picSEEK - Collaborative Filtering for Context-Based Image Recommendation

A.R Widisinghe, R.A.D.R Ranasinghe, H.K.C Kulathilaka, R.I Kaluarachchi and K. Wimalawarna
Department of Computer Science and Engineering, University of Moratuwa, Sri Lanka.

Abstract— **As the World Wide Web becomes a large source of images, the image recommendation system has got a great demand. There are several image recommendation systems for both commercial and academic areas, which deal with the user preference as fixed. However, since the Images preferred by a user may change depending on the contexts, the conventional systems have inherent problems. This paper proposes a context-aware image recommendation service (picSEEK) that exploits the collaborative-filtering to recommend appropriate images with respect to the context. We have analyzed the recommendation process and performed a subjective test to show the usefulness of the proposed system.**

Keywords- context-awareness, image recommendation system, collaborative-filtering, image retrieval

I. Introduction

Information recommendation has become an important research area since the first papers on collaborative filtering published in the 1990s [1]. Extensive work has been done in both industry and academia on developing new approaches on recommendation systems over the last decades [2].There has been shown a roughly exponential growth in interest in image retrieval and closely related topics for last five years [3].For example, in a marketing domain, visual documents have been recognized as efficient means in advertisements since they can convey meanings that cannot be expressed using words [4].

There are two basic approaches to image retrieval: 1) content-based image retrieval (CBIR) and 2) metadata based image retrieval [3].

A. Content-based Image Retrieval(CBIR)

In CBIR the images are retrieved describing their content. At the lowest level, features such as color, texture, shape, and spatial location are used. At a higher conceptual level, images with an object of a given type or a given individual are searched. In retrieval, a user expresses the information need by formulating a search query generally in the form of image examples or textual descriptions. An example of the CBIR approach on the web is the PicSOM system. Given a set of reference images, PicSOM is able to retrieve another set of images which are similar to the given one [5].

B. Metadata-based Image Retrieval

In the metadata-based approach image retrieval is based on textual descriptions

about pictures. Typical image search engines such as Google or Yahoo image search which are based primarily on surrounding metadata such as file names and HTML text are few examples. In practice, this approach is usually employed in image retrieval due to the great challenges of the CBIR approach, when dealing with conceptually higher levels of content.

But in both approaches there exists a semantic gap-the lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data has for a user in a given situation [3]. Semantic web ontology techniques and meta data languages has contributed to this problem by providing means for annotating image meta data according to the ontology (such as Events, Persons, and Places) which provides the terminology and concepts by which metadata of the images is expressed. The ontology together with the image metadata forms a knowledge base, which can facilitate new semantic information retrieval services [6].

Main goal of most of the above approaches is to respond to the user's search query. But there is another kind of interests i.e., long term or permanent such as desires, tastes and preferences of each user [4]. And it is also important to incorporate the contextual information into the recommendation process [7]. This paper proposes a context-aware image recommendation service which exploits collaborative filtering techniques for context aware recommendation that is capable of using generally in different kind of context domains.

II. Related Work

A. Collaborative Filtering

Collaborative Filtering (CF) is a technology that has emerged in e-Commerce applications to produce personalized recommendations for users. CF works by combining the opinions of people who have expressed inclinations similar to yours in the past to make a prediction on what may be of interest to you now [8]. One well known example of a CF system is Amazon.com. An implementation of such CF system could build user profiles from feedback on items made overtime. It then finds likeminded users by "weighing" the active user against every other user with respect to the similarity in their ratings given to the same items. All the neighboring likeminded users' ratings are then combined into a prediction by computing a weighted average of the ratings. . It is assumed that the predicted vote of the active user *a* for item *i* ($P_{a,i}$) is a weighted sum of the votes of the other users.

$$P_{a,i} = \overline{v_a} + k \sum_{j=1}^{n} w(a,j).(v_{j,i} - \overline{v_i})$$

Where *n* is the number of users in the CF database with non zero weights. The weights *w(a,j)* reflect distance, correlation or the similarity between each user *i* and the active user. *k* is the normalizing factor such that absolute values of weights sum to unity[8].

In general formulation of statistical collaborative filtering Pearson correlation coefficient was defined as the basis for the weights [10]. The correlation between user *a* and *j* is:

$$w(a,j) = \frac{\sum_i (v_{a,i} - \overline{v_a}).(v_{j,i} - \overline{v_j})}{\sqrt{\sum_i (v_{a,i} - \overline{v_a})^2 . (v_{j,i} - \overline{v_j})^2}}$$

Where the the summations over are over the items for which both user *a* and *j* have recorded votes.

Recommender systems rely on different types of input. Most convenient is the high quality explicit feedback, which includes explicit input by users regarding their interest in products. For example, Netflix collects star ratings for movies and TiVo users indicate their preferences for TV shows by hitting thumbs up/down buttons. However, explicit feedback is not always available. Thus, recommenders can infer user preferences from the more abundant implicit feedback, which indirectly reflect opinion through observing user behavior [9]. Types of implicit feedback include purchase history, browsing history, search patterns, or even mouse movements. In a context aware image search implicit feedback includes- what is the selected image for a particular search request in the context where user exists.

Thus far, CF has mostly been applied to applications for which the context is static; hence the recommendations do not change. In the dynamic environment of like in ubiquitous computing, users' decisions can be influenced by many things in their surrounding context. For instance, when people travel, their preferred activities may largely depend on the weather. Existing CF systems could not model this complexity of context. They are as likely to recommend mountain routes for a person who likes hiking whether it rains or shines. And in image recommendation (say in a blogging tool), for a same search query two users may prefer two different images depending on the article he is writing. i.e. for a search query 'student' a user who is writing an article about medicine may prefer a medical student while a user who is writing an article about education might prefer a school boy. This shows that interest of a user differs according to the context that user exists.

We propose a context aware CF system that can predict a user's preference for an image in any context environment by leveraging past experiences (implicit feedbacks of selected images by users) of likeminded users in similar context. In the sections to follow, we will propose algorithmic extensions to the CF process that incorporate pervasive context in making predictions and provide an overview of picSEEK.

III. Context-based Image recommendation

picSEEK is a Context-Based Image Recommender service which also provides most innovative image search. It is basically a web service which can be used by any kind of web based application, where there are user accounts and users searching images for different purposes, to perform image search.

For the discussion below consider the following scenario as the backdrop to the discussion. Scenario: A blogging tool (ABC) has users that write articles on different domains. And the user search web in order to add pictures to their articles. Hence the ABC intends to use our service to offer their users better images which have higher relevance to the context that user exist.

In the Scenario described above. ABC goes the picSEEK website and configures the services they required and context model they will provide when using our web service and register with our web service. Figure 1 shows how picSEEK provides context-based image search and other image recommendation services to its clients.

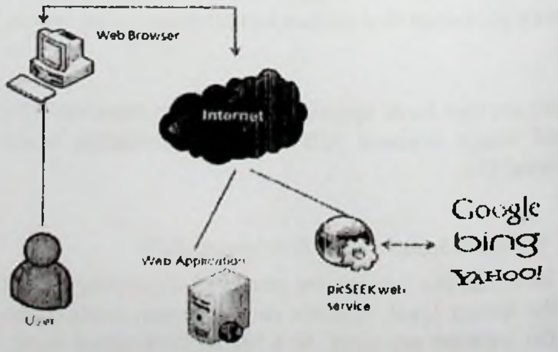

Figure 1. Overview of the system

When a user of ABC wants to add images to their article WordPress can send picSEEK a request including the context details and search keyword. For this it should be intelligent enough to analyze the context where user exists by referring to the content of the article. After getting a request from client picSEEK will recommend top most images which match the search query and interests of user and also which have the highest relevance to the context. This is carried out by running our high performance context based algorithms on our knowledge base which learns from user feed-backs. And if the request is completely new to the knowledge base picSEEK will query any common search engine which is preferred by the client and show the results directly.

Service offers implementation of two main recommendation algorithms which has been developed by modifying and adding context awareness to existing collaborative-filtering algorithms. And they will be explained in the following sub sections.

A. *Context-Aware CF for Implicit Feedback*

In Classical Collaborative Filtering there are mainly two tasks.

1) Calculate User Similarity: - Compare users by their individual ratings and define neighbors (Pearson correlation coefficient)
2) Generation of Recommendation :- Prediction for user u, based on neighbor's ratings (weighted by correlation)

But in real world relevance of a particular rating for the current prediction depends on user and also the context similarity as well. Hence in order to model context in a CF system, a user's choice or preference needs to be associated with the context in which the user made that choice. This means that the current context needs to be known each time the user makes a choice. The same applies for the reciprocal: when a user asks for recommendations, we need the current context and evaluate what others have chosen previously in a similar context. Implies that predicted rank should also be weighted by the context similarity or the relationship between the current context and the context in which the user selected the item. For this we can introduce an additional task to CF process.

1) Calculate Context Similarity
2) Calculate User Similarity
3) Generation of Recommendation

But our algorithm associates with an implicit feedback problem. Because Pearson coefficient uses a rating value as a parameter to the equation a model like Pearson coefficient cannot be directly used to calculate similarities between contexts and users. In our scenario users don't give an explicit rank to the selected image. The only feedback system gets is the image that user selected from the recommended results. Hence we had to introduce another mechanism to calculate similarities. Cosine Similarity is another model to calculate relationship between two users or items.

Cosine-based Similarity - In this case, two items are thought of as two vectors in the *m* dimensional user-space. The similarity between them is measured by computing the cosine of the angle between these two vectors. Formally *m* × *n* ratings matrix similarity between items *i* and *j*, denoted by *sim (i,j)* is given by

$$sim\ (i,j) = \frac{V_i \cdot V_j}{|V_i| \times |V_j|}$$

Where V_i and V_j are rating vectors of items *i* and *j*. Similarly, similarity can be calculated for two users using their rating vectors. In that case rating vector of the user can be denote by

$$V_u = 3 \times I_1 + 5 \times I_2 + 4 \times I_3$$

Where 3, 5 and 4 are ranks given by user u to items I_1, I_2 and I_3 respectively. In implicit feedback systems there are no rank values. As an example our system has a binary feed backs (rank value is either 1-if selected or 0 – is user has never selected that image). Say that user u_1 has selected images I_2 and I_3 then the $V_u = 0 \times I_1 + 1 \times I_2 + 1 \times I_3 = I_2 + I_3$. Since the same image can be selected by the same user more than once, that frequency can also be consider when calculating similarities.

$$V_u = f1 \times I_2 + f3 \times I_3$$

Now the vectors are very much similar to the vectors in classical CF process. Hence Cosine similarity model can be used as the basic concept for calculating user and context similarities in our implicit feedback algorithms.

B. *Context Definition*

In a standard CF system, a user's profile is made up of a set of items with at most one rating assigned to each item. In a dynamic environment like in our problem, a user's preference towards an item may change with the context. To capture this, a snapshot of the context needs to be stored along with each user-selection. In the problem of image search and recommendation context can be categorized in to two depending on the client application.

1. Structured context: - Context can be expressed using some structured context types.

Eg: Online Diagramming

Context Type/Context Dimension	Context Values
Diagram Type	Business Workflow
Used Object Style	Sketchy
Used Object Color	Black & White

A snapshot of context is a composite of different types of context data from various sources. Consequently, various context data can be available or not, depending on what is accessible in the current environment. This yields the requirement that different context types should be managed independently, and their combined impact be calculated algorithmically.

2. Unstructured context: - Context values cannot be associated with a clearly defined context type. A snapshot of context is a composite of different context data.

E.g.: in Online Blogging, context is expressed using a set of keywords-most frequently used words that are collected from the content of the blog.

Considering all the above facts picSEEK has introduced two different CF algorithms each support one of the above mentioned context types.

1. Structured-Context Collaborative Filtering
2. Tag-based Collaborative Filtering

C. *Structured-Context Collaborative Filtering*

- Context Similarity

The goal of calculating context similarity is to determine which images are more relevant for the current context. The similarity of the context in which an image is selected with the current context of the active user determines the relevance of that image to the current context. Consequently, for each context type there needs to be a *quantifiable measure* of the similarity between two context values.

Context types can vary widely and it would be difficult to manually define a similarity function for each context type. Therefore, we devised an automated method to compare the relevance of one context value to another for the same context type. We make the assumption that if user preferences towards an image do not differ much in different contexts, then the images given in one context would also apply for the other. So if the images that have being selected are similar for two different context values, then these two values are very relevant to each other.

$$Sim_t(C\,.X) = \frac{CVector_t(C).CVector_t(X)}{|CVector_t(C)|.|CVector_t(X)|}$$

The above equation calculates the similarity of two context values C, X at context dimension t. It is the calculation of the cosine similarity between two vectors each representing the occurrence of values C, X at dimension t in our previous interactions. This returns the relevance of two context values in a context dimension C_t over all the images users have selected in this context.

The context for C at t would be in the following format. Assume that several users have selected i_j, i_k, i_l images while working in contexts described by C at context dimension t previously using our service.

$$CVector_t(C) = 2i_j + i_k + 4i_l$$

This implies that image i_j, i_k, i_l has been selected respectively 2, 1, 4 times each for contexts described by C at context dimension t previously.

- Incorporating Context into Prediction

In the normal CF process, a prediction is calculated by combining the neighbors' ratings into a weighted average of the ratings, using the neighbors' similarities as weights. But picSEEK uses implicit feedbacks. Hence we have come up with a simple implicit feedback model for the rank.

$$r_{i,X} = 1 + \propto f_{i,X}$$

In above model $r_{i,X}$ is the rating of image i in the context of X where $f_{i,X}$ is the frequency of that image within that context (number of times that image i has selected within in the context X). In our experiments, setting $\alpha = 0.04$ was found to produce good results.

In the context-aware CF system, each rating has an associated context. The similarity between the contexts of past interactions with the active user's context determines how relevant this image is, so it must be incorporated into the weight.

We define $R_{u,i,c}$ as the weighted rating for the user u on an item i in context c, where c is the current context of the active user. This rating is weighted as with respect to the similarity between context x in which the rating r was given and the context c of the active user. The context is multi-dimensional so we assume linear independence and calculate the similarity for each dimension separately.

$$R_{u,i,c} = k \sum_{x \in C} \sum_{t=1}^{2} r_{i,x}\,.sim(c,x,i)$$

Here k is a normalizing factor such that the absolute values of the weights sum to unity.

- Generation of Recommendation

Now these equations can be used as the context sensitive version of user's implicit rating to generate prediction, i.e.,

$$P_{a,i,c} = k \sum_{u=1}^{n} R_{u,i,c}\,.Similarity(U_a\,, U_u)$$

Where $P_{a,i,c}$ is the predicted rating of active user a in context c for item i. This calculation combines all the weighted ratings, with respect to similarity in context, of all the neighbors, which is then again weighted with respect to the similarity of user- *Similarity(U_a ,U_u)* , to give an overall prediction for the active user on an image in the current context.

D. *Tag-based Collaborative Filtering*

Consider an Online blogging application where the working environment cannot be described in a structured manner. In this case the working environment can be described by the key content of the blog which we call here as tags. When a blogging user searches for an image, a tag set can be extracted by analyzing the content of his blog, which can be used to calculate the tag similarity measure.

Tag based recommendation algorithm is based on the concept that an image can be described using a set of tags. When a user searches for an image using a set of tags, the most relevant images for him can be recommended by calculating a similarity measure of the current tags and the tags by which each image is tagged on.

Assume this paragraph is a part is of a blog content that the user is currently working on, "A university is an institution of higher education and research. University provides both undergraduate education and postgraduate education." The *T - TagVector* describing the current context is as follows. This is generated after neglecting the words that has no key contribution to the direction of the context.

$$T_{current} = (3)education + (2)university + (1)institution$$

By looking in to previous feedbacks from users who selected image *i* in their interactions we come up with a *TagVector* for the image *i*. In this case image *i* have been previously selected for contexts that were described by tags education, university, undergraduate, study and student.

$$T(i_j) = (8)education + (4)university + (1)undergraduate + (6)study$$

- Tag Similarity/ Context Similarity

Tag similarity is the measurement of similarity between two contexts. Mainly we calculate this for taking the similarities between current context and the contexts for which the target image has selected. We use cosine similarity with respect to implicit feedback logic.

$$Similarity\left(T_{current}, T(U_b, i)\right) = \frac{T_{current} \cdot T(U_b, i)}{|T_{current}| . |T(U_b, i)|}$$

- User similarity

Tag-based user similarity between two users is calculated based on their similarity of tagging an image. It is reasonable to assume that two users who tags the same image with same set of tags are likely to be similar minded. So with regard to that image, these two users should have a higher similarity value. If this is the case for all the images which has been tagged by both A and B users, A and B are very like to be similar minded. This is our basis of Tag based user similarity prediction.

T (U_a,i) represents all the tags that user U_a has provided in his previous interactions with our service at times when he selected *i* from our recommendations. The following is the mathematical model we came up for user similarity predication.

$$T(U_a, i) = (8)education + (4)university + (1)undergraduate + (6)study$$

Hence to represent a certain users' behavior say U_a we have a set of such TagVectors one for each image that U_a has ever selected. Hence when we need to calculate the similarity between two users, what we do is we calculate the similarity of their behaviors or the relative overlap of their behaviors. As the behavior is described by a set of tags provided for each image by each user we calculate the overlap of those tags for each image and take their mean as the measure of their similarity.

$$Similarity(U_a, U_b) = \frac{1}{2(n)} \left| \sum_{j=common\ images} \frac{|T(U_a, i) \cap T(U_b, i)|}{|T(U_a, i) \cup T(U_b, i)|} \right|$$

n is the number of commonly selected images by the two users which is used as a normalization factor

- Generation of Recommendation

Now in order to predict the rank for a target image *i* following model can be used.

$$P_{a,i,c} = \sum_{u=1}^{n} r_{i,c} . Sim\left(T_{current}, T(U_b, i)\right) . Sim(U_a, U_b)$$

Here *u* denotes all the users who have selected the image *i* for the same keyword.

IV. Evaluation

We evaluate a scenario where we generate for each user an ordered list of images for a particular domain (say medical student), sorted from the one predicted to be most preferred till the least preferred one. Then, we present a prefix of the list to the user as the recommended Images. It is important to realize that we do not have a reliable feedback regarding which images are unloved, as not selecting an image can stem from multiple different reasons. In addition, we are currently unable to track user reactions to our recommendations. Thus, precision based metrics are not very appropriate, as they require knowing which programs are undesired to a user. However, selecting an image is an indication of liking it, making recall-oriented measures applicable.

We denote by $rank_{u,i}$ the percentile-ranking of image *i* within the ordered list of all programs prepared for user *u*. This way, $rank_{u,i}$ = 0% would mean that program *i* is predicted to be the most desirable for user *u*, thus preceding all other programs in the list. On the other hand, $rank_{u,i}$ = 100%

indicates that image *i* is predicted to be the least preferred for user *u*, thus placed at the end of the list. (We opted for using percentile-ranks rather than absolute ranks in order to make our discussion general and independent of the number of images.) Our basic quality measure is the expected percentile ranking of a selecting an image in the test period, which is:

$$\overline{rank} = \frac{\sum_{u,i} r_{u,i} \times rank_{u,i}}{\sum_{u,i} r_{u,i}}$$

Lowest of *rank* values are more desirable, as they indicate ranking actually selected images closer to the top of the recommendation lists. Notice that for random predictions, the expected value of $rank_{u,i}$ is 50% (placing *i* in the middle of the sorted list). Thus, *rank* > 50% indicates an algorithm no better than random.

A. Evaluation Results of Structured-Context Collaborative Filtering

We have tested two versions of this algorithm by considering and without considering user awareness. User is asked to search a student from our system giving context details within the domain medicine. Following are some of the domains used to search images.

- Education- Medicine- Family Care
- Education- Medicine- Critical Care
- Education- Medicine- Geriatrics
- Education- Medicine- Emergency Medicine
- Education- Medicine- Pediatrics
- Education- Medicine- Anesthesiology
- Education- Medicine- Dental
- Education- Medicine- Nursing

And we have observed results increasing the number of relevant images for medical domain by 10 till 100 and by increasing the number of other images for same keyword from 40-400. And Figure R2 illustrates how the expected percentile ranking changes as number of images increases.

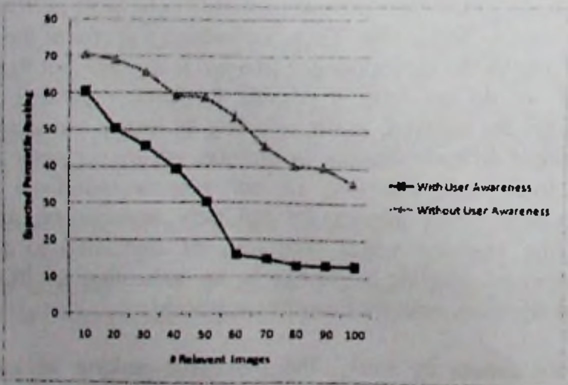

Figure 2. Performance graph of Structured-Context Collaborative Filtering

Figure 2 shows the measured values of rank with different number of images, and also the results by the user aware and non user aware models. We can clearly see that expected percentile ranking is decreasing as the image database grows (number of relevant images is increasing). Better results are obtained by our user aware model, which offers a more principled approach to the problem. Expected percentile ranking are getting deceased with a higher rate in user aware model. This shows that the user aware algorithm has been able to recommend images which are relevant to the context and which are mostly preferred by the user.

B. Evaluation Results of Tag-Based Collaborative Filltering

Same as in section *A* we have tested two versions of Tag-based algorithm by considering and without considering user awareness. User is asked to search a student from our system giving context tags (some words that represent the context where user exists) within the domain medicine. Observed results are shown in Figure 3.

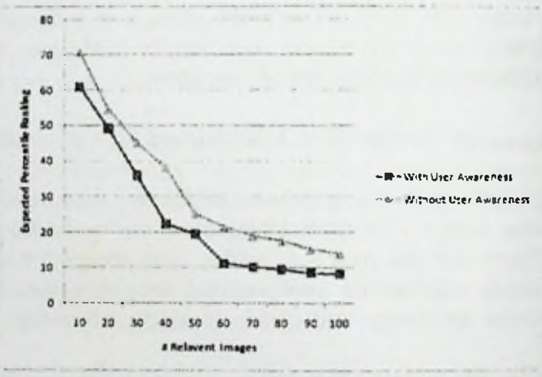

Figure 3. Performance graph of Tag-based Collaborative Filtering

By looking at the above diagram it is clearly visible that both versions of tag-based algorithm shows better performance gain that the structured-context algorithm. Context provided to the tag based algorithms is unstructured. i.e. user can provide any number of words that is relevant to the context. Because of this feedback has much implicit information about images. Hence the learning rate of the algorithm is very high. Same as in structured-context algorithm use based version of tag-based algorithms shows a better performance gain in expected percentile ranking as it provide better recommendation considering the user similarities. i.e Images selected by similar users comes on to the top of the recommendation list, and there is a high possibility if selecting an image which is on top by the current user as they have similar interests.

V. Conclusion & Future work

This paper describes two new contextual collaborative filtering models based for image recommendation. In this model, areas of traditional CF have been changed: user similarity calculation and score prediction calculation. And we have incorporated another measurement called context similarity into prediction. These new models effectively consider the context in which a user likes a resource. Given that the context is considered, effective recommendations can be made. They can be made over a larger domain, not just image recommendation but movies music and domains such as the entire internet. Even users who have differing interests can be effectively recommended for. For example Movies cover many genres and consequently, not considering the context of why a user likes a resource is not very effective.

picSEEK has presented an novel methodology for the flexible and user-friendly retrieval of images, combining context awareness and results have been show that it has overcome the restrictions of conventional methods. From here we are planning to improve the efficiency of our algorithms by using load balancing and clustering. Additionally, further research into tag expansion through natural language processing methods is needed.

References

[1] U. Shardanand and P. Maes, "Social information filtering Algorithms for automating 'word of mouth'," in *Proceedings of the ACM CHI 95 Human Factors in Computing Systems Conference May 7-11, 1995, Denver, Colorado*, vol. 1, pp. 201-217.

[2] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems. A survey of the state-of-the-art and possible extensions." *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734-749, 2005.

[3] R. Ritendra Datta, D. Joshi, J. Li and James Z. Wang, " Image Retrieval: Ideas, Influences, and Trends of the New Age." *ACM Computing Surveys*, vol. 40, no. 2, article 5, pp. 1-60, 2008.

[4] S. Boutemedjet and D. Ziou, "A Graphical Model or Context-Aware Visual Content Recommendation," *IEEE Transactions on Multimedia*, vol. 10, no 1, pp. 52-62, January 2008.

[5] J. Laaksonen, M. Koskela, S. Laakso and E. Cja, "PicSOM content-based image retrieval with self-organizing maps," *11th Scandinavian Conference on Image -Pattern Recognition Letters*, vol 21, Issues 13-14, December 2000, pp 1199-1207.

[6] E. Hyvonen, A. Styrman, and S Sarela. "Ontology-based image retrieval," *In Towards the semantic web and webservices, Proceedings of XML Finland 2002 Conference*, pp15–27,2002.

[7] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin, "Incorporating Contextual Information in Recommender Systems Usinga Multidimensional Approach," *ACM Trans. Information Systems*, vol. 23, no.1, Jan. 2005.

[8] John S. Breese, David Heckerman, and Carl Kadie,"Empirical Analysis of Predictive Algorithms for Collaborative Filtering." *in Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pp 43–52, July 1998.

[9] D.W. Oard and J. Kim,"Implicit Feedback for Recommender Systems", *In Proceedings of the 5th DELOS Workshop on Filtering and Collaborative Filtering*, pp 31–36,1998.

[10] P Resnick, N Iacovou, M Suchak, P Bergstrom, and J Riedl, "An open architecture for collaborative filtering of netnews," *In the proceedings of the ACM Conference on Computer Supported Cooperative Work*, pp. 175-186, 1994.