Developing Richly Textured Customer Profiles

•••

Final Presentation
Fall 2018
Bruce Hao

Agenda

1. Current landscape
2. Resulting challenges
3. Path forward
4. Design & implementation overview
5. Results summary
6. A few words of caution

Current landscape

- Social network/media platform APIs have become much more restrictive
  - Who can use the APIs; Which users are data available for; What user data can be accessed; What authorizations must users provide
- User profiles and persona data are more valuable than ever
  - Growth in IoT, AI/ML, NLP, location-based services, etc., opportunities to meet users at the right time, at the right place, with the right thing abound
- Balance between commercial use of user data and user privacy is being reevaluated

Resulting challenges

Companies whose business models depend on the data provided by these social network/media platform APIs are scrambling to fill this massive data gap or have already gone out of business

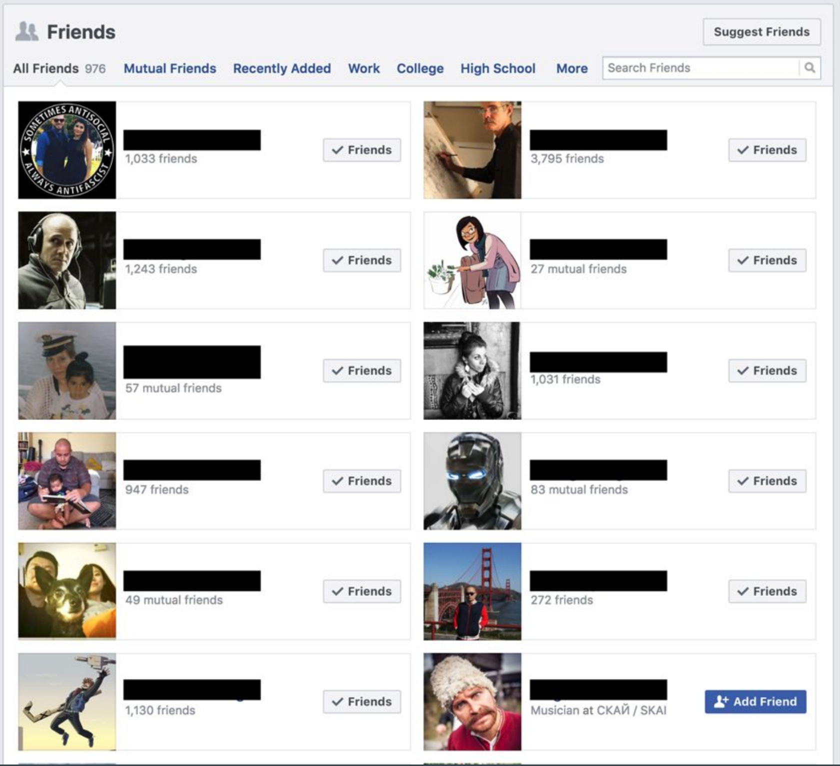
Path forward

- The social network/media platform APIs were the easiest way to access structured user data. While the APIs no longer share these data, the data are still publicly available on these platforms
- As such, we can use a combination of network analyses, text and image processing, recommendation engines, and a host of other techniques to more than fill the gap

Design & implementation overview

Snowball technique to compile a network of users

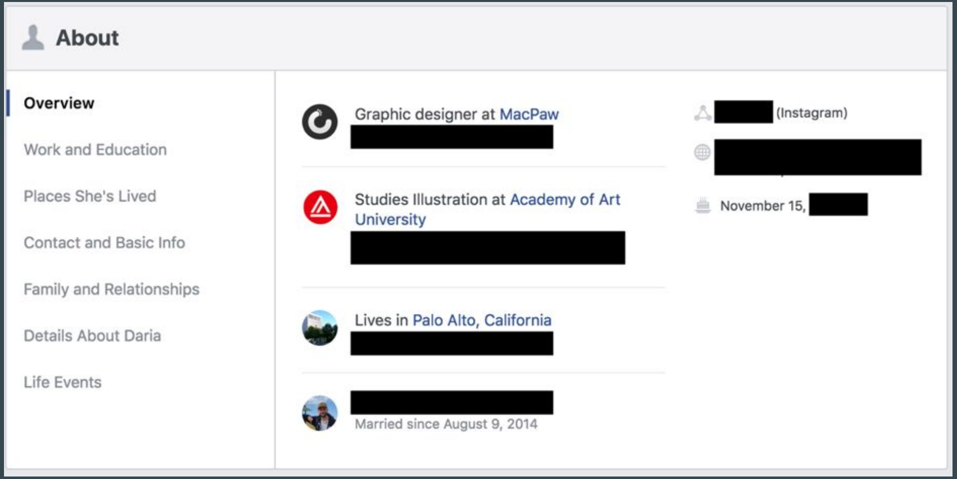
- Starting with disparate users, we can use each user’s friends to snowball out network of users
- In this case, we built a network of a few thousand users and then pruned those users with only a few connections; ultimately, we ended up with 82 users
- Nodes in the graph represent users, and edges represent relationships between the users



Design & implementation overview

Web-scrape basic user attributes

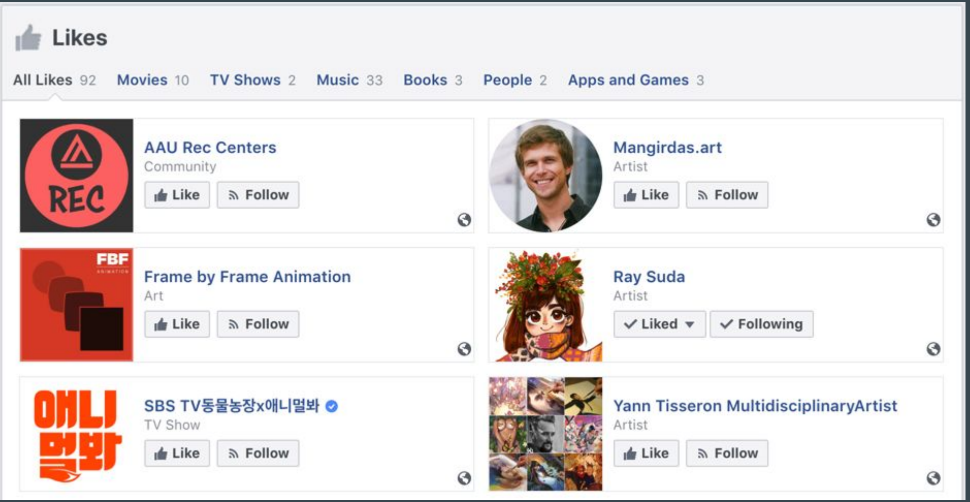
- We can scrape basic user information from the “About” page of each user profile
- Information on the “About” page includes things like occupation and place of work, major and place of study, list of places of residence, marriage status, birth date, etc.
- Finally, we add this information as attributes to each node



Design & implementation overview

Web-scrape user preferences

- We can scrape user preference information from the “Like” page of each user profile
- Information on the “Like” page includes everything a given user has Liked and a summary of the number of Likes within a given category, e.g. Movies, TV Shows, etc.
- In this exercise, we only capture the summary Like information as attributes to the nodes within the graph



Design & implementation overview

Summary of collected data

- Of 82 users in the data set...
- 38 reported year of birth (if not exact DOB)
- 74 reported places of residence
- 74 reported places of study
- 56 reported at least one Like category across 9 total categories, i.e. Movies, Music, Books, TV Shows, Sports Teams, Apps and Games, Athletes, Restaurants and People

Attribute	Total Users	Users Reporting Attribute	Variety of Attribute Values
Year of Birth	82	38	18
Place of Residence	82	74	18
Place of Study	82	74	98
Like Categories	82	56	9

Design & implementation overview

Data cleansing & standardization

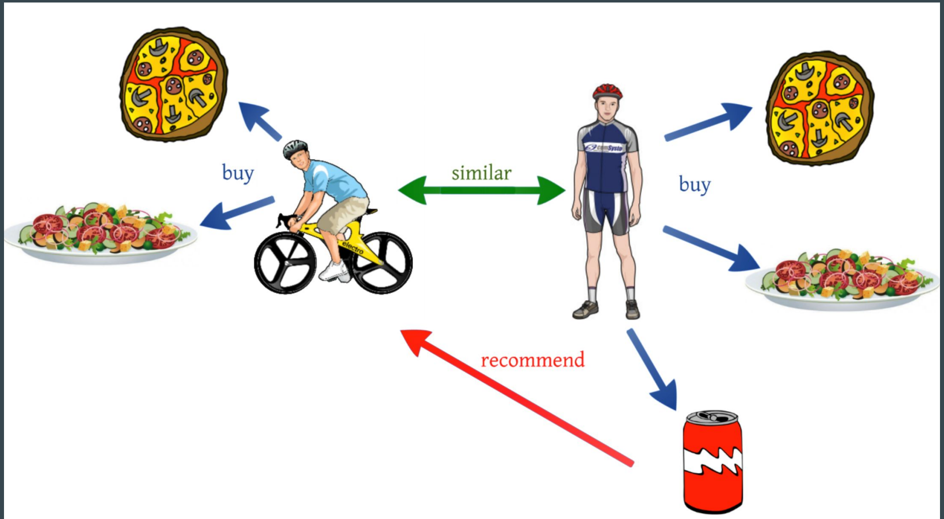
- Reported data comes in a variety of formats
- For example, studies information sometimes includes institution, major, class year, start/end dates, etc.
- For each of the reported categories captured, simple cleansing & standardization scripts were developed to ensure a clean data set
- Also, preference ratings were normalized so that all rating were between 0-1

```
schools_list_clean = []
for study in schools_list:
    # most recent school
    study = re.sub(r"Jan |Feb |Mar |Apr |May |Jun |Jul |Aug |Sep |Oct |Nov |Dec ", "", study)
    if len(re.findall("(Past:|Class of|\d{4} to)", study)) == 0:
        school = re.findall(" at (.*)", study)
        schools_list_clean.append(school[0])
    else:
        school = re.findall(" at (.*)\s(Past:|Class of|\d{4} to)", study)
        schools_list_clean.append(school[0][0])
    # previously attended schools
    if len(re.findall(" Past: (.*)", study)) > 0:
        study_and = re.findall(" Past: (.*)", study)
        if len(re.findall("(.*) and (.*)", study_and[0])) > 0:
            schools_list_clean.append(re.findall("(.*) and (.*)", study_and[0])[0][0])
            schools_list_clean.append(re.findall("(.*) and (.*)", study_and[0])[0][1])
        else:
            schools_list_clean.append(study_and[0])
schools_list_clean = list(set(schools_list_clean))
return(schools_list_clean)
```

Design & implementation overview

Apply user-based collaborative filtering (UBCF) to fill gaps in persona data

- Many users provided limited Like information or none at all
- Thus, there is an opportunity to use a recommendation engine (UBCF in this case) to fill in these preference data gaps
- UBCF basically 1) uses available information to determine user-to-user similarity and then 2) combines that with user-to-item ratings to predict missing user-to-item ratings



Design & implementation overview

Starting with a user-to-item matrix, i.e. rows of users, columns of attributes and cells with preference ratings...

	user	Live: Seoul, Korea	Live: San Francisco, California	Study: Academy of Art University	YOB: 1994	Live: Nonthaburi	Study: Plernpattana	Study: Plearnpattana	Like: Movies	Like: Music	...	Study: Lancaster High School	Study: Gaithersburg High School
0	[redacted]	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0	0	...	0.0	0.0
1	[redacted]	0.0	0.0	1.0	1.0	1.0	1.0	1.0	0	0	...	0.0	0.0
2	[redacted]	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1	2	...	0.0	0.0
3	[redacted]	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0	0	...	0.0	0.0
4	[redacted]	0.0	1.0	1.0	0.0	0.0	0.0	0.0	16	36	...	0.0	0.0

Design & implementation overview

Calculate user-to-user similarity (cosine similarity in this case) to produce user-user similarity matrix, i.e. rows and columns of users and cells of similarity measures

user	[redacted]								
user									
[redacted]	1.000000	0.258199	0.269272	0.666667	0.634600	0.577350	0.252144	0.761639	0.666667
	0.258199	1.000000	0.208577	0.258199	0.245779	0.447214	0.195310	0.294982	0.258199
	0.269272	0.208577	1.000000	0.269272	0.300705	0.233197	0.236937	0.335876	0.269272
	0.666667	0.258199	0.269272	1.000000	0.634600	0.577350	0.252144	0.761639	0.666667
	0.634600	0.245779	0.300705	0.634600	1.000000	0.549580	0.303297	0.811580	0.634600

Design & implementation overview

Take dot product of preceding matrices to produce a prediction matrix, i.e. rows of users, columns of predicted attributes and cells of predicted preference ratings

user	Like: Movies	Like: Music	Like: Books	Like: TV Shows	Like: Sports Teams	Like: Apps and Games	Like: Athletes	Like: Restaurants
[redacted]	6.161584	9.808039	2.175187	3.492303	0.184991	1.778376	0.469863	0.530594
	2.650614	4.233775	0.984794	1.424002	0.077590	0.775556	0.142407	0.214070
	3.147343	5.146681	1.983260	1.758199	0.120049	0.989071	0.219058	0.256632
	5.966542	9.669805	2.171400	3.467686	0.184991	1.770801	0.467970	0.530594
	6.665305	11.363438	2.451143	3.900689	0.301461	1.960371	0.542032	0.577078

Design & implementation overview

Review accuracy of predicted preference ratings

- Since some users provide actual preference ratings, we have “in-sample” data with which we can validate prediction accuracy
- Here, we look at correlations between actual and predicted ratings as an indication of accuracy
- Correlations are mostly in the 0.6 range; these are somewhat conservative since we considered users who provided even just one preference as in-sample, for the sake of expediency

Attribute	Actual-Predicted Correlation
Movies	0.62
Music	0.62
Books	0.65
TV Shows	0.62
Sports Teams	0.83
Apps and Games	0.61
Athletes	0.56
Restaurants	0.51
People	0.94

Results summary

- Certainly room for improvements in prediction accuracy, but...
- We were successfully able to create personas for users that did not report any or incomplete preference information using data from their social network profiles
- These user personas could have been much richer given more time and resources. Areas for further improvement include using the detailed Like data and including data from other social network/media data sources

A few words of caution

- Relying so heavily on a few dominant social network/media platforms for data concentrates business model risk and should be actively diversified (somehow)
- Fragility of web-scraping requires monitoring overhead, but can be done, e.g. Mint
- Ensuring real user utility, control and transparency to mitigate growing (and valid) concerns about user privacy