Combining content based and collaborative filter in an online musical guide

Nandita Dube, Larisa Correia, Dhvani Parekh, Radha Shankarmani

Abstract— The explosive growth of web content makes obtaining useful data difficult, and hence demands effective filtering solutions. Collaborative filtering combines the informed opinions of humans to make personalized, accurate predictions. Content-based filtering uses the speed of computers to make complete, fast predictions. In this work, we present a recommendation approach that combines the coverage and speed of content-filters with the depth of collaborative filtering. We apply our research approach to an online musical guide an as yet untapped opportunity for filters, useful to the wide-spread music populace. We present the design of our filtering system and describe the results from preliminary experiments that suggest merits to our approach.

Index Terms— Content based filtering, Collaborative filtering, recommendation, MD5, Google Custom Search API, Naive Bayes filter, music.

1 INTRODUCTION

In this age of information, we want to know the current events, important ideas and smart opinions that are circulating in our world. That is-what's happening and what's interesting in our area of interest. There is a considerable amount of free content on the Web related to Music, but comparatively few tools to help us organize or mine such content for specific purposes. Thus, it's increasingly difficult and time-consuming to find the content we want. This complexity of choice reinforces the necessity of filtering systems that assist users in finding and selecting relevant results. There have been attempts made at filtering musical content, but these attempts have not completely countered the weaknesses of human and computer filters [1].

While humans are generally smart at deciding what information is good and why, they are relatively slow compared to the amount of information that is out there to process. And while computers have the power and connectivity to reach out to trillions of data bytes, they are stupid in deciding what is relevant and what is not.

Collaborative filtering applies the speed of computers to the intelligence of humans. The motivation for collaborative filtering comes from the idea that people often get the best recommendations from someone with similar tastes to themselves. Collaborative filtering explores techniques for matching people with similar interests and making recommendations on this basis. Previously, Ringo has used pure collaborative filtering to recommend music to users [1].

However, collaborative filtering alone can prove ineffective for various reasons:

Data sparsity: One typical problem caused by the data sparsity is the cold start problem. As collaborative filtering methods recommend items based on users' past preferences, new users will need to rate sufficient number of items to enable the system to capture their preferences accurately and thus provides reliable recommendations. Similarly, new items also have the same problem. When new items are added to system, they need to be rated by substantial number of users before they could be recommended to users who have similar tastes with the ones rated them.

Gray sheep: Gray sheep refers to the users whose opinions do not consistently agree or disagree with any group of people and thus do not benefit from collaborative filtering. These individuals will rarely, if ever, receive accurate collaborative filtering predictions, even after the initial start up phase for the user and system.

Limited diversity: Collaborative filters are expected to increase diversity because they help us discover new products. Some algorithms, however, may unintentionally do the opposite. Because collaborative filters recommend products based on past sales or ratings, they cannot usually recommend products with limited historical data. This can create a rich-get-richer effect for popular products, akin to positive feedback. This bias toward popularity can prevent what are otherwise better consumer-product matches.

Talking about pure content based filtering methods; they are based on a description of the item and a profile of the user's preference. In a content-based recommender system, keywords are used to describe the items; beside, a user profile is built to indicate the type of item this user likes. In other words, these algorithms try to recommend items that are similar to those that a user liked in the past (or is examining in the present) [2]. In particular, various candidate items are compared with items previously rated by the user and the best-matching items are recommended. Content based filtering methods are less affected by the above mentioned problems of collaborative filtering as their techniques apply across all documents. For example, a filter that predicts high interest articles with the word 'Shreya' in it, will give the prediction even before anyone has read the article.

Despite these strengths, content based filtering alone is not adequate to reach the desired goal of relevant filtering. Unlike humans, content based filtering systems have difficulty in differentiating between high quality and low quality information that is on the same topic. Also, as the number of items increases, the number of keywords used to describe a user profile increases, making it difficult to predict accurately for a given user.

Experiments have shown that collaborative filtering can be enhanced by content based filtering. [6, 7] By using a combination of the two, we can realize the benefits of content based filtering by eliminating the early rater problem or the cold start problem, and give early predictions for all users and items, while also gaining the advantages of accurate collaborative filtering predictions as the number of items and users increases.

2 SYSTEM

The system works as a web based application which requires every user to register with a unique username along with a password. The user can search for music artists and obtain relevant results through his account and his history is stored in the database. The history is scanned and an account of preferences of every user, based on the genre and era is made. The system then recommends most likely artists which the user may like, after classifying him into appropriate categories.

A. Narrowing search results

- The general search results provided by search engines like Google include all possible links whose content contains the keyword entered by the user. In such a scenario, it becomes difficult to find results relevant to user query. So, it becomes important to narrow down search results provided to the user. The system aims at providing search results closest to the user's musical interests by providing links to articles and blog entries from selected websites. The choice of these websites was done by an extensive online research of browsing trends of users. A list of relevant websites was made, which included online blogs like WordPress, Blogger, Tumblr, Medium, Svbtle, SETT, Ghost, Squarespace, Typepad, PostHaven and newspapers like The Times of India.

- We have made a custom search engine using the Google Custom Search API for this purpose. The system uses this search engine to provide user search results. In this way, irrelevant links are not displayed and user search experience is improved.
- Google Custom Search Engine allows web developers to build a tailored search experience using the core Google search technology, and it allows you to prioritize or restrict search results based on settings that you specify from a control panel. Several features allow you to manage the way your custom search engine responds from within your particular intranet; the control panel gives you the power to manipulate settings that will fine tune the search results your user's request.

B. Storage of user information and content based recommendation

- Every time a user logs in to the system and searches for an artist or an album, the search keyword is stored in the database along with the unique user ID given to him. The trend of searches performed by every user is analysed to find out his preference of genre and era of music. Let U be the set of all system users who have registered $U = \{U_1, U_2, U_3, \dots\}$
  Let U_i be a user whose search history is an ordered set $U_h = \sum\{(S_i, F_i)\}$
  Where S_i is the artist searched and F_i is frequency of search for S_i
  The preference of U_i is a set P having all S_i corresponding to F_i which have value greater than a threshold t.

- The artists or albums most frequently sought after by the user are noted. User preference of genre and era of music is found out by mapping the most frequently sought after artist or album to a reference dataset meant for this purpose. The dataset is updated timely as and when new music information is discovered. Also, the most recent search trends of the user are identified. This is done keeping in mind the fact that user interests change over time. As the users of the system increase, we have training data to classify users into genres and eras of interest [4].

C. Classification of system users

The Simple Bayesian classifier is one of the most successful algorithms on many classification domains. Despite of its simplicity, it is shown to be competitive with other complex approaches especially in content based filtering. Making the 'naive' assumption that features are independent given the class label, the probability of an item belonging to class *j* given its *n* feature values, $p(class|f_1, f_2, \dots f_n)$ is proportional to:

$$P(class_j) \prod p(f_i | class_j)$$

Based on the training data, the classification of a user as like or dislike can be made as:
Let Like be *l* and Dislike be *d*
Let $F=(f_1, f_2, f_3, \dots)$ be set of parameters which contribute to identification of the class.
Let *X* be the sample of user history, having values of the required parameters.
$p(X|l) \cdot p(l) = p(f_1|l) \cdot p(f_2|l) \dots \cdot p(l)$

$p(X|d) \cdot p(d) = p(f_1|d) \cdot p(f_2|d) \ldots \cdot p(d)$

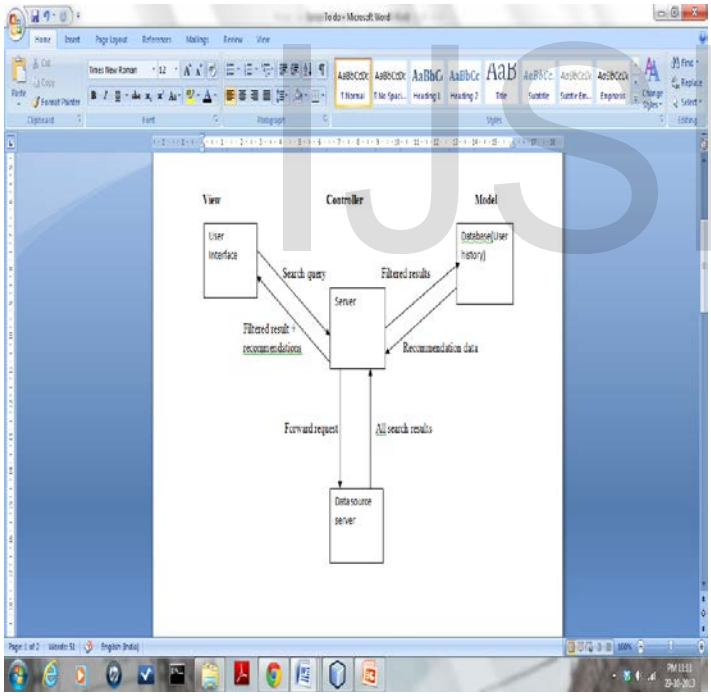
The class which maximises the probability is chosen.

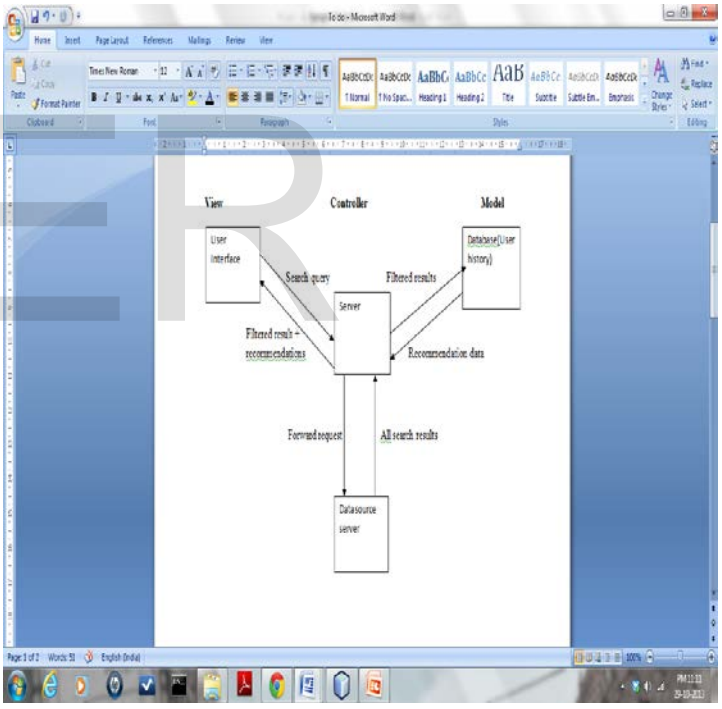
D. Recommendation

It is quite probable that users belonging to the same class in terms of genre and era will have similar tastes. The searches of all users belonging to the same class as the target user, which are not common to searches of the target user are recommended to the target user. Also, since the system keeps track of latest preference of the user. Hence, artists similar to latest searched artist can be similarly recommended.

The content based recommendation is done from the reference dataset which maps user preferences to genre and era in that genre. Thus, artists belonging to the same genre and era in that genre, as the user preference are recommended. For collaborative recommendation, ratings from users are taken for preferred artists. Similar users are found out by calculating Pearson,s co efficient of correlation for user ratings. The dataset is updated timely as and when new music information is discovered. Also, the most recent search trends of the user are identified. This is done keeping in mind the fact that user interests change over time. As the users of the system increase, we have training data to classify users into genres and eras of interest.

3 SYSTEM ARCHITECTURE





IJSER

4 TECHNOLOGY USED

PHP and XAMPP

PHP is an acronym used for "PHP Hypertext Preprocessor". It is a widely used, open source scripting language. PHP scripts are executed on the server. PHP files can contain text, HTML, CSS, JavaScript and PHP code. PHP code is executed on the server and the result is returned to the browser as plain html. [11]

The system uses PHP to generate dynamic page content, send and receive cookies. Also it serves to add, delete and modify the database of our system.

A. phpMyAdmin:

phpMyAdmin is a free software tool written in PHP, intended to handle the administration of MySQL over the Web. phpMyAdmin supports a wide range of operations on MySQL. Our system uses SQL to manage the database. Frequently used operations (managing databases, tables, columns, relations, indexes, users, permissions, etc) can be performed via the user interface in phpMyAdmin, while you still have the ability to directly execute any SQL statement.

B. Apache HTTP server:

The Apache HTTP Server is an effort to develop and maintain an open-source HTTP server for modern operating systems like Windows NT and Unix. Its goal is to provide a secure, efficient and effective server that provides HTTP services in sync with the current HTTP standards. Our system uses Apache as the server to host the web application, though there is a requirement of a dedicated host server when the website goes live online. [10]

C. Google Custom Search Engine:

Google Custom Search enables you to create a search engine for your website, your blog, or a collection of websites. You can configure your engine to search both web pages and images. You can fine-tune the ranking, add your own promotions and customize the look and feel of the search results. You can monetize the search by connecting your engine to your Google AdSense account. In our system, we use Google Custom Search API to create a search engine custom to the requirements of the target user. This search engine shows search results from target websites. [9]

D. Password encryption:

The MD5 message-digest algorithm is a widely used cryptographic hash function producing a 128 bit (16 byte) hash value, typically expressed in a text format as a 32 digit hexadecimal number. MD5 has been used in a wide variety of cryptographic applications, and is also commonly used to verify data integrity. [8]

The MD5 algorithm is an extension of the MD4 message-digest algorithm. MD5 is slightly slower than MD4, but is more "conservative" in design. MD5 was designed because it was felt that MD4 was perhaps being adopted for use more quickly than justified by the existing critical review; because MD4 was designed to be exceptionally fast, it is "at the edge" in terms of risking successful cryptanalytic attack. MD5 backs off a bit, giving up a little in speed for a much greater likelihood of ultimate security. It incorporates some suggestions made by various reviewers, and contains additional optimizations. The MD5 algorithm is being placed in the public domain for review and possible adoption as a standard.

The generalised algorithm for MD5 hashing:

IJSER

1. Pad message so its length is 448 mod 512
2. Append a 64-bit original length value to message
3. Initialise 4-word (128-bit) MD buffer (A,B,C,D)
4. Process message in 16-word (512-bit) blocks:
    1. Using 4 rounds of 16 bit operations on message block & buffer
    2. Add output to buffer input to form new buffer value
5. Output hash value is the final buffer value

Every user of our website has a password to login to the website. It is of utmost importance to encrypt the password and store it in the system database in order to prevent security attacks.

5 FEATURES CONTROLLED BY THE SYSTEM

1.1 Function: Registration
Description:Takes user registration including username, password, secret name, email id
A mail is sent to the user confirming the registration.
Input: Username and Password
Output: Unique Registration Status

1.2 Function: Login
Description: Logs in the user. Also has a password change module
Input: Username and Password
Output: Authentication process

IJSER

1.3 Function: Search
Description: Feature to search for artists. Implemented using Google Custom Search API
Input: Keyword
Output: Search Results

1.4 Function: View Recommendation
Description: Recommendations to the user, based on his search history and similarity with other users
Input: User search history
Output: Recommendation

1.5 Function: Logout
Description: Logs out the user by ending the current session.
Output: Logged out

6 CONCLUSION

In our work, we present an approach which combines the effectiveness of content based and collaborative filters on an online musical search. The system helps the users to obtain relevant information about music as also, it recommends to them, other search topics after evaluating similar users and topics, which may be of their interest.

The system is a new approach to combining collaborative and content based filtering in the field of music. It uses direct learning from the web as well as indirect learning from history data of the system.

The system can be further extended to recommendation of relevant articles on artists or music albums presented by the user.

REFERENCES

[1] Mark Claypool, Anuja Gokhale, Tim Miranda, Pavel Murnikov, Dmitry Netes, Matthew Sartin, "Combining Content-Based and Collaborative Filters in an Online Newspaper"

[2] Daniel Lemire, Sean McGrath, "Implementing a Rating-Based Item-to-Item Recommender System in PHP/SQL", May 20, 2013.
[3] Jiyong Zhang, Pearl Pu, "A Recursive Prediction Algorithm for Collaborative Filtering Recommender Systems"
[4] Royi Ronen, Noam Koenigstein, Elad Ziklik and Nir Nice, "Selecting Content-Based Features for Collaborative Filtering Recommenders"
[5] Nathaniel Good, J. Ben Schafer, Joseph A. Konstan, Al Borchers, Badrul Sarwar, Jon Herlocker, and John Riedl, "Combining Collaborative Filtering with Personal Agents for Better Recommendations"
[6] Marko Balabanovic, Yoav Shoham, "Content based Collaborative recommendation", Communications of the ACM, 40(3), March 1997.
[7] Joshua Alspector, Alexander Kolcz, Nachimuthu Karunanithi, "Comparing feature based and clique based user models for movie selection", In proceedings of the third ACM conference on Digital Libraries, pages 11-18, 1998.
[8] www.wikipedia.com
[9] https://developers.google.com/custom-search
[10] httpd.apache.org
[11] https:// www.w3schools.com
IJSER