Capstone Project

A location content-aware recommender system

1. Introduction

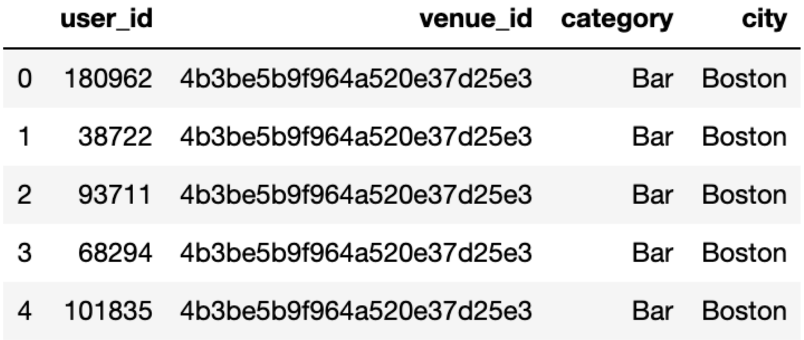
Newly emerging location-based and event-based social network services, such as Foursquare provide us with a new platform to understand users preferences based on their activity history, and thus, we are able to recommend event or/and venues based on the users preferences.

For this capstone data science project i decided to implement a recommender system. There many different algorithm to do this,on of the most used are traditional collaborative filtering-based recommender systems. But, this method is limited since a user can only visit a limited number of venues/events and most of them are within a limited distance range, so the user-item matrix is very sparse.The problem becomes more challenging when people travel to a new city where they have no activity history.

LCARS can alleviate these problems by considering both user preferences and local preferences into recommendation.

2. Data set Presentation

The data set on which the model was learned consists of 4 columns : user Id (Anonymized), Venue Id (Foursquare), category and the city , 161,468 rows, each row represent a user interest (= a checkin) of 646 users , from three big cities : Chicago, New York City and Boston .



	user_id	venue_id	category	city
0	180962	4b3be5b9f964a520e37d25e3	Bar	Boston
1	38722	4b3be5b9f964a520e37d25e3	Bar	Boston
2	93711	4b3be5b9f964a520e37d25e3	Bar	Boston
3	68294	4b3be5b9f964a520e37d25e3	Bar	Boston
4	101835	4b3be5b9f964a520e37d25e3	Bar	Boston

Figure 1 5 firsts user interests

Here is some descriptive statistic about the data set :
Average number of users per user : 13.6 (Chicago), 12.1 (Boston), 10.2 (New York)
Nb of users : 11894
Nb of venues : 47969
Nb chekins from Boston : 54279
Nb chekins from New York : 45866
Nb chekins from Chicago : 61323

This data set was constructed from a larger data set that includes long-term (about 18 months from April 2012 to September 2013) global-scale check-in data collected from Foursquare and contains 33,278,683 checkins by 266,909 users on 3,680,126 venues (in 415 cities in 77 countries). Those 415 cities are the most checked 415 cities in the world, each of which contains at least 10000 check-ins), It contains three files in tsv format.

- File dataset_TIST2015_Checkins.txt contains all check-ins with 4 columns, which are:
1. User ID (anonymized)
2. Venue ID (Foursquare)
3. UTC time
4. Timezone offset in minutes (The offset in minutes between when this check-sin occurred and the same time in UTC, i.e., UTC time + offset is the local time)

- File dataset_TIST2015_POIs.txt contains all venue data with 7 columns, which are:
1. Venue ID (Foursquare)
2. Latitude
3. Longitude
4. Venue category name (Foursquare)
5. Country code (ISO 3166-1 alpha-2 two-letter country codes)

- File dataset_TIST2015_Cities.txt contains all 415 cities data with 6 columns, which are:
Venue category ID (Foursquare)
1. City name
2. Latitude (of City center)
3. Longitude (of City center)
4. Country code (ISO 3166-1 alpha-2 two-letter country codes)
5. Country name
6. City type (e.g., national capital, provincial capital)

References :
This data set was formed by, Yang, Dingqi and Zhang, Daqing and Zheng, Vincent. W. and Yu, Zhiyong, for the article *Modeling User Activity Preference by Leveraging User Spatial Temporal,* in the journal IEEE Transactions on Systems, Man, and Cybernetics: Systems},

3. Methodology

The project is divided in three part, the first part consists in data processing, which includes the formation of the data set and the data wrangling necessary to apply LCARS algorithm., the second part is about the implementation of the LCARS model, where I explain LCARS model and why I used this model.

3.1. Data Processing

The data processing section start with a first step of data collecting and pre-processing, where the code is available in the notebook:LCARS-data-collecting. This step consists in collecting and selecting data from the given data set to form a smaller data set because of my limited computer ressources, I can not hande the whole data set. Here is the 5 tasks to form the users profiles data set :

1. Start with import venues data set (dataset_TIST2015_POIs.txt), and keep the venues from US only.

2. Import cities data set and (dataset_TIST2015_Cities.txt), and use the algorithm of the k nearest neighbors with *scikit learn* to assign to each venues the nearest city from its latitude and longitude.
3. Arbitrary select 3 cities, in our case they are Boston, Chicago and New York cities, and for each of these cities we create a *pandas* dataframe, and merge with the checkins dataset (dataset_TIST2015_Checkins.txt) for each one. Thus we have our user profile dataset. But it is not finished yet.
4. for simplicity's sake, i decided, for user that have several checkins of just one venue, to keep only one occurence of each venue.
5. The data set is still too large, so i reduced it by selecting a limited and equal number (4500) of users in each city.

3.2. Presentation of the recommender system: LCARS

LCARS, a Location content-aware algorithm recommender system that offers a particular user a set of venues (e.g., restaurants) or events (e.g., concerts and exhibitions) by giving consideration to both personal interest and local preference. This recommender system can facilitate people’s travel not only near the area in which they live, but also in a city that is new to them.

3.2.1. The architecture Framework of LCARS

LCARS consists of two components: offline modeling and online recommendation. The offline modeling part, called LCA-LDA, is designed to learn the interest of each individual user and the local preference of each individual city by capturing item co-occurrence patterns and exploiting item contents. The online recommendation part automatically combines the learnst interest of the querying user and the local preference of the querying city top produce the top-k recommendations.

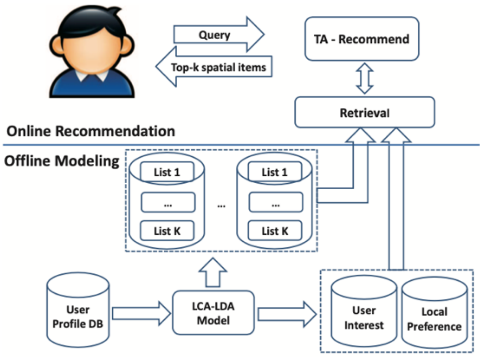


Figure 1: The Architecture Framework of LCARS

3.2.2. LCA-LDA

The proposed offline modeling part, LCA- LDA, is a location-content-aware probabilistic mixture generative model that aims to mimic the process of human decision making on spatial items. This mixture model considers the user’s personal in- terest and the influence of local preference in a unified manner, and automatically leverages the effect of the two factors. Specifically, given a querying user u with a querying city lu, the likelihood that user u will prefer item v when traveling to city lu, is sampled from the following LCA-LDA model.

$$P(v|\theta_u, \theta'_{l_u}, \phi, \phi') = \lambda_u P(v|\theta_u, \phi, \phi') + (1-\lambda_u) P(v|\theta'_{l_u}, \phi, \phi') \quad (1)$$

where P(v|θu,ϕ,ϕ) is the probability that spatial item v is gen- erated according to the personal interest of user u, denoted as θu, and P (v|θ′ , ϕ, ϕ′) denotes the probability that spatial item v is generated according to the local preference of lu, denoted as θlu . The parameter λu is the mixing weight which controls the moti- vation choice. That is, when deciding individual preference on v.

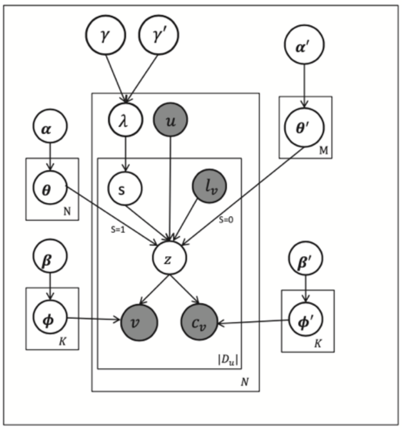
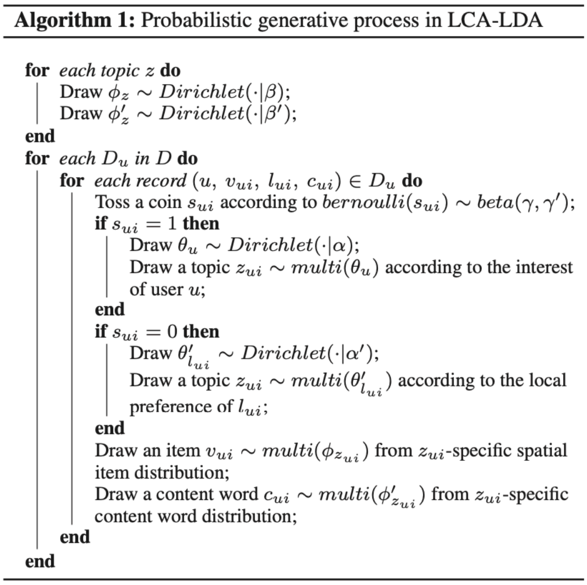


Figure 2: The Graphical Representation of LCA-LDA

3.2.3. LCA-LDA : Inference

Just as presented in the paper of, I we use col- lapsed Gibbs sampling to obtain samples of the hidden variable assignment and to estimate unknown parameters {θ, θ′ , ϕ, ϕ′ , λ}.



Algorithm 1: Probabilistic generative process in LCA-LDA

```
for each topic $z$ do
    Draw $\phi_z \sim Dirichlet(\cdot|\beta)$;
    Draw $\phi'_z \sim Dirichlet(\cdot|\beta')$;
end
for each $D_u$ in $D$ do
    for each record $(u,\ v_{ui},\ l_{ui},\ c_{ui}) \in D_u$ do
        Toss a coin $s_{ui}$ according to $bernoulli(s_{ui}) \sim beta(\gamma, \gamma')$;
        if $s_{ui} = 1$ then
            Draw $\theta_u \sim Dirichlet(\cdot|\alpha)$;
            Draw a topic $z_{ui} \sim multi(\theta_u)$ according to the interest
            of user $u$;
        end
        if $s_{ui} = 0$ then
            Draw $\theta'_{l_{ui}} \sim Dirichlet(\cdot|\alpha')$;
            Draw a topic $z_{ui} \sim multi(\theta'_{l_{ui}})$ according to the local
            preference of $l_{ui}$;
        end
        Draw an item $v_{ui} \sim multi(\phi_{z_{ui}})$ from $z_{ui}$-specific spatial
        item distribution;
        Draw a content word $c_{ui} \sim multi(\phi'_{z_{ui}})$ from $z_{ui}$-specific
        content word distribution;
    end
end
```

As for the hyperparameters α, α′, β, β′, γ and γ′ , for simplicity, we take a fixed value (i.e., α = α = 50/K, β = β′ = 0.01, γ = γ′ = 0.5) .

I set the number of topics up to 50. Because I have few data and less content information compared to original set up described in the paper. The model converge with 300 iterations.

4. Experimental Results

The evaluation method is described in the paper section 3.1.3, and also in the notebook. Basically, it's about calculating the Recall.

I selected 3 user profiles on which I've tested the effectiveness of the recommendation:
--- User 1797 (=user Id) Profile ---
User from Chicago travel to New York
Number of checkins in up2 : 148
Nb of checkins in NYC : 4
Nb of checkins in Chicago : 144
Nb of checkins in Boston : 0

-- User 1583 Profile --
User from NYC travel to Chicago
Number of checkins in up2 : 101
Nb of checkins in NYC : 89
Nb of checkins in Chicago : 12
Nb of checkins in Boston : 0

--- User 2954 Profile --
User from Chicago travel to NYC
Number of checkins in up2 : 125
Nb of checkins in NYC : 8
Nb of checkins in Chicago : 117
Nb of checkins in Boston : 0

For each of these user profiles, we divide a user's activity history into a test set and a training set under two different dividing strategies :
1 - For the first strategy, we select all spatial items visited by the user in a non-home city as the test set and use the rest of the user's activity history in other cities as the training set.
2 - For the second setting, we select all spatial items visited by the user in its home city as the test set and use the rest of the user's activity history in other cities as the training set.

According to the above designed dividing strategies, we split the user activity history S into the training data set Straining and the test set Stest. To evaluate the recommender models, we adopt the testing methodology and the measurement Recall@k

The computation of Recall proceeds as follows. We define hit for a single test case as either the value 1 if the test spatial item v appears in the top-k results, or else the value 0. The overall Recall are defined by averaging all test cases:

Recall = hits / |Stest|
where hits denotes the number of hits in the test set, and |Stest| is the number of all test cases.

User Profile (uid)	Recall (querying cities are new to querying users)	Recall (querying cities are the home cities of querying users)
1797	0.5	0.25
1583	0.25	0.16
2954	0.63	0.20
Average=	0.46	0.20

We compare the results of our experiments with those of the original paper :

As shown in Figure 3(a) where querying cities are new cities, the recall of LCA-LDA is about 0.42 when k = 20 (i.e., the model has a probability of 42% of placing an appealing event within the querying city in the top-20).

That is a better score than the results i reported. Because, on the DoubanEvent dataset is larger and includes content informations. foursquare data set only includes cagegory name.

Also, we can see that in my case, that the model is more efficient where the querying city is a new city to the user than where the querying city is his home city. I cannot explain that.

5. Conclusion

To conclude, the implementation of the location-content-aware recommender system was very challenging. Although the results are quite satisfactory, they are not as good as those reported in the paper. This is mainly due to the data sets, which represent only a sample of the original data set. Moreover, the foursquare data, unlike DoubanEvent, does not contain any content information which makes the LCA-LDA model less relevant.
I strongly recommend seeing the original paper to know more about LCARS.

References.

LCARS: A Location-Content-Aware Recommender System, Hongzhi Yin Yizhou Sun, Bin Cui† Zhiting Hu, Ling Chen. https://www.cs.cmu.edu/~zhitingh/data/kdd13_lcars.pdf
A Theoretical and Practical Implementation Tutorial on Topic Modeling and Gibbs Sampling William M. Darling School of Computer Science University of Guelph.
http://u.cs.biu.ac.il/~89-680/darling-lda.pdf