

Recommender System in Location Based Social Network

Dr. Bidyut Kumar Patra

Assistant Professor

Department of Computer Science and Engineering
National Institute of Technology Rourkela

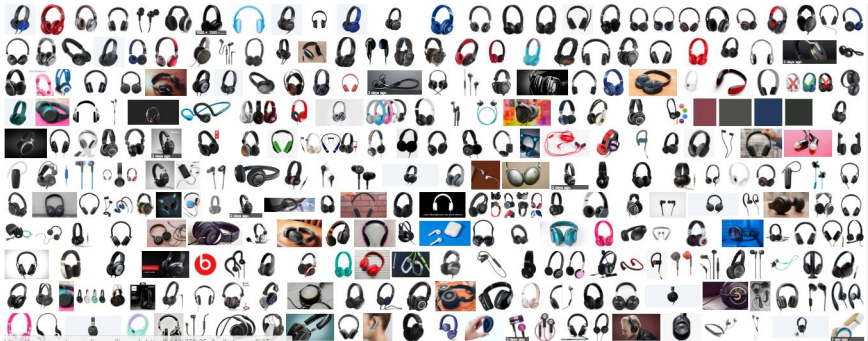
August 21, 2020

Outline

- 1 Introduction
 - Information Overload Definition
 - Recommender Systems
 - Types of Recommender Systems
 - POI Recommendation

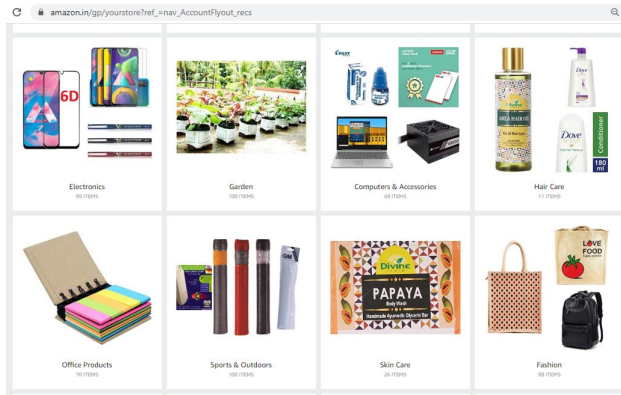
Information Overload

Information overload is a scenario where the consumers are unable to make decision on a product due to too much of available information.



Recommender Systems

Recommender systems help people cope with information overload problem in this digital era.



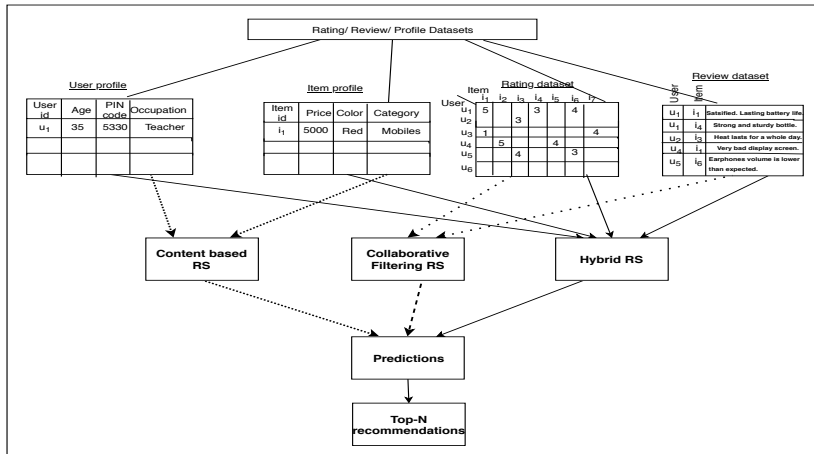
Applications of Recommender Systems

- Movies, series and music in Over The Top (OTT) platforms
- e-learning courses
- Electronic products, furniture and apparel
- Books
- Tours and hotels
- Crops for cultivation

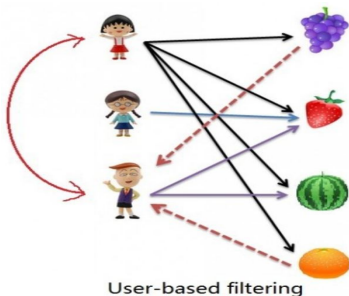
A few applications which use recommender systems:

Netflix, Amazon, Youtube, Spotify, Bookcrossing, Yahoo! Music and Movies, Disney, TripAdvisor, Expedia

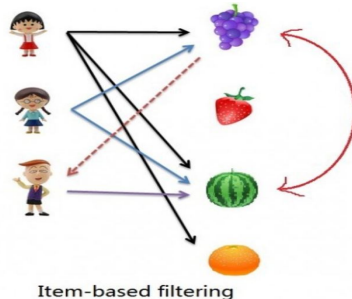
Types of Recommender Systems - Overview



Neighborhood based approaches - User based and Item based Collaborative Filtering



$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum_{v \in N(u)} \text{sim}(u, v) * (r_{vi} - \bar{r}_v)}{\sum_{v \in N(u)} |\text{sim}(u, v)|}$$



$$\hat{r}_{ui} = \bar{r}_i + \frac{\sum_{j \in N(i)} \text{sim}(i, j) * (r_{uj} - \bar{r}_u)}{\sum_{j \in N(i)} |\text{sim}(i, j)|}$$

<https://medium.com/@cfpinela/recommender-systems-user-based-and-item-based-collaborative-filtering-5d5f375a127f>

User-based Collaborative Filtering

- 1 Find similar users of the active user (u_4) who rated target item (i_4): u_1, u_2, u_3, u_5

	i_1	i_2	i_3	i_4	i_5
u_1	5		4	1	
u_2		3		3	
u_3		2	4	4	1
u_4	4	4	5		
u_5	2	4		5	2

- 2 Rating prediction:

$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum_{v \in N(u)} \text{sim}(u, v) * (r_{vi} - \bar{r}_v)}{\sum_{v \in N(u)} |\text{sim}(u, v)|}$$

- 3 Item i_4 is recommended to the user u_4 , if the predicted rating value is greater than a threshold.

Item-based Collaborative Filtering

- 1 Find similar items of the target item which are also rated by active user.

	i_1	i_2	i_3	i_4	i_5
u_1	5		4	1	
u_2		3		3	
u_3		2	4	4	1
u_4	4	4	5		
u_5	2	4		5	2

- 2 Rating prediction:

$$\hat{r}_{ui} = \bar{r}_i + \frac{\sum_{j \in N(i)} \text{sim}(i, j) * (r_{uj} - \bar{r}_u)}{\sum_{j \in N(i)} |\text{sim}(i, j)|}$$

- 3 Item i_4 is recommended to the user u_4 , if the predicted rating value is greater than a threshold.

Traditional Similarity Measures in Neighborhood based CF

Pearson

Correlation (PC)

$$sim^{PC}(u, v) = \frac{\sum_{j \in I(u, v)} (r_{uj} - \bar{r}_u) * (r_{vj} - \bar{r}_v)}{\sqrt{\sum_{j \in I(u, v)} (r_{uj} - \bar{r}_u)^2} * \sqrt{\sum_{j \in I(u, v)} (r_{vj} - \bar{r}_v)^2}}$$

$I(u, v)$ is the set of co-rated items by users u and v
 \bar{r}_u is the average rating of user u , respectively

Jaccard (Jac)

$$sim^{jac}(u, v) = \frac{|I(u) \cap I(v)|}{|I(u) \cup I(v)|}$$

$I(u)$ is the set of items rated by user u
 $I(v)$ is the set of items rated by user v

Cosine

Similarity (CS)

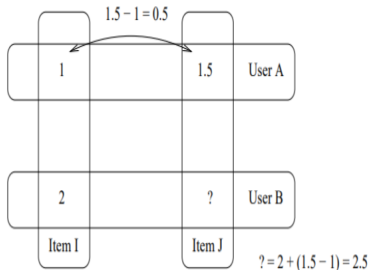
$$sim^{CS}(i, j) = \frac{\sum_{u \in U(i, j)} r_{ui} * r_{uj}}{\sqrt{\sum_{j \in U(i, j)} r_{ui}^2} * \sqrt{\sum_{j \in U(i, j)} r_{uj}^2}}$$

$U(i, j)$ is the set of users who rated items i and j

Adjusted Cosine
Similarity (ACS)

$$sim^{ACS}(i, j) = \frac{\sum_{u \in U(i, j)} (r_{ui} - \bar{r}_u) * (r_{uj} - \bar{r}_u)}{\sqrt{\sum_{j \in U(i, j)} (r_{ui} - \bar{r}_u)^2} * \sqrt{\sum_{j \in U(i, j)} (r_{uj} - \bar{r}_u)^2}}$$

Model based approach - Slope one predictors [1]



- ① Basic slope one: Deviation between items i and j

$$dev_{ij} = \sum_{u \in U} \frac{r_{ui} - r_{uj}}{|U_{ij}|} \quad \hat{r}_{ui} = \bar{r}_u + \frac{\sum_{j \in I_u} dev_{ij}}{|I_u|}$$

Model based approach - Slope one predictors [1]

- 1 Weighted slope one predictor:

$$\hat{r}_{ui} = \frac{\sum_{j \in I_u} (r_{uj} + dev_{ij}) \times |U_{ij}|}{\sum |U_{ij}|}$$

- 2 Bipolar approach:

$$I_u^{like} = \{i \in I_u | r_{ui} \geq \bar{r}_u\} \quad I_u^{dislike} = \{i \in I_u | r_{ui} < \bar{r}_u\}$$

$$U_{ij}^{like} = \{u \in U | i, j \in I_u^{like}\} \quad U_{ij}^{dislike} = \{u \in U | i, j \in I_u^{dislike}\}$$

$$\hat{r}_{ui} = \frac{\sum_{i \in I_u^{like}} \hat{r}_{ui}^{like} \times |U_{ij}^{like}| + \sum_{i \in I_u^{dislike}} \hat{r}_{ui}^{dislike} \times |U_{ij}^{dislike}|}{\sum_{i \in I_u^{like}} |U_{ij}^{like}| + \sum_{i \in I_u^{dislike}} |U_{ij}^{dislike}|}$$

Model based approach - Matrix Factorization

User\ Item	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6
User 1	X	5	3	4	-	4
User 2	2	2	1	-	-	4
User 3	5	4	-	5	-	-
User 4	-	-	3	-	5	1
User 5	3	-	4	5	1	-



User 1	0.50	0.05	0.63
User 2	0.62	0.02	0.78
User 3	0.05	0.84	0.16
User 4	0.25	0.25	0.50
User 5	0.63	0.55	0.01



Item 1	Item 2	Item 3	Item 4	Item 5	Item 6
0.91	1.25	0.86	0.33	0.55	0.68
0.23	0.36	0.51	0.60	0.79	1.02
0.55	0.42	0.45	0.06	0.02	0.09

$$\hat{r}_{ui} = p_u q_i$$

Model based approach (contd.)

- **Matrix factorization techniques for recommender systems - Koren, Bell & Volinsky [2]**

$$\min_{q,p} \sum_{r_{ui} \neq \phi} (r_{ui} - p_u q_i)^2 + \lambda_1 (\|q_i\|^2 + \|p_u\|^2)$$

Model based approach (contd.)

- **Matrix factorization techniques for recommender systems - Koren, Bell & Volinsky [2]**

$$\min_{q,p} \sum_{r_{ui} \neq \phi} (r_{ui} - p_u q_i)^2 + \lambda_1 (\|q_i\|^2 + \|p_u\|^2)$$

$$\hat{r}_{ui} = \mu + b_i + b_u + p_u q_i$$

$$b_i = \frac{\sum_{u \in U(i)} (r_{ui} - \mu)}{\lambda_2 + |U(i)|}$$

$$b_u = \frac{\sum_{i \in I(u)} (r_{ui} - \mu - b_i)}{\lambda_3 + |I(u)|}$$

Important Areas in Recommender Systems

- Long tail item recommendations - items which received very few ratings.

Important Areas in Recommender Systems

- Long tail item recommendations - items which received very few ratings.
- Cold-start user/ item recommendations - users or items recently added into the system.

Important Areas in Recommender Systems

- Long tail item recommendations - items which received very few ratings.
- Cold-start user/ item recommendations - users or items recently added into the system.
- Diversity - recommending diversified item list to the users to maximize liquidation.

Important Areas in Recommender Systems

- Long tail item recommendations - items which received very few ratings.
- Cold-start user/ item recommendations - users or items recently added into the system.
- Diversity - recommending diversified item list to the users to maximize liquidation.
- Group recommendations - Providing recommendations to a group of users - example tour, movie, etc..

Important Areas in Recommender Systems

- Long tail item recommendations - items which received very few ratings.
- Cold-start user/ item recommendations - users or items recently added into the system.
- Diversity - recommending diversified item list to the users to maximize liquidation.
- Group recommendations - Providing recommendations to a group of users - example tour, movie, etc..
- Streaming recommender systems - computing user preferences on the fly.

Important Areas in Recommender Systems

- Long tail item recommendations - items which received very few ratings.
- Cold-start user/ item recommendations - users or items recently added into the system.
- Diversity - recommending diversified item list to the users to maximize liquidation.
- Group recommendations - Providing recommendations to a group of users - example tour, movie, etc..
- Streaming recommender systems - computing user preferences on the fly.
- Session-based recommender systems - capturing short-term transactional patterns of the users.

Important Areas in Recommender Systems

- Long tail item recommendations - items which received very few ratings.
- Cold-start user/ item recommendations - users or items recently added into the system.
- Diversity - recommending diversified item list to the users to maximize liquidation.
- Group recommendations - Providing recommendations to a group of users - example tour, movie, *etc.*.
- Streaming recommender systems - computing user preferences on the fly.
- Session-based recommender systems - capturing short-term transactional patterns of the users.
- Community based recommender systems - utilizing trust and social media information to recommend friends, new connections, *etc.*.

Cold-start Item Recommender Systems

- User cold-start recommendations: Users who recently joined the system.

Cold-start Item Recommender Systems

- User cold-start recommendations: Users who recently joined the system.
- Item cold-start recommendations: New items which do not have any ratings in the system.

Cold-start Item Recommender Systems

- User cold-start recommendations: Users who recently joined the system.
- Item cold-start recommendations: New items which do not have any ratings in the system.
- Trust networks are exploited to understand cold-start users' preferences over the items [3].

Cold-start Item Recommender Systems

- User cold-start recommendations: Users who recently joined the system.
- Item cold-start recommendations: New items which do not have any ratings in the system.
- Trust networks are exploited to understand cold-start users' preferences over the items [3].
- Deep learning based approaches are proposed to learn the user and item features using user profile information and meta-data about the items [4], [5].

Session-based Recommender Systems

- Long-term user models are often not available for a larger fraction of the users.

Session-based Recommender Systems

- Long-term user models are often not available for a larger fraction of the users.
- Users could be first-time on the platform, *i.e.*, cold start users.

Session-based Recommender Systems

- Long-term user models are often not available for a larger fraction of the users.
- Users could be first-time on the platform, *i.e.*, cold start users.
- Session based recommender systems rely solely on user's actions in an ongoing session.

Session-based Recommender Systems

- Long-term user models are often not available for a larger fraction of the users.
- Users could be first-time on the platform, *i.e.*, cold start users.
- Session based recommender systems rely solely on user's actions in an ongoing session.
- Deep learning based approaches are proposed to predict the next item to be recommended in a session [6], [7].

Mobility - Social networks - Recommender system

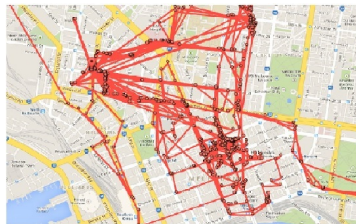
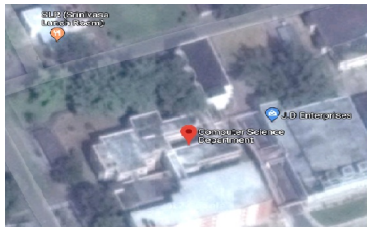
- Mobility is a basic need for a human being
- It has a very high degree of freedom and variation
- Analysis of the data extracted from movement of individuals over a period of time
- Various spatio-temporal components involved in human mobility analysis are: locations visited, distance travelled, number of locations, frequency of visits, time of visit, etc.
- Requires a large volume of spatio-temporal mobility data
- One important source is the Online social networks

Online Social Networks

- Internet usage has grown rapidly since last three decades
- Users of internet share enormous volume of information through blogs, forums, digital images/videos and various social media
- Social networks have the largest user generated content
- Sharing real-time experiences on social network is the current trend
- Smart devices have enabled sharing information ubiquitously
- Analysis of this data has many folds advantages

Social Networks + Locations

- ① Adds a new dimension to the social networks
 - Geo-tagged user generated media: texts, photos, and videos
- ② Bridges the gap between the virtual world and physical world
 - Sharing real-world experience.
 - Consume online information in physical world.



Point location and Trajectory.

Virtual world

Sharing &
UnderstandingInteractions
↕Generating &
Consuming

Physical world



Virtual world and physical world correlation.

Location-based Social Networks (LBSN)

- Share location-embedded content
- Connect with friends and nearby users
- Share multimedia content such as images and videos
- Perform review and rating at the checked-in POIs
- **Check-in:** A facility to share time and location publicly

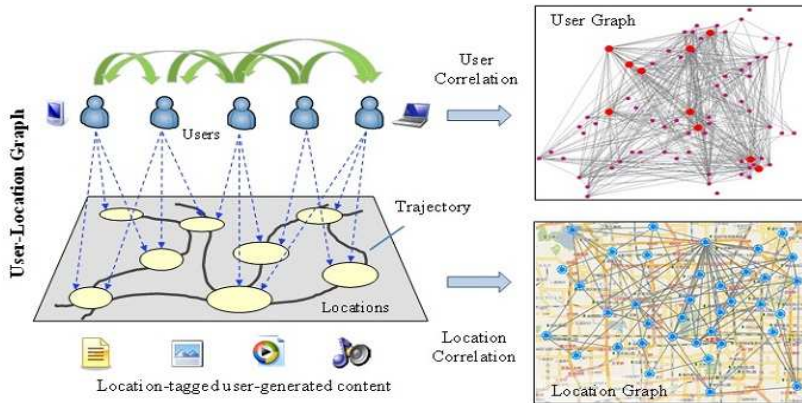
Popular LBSNs

- Foursquare (Swarm: dedicated mobile app for check-in)
- Gowalla (acquired by Facebook in 2011)
- Google Map (mobile application, formerly Google Latitude)
- Microsoft Geolife
- Yelp



Examples of various Location-based Social Networks.

LBSN Information layout



Information layout of a Location-based Social Network.

Research in LBSNs

- **POI or Location recommendation**
- **Friend recommendation**
- Event recommendation
- Content recommendation

Item vs POI recommendation

- Personalization is the primary factor in any recommendation
- Following are explored in both item and POI recommendations
 - i. Collaborative filtering
 - ii. Liking of similar users
 - iii. Implicit rating at features
 - iv. Popularity, etc.

New in POI Recommendation

- Geographical influence
- Proximal locations
- Selecting the expert users is challenging
- Friend-based collaborative filtering

General approaches

- **Memory-based approaches:**
 - i. POI-based collaborative filtering
 - ii. User-based collaborative filtering
- **Model-based approaches:**
 - i. Matrix factorization
- **Hybrid approaches:**
 - i. Reviews
 - ii. Social influences
 - iii. Geographical influences

M. Ye et. al.[8]: Location Recommendation for Location-based Social Networks:

Friend-based Collaborative Filtering (FCF)

Compute social friends by applying Jaccard similarity between a pair of users.

M. Ye et. al.[8]: Location Recommendation for Location-based Social Networks:

Friend-based Collaborative Filtering (FCF)

Compute social friends by applying Jaccard similarity between a pair of users.

Apply User based CF for finding rating on non-visited location.

$$\hat{r}_{ij} = \frac{\sum_{u_k \in U_i} r_{kj} \times w_{ik}}{\sum_{u_k}$$

U'_i = Set of friends of user i ; w_{ik} = similarity between friend u_k and u_i

Geo-measured friend-based collaborative filtering [8]

- Nearby friends tend to share common visited locations
- Similarity between friends can be weighted by distance
- It uses linear regression method upon power-law distribution of distances between friends to learn a friend similarity model
- $y = \alpha x^\beta$; $\log_{10} y = w_0 + w_1 \log_{10} x$

$$w_{i,k} = \frac{y(x = d(u_i, u_k), \alpha, \beta)}{\sum_{u_k \in U_i} y(x = d(u_i, u_k), \alpha, \beta)}$$

$w_{i,k}$ = similarity between users u_i and u_k

y = variable of common location ratio

x = variable of distance between friends

Friend-based collaborative filtering [9]

- Various platforms provide option to make friendship among its fellow users
- Preferences of such virtual friends can be exploited for POI recommendation
- In user-based CF, rating similarity of friends are considered
- Friend-based CF exploits how the active user is influenced by the mobility of its virtual friends

Social influence [9]

$$SI_{u_i, u_z} = \sigma * \left| \frac{F_i \cap F_z}{F_i \cup F_z} \right| + (1 - \sigma) * \left| \frac{L_i \cap L_z}{L_i \cup L_z} \right|$$

SI_{u_i, u_z} = social influence of u_i over u_z

F_i, F_z = friends of u_i and u_z

L_i, L_z = checked-in locations for u_i and u_z

σ = turning factor in the range $[0,1]$

Friend-based collaborative filtering [9]

$$\text{FBCF}_{u_i, l_j} = \frac{\sum_{z \in U} (SI_{u_i, u_z} * T_{u_z, l_j})}{\sum_{z \in U} SI_{u_i, u_z}}$$

SI_{u_i, u_z} = cosine similarity between users u_i and u_z

$T_{u_z, l_j} = 1$ if u_z has checked-in at l_j , else 0

Location-based collaborative filtering [10]

- Assumes that users visit similar locations
- Location profiles of users are exploited

$$\text{score}(u, \ell) = \frac{\sum_{\ell' \in L} \cos(\mathbf{p}_\ell, \mathbf{p}_{\ell'}) \cdot w_{u, \ell'}}{\sum_{\ell' \in L} \cos(\mathbf{p}_\ell, \mathbf{p}_{\ell'})}$$

$$\mathbf{p}_\ell = (w_{u_1, \ell}, w_{u_2, \ell}, \dots, w_{u_{|U|}, \ell})$$

$$w_{u_i, \ell} = \text{freq}(i, \ell) / \sum_j \text{freq}(j, \ell)$$

Cold-start problem

- Personalization depends upon historical data
- A new user does not poses historical data
- Such scenarios are called Cold starts
- Types of cold-starts
 - i. User cold-start
 - ii. POI cold-start

Problems in POI recommendation

- **User cold-start:**

- i. Users having no historical data
- ii. Difficult to find their preferences
- iii. Personalized recommendation is difficult

- **POI cold-start:**

- i. POI having no historical data
- ii. Difficult to find the good features
- iii. Never selected for personalized recommendation

Problems in POI recommendation

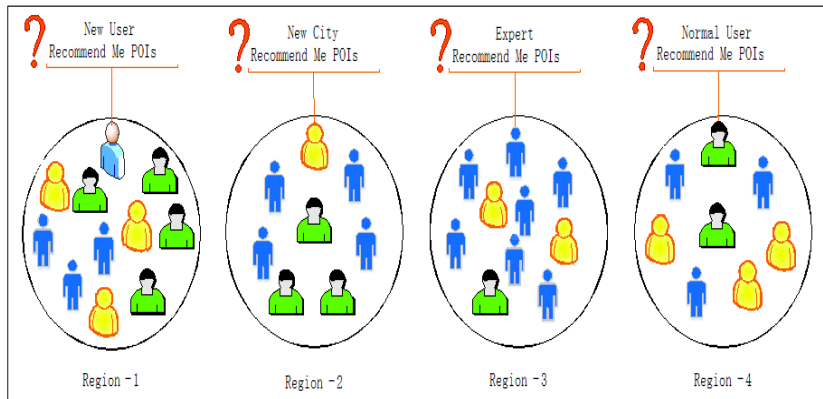
- **New city problem:**

- i. Users having no historical data
- ii. Difficult to find their preferences
- iii. Personalized recommendation is difficult

Problems in POI recommendation

- **POI long tail:**
 - i. Less number of categories among POIs
 - ii. More number of POIs for each category
 - iii. Unpopular POIs are never recommended
- **Selecting experts:**
 - i. Influence within a geographical area
 - ii. Rating at implicit features and number of friends
 - iii. Both these two factors are not considered during item recommendation
 - iv. POI Recommendation to an expert

POI recommendation scenarios

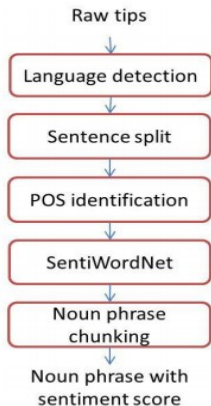


Different scenarios for POI Recommendations.

Exploring reviews [12]

- Language detection component filters out non-English tips
- Reviews are split into sentences and the part-of-speech
- Adjectives, nouns, verbs, etc. are identified
- Sentiment score for each word is obtained from SentiWordNet [11]
- Noun phrase chunking technique is used to extract phrases
- Overall sentiment score for each word is computed by sum of all the sentiment scores of each word in a review
- Positive and negative phrases are identified based on the sentiment scores

Exploring reviews [12]



User u at Venue v ($ss = 0.3$)

Good place in center New York, I went there last Sunday night and had great spaghetti with reasonable price. But I had a very long waiting time , almost one hour just for appetizer !!!



Great spaghetti
Reasonable price
Good place



Center New York
Last Sunday night
Appetizer



Long waiting
time

Sentiment analysis from reviews [12].

Modelling User Preferences [13, 14]

- Matrix factorization is the standard approach
- User (U) \times POI (I) rating matrix Q

	i1	i2	i3	i4	i5
u1		5	2	1	
u2	3		1		4
u3		2	4	5	1
u4	2	2	1		1
u5	2		1	5	4

- Unobserved ratings are denoted in green

Modelling User Preferences [13, 14]

- User (U) \times Feature (F) rating matrix M

	f1	f2	f3	f4	f5
u1		5	4	3	3
u2	1	1	5		4
u3	1		3	3	1
u4	5	5	4	4	
u5	5		4	2	1

- Features are extracted from reviews given by a user at all POIs
- Aggregate rating is used at each entry
- Unobserved ratings are denoted in green.

Modelling User Preferences [13, 14]

- Matrix factorization is performed on M

	f1	f2	f3	f4	f5
u1	2	5	4	3	3
u2	1	1	5	4	4
u3	1	1	3	3	1
u4	5	5	4	4	4
u5	5	3	4	2	1

- Features with high ratings are considered
- Top-W features for each user are identified
- User preferences over features are identified

Modelling User Preferences [14]

- Feature (F) \times POI (I) rating matrix N

	f1	f2	f3	f4	f5
u1	2	5	4	3	3
u2	1	1	5	4	4
u3	1	1	3	3	1
u4	5	5	4	4	4
u5	5	3	4	2	1

- Features are extracted from reviews given by all users at a POI
- Aggregate rating is used at each entry
- Top- V features are identified for each POI.

Recommendation to cold-start users

$$ER_p^u = \frac{1}{|E|} \sum_{e \in E} r_{ep}$$

ER_p^u = estimated rating ER on POI p of region g by new user u

E = all experts of a region g

r_{ep} = rating of an expert e of region g on POI p

Recommendation to user in new city

- Select top- I influential users (experts)
- Rank the top- I users based on similarity
- Estimated rating is computed as,

$$ER_p^u = \frac{1}{|I|} \left[r_{i_1 p} + \sum_{j=2}^d \frac{r_{ij} p}{\log_2(j)} \right]$$

where,

I = socially influential users

$r_{i_1 p}$ = rating given to POI p by users i_1

d = number of users in I

Recommendation to expert user

$$ER_p^u = \bar{r}_p + \frac{\sum_{x \in X} \text{sim}(p, p_x) * (r_{ux} - \bar{r}_x)}{\sum_{x \in X} |\text{sim}(p, p_x)|}$$

\bar{r}_p = average rating at POI p

X = set of all POIs rated by target user u

$\text{sim}(p_1, p_x)$ = similarity score between POIs p and p_x

r_{ux} = rating at POI p_x by user u

Recommending top-K POIs

- Distance of user from POIs is considered
- Score is computed using rating and distance

$$Score_{p_1}^{u_1} = \frac{ER_{p_1}^{u_1}}{\text{dist}(u_1, p_1)}$$

- Haversine formula is used to compute distance between user u_1 and POI p_1
- POIs with high scores are selected for recommendation

Recommending top-K POIs

1. Google Local Reviews [15]
2. YFCC100m multimedia dataset [16]
3. Gowalla social network [17]
4. Foursquare social network [18]
5. Brightkite social network [17]
6. Mobile Data Challenge GPS data [19]
7. T-drive GPS data [20]
8. Geolife trajectory dataset [21]
9. Yelp social network [22]
10. Tripadvisor dataset [23]

Recommending top-K POIs

- **Recommender system**

1. Precision
2. Recall
3. F-measure

- **Ranking recommended POIs**

1. Mean reciprocal rank
2. Discounted cumulative gain
3. Normalized discounted cumulative gain

Conclusion and Future directions

- Time factor can be incorporated in POI recommendations
- Perform a subjective analysis on how the cold-start POIs of a city are perceived by the cold-start users of the city
- Utilize the neighboring non cold-start point-of-interests for estimating the probability of the target user to visit the cold-start point-of-interest
- Recommending nearby events based on user interests

References

- [1] Daniel Lemire and Anna Maclachlan.
Slope one predictors for online rating-based collaborative filtering.
In Proceedings of the International Conference on Data Mining, pages 471–475.
SIAM, 2005.
- [2] Yehuda Koren, Robert Bell, and Chris Volinsky.
Matrix factorization techniques for recommender systems.
Computer, 42(8):30–37, 2009.
- [3] Paolo Massa and Bobby Bhattacharjee.
Using trust in recommender systems: an experimental analysis.
In Proceedings of the International Conference on Trust Management, pages 221–235. Springer, 2004.
- [4] Jian Wei, Jianhua He, Kai Chen, Yi Zhou, and Zuoyin Tang.
Collaborative filtering and deep learning based recommendation system for cold start items.
Expert Systems with Applications, 69:29 – 39, 2017.

References

- [5] Guibing Guo, Jie Zhang, and Daniel Thalmann.
Merging trust in collaborative filtering to alleviate data sparsity and cold start.
Knowledge-Based Systems, 57:57–68, 2014.
- [6] Balázs Hidasi, Massimo Quadrana, Alexandros Karatzoglou, and Domonkos Tikk.
Parallel recurrent neural network architectures for feature-rich session-based recommendations.
In Proceedings of the 10th ACM conference on recommender systems, pages 241–248, 2016.
- [7] Balázs Hidasi and Alexandros Karatzoglou.
Recurrent neural networks with top-k gains for session-based recommendations.
In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pages 843–852, 2018.
- [8] Mao Ye, Peifeng Yin, and Wang-Chien Lee.
Location recommendation for location-based social networks.
In SIGSPATIAL International Conference on Advances in Geographic Information Systems, pages 458–461, 2010.

References

- [9] Mao Ye, Peifeng Yin, Wang-Chien Lee, and Dik-Lun Lee.
Exploiting geographical influence for collaborative point-of-interest recommendation.
In Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval, pages 325–334, 2011.
- [10] Hao Wang, Manolis Terrovitis, and Nikos Mamoulis.
Location recommendation in location-based social networks using user check-in data.
In Proceedings of the 21st ACM SIGSPATIAL international conference on advances in geographic information systems, pages 374–383, 2013.
- [11] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani.
Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining.
In Lrec, volume 10, pages 2200–2204, 2010.
- [12] Dingqi Yang, Daqing Zhang, Zhiyong Yu, and Zhu Wang.
A sentiment-enhanced personalized location recommendation system.
In ACM Conference on Hypertext and Social Media, pages 119–128, 2013.

References

- [13] Betim Berjani and Thorsten Strufe.
A recommendation system for spots in location-based online social networks.
In Workshop on social network systems, pages 1–6, 2011.
- [14] Pramit Mazumdar, Bidyut Kr Patra, and Korra Sathya Babu.
Handling cold-start scenarios in point-of-interest recommendations through crowdsourcing.
In Proceedings of the ACM India Joint International Conference on Data Science and Management of Data, pages 322–324, 2018.
- [15] Julian McAuley.
Recommender Systems Datasets, 2020 (accessed August 20, 2020).
- [16] Bart Thomee, David A. Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li.
Yfcc100m.
Communications of the ACM, 59(2):64–73, 2016.

References

- [17] Eunjoon Cho, Seth A Myers, and Jure Leskovec.
Friendship and mobility: user movement in location-based social networks.
In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 1082–1090, 2011.
- [18] Dingqi Yang, Daqing Zhang, and Bingqing Qu.
Participatory cultural mapping based on collective behavior data in location-based social networks.
ACM Transactions on Intelligent Systems and Technology (TIST), 7(3):1–23, 2016.
- [19] Niko Kiukkonen, Jan Blom, Olivier Dousse, Daniel Gatica-Perez, and Juha Laurila.
Towards rich mobile phone datasets: Lausanne data collection campaign.
International Conference on Pervasive Services, 68, 2010.
- [20] Jing Yuan, Yu Zheng, Xing Xie, and Guangzhong Sun.
Driving with knowledge from the physical world.
In ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 316–324, 2011.

References

- [21] Yu Zheng, Hao Fu, Xing Xie, Wei-Ying Ma, and Quannan Li.
Geolife GPS trajectory dataset - User Guide, July 2011.
- [22] Yelp.
Yelp Open Dataset, 2020 (accessed August 20, 2020).
- [23] Hongning Wang, Yue Lu, and ChengXiang Zhai.
Latent aspect rating analysis without aspect keyword supervision.
In ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 618–626, 2011.

Acknowledgement

- Dr. Pramit Mazumdar, Post-doctoral Fellow, Università degli Studi Roma TRE, Rome, Italy. [Ph.D., 2018]
- Dr. Rama Syamala Sreepada, Post-doctoral Fellow, University of British Columbia, Canada. [Ph.D., 2020]

Thank You.

E-mail: patrabk@nitrkl.ac.in; bidyut76@gmail.com

`https:`

`//sites.google.com/site/patrabidyutkr/`