrieveddocuments\}\rvert}{\lvert\{\text{ relevant documents }\}\rvert}$
F1	$2*\dfrac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$

Performance of the proposed recommendation system is evaluated using the F1 measure. Accuracy of the recommendation is obtained as 90.148%. Calculations were made according to the K-NN and Pearson similarity methods. We can also evaluate system performance with different methods such as response time, which takes into account the elapsed time for calculating the recommendation using the training set. Response time of our system is approximately 35 min and this result shows that the system makes a recommendation to a user in approximately 0.06 s. If we consider that our application is web- and mobile-based, the system responds to users very quickly.

6. Conclusions

In this paper, a mobile and web application-based recommendation system has been proposed, and its implementation has been given for indoor decoration. We have presented the methods used for the mobile part and the web part of the system. The performance of the K-NN and correlation threshold methods as second-stage filtering (collaborative filtering) has been evaluated for the feasibility of implementing them into our web application part.

Results have shown the feasibility of using the K-NN method with the Pearson correlation coefficient. We have also compared the image quantization algorithms, and the scalar quantization algorithm was found to be most suitable in terms of implementation time and memory usage for a mobile device in our system.

In a further study, the system can be improved by incorporating a content-based filtering method in the recommendation engine. The main features and labels of the products can be used by this method at the expense of complexity in the system. Users can also be grouped taking into account various features in the product selection, and later while calculating the similarities, the users only in the same group can be taken into account.

References

[1] B. Karagöz, B. Çağlar, "Perakende sektöründe mobil pazarlama uygulamalarına karşı tüketici davranışlarının incelenmesi", AJIT-E, Vol. 2, pp. 1–1, 2011 (in Turkish).

[2] S. Barutçu, M. Öztürk Göl, "Mobil reklamlar ve mobil reklam araçlarına yönelik tutumlar", Journal of KMU İİBF, Vol. 17, pp. 24–41, 2009 (in Turkish).

[3] M. J. Pazzani, "A framework for collaborative, content based and demographic filtering", Artificial Intelligence Review, Vol. 13, pp. 393–408, 1999.

[4] J.S. Breese, D. Heckerman, C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering", Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, pp. 43–52, 1998.

[5] C. S. Firan, W. Nejdl, R. Paiu, "The benefit of using tag based profiles", Web Conference LA-WEB 2007, pp. 32–41, 2007.

[6] X. Yu, S. Sun, "Research on personalized recommendation system based on web mining", International Conference on E-Business and E-Government, pp. 346–349, 2010.

[7] C. Hang, Z. Meifang, "Improve tagging recommender system based on tags semantic similarity", 3rd International Conference on Communication Software and Network, pp. 94–98, 2011.

[8] R. Lian, "The construction of personalized web page recommendation system in e-commerce", International Conference on Computer Science and Service System, pp. 2687–2690, 2011.

[9] B.N. Miller, Toward a Personal Recommender System, PhD, University of Minnesota, Minneapolis, MN, USA, 2003.

[10] R. Rafter, Evaluation and Conversation in Collaborative Filtering, PhD, University College Dublin, Dublin, Ireland, 2010.

[11] D. Goldberg, D.M. Nichols, B. Oki, D. Terry, “Using collaborative filtering to weave an information tapestry”, Communications of the ACM, Vol. 35, pp. 61–70, 1992.

[12] P. Resnick, N. Lacovou, M. Suchak, P. Bergstrom, J. Riedl, “GroupLens: an open architecture for collaborative filtering of netnews”, ACM Conference on Computer-Supported Cooperative Work, pp. 175–186, 1994.

[13] U. Shardanand, P. Maes, “Social information filtering: algorithms for automating ‘word of mouth”’, ACM SIGCHI Conference on Human Factors in Computing Systems, pp. 210–217, 1995.

[14] L.J. Herlocker, J.A. Konstan, A. Borchers, J. Riedl, “An algorithmic framework for performing collaborative filtering”, 22nd Annual International ACM SIGIR Conference, pp. 230–237, 1999.

[15] K. Goldberg, T. Roeder, D. Grupta, C. Perkins, “Eigenstate: a constant time collaborative filtering algorithm”, Information Retrieval, Vol. 4, pp. 133–151, 2001.

[16] M.T. Orchard, C.A. Bouman, Color Quantization of Images, IEEE Transactions on Signal Processing, Vol. 39, pp. 2677–2690, 1991.

[17] D.S. Bloomberg, “Color quantization using octrees”, http://www.leptonica.org/papers/colorquant.pdf, 2008.

[18] G. Baolong, F. Xiang, “A modified octree color quantization algorithm”, First International Conference on Communications and Networking, pp. 1–3, 2006.

[19] W. Chen, W. Ding, “An improved median cut algorithm of color image quantization”, International Conference on Computer Science and Software Engineering, pp. 943–946, 2008.

[20] G. Linden, B. Smith, J. York, “Amazon.com recommendations: item-to-item collaborative filtering”, IEEE Internet Computing, Vol. 7, pp. 76–80, 2003.

[21] L. McGinty, B. Smyth, “Comparison-based recommendation”, ECCBR 2002, pp. 575–589, 2002.

[22] R.M. Bell, Y. Koren, C. Volinsky, The BellKor Solution to the Netflix Prize, http://www2.research.att.com/∼volinsky/netflix/ProgressPrize2007BellKorSolution.pdf, 2008.

[23] R. Rafter, B. Smyth, “Towards a domain analysis methodology for collaborative filtering”, Proceedings of ECIR 01, pp. 1–14, 2001.

[24] R. Rafter, B. Smyth, Item Selection Strategies for Collaborative Filtering, New York, Morgan Kaufmann Publishers, 2003.

[25] G.Yildiz, F. Kaya, A. Kavak, “A mobile application and web recommendation system for assisting indoor decoration”, Innovations in Intelligent Systems and Applications (INISTA), pp. 1–5, 2012.

[26] F. Kaya, İşbirlikçi Filtrelemeye Dayalı Web Tabanlı Öneri Sistemi Geliştirilmesi, MSc, Kocaeli University, Kocaeli, Turkey, 2012 (in Turkish).

[27] G. Yıldız, Renk Analizi Tabanlı Öneri Sistemleri için Mobil Uygulama Geliştirme, MSc, Kocaeli University, Kocaeli, Turkey, 2012 (in Turkish).

[28] MovieLens, Department of Computer Science and Engineering at the University of Minnesota, http://www.grouplens.org/node/73, 2012.

[29] F. Kaya, G. Yıldız, A. Kavak, “Evaluation of collaborative filtering techniques for mobile and web application based recommendation”, 9th International Conference on Electronics, Computer and Computation, pp. 314–317, 2012.

[30] B.M. Sarwar, G. Karypis, J.A. Konstan, J. Riedl, “Analysis of recommendation algorithms for e-commerce”, 2nd ACM Conference on Electronic Commerce, pp. 158–167, 2000.

[31] D. Billsus, M. Pazzani, “Learning collaborative information filters”, Machine Learning: Proceedings of the 15th International Conference, 1998.

[32] B.M. Sarwar, G. Karypis, J.A. Konstan, J. Riedl, “Application of dimensionality reduction in recommender system - a case study”, ACM WebKDD 2000 Workshop, 2000.