

THE RECOMMENDATION ALGORITHM FOR AN ONLINE ART GALLERY

WALDEMAR KARWOWSKI^{a)}, JOANNA SOSNOWSKA, MARIAN RUSEK^{a)}

^{a)} *Department of Informatics, Warsaw University of Life Sciences (SGGW)*

The paper discusses the need for recommendations and the basic recommendation systems and algorithms. In the second part the design and implementation of the recommender system for online art gallery (photos, drawings, and paintings) is presented. The designed customized recommendation algorithm is based on collaborative filtering technique using the similarity between objects, improved by information from user profile. At the end conclusions of performed algorithm are formulated.

Keywords: algorithms, recommender system, collaborative filtering

1. Introduction

Nowadays most of web pages are created not only by the programmers and administrators but also by the users. This direction of the development of the Internet started at the beginning of 21 century with the possibility of commenting web content by users, and is known as Web 2.0 described in the Tim O'Reilly article from the year 2005 [1]. With the ability to publish and edit the content by any user, everybody can be a co-author of a portal. Main manifestation of this idea is the popularity of blogs, social networking and wiki services. Of course users' impact on the content of the websites is not restricted to enable users to self-manage the content and appearance of the pages. Users provide their personal data and can be identified during their activity. This means that Internet services have a lot of information about users, for example history of activity or interest of topics. In the

age of the Internet, the customer is faced with the problem of excess offers and information. Similar situation appears, if user is looking for interesting information on the social networking site. Currently many systems provide mechanisms for automatically displaying personalized content on the basis of the data contained in the user profile and the history of his activity on the portal. Examples of such mechanisms are recommendations to help user finding interesting new content, services with similar or related topics and people interested in such topics. The most common applications of personalization mechanisms are media-sharing services such as videos (YouTube) or images (DeviantArt), thematic portals with reviews of movies (FilmWeb) computer games and books and, above all, e-commerce systems and online stores, of which the largest and most famous is the amazon.com. Such systems are called recommendation or recommender systems. However, recommendation systems have not appeared together with Web 2.0, they are much older. The first recommendation systems concepts already appeared before year 1980, much earlier than the first Internet portal. A prototype system of recommendations was “electronic Librarian” Grundy [2], book proposals system for reader, based on the information provided by the reader and the preset “stereotypes” about readers taste. The first real running, but partially manual, recommendation system was Tapestry [3]. It allowed the user to query for items in an information domain and had the task of filtering the documents, for example messages in internal e-mail systems used by corporations. The purpose of its use was to release users from the unnecessary messages. Shortly after it appeared fully automatized filtering systems GroupLens [4]. It was locating relevant opinions automatically and aggregating them to provide recommendations to identify Usenet articles which are likely to be interesting to a particular user. Generally a recommendation process is closely related to the filtering of information, in mentioned examples, recommendation was designed to help user choose proper documents by filtering. Today, recommendation more frequently may be a way to offer the product to the purchaser.

The aim of the work was an implementation of the simple recommendation system for an online art gallery MyArtGallery. MyArtGallery is typical Web 2.0 service and was created in ASP.NET MVC 5 technology as a part of first degree thesis. The main functionality of the MyArtGallery is the ability to publish users’ work in various fields of the wider art. Main gallery functionalities are among other commenting on the work of other users, add images to own collection of favorites, download images in the selected resolution on own PC and much more. The images are divided into genres (categories) and are described with keywords (tags). The application also offers the ability to search for images based on specified criteria. Analysis of the first version of the MyArtGallery showed that a recommendation system would be useful. The recommendation system was needed to facilitate the user to discover images that may be of interest to him.

The rest of this paper is organized as follows: in Sect. 2 the concept of a recommender system, together with most important recommendation technics and algorithms is presented. In Sect. 3 problems connected with recommendation for an online art gallery is discussed. In Sect. 4 an original recommendation algorithm for an online art gallery is precisely described. We finish with summary and brief remarks in Sect. 5.

2. Recommendation systems and algorithms

Assisting the user in making decisions is very important because of the widespread information overload. Information overload comes from the fact that modern man meets daily with much more information than he is able to process, i.e. understand and remember. This problem is much older than the World Wide Web; however thanks to the dynamic development of the Internet, it has become particularly disruptive. The consequence of the increasingly more widespread access to the Internet in all parts of the world in conjunction with the use of Web 2.0 philosophy is the fact that the amount of new information grows with the number of users. The book dedicated to the problem of information overload [5] describes yet another variation of this phenomenon—overload messages (the message) resulting from the popularity of new forms of communication such as social networking sites, post office email and mobile technologies. According to the Internet Live Stats (<http://www.internetlivestats.com/one-second/>), within each second it is published more than 7 thousand entries on the social networking site Twitter, on Instagram is published more than 700 new photographs, and is sent more than 2.5 million e-mails (October 2017). In this situation, the possibilities offered by traditional search engines are inadequate. The use of traditional search engines involves the necessity of independent browsing hundreds or even thousands of pages of results. Users need recommendations from trusted sources to make decisions; this means that information filtering systems are very important.

Definitions of recommendation systems are rather descriptive. According to [6]: “The goal of a Recommender System is to generate meaningful recommendations to a collection of users for items or products that might interest them”. Currently the most common are contacts between sellers and buyers and Recommendation Systems become one of the most powerful and popular tools in electronic commerce. In other words “Recommender Systems have evolved to fulfill the natural dual need of buyers and sellers by automating the generation of recommendations based on data analysis” [6]. It is possible because sellers and site owners have a large collection of data gathered about users that allows for deeper analysis of how a user interacts with topics, items etc. From the other side users need to personalize their online environment to overcome information overload. We can define Recommender systems as tools to help people make decisions in complex

information spaces [7]. According to [8]: “Recommender Systems are software tools and techniques providing suggestions for items to be of use to a user. The suggestions provided are aimed at supporting their users in various decision-making processes, such as what items to buy, what music to listen, or what news to read”. Most important functions of recommender systems are listed in [8]. From the service providers’ point of view there are: increase the number of items sold, sell more diverse items, increase the user satisfaction, increase user fidelity, better understand what the user wants. From the users’ point of view there are: find some good items, find all good items, annotation in context, recommend a sequence, recommend a bundle, just browsing, find credible recommender, improve the profile, express self, help others, influence others. In order to implement its core functionality, identifying the useful items for the user, a Recommendation System must predict that an item is worth recommending. In order to do this, the system must be able to predict the utility of some of them, or at least compare the utility of some items, and then decide what items to recommend based on this comparison [8].

Process of generating a recommendation depends on, among others, from the system destination, implemented functionality and availability of data on the user activities. Many recommendation algorithms use the history of reviews or other activities that could be construed as equivalent to evaluate items. According to [6] Recommender Systems can be broadly categorized as two types. In Collaborative Filtering systems a user is recommended items based on the past ratings of all users collectively. The second type is Content-based recommending where recommend items are similar in content to items the user has liked in the past, or matched to attributes of the user. Moreover [6] define many Hybrid approaches which combine both collaborative and content based approaches. In [9] recommenders systems are classified as collaborative filtering and knowledge-based approaches. Collaborative filtering is a real-time personalization technique that leverages similarities between people to make recommendations. In contrast, a knowledge-based recommender system exploits its knowledge base of the product domain to generate recommendations to a user, by reasoning about what products meet the users’ requirements. Much wider taxonomy was provided in [10] it distinguishes between four different classes of recommendation techniques based on knowledge source of recommendation approaches. Collaborative: the system generates recommendations using only information about rating profiles for different users. Content-based: the system generates recommendations from two sources: the features associated with products and the ratings that a user has given them. Demographic: a demographic recommender provides recommendations based on a demographic profile of the user. Recommended products can be produced for different demographic niches, by combining the ratings of users in those niches. Knowledge-based: a knowledge-based recommender suggests products based on inferences about a user’s needs and preferences. This knowledge will sometimes contain explicit functional

knowledge about how certain product features meet user needs. Additionally [8] distinguish community-based approach. This type of system recommends items based on the preferences of the users friends. Evidence suggests that people tend to rely more on recommendations from their friends than on recommendations from similar but anonymous individuals.

3. Recommendation problems for an online art gallery

The first version of MyArtGallery was implemented in ASP.NET MVC technology. The basic functionality of the MyArtGallery is the ability to publish users work in the field of visual art, for example: drawing, painting, photography and all other forms of art, which can be provided in the form of a digital image by photograph or scan. This ability is available only for registered users. The user account contains basic personal information such as, first name, last name, date of birth, gender, user name, e-mail address, avatar and information about interests. The publication process involves uploading image to the server from user computer or submission the image Web address, and entering in the information such as the title, short description of the image, the list of keywords and genre of art (category) from the selection list. The application has the ability to edit the information and image file and offers many additional functions such as commenting on the images of other users, create a collection of user's favorite images, downloading graphic files in the selected resolution to user's computer and reporting to administrator about images and comments which are illegal with the principles of the community. The functionality implemented in the first version also included the ability to search for images or user profiles according to the selected criteria. The search engine has two modes: simple and advanced.

SZUKAJ OBRAZÓW - ZAAWANSOWANE

Tytuł

Autor

Opis
czarny kot

Słowa kluczowe
kot, kitten, cat, czarny

Wszystkie kategorie

Kategoria
Fotografia

Posortuj wyniki według
Data dodania

Szukaj

Figure 1. Advanced search. Source: own preparation

Simple search means searching to find all the information about the images or profiles according to given word or phrase. The functionality of the advanced search is designed to allow users to discover art based on several different criteria. User fills the form (Fig.1) with the following criteria: the selected genre of art, whose description contains a given phrase, and is tagged by chosen keywords. Results can be sorted against date or popularity. However, both search methods have proven to be insufficient, because using the information in text form is associated with multiple disadvantages. First of all full text search is not always applicable because user can leave description field almost empty. Keywords (tags), and categories allow user to quickly search for images without the use of expensive computationally intensive searching text algorithms. However, this does not eliminate the underlying problem resulting from the application of a classic search engine, which is the need to accurately determine the search criteria that requires good orientation in the topic (e.g., frequent links between the keywords). In addition, image tags in the MyArtGallery are supplied by users; connecting image with keywords is very subjective. One user may provide significant keywords but the other rather random one. Another element to be taken into account is the problem of the different priorities of individual users, for one user more important is what the image shows and for another technique, in which item was made.

Mentioned above problems have resulted in the need for adding the recommendation system. The main goal was primarily to facilitate the user to discover interesting images and thus broaden his interest in the arts. The choice of algorithm for art gallery is a special challenge due to the nature of the published content, because art items evaluating is very subjective. For example, user might not like the illustration connected with favorite book drawn in a style that user does not like. Recommendations based on a single image should contain the art of similar themes and genre, but yet unknown for the currently logged user. Recommendations for a specific user should combine items similar to those that he already knows, and new for him but often liked by users with similar taste. It is also important to maintain a balance between current user interests and topics interesting him in the past.

Content-based solutions work well for many kinds of content. However, the use of a description of the image and keywords is very inaccurate for image recommendations. In addition, this method is very sensitive to errors such as misspelled category. Contrary common filtering recommendation algorithm based on the relationship between images makes the quality of the recommendations independent of the image description. However, this solution has a few restrictions particularly undesirable in the case of an art gallery like favoring the most popular images and popular types of art. Such recommendations are not good for users interested in niche genres of art. This problem is particularly visible for systems with a relatively small amount of data in the database.

The best choice was to create a hybrid solution, which combine the diversity of recommendations and their compatibility with the user's current interests. Recommendation system for the MyArtGallery is designed on the basis of the common filtering algorithm extended with the concepts of knowledge-based algorithm [11]. Common filtering enables us to generate a recommendation without relying on information about the images. Algorithm based on the knowledge compensates the basic restrictions associated with the common filtering. A solution of this type works well in e-commerce systems [12], and one of the objectives was to try the similar solution in the noncommercial social media system. It was decided to implement two types of recommendations: images similar to the currently displayed image and recommendations for the user, based on his favorite images. Both algorithms are influenced by the solution described in [13]. They are based on generated list of neighbors i.e. images often added to favorite lists by users interesting in in the past a certain image or group of images. List of images similar to the currently displayed image is created on the basis of the classic list of neighbors i.e. images with the highest similarity values to the currently displayed image. Recommendations based on user's interest are selected from lists of neighbors for each user images and from a list of his favorites.

4. Recommendation algorithm for an online art gallery

The process of generating recommendations for the user is divided into three basic steps. The first of these is to create a list of candidates. Then a list of “neighbors” for a sample image or a list of images is created with the use of common filtering based on the items. At the end the received recommendations are ordered on the basis of the values of similarity and additional information. The first step, initial filtering data, consists on preparing data for analysis and includes the creation of a list of candidates or images to be taken into account by the algorithm. Finally a list of all the images in the Gallery is limited by removing user paintings, his list of favorites and such, for which recommendations were earlier rejected by the user. The second step, creating a list of neighbors and a choice of n top recommendations is performed as the following. In both types of recommendations, the recommendations are generated using common filtering based on the links between images. The difference is how to create a list of neighbors and criteria for the selection of the n best results (it was set for MyArtGallery $n = 20$). Instead of the classic concept of the selection of the best neighbors based on the value of the similarity, recommendation algorithm for the MyArtGallery is using several criteria to organize potential recommendations. In addition to the value of the similarity, algorithm takes into account information about the categories and tags supplied by the user, so it can be referred to as a hybrid solution that combines the elements of the com-

mon filtering and the concept of knowledge-based recommendations. This approach allows user to generate a satisfactory recommendation, regardless of the amount of own work and his favorites and number of links with existing user interests.

First type of recommendations is based on one sample image. When a user displays a page of the selected image, below the image a list entitled “similar images” is presented. Recommendations for a single image are not stored in the database, it is created dynamically. The first step of the algorithm is to create a user ratings matrix (see table 1).

Table 1. Sample matrix of ratings – filtering based on items (images)

	Sample Image	Image1	Image2	Image3	Image4
User 1	1	0	1	1	0
User 2	1	1	0	0	1
User 3	1	1	1	1	1
User 4	1	1	1	0	0

The rows of the matrix correspond to users whose list of favorites contains the sample image, columns – images from the list of candidates added to favorites by at least one of user. If user i added an image j to own favorites, rating at the intersection of the row and column has the value 1, otherwise it has the value 0. The next and most important step of the algorithm is to generate a list of neighbors for the image. For each column of the matrix of ratings, similarity to the sample image is counted. As a measure of similarity is the Jaccard coefficient was chosen for vectors of likes of comparing images i.e. columns of the matrix. The primary argument for the choice of Jaccard measure of similarity was the fact that in the system using the favorites list “ratings” are binary values. Compared sets have the following form:

- column of sample image - consists of all ones, the number of components is equal to the number of users who have added the image to their favorites.
- column of the ratings matrix for j -th image.

The common part of the sets (columns) are components that both have a value of 1. To compare images based on interest history it is enough comparing the number of ones in the corresponding columns of the matrix. The pattern in the image of j -this image simplifies to equation 1:

$$\text{similarity}(j) = \frac{\text{number of values 1 in column } j}{\text{number of sample image likes}} \quad (1)$$

The undoubted advantage of this measure of similarity is relatively small computational complexity - for each image it is performed one division, and addition at most as many "likes" as sample image. Because column of sample image likes always consists of all ones, the value in the denominator is the number of users who have added the image to their favorites. The identifiers of the images and the corresponding values of similarity are stored in a dictionary of potential recommendations and sorted in descending order and finally narrowed down to the $2n$ elements with the highest values of similarity. Thanks to that data analyzed in the further part of the algorithm is much smaller.

The next step is to create on the basis of the dictionary of potential recommendations new dictionary which assigns each identifier an array of numbers. These numbers are measures of the similarity of the two images: one image i contained in the dictionary of potential recommendations and the sample image.

For an identifier i array contains:

- the value of the similarities (Jaccard) for the image with identifier i from the dictionary of potential recommendations,
- the ratio of the number of common tags for images being compared to the number of all image tags with identifier i ,
- a value of 0 or 1 represents the membership of comparable images to the same category of art (this value is also 1 if the image belongs to one of the subcategories of the category image sample).

This dictionary is first ordered in descending order by the common tags and limited to the n elements corresponding to the images with the highest content of common tags. Then the resulting dictionary is ordered according to the category. If images have the same content of common tags, as the first positions are images in the same category of art, what the sample image. For the image, which nobody has added to the favorite measure, dictionary contains the identifiers of all images from the list of candidates and tables consisting of only two values corresponding to the common tags and belonging to the category. List of recommendations appears in view is the list of images from the list of candidates, whose identifiers are included in the ultimate recommendations dictionary. The order of the images is the same as the order of the elements of the dictionary.

Recommendations based on the profile of the logged-on user are generated on the basis of a list of user's favorite images, and his own images. They are generated each time the user navigates to the home page of the application. Before the start of the algorithm, existing recommendations for a particular user, in addition to the recommendations rejected by user, are removed from the database. The basis of the algorithm is a comprehensive dictionary of potential recommendations, initialized as an empty dictionary, similar to the dictionary for a single image. Then, for each image from the history of user activity is performed the following procedure. In

step 1 dictionary of similarities for the current image is created– the same way as for a single image. In step 2, created dictionary is narrowed down to the elements with the highest values of similarity. In step 3, identifiers of images and assigned to them the values of similarity are added to the dictionary of potential recommendations. If the algorithm encounters an identifier that is already in the dictionary, the value of the similarity is summarized with the value assigned to the same key value. It means that the images similar for more than one image are more likely to find in the list of recommendations. Then in the same way second dictionary is created, with images potentially uninteresting for a particular user (anti-recommendations) based on images whose recommendations were rejected in the past by the user. It is limited to the n elements with the highest values of similarity. Items whose keys are included in this dictionary are removed from the dictionary of potential recommendations, and dictionary of potential recommendations is reduced to the 2n elements with the highest values of similarity.



Figure 2. User profile - preferences

The last stage of recommendation algorithm involves extracting the n best recommendation based on dictionary of potential recommendations and preferences from profile for particular user (Fig. 2). If user did not provide any additional information about his preferences, recommendations are generated based on the n items with the highest similarity values extracted from the dictionary of potential recommendations. If at least one of the four lists included in the profile of user interests is not empty, the new dictionary is created. Its keys are the image identifiers from the dictionary of potential recommendations and its values are arrays of numbers. The following conditions must be met to the identifier of the image found

in this dictionary: the image cannot belong to the category ignored by user and user list of tags can contain a maximum of 10% of the tags that are marked by a given user as uninteresting.

Array for an element with the identifier i consist of three elements:

- the similarity values for the image i ,
- the number for the ratio of the number of image tags in the list of tags preferred by the user to the number of all tags for image i ,
- a value of 0 or 1 which indicates a membership category of image i to the list of categories preferred by the user or their subcategories.

Extracting n the best recommendations from the dictionary is performed in the same way as in the case of recommendations for a single image. If the list of similarities is empty (for example, for a user who hasn't any own images, and nothing is added to his favorites), the dictionary value is created based on the entire list of candidates. Due to the lack of similarity values, an array for identifier i consist only two numbers calculated based on user profile. The result is a list of images with the highest content of the tags listed in the profile of user interests and preferred categories. The last stage of generating recommendations is to create recommendation from the final version of the dictionary and save them in the database in the order specified by the algorithm.

5. Conclusions and future work

The recommendation system has been tested by using 30 user accounts with the sample data. Users' interests have been chosen so way that some topics often occur together (e.g. nature-photography, science fiction-fanart, traditional art-portrait). Evaluating of the algorithm consisted of comparing how much the recommended images correspond to the interests of the user. The second aspect of evaluating was to determine whether the topic of recommendations that go beyond the current interests of the selected user is related to the common relationship between user preferences. For the first type of recommendation test were performed for a selected set of examples. They have shown that the use of a recommendation based on the relationship between users interested in a given image, even if the image description is not very accurate, makes that the algorithm is able to find images with similar themes. Using a tag description does not dramatically change the results of a recommendation, but only increases precision. Tests for recommendations based on the user's profile have been tried for all user profiles created. It was shown that profile-based recommendation functionality allows users expand their interests and receive suggestions that match their current preferences. However, for this type of recommendation it was necessary to clarify the user profile and

to reject the part of the proposed images. After such steps, the quality of the results was much better. It was concluded that the most important future improvement should be possibility of remember tags often encountered together. Remembered links can be used to generate hints of possible tags when a user fills out a preferences form.

REFERENCES

- [1] O'Reilly T. (2005) *What is Web 2.0*, <http://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html>.
- [2] Rich E. (1999) *User Modeling via Stereotypes*, Cognitive Science Volume 3, Issue 4 October 1979, 329–354.
- [3] Goldberg D., Nichols D., Oki B. M., and Terry D. (1992) *Using collaborative filtering to weave an information tapestry*, Communications of the ACM, vol. 35, no. 12, 61–70.
- [4] Resnick P., Iacovou N., Suchak M., Bergstrom P., and Riedl J. (1994) *GroupLens: an open architecture for collaborative filtering of netnews*, in ACM CSCW '94, 175–186.
- [5] Strother J. B., Ulijn J. M., Fazal Z. (2012) *Information Overload. An International Challenge for Professional Engineers and Technical Communicators*, John Wiley & Sons.
- [6] Melville P., Sindhvani V. (2010) *Recommender Systems. Encyclopedia of Machine Learning*, Springer.
- [7] Rashid A.M., Albert I., Cosley D., Lam S.K., McNee S.M., Konstan J.A. et al. (2002) *Getting to know you: learning new user preferences in recommender systems*. In: Proceedings of the international conference on intelligent user interfaces, 127–34.
- [8] Ricci F., Rokach L., Shapira B., Kantor P. B. (2010) *Recommender Systems Handbook*, Springer.
- [9] Tran T. (2007) *Combining Collaborative Filtering and Knowledge-Based Approaches for Better Recommendation System*. Journal of Business and Technology.
- [10] Burke, R. (2007) *Hybrid web recommender systems*. In: The AdaptiveWeb. Springer Berlin / Heidelberg 377–408.
- [11] Sosnowska J. (2016) *Recommender system for an online art gallery*, MSc thesis, Warsaw University of Life Sciences.
- [12] Burke R. (1999) *Integrating Knowledge-based and Collaborative-filtering Recommender Systems*. AAAI Technical Report WS-99-01.
- [13] Linden G., Smith B., York J. (2003) *Amazon.com Recommendations. Item-to-item collaborative filtering*. IEEE Internet Computing. January-February 2003.