LOVELY PROFESSIONAL UNIVERSITY

FINAL REPORT

CSE404

Media Recommendation Engine

SUBMITTED BY:

Section – k18gx G2

NAME	ROLL NO.	REG NO.
AKASH KUMAR	46	11815467
ANURAG KUMAR	61	11802361

Project: Media Recommendation Engine

Abstract

A recommendation engine is a system that suggests products, services, information to users based on analysis of data. Notwithstanding, the recommendation can derive from a variety of factors such as the *history of the user* and the *behavior of similar users*. We used collaborative filtering model with both user-based and item- based strategies, matrix factorization model and a graph-based Network Inference model as our rating prediction models. Recommendation systems are quickly becoming the primary way for users to expose to the whole digital world through the lens of their *experiences*, *behaviours preferences* and *interests*. And in a world of information density and product overload, a recommendation engine provides an efficient way for companies to provide consumers with *personalised information* and *solutions*.

Overview

Introduction

Recommendation system has been widely applied to e-commerce and personalized recommending services today, such as recommended friends on Facebook, video recommending on Youtube and music recommendations on Itunes and so on. The benefits that a well-designed recommender system could contribute to business is significant. Since users tend to have higher incentive being in- terested in the items which satifies their tastes and needs rather than random items, personalized recommendations could help increase sales in retailing and improve the users' experience with services.

Suppose we could predict the numerical rate that a user will give to a prod- uct, we could have a better understanding on the preference and taste of this user when making any recommending decisions. The predicted rate could help provide essentially strong evidence to improve the performance of the entire recommending decisions. In our project, we explored several popular rate-prediction models in recommender system and evaluted and compared which achieved highest possible recommendation

Recommendation Algorithm :

Given the user profile, P(u) = (N(u),T(u)), we suggest items to the user that are related to people and/or tags in his profile. The recommendation score of item i for user u is determined by: ∑∈− = · · ( ) ( ) (,) [(,) (,) v N u d i RS u i e β w u v w v i α ∑∈+ − · ( ) (1) (,) (,)] t T u β w u t w t i where d(i) is the number of days since the creation date of i; α is a decay factor (set in our experiments to 0.025, as in [17]); β is a parameter that controls the relative weight between people and tags, and is used in our experiments to evaluate different recommenders; w(u,v) and w(u,t) are the relationship strengths of u to user v and tag t, as given by the user profile; w(v,i) and w(t,i) are the relationship strengths between v and t, respectively, to item i, as determined by SaND, based on direct relations as described in Figure 1. User-item direct relation types are weighted as in previous studies [1,5,17]: authorship (0.6), membership (0.4), commenting (0.3), and tagging (0.3). Tag-item relations are weighted relative to the number of users who applied the tag on the item, normalized by the overall popularity of the tag, as in [1]. Ultimately, the recommendation score of an item, reflecting its likelihood to be recommended to the user, may increase due to the following factors: more people and/or tags within the user's profile are related to the item; stronger relationships of these people and/or tags to the user; stronger relationships of these people and/or tags to the item; and freshness of the item. We exclude items that are found to be directly related to the user. For example, we will not recommend an item on which the user has already commented or has already tagged.

Recommender Widget:

UI widget for item recommendations based on the algorithm described in the previous section. The user is presented with a number of items (three, in this example) that may include a mix of the five LC item types. Each item has a title that links to the original document, and a short description when available

Recommended Items Survey :

Methodology

The main part of our evaluation is based on an extensive user survey, designed to compare the people-based recommender (PBR), the tag-based recommender (TBR), and two combinations of these two recommenders (PTBRs). Participants of the survey were asked to evaluate 16 recommended items in two randomly ordered phases (each phase included eight items): with and without explanations. Each participant was assigned to one of five groups in a round-robin order, receiving recommendations based on one of the following five recommenders: (1) PBR (β=1 in the equation in Section 3.4); (2) TBR (β=0); (3) or-PTBR—each item may be recommended due to related people, related tags, or both (β=0.5); (4) and-PTBR—each item is recommended due to at least one person and at least one tag in the user's profile (β=0.5 with the constraint that both parts of the summation in brackets are nonzero); and (5) POPBR—popular item recommendation (as a benchmark). The popularity of items was determined based on the number of people they were directly related to in SaND, and on the items' freshness. For explanations, we pointed out the types and numbers of the different direct relations with people as well as the last-update date. For example, an explanation for a popular item would be: "tagged by 57, commented by 12, last updated Jan. 17th, 2010". Recommended items in each of the two phases were presented using the widget described in Figure 2, allowing to rate them as "Very Interesting", "Interesting", "I already know this", or "Not Interesting". Our target population for the survey consisted of 1,410 LC users who were directly related to at least 30 other people, 30 tags, and 30 items. We note that this group does not represent the entire population of our organization, but rather active users of the LC system, who are the target population for our recommender system. A link to the survey with an invitation to participate was sent to each of these 1,410 individuals. In addition, we ran the five recommenders for each of these users to retrieve the top 16 items, and calculated average overlap between the items returned from the different recommenders. The average overlap across the 1,410 users between the items returned by the PBR and the TBR was 1.58%, indicating that these two recommenders return very dissimilar items. The POPBR had very low overlap with all other recommenders, ranging from 0.87% to 1.83%. Overlap between the two PTBRs was 38.6%. The or-PTBR had higher overlap with the PBR (57.3%) and the TBR (32.6%) than the and-PTBR (24.1% and 9.7%, respectively). This indicates that the or-PTBR recommends mostly items that are either recommended by the PBR or the TBR, while the and-PTBR recommends more items that are further down the list of the PBR and the TBR. 4.2.2 Results In total, 412 participants completed our survey, originating from 31 countries and spanning the different organizational units: 32% sales, 28% software, 18% services, 11% headquarters, 4% research, 4% systems, and 3% others.

DISCUSSION AND FUTURE WORK :

The results presented in the previous section indicate that using tags for social media recommendation can be highly beneficial. The combination of directly used tags and incoming tags produces an effective tag-based user profile. A TBR that makes use of this profile yields significantly more interesting recommendations than the most effective PBR presented in a previous work . In addition, the items produced by the TBR are almost completely disjoint from the items produced by the PBR (less than 2% average overlap across the top 16 items), indicating that related tags produce very different recommendations as compared to related people. Combining both related tags and people in the user profile does not significantly increase the interest in recommended items over a pure tag-based approach; however, it significantly lowers the percentage of already known items, increases the diversity of item types, and makes explanations more effective. The higher effectiveness of the TBR over the PBR may be attributed to the fact that tags are better filters for topics of interest than are people. People related to the user may broaden the scope of recommended items (and increase diversity), yet they are also likely to add irrelevant items, as they may have interest areas that are different from those of the user. In our previous work on PBRs , some of the feedback we received highlighted the need for additional filtering based on topics
extensive experimentation is conducted to compare people-based and tag-based recommenders as well as their hybridizations. We show that a combination of directly used tags and tags applied by others is most effective in representing the user' s topics of interest. A recommender based on this tag profile yields items that are significantly more interesting to the user than the most effective people-based recommender demonstrated in a previous work [17]. Combining related people and tags in the user profile improves the results slightly further, leading to a 70:30 ratio between interesting and non-interesting items when explanations are included. In addition, a hybrid people-tag-based recommender has other advantages, such as low proportion of expected items, high diversity of item types, richer explanations, and the simple fact that for some users, recommendations based on people work better, while for others, recommendations based on tags are more effective. Future work should thoroughly examine whether the results presented here can be further improved by means such as integration of other recommenders (e.g., content-based or popularity-based), execution of more sophisticated algorithms (e.g., clustering of people, tags, or items), or optimization of the parameters used by the recommender engine.

Motivation:

The key reason why many people seem to care about recommender systems is *money*. For companies such as Amazon, Netflix, and Spotify, recommender systems drive significant engagement and revenue. But this is the more cynical view of things. The reason these companies (and others) see increased revenue is because they deliver actual *value* to their customers – recommender systems provide a scalable way of personalising content for users in scenarios with many items.

Another reason why data scientists specifically should care about recommender systems is that it is a true data science problem. That is, at least according to my favourite definition of data science as the intersection between software engineering, machine learning, and statistics. As we will see, building successful recommender systems requires all of these skills (and more).

AIM:

The main **aim** of implementing a **recommendation engine** is for the customer to buy more products. And if it doesn't then it defeats the purpose of having a **recommendation engine**. Now since a product **recommendation engine** mainly runs on data.

different types of recommendation systems, each use case is different from the next, as each aims to solve a different business problem. Let's consider a few examples:

1. Movie/Book/News Recommendations — Suggest new content that increases user engagement. The aim is to introduce users to new content that may interest them and encourage them to consume more content on our platform.

2. Stock Recommendations — Suggest stocks that are most profitable to the clients. The recommendations may be stocks that they have traded in historically. Novelty does not matter here; profitability of the stock does.

3. Product Recommendations — Suggest a mix of old and new products. The old products from users' historical transactions serve as a reminder of their frequent purchases. Also, it is important to suggest new products that the users may like to try.

OBJECTIVES:

The **objective of recommender systems** is to provide **recommendations** based on recorded information on the users' preferences. These **systems** use information filtering techniques to process information and provide the user with potentially more relevant items.

Recommender systems are widely used in several different domains for the recommendation of articles, music, movies, and even people. Portals such as Amazon and Sub ma r i no use recommender systems to suggest products to their customers. Meanwhile, social networks such as LinkedIn and Facebook use them to suggest new contacts.

To accomplish that, the most used techniques employed in recommender systems are The collaborative filtering and content-based systems. The collaborative filtering does not take into account the type of items, nor their attributes. It takes exclusively into account the expressed opinion about the other items in order to make recommendations. Meanwhile, content-based filtering uses the knowledge it has of the items and their attributes to make recommendations.

SCOPE:

IT is far more limited scope has in A **recommender system**, or a **recommendation system** (sometimes replacing 'system' with a synonym such as platform or engine), is a subclass of information filtering system that seeks to predict the "rating" or "preference" a user would give to an item. They are primarily used in commercial applications.

Recommender systems are utilized in a variety of areas and are most commonly recognized as playlist generators for video and music services like Netflix, YouTube and Spotify, product recommenders for services such as Amazon, or content recommenders for social media platforms such as Facebook and Twitter These systems can operate using a single input, like music, or multiple inputs within and across platforms like news, books, and search queries. There are also popular recommender systems for specific topics like restaurants and online dating. Recommender systems have also been developed to explore research articles and experts,collaborators,[and financial services.

RELATED WORK:

The origin of recommender systems can be traced back to the extensive work in cognitive science, approximation theory, information retrieval, forecasting theories, and consumer choice modeling in marketing. In the most common formulation, the recommendation problem is the problem of estimating ratings for a set of items that have not yet been seen by a given user where, intuitively, a rating should measure how much the user is interested in the item itself. Once we can estimate ratings for the as yet unrated items, we can recommend the items with the highest estimated ratings to the user. Such ratings can be estimated in many different ways, using methods from machine learning, approximation theory, and various heuristics.

In content-based recommender systems [Pazzani and Billsus 2007], the utility r(u o) of an item is estimated using the utility r(u, o_i) assigned by user u to items o∈ O that are in some way similar to item o. For example, in a movie recommendation application, in order to recommend movies to user u, the recommender system tries to understand the commonalities among the movies that user u has rated highly in the past (actors, directors, genres, etc.). Then, only the movies that have a high degree of similarity to the user's preferences would be recommended. The content-based approach to recommendation relies on information retrieval and information filtering techniques and pioneering systems focused on recommending items containing only textual information. Among such approaches, one of the most common is IPCC [Su and Khoshgoftaar 2009], which is a content-based strategy that finds items similar to the test item, and assigns similarity weights to them based on the computation of Pearson correlation coefficients. The main improvement over traditional information retrieval approaches comes from the use of user profiles that contain information about each user's preferences and needs, which can be elicited explicitly, through questionnaires, or implicitly learned from their behavior over time. Another very common class of content-based techniques are those used in Random Walk Recommender (RWR) systems [Yildirim and Krishnamoorthy 2008] that first infer transition probabilities between items based on their similarities and use the transition probability matrix as a similarity matrix. This approach uses a finite length random walk for rating predictions, where items are represented as nodes of a graph and edge labels denote similarity between items.

In recent times, the content-based approach has been applied to other types of multimedia data, beyond text. For example, Mai del et al. [2008] propose a method based on ontologies for ranking reevancy in the electronic paper domain, while in Hijikata et al. [2006], content based filtering has been applied to music data using decision trees. In the domain of multimedia sharing systems, Musial et al. [2008] propose a recommender system that uses two ontologies (one for multimedia objects and one for users). One of the main drawbacks of these techniques is that they do not benefit from the great amount of information that could be derived by analyzing the behavior of other users. Moreover, the content must either be in a form that can be automatically parsed or features should be assigned to items manually. In fact, while information retrieval techniques work reasonably well in extracting features from text documents,

automatic feature extraction is inherently challenging in other domains. Additionally,
Evaluation Our evaluation aims at comparing five types of recommenders: a people-based recommender (PBR); a tags-based recommender (TBR); two types of a hybrid recommender (PTBR): a combination of people or tags (or-PTBR), and a combination of people and tags (and-PTBR, suggesting only items related to both people and tags); and a popularity-based recommender (POPBR), as a benchmark. To the best of our knowledge, this is the first comprehensive study to compare people-based recommenders with tag-based recommenders and their hybridizations. Our evaluation involves the following elements: (1) an offline comparison of the recommended items yielded by the five recommenders over 1,410 LC users, to examine the diversity across the recommenders, and in particular to compare the items stemming from related people with the items stemming from related tags; (2) a user survey with 65 participants who were asked to evaluate tags as indicators of topics of interest, based on four different methods: indirect tags, used tags, incoming tags, and a combination of both used and incoming tags; (3) the main element of our evaluation is a survey of over 400 LC users, who were randomly divided into five groups, receiving recommendations based on the five recommenders. All groups received recommendations in two phases—without explanations and with explanations. Participants were asked to provide feedback on their interest in the recommended items. Our primary results show that the combination of incoming tags and used tags is the most effective in representing a user's topics of interest, with users rating nearly 70% of the topics as very interesting. Recommendations based on a TBR, with a tag profile that combines incoming and used tags, are rated significantly more interesting than the most effective PBR studied in our previous work. Recommended items are shown to be highly different between the PBR and the TBR, with less than 2% overlap. A hybrid PTBR recommender including explanations improves the results slightly further, leading to an over 70:30 ratio between interesting and non-interesting items. It also presents other potential benefits over a TBR, such as a lower percentage of already known items and higher diversity of item types. In the next section, we discuss how existing work relates to our research. We then present our recommender system, followed by a detailed description of our experiments and their results. We conclude by discussing our findings and suggesting future work.

Social Media Platform: Our research platform for personal recommendation is Lotus Connections (LC) [18]—a social software application suite for organizations. It includes seven social media applications: profiles (of all employees), activities, bookmarks, blogs, communities, files, and wikis. We focus on recommending items of the last five applications, disregarding the first two, since profiles pose a different challenge regarding people recommendation [16], and an activity is generally restricted to a limited number of users. In our work, recommended items may originate from one of the following five applications, which are part of LC's deployment within our organization: (1) social bookmarking application, which allows users to store and tag their favorite web pages. It includes 900K bookmarks with 2M tags by 21K users; (2) blogging service that contains 7.5K public blogs, 130K entries, 350K tags and 17K users; (3) online community system that contains 6K public communities, each with shared resources (such as feeds and discussion forums), with a total of 174K members and 19.5K tags; (4) system for file sharing with 15K public files (presentations, photos, articles, etc.), 24K tags, and 8K users; and (5) wiki system with 3K public wikis including 20K pages edited by 5K users, and with 10K tags.

Relationship Aggregation: SaND [5,27] is an aggregation system that models relationships among people, items, and tags, through data collected across the enterprise, and in particular across all LC applications. SaND aggregates any kind of relationships between its three core entities—people, items, and tags. The implementation of SaND is based on a unified approach [1], in which all entities are searchable and retrievable. As part of its analysis, SaND builds an entity-entity relationship matrix that maps a given entity to all related entities, weighted according to their respective relationship strengths. The entity-entity relationship strength is composed of two types of relations: • Direct Relations: Figure 1 shows all direct relations among entities that are modeled by SaND. Particularly, a user is directly related to: (1) another person: as a friend, as a tagger of or tagged by that person, or through the organizational chart (direct manager or employee); (2) an item (e.g., a shared file or a community): as an author, a commenter, a tagger, or a member; or (3) a tag: when used by the user or applied on the user by others. In addition, an item is directly related to a tag if it has been tagged with it. SaND does not currently model any direct tag-tag and item-item relations. • Indirect Relations: Two entities are indirectly related if both are directly related to another common entity. For example, two users are indirectly related if both are related to the same user, e.g., if both have the same manager or friend, or if both have tagged or were tagged by the same

IMPLEMENTATION:

14th ACM Conference on Recommender Systems

Rio de Janeiro, Brazil, 22nd-26th September 2020

RECSYS 2020 The ACM Recommender Systems conference (RecSys) is the premier international forum for the presentation of new research results, systems and techniques in the broad field of recommender systems. Recommendation is a particular form of information filtering, that exploits past behaviors and user similarities to generate a list of information items that is personally tailored to an end-user's preferences. As RecSys brings together the main international research groups working on recommender systems, along with many of the world's leading e-commerce companies, it has become the most important annual conference for the presentation and discussion of recommender systems research. RecSys 2020, the fourteenth conference in this series, will be held in Rio de Janeiro, Brazil. It will bring together researchers and practitioners from academia and industry to present their latest results and identify new trends and challenges in providing recommendation components in a range of innovative application contexts. In addition to the main technical track, RecSys 2020 program will feature keynote and invited talks, tutorials covering state-of-the-art in this domain, a workshop program, an industrial track and a doctoral symposium.

Published papers will go through a rigorous full peer review process. The conference proceedings, which will be available via the ACM Digital Library, are expected to be widely read and cited

REFERANCE:

http://dataconomy.com/2015/03/an-introduction-to-recommendation-engines/

https://www.toptal.com/algorithms/predicting-likes-inside-a-simple-recommendation-engine

http://www.sciencedirect.com/science/article/pii/S1110866515000341

https://mapr.com/blog/recommendation-engines-driving-customer-interactions-next-best-action/

[4] Burke, R. 2002. Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-Adapted Interaction 12, 4 (2002), 331-370.

[5] Carmel, D., Zwerdling, N., Guy I., Ofek-Koifman, S., Har'el N., Ronen, I., Uziel, E., Yogev, S., & Chernov, S. 2009. Personalized Social Search based on the User's Social Network. Proc. CIKM '09, 1227-1236.

[6] Claypool, M., Le, P., Wased, M., & Brown, D. 2001. Implicit Interest Indicators. Proc. IUI '01, 33–40.

{7] Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., & Sartin, M. 1999. Combining Content-Based and Collaborative Filters in an Online Newspaper. Workshop on Recommender Systems, SIGIR '99.

[8] DiMicco, J., Millen, D. R., Geyer, W., Dugan, C., Brownholtz, B., & Muller, M. 2008. Motivations for Social Networking at Work. Proc. CSCW '08, 711-720.

[9] Farrell, S., & Lau T. 2006. Fringe Contacts: People Tagging for the Enterprise. Workshop on Collaborative Web Tagging, WWW '06.

[10] Firan, C. S., Nejdl, W., & Paiu, R. 2007. The Benefit of Using Tag-Based Profiles. Proc. LA-WEB ’07, 32-41. [11] Freyne J., Jacovi, M., Guy I., & Geyer W. 2009. Increasing Engagement through Early Recommender Intervention. Proc RecSys ’09, 85-92.