Recommendation Engine: ATC Connect

MARCH | 2018

Presented by Ankit Shukla
Business and Analytics Advisors Practice

World Wide Technology
www.wwt.com

Table of Contents

Abstract 3

Business Justification 3

Experimental Setup 4

Methodology 4

Results 8

Conclusion 9

References 9

Abstract

Recommendation engines have become a staple for any website or mobile app offering content and products to its users. In the digital age, when users have access to large amounts of information, it is of utmost importance that a service provides relevant content and products to its users. World Wide Technology (WWT) has developed a mobile application named "ATC Connect" to provide customers and employees easy access to all aspects of the Advanced Technology Center (ATC). A key goal of the ATC connect application is to engage customers with the ATC capabilities and enhance their experience at Executive Briefing Conferences (EBCs). A major feature of the application provides access to relevant articles and blog posts that may be of interest to a prospective or current customer. In this paper, a hybrid recommendation engine is built for recommending relevant articles to the users of the WWT ATC Connect mobile app.

Business Justification

Recommendation engines are algorithms that are used to recommend relevant content to the end user. Recommendation engines were initially developed as a distinguishing feature of a software product, but in today's digital age they have become a necessity. This necessity, however, comes with its own challenges. First, relevant content is a very subjective term and differs for each user based on his or her personal preferences. This requires development of an algorithm that understands the consumer and recommends content tailored to meet their needs.

Recommendation systems are widely used by Internet companies like Amazon and Netflix. They have become an important research topic in data science and decision support systems. These systems make use of the massive and detailed data captured by these Internet majors in order to analyze the browsing and purchasing patterns and profile them accordingly. This understanding of consumer behavior is then used by the algorithm to make customized recommendations of products which are more likely to be purchased by a user.

Recommendation systems are broadly classified into two categories - content based (CB) and collaborative filtering (CF). CB systems analyze users by profiling their historical behavioral data. This data, however, can be extremely massive and diverse and may pose a challenge profiling it manually. For such cases, a CF model is used, which is based on the principle that people who share interests in certain things will probably have similar tastes in other things. A new class of recommendation systems, called a hybrid recommendation engine, makes use of both CB and CF to make recommendations. It is this type of recommendation system that has been built for the ATC Connect application and discussed in this paper.

Experimental Setup

The hybrid recommendation engine built for the ATC Connect application is implemented in the R programming language. The three data sources that were used as inputs were transferred from their source application into a columnar Vertica database for analysis. The data is fetched once a day using REST APIs in JSON format. This JSON data is then processed by the R code to generate article recommendations for the application users. The R implementation of the user based collaborative filtering (UBCF) is provided in the recommenderlab package 1 (reference) by Michael Hahsler and Bregt Vereet. This is the package used to generate all of the UBCF recommendations. The CB recommendations are generated by a WWT proprietary algorithm that is again implemented in R. These recommendations are then stored in an Oracle database from where they are updated in the ATC Connect app through an API.

Methodology

The methodology used for this work is a four step process –

1. Data input
2. Data processing
3. User-article matrix
4. UBCF

Each step is discussed in detail below.

DATA INPUT

The following 3 different data sources are used to build the master dataset which serves as an input for UBCF – Digital Insights data, Salesforce (SF) data and Wire data.

Digital Insights Data: This data captures the user activity on the ATC Connect app. This activity varies from browsing practices to reading articles to attending an EBC. It provides information about users scheduled to attend EBCs in the upcoming months. It also has user profile information including user preferences that a user enters while signing up on the app for the first time.

SF Data: SF is a widely used cloud based customer resource management (CRM) tool to generate and keep track of customers and sales opportunities. The past/potential purchase history of each account is used to gauge interest areas of the individual users associated with those accounts.

Wire Data: This data captures online user interaction metrics for the articles published on the WWT portal. These online user metrics include clicks, views, impressions and more, which help determine the popularity of every individual article among the general populace.

These three data sources are pulled from their respective Vertica tables using the REST APIs in JSON format once a day. This data is then processed, transformed and merged together to build the final User – Article matrix that acts as the input for UBCF.

DATA PROCESSING

In most data science work, the raw input data is not fed directly into the algorithm. It typically goes through transformation processes such as deduplicating, filtering, reshaping, transforming and more to convert it into our desired input. The processing steps for each dataset is described below:

Digital Insights Data

- Converting JSON formatted data to tabular format
- De-duplicating the data to get unique records
- Creating article ID, articles read and user information mapping tables
- Filtering out records without any timestamps
- Segregating user application activity into broad buckets (e.g. any article related activity was categorized into article bucket, anything to do with practice into practice, etc.)
- Collating user preferences
- Selecting relevant columns and filtering out records with missing category data
- Removing generic EBC categories (e.g. lunch, travel, etc.) and exploding the data to create one category for each data record
- Mapping every category to a practice and assigning practices to every user activity accordingly
- Calculating correlations between practices and categories for articles read and assigning practices to articles accordingly

SF Data:

- Converting the JSON data to tabular format
- Filtering out relevant columns
- De-duplicating the data to get unique records
- Aggregating the data by practice and account
- Normalizing the practice names

Wire Data:

- Converting the JSON data to tabular format
- Filtering out relevant columns
- De-duplicating the data to get unique records
- Aggregating the data by articles

User-Article Matrix

A major challenge with CF is the cold start problem. Initially, with any application or website, there is not enough ratings by users to generate recommendations. This is also a challenge for the data provided for this work. Since the application is relatively new, there are few users that have read articles. Moreover, user ratings are non-existent for use in UBCF. This problem is tackled by generating simulated ratings for every user for every article.

The Digital Insights data is used to compute an application activity score which is the ratio of a user's activity in each practice to his or her overall activity. Similarly, a score for each practice is provided for every user from the SF data. This is a binary score (0 or 1) depending on the presence/absence of a practice in the SF data for an account. The user preferences that every individual user provides in his or her profile are also considered as a gauge for the user's interest in a practice. For users who are scheduled to attend an EBC, the number of weeks until the upcoming EBC is also considered as a measure of interest in a practice. Lastly, the popularity of the articles is determined by the clicks on the article from the wire data. All of this is combined to generate a user score for every user, which is the metric measuring the interest of a user in a practice. This score acts as pseudo ratings and is used as the input for UBCF. The below equation illustrates this process mathematically:

$$\text{User Score} = (0.2 * \text{SF Opportunity}) + (0.2*(\text{events}/\sum\text{events})) + (0.3*e^{-\text{EBC}}) + (0.3*\text{Tech Interests}) + (\text{Article Score}) \quad [1]$$

Where:

SF Opportunity = SF Opportunity in a practice
event = User activity in a practice on the ATC Connect app
EBC = Weeks until EBC
Tech Interests = User Preferences in the app
Article Score = $n/\sum n$

n = Number of clicks on an article in a practice
$\sum n$ = Sum of all clicks on every article of that practice

Table 1 shows an illustrative example of the User – Article matrix

TABLE 1:

User – Article Matrix Example

User	Article 1	Article 2	Article 3	Article 4	Article 5
Sujit Kulkarni	1.25	0.3		1.2	0.75
Mark Bernanrdo	.0.8		1.2		0.34
Ezer Campos		1.4	0.86		
David Kohler	1	1.2	0.24	1.7	0.46

USER BASED COLLABORATIVE FILTERING (UBCF)

The UBCF technique tries to mimic word of mouth by predicting the items highly rated by users similar to the active user. The assumption is that users who share interests in certain areas will probably have similar tastes in other areas. Similar users can be computed using a number of techniques such as Pearson correlation coefficient and cosine similarity. These similarity measures are defined between two users, u_x and u_y as:

Pearson Correlation:

$$\mathrm{sim}_{\mathrm{Pearson}}(\vec{x},\vec{y}) = \frac{\sum_{i\in I}(\vec{x}_i\bar{\vec{x}})(\vec{y}_i\bar{\vec{y}})}{(|I|\text{-}1)\mathrm{sd}(\vec{x})\mathrm{sd}(\vec{y})} \qquad [2]$$

Cosine Similarity:

$$\mathrm{sim}_{\mathrm{Pearson}}(\vec{x},\vec{y}) = \frac{(\vec{x}\bullet\vec{y})}{||\vec{x}||||\vec{y}||} \qquad [3]$$

Where:

$\vec{x} = r_x$ (row vector in the rating matrix)
$\vec{y} = r_y$ (row vector in the rating matrix)
sd = standard deviation.

The prediction of an item *i* for a user *u* is calculated by computing the weighted sum of different user ratings on item *i*. The prediction P_u is calculated using:

$$P_{ui} = \frac{\sum_v (r_{v,i} * s_{u,v})}{\sum_v s_{u,v}} \quad [4]$$

Where:

$r_{v,i}$ is the rating of user *v* on item *i*
$S_{u,v}$ is the similarity between the users *u* and *v*

Results

This recommendation engine provides article recommendations for the ATC Connect applications users. The number of recommendations for every user is a dynamic parameter and can be set as per the requirement. Once the recommendations are generated, any articles that the user has already read are removed. Currently, four new article recommendations are provided for every application user.

Note that not all the application users are provided with a UBCF recommendation. Some users are provided a recommendation based solely on their user score, providing the top four. These recommendations are called “popular” since we are recommending popular articles to the user. After combining the popular and UBCF recommendations, other user details are pulled in such as user name, user email address, and account with which the user is associated. The output from the recommendation engine is stored in a table in a relational database. A REST API is used to pull the recommendations from the table and show it to the user on the ATC Connect app as shown in Table 2.

TABLE 2:

A typical recommendation output for example users.

User Name	WWT User Id	Article URL	Source	Flag
Sarah Goellner	3621	/all-blog/cisco-announces-new-high-performance-32g-fibre	UBFC	Y
Sarah Goellner	3621	/all-blog/dont-fear-the-migrator-the-myths-and-realities-of	UBFC	Y
Sarah Goellner	3621	/all-blog/how-to-protect-against-drone-threats/	UBFC	Y
Sarah Goellner	3621	/all-blog/unboxing-new-prem-hybrid-cloud-experience/	UBFC	Y
Chad Bockert	221	/all-blog/2017-mobility-trends/	popular	Y
Chad Bockert	221	/all-blog/bus-central/	popular	Y
Chad Bockert	221	/all-blog/grab-your-capes-and-meet-us-at-cisco-live-2017/	popular	Y
Chad Bockert	221	/all-blog/watch-our-tec37-retail-and-mobility-trends-to-watch	popular	Y

Conclusion

A hybrid recommendation engine was developed to recommend relevant article for the users of ATC Connect mobile application. A content based WWT proprietary algorithm is first used to simulate user ratings for all articles. These ratings are then used as input for UBCF to generate final article recommendations for all the app users.

While the focus of this article is on the ATC Connect application, this recommendation service can be deployed on other similar web and mobile based applications that want the ability to recommend relevant content and products to their consumers. In addition, other emerging recommender techniques, such Neural Networks, could be explored. Different types of neural networks like CNNs, RNNs and Auto-Encoders are being investigated for recommending a wide range of content like videos, music and articles, and may provide a better and more accurate recommender system.

References

1. https://cran.r-project.org/web/packages/recommenderlab/vignettes/recommenderlab.pdf

2. https://cseweb.ucsd.edu/~jmcauley/cse255/reports/wi15/Guanwen%20Yao_Lifeng_Cai.pdf