End-term Project Report: Personalized Location Recommendation by Aggregating Multiple Recommenders in Diversity

Team Name: nogen Team Members: 22M1431,22M1432

Abstract

This project report contains the details of our project, which aims to improve location-based recommenders by making them personalized. We tried to combine different recommenders, which recommend locations based on different principles. We combine them based on which recommender is good for the particular user. In particular, in this report, we explain the ML method used for the project, the details of the training procedure, and details of inference and discuss about experiments conducted. We have tried a new LURA ML method, and superior results are presented using the proposed method.

1 Introduction

Location recommendation is a significant aspect of social networking applications and services that rely on location. It provides a better user experience and contributes to the business.

When it comes to evaluating user preferences for a specific item or location, there are two primary methods - explicit ratings and implicit check-in records. While explicit ratings clearly indicate whether a user likes or dislikes an item, implicit check-in records can be a bit more complex. Location records are implicit. One of the unique characteristics of check-in records is that they do not provide any negative feedback. Users only check in when they visit a location, and they do not have the option to indicate any negative experiences.

Most existing methods take unified perspectives toward the recommendation problem; they are based on the theory one model fits all. Collaborative recommender systems try to predict the utility of items for a particular user based on the items previously rated by other similar users. For example, in a movie recommendation application, in order to recommend movies to user u, the collaborative recommender system tries to find the “peers” of user u, i.e., other users that have similar tastes in movies. As a result, only the films that are highly favored by the "peers" of user C would be suggested.

Location-based algorithms often prioritize popularity and similarity measures, which may overlook lesser-known or less similar options that could better match individual users' preferences. However, analysis of real-world location-based social networks has demonstrated that users' decisions on where to visit depend on a various factors unique to the individual. As a result, it is crucial to provide personalized recommendations that cater to the specific interests and preferences of each user.

Personalized recommendations are essential for location-based services because users expect recommendations that reflect their unique interests and behavioral patterns. A one-size-fits-all approach does not work for all users, as their preferences and interests can vary significantly. Therefore, location-based services that provide personalized and diverse recommendations can gain a competitive advantage in the market by satisfying the needs of individual users.

This can be done of using different recommenders and combining them, which will give personalized recommendations to user.

We provide a survey of existing literature in Section 2. Our proposal for the project is described in Section 3. We give details on experiments in Section 5. A description of future work is given in Section 7. We conclude with a short summary and pointers to forthcoming work in Section 8.

2 Literature Survey

G. Adomavicius and A. Tuzhilin's paper, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," provides an in-depth analysis of the various techniques used in recommender systems, including content-based filtering, collaborative filtering, knowledge-based recommender systems, and hybrid recommender systems.[2]

One of the key takeaways from the article is the challenges faced by these systems, such as the cold start problem, sparsity problem, and scalability problem. These challenges are particularly pronounced in the context of large-scale systems that need to cater to a diverse range of users with varying preferences and interests.

To address these challenges, the authors suggest possible extensions to recommender systems, such as context-aware, social, and privacy-preserving recommendations. Context-aware recommendations, for instance, consider the user's current context, such as their location, time of day, and recent activities, to provide more relevant recommendations. Social recommendations, on the other hand, leverage the user's social network to give recommendations.

Our project draws inspiration from a closely related work by Ziyu Lu et al. [1]. The paper titled "Personalized Location Recommendation by Aggregating Multiple Recommenders in Diversity" presents a comprehensive literature survey on personalized location recommendation systems.

The article discusses the challenges and limitations of traditional location-based recommendation systems and highlights the importance of incorporating personalized recommendations that cater to individual users' preferences and interests. It proposes a novel approach that aggregates multiple recommenders in diversity to provide more accurate and diverse recommendations

The authors took eleven recommenders-

1) **R1**: user-based cf (UCF). W(u,v) could be the cosine similarity between users u and v, i.e., Wu,v = cos (cu, cv), 2) **R2**: friend-based CF (FCF)- The similarity between users could be reflected by their common friends. In this Jaccard index is used. 3) **R3**: friend-location cf (FLCF)- It takes into account both the common friends and commonly visited locations of the two people. It also measures the closeness of two friends. 4) **R4**: geo-distance CF (GCF). The idea behind GCF is that friends who are geographically closer to us tend to influence us more than those who are farther away. 5) **R5**: category cf (CCF)- Considers users as keywords and location categories as documents; between user u and category C there can be a relevance score, rel(u, C). 6) **R6**: Item-based CF(ICF)-similar to UCF, weight is cosine similarity between two locations.7) **R7**: time-weighted CF (TCF)- recent check-in records are relatively more reliable (high weight) if we consider any evolution of user preference.8) **R8**: power-law model (PLM)-user preferences follow a power law distribution, which means that a user highly prefers a small number of locations, while the majority of locations are less preferred.9) **R9**: kernel density model (KDM)- For a user u with past check-ins at locations lu = { l1, l2, • • • , ln }, this factor estimates the probability of u visiting a new location using kernel techniques..10) **R10**: spatial kernel density model (SKDM)- This is author’s improvement over R9. They consider distances between latitudes and longitudes.11) **R11**: implicit matrix factorization (IMF): modification of the conventional MF for implicit user feedback.

They created a common framework (**LURA**) to combine the recommenders based on the user to provide user-personalized recommendations.

In R9 recommender, iGSLR algorithm is used[3]. It was used by Zhang et al., The iGSLR algorithm uses kernel density estimation to model the probability distribution of a user's check-in history and then recommends locations based on the density of other users' check-ins in the same area. The algorithm takes into account the user's personal preferences as well as the popularity and geographical proximity of the locations being recommended. The results of the study suggest that iGSLR has the potential to be an effective tool for location-based marketing and personalized advertising.

3 Methods and Approaches

This part of project report contains the details of our project method, which aims to give recommendations of locations for a user. We will discuss the machine learning method used in the project, as well as the specifics of the training process and inference. Additionally, we will provide details on the experiments conducted. Our approach employs a machine learning technique and the outcomes are demonstrated utilizing this proposed methodology.

Method:

ALGORITHM 1 LURA

The name “LURA” comes from its execution cycle of Learn-Update-Recommend-Aggregate.

Input:

- G^t = Data of the user till the t^{th} (current) day;
- $R^{t-\Delta t}$ - a set of n recommenders, trained using $G^{t-\Delta t}$
- u - a user

Output: N recommended locations for user u at time t.

At time $t - \Delta t$ each recommender $R_i^{t-\Delta t}$ has recommended N locations to u.

At time $t - \Delta t$ each recommender $R_i^{t-\Delta t}$ has recommended N locations to u.

Figure 1 LURA Framework

Here we need to maximize the likelihood, which is the probability of observing P_u^t given preference α_u^t. Assuming that pairs (l, l') ∈ P_u^t are independent, then

$$\Pr\left\{P_u^t \,\middle|\, \boldsymbol{\alpha}_u^t\right\} = \prod_{(\ell,\ell')\in P_u^t} \Pr\left\{\ell >_u \ell' \,\middle|\, \boldsymbol{\alpha}_u^t\right\}$$

Computing Pr {l $>_u$ l' |α_u} is nontrivial, but intuition is that this probability should be proportional to the difference between scores s(u, l; α_u) and s(u, l' ; α_u)

$$\Pr\left\{\ell >_u \ell' \left|\boldsymbol{\alpha}_u\right.\right\} = \sigma\left(d_u^{\ell,\ell'}\left(\boldsymbol{\alpha}_u\right)\right)$$

Where σ is the logistic function that can generate a probability distribution in the range of [0,1], $\sigma(x) = 1/(1+e^{-x})$, and $d_u^{l,l'}(\alpha_u) = s(u, l; \alpha_u) - s(u, l' ; \alpha_u)$.

$$\ln\left(\Pr\left\{P_u^t\left|\boldsymbol{\alpha}_u\right.\right\}\right) = \sum_{(\ell,\ell')\in P_u^t} \ln\left(\sigma\left(d_u^{\ell,\ell'}\left(\boldsymbol{\alpha}_u\right)\right)\right)$$

Algorithm 2 Stochastic Gradient Descent

Input: Let $P_u^t = L_+^t \times L_-^t$; $\alpha_u^{t-\Delta t}$; number of iterations M; number of samples K

Output: updated user preference α_u^t

1: for j = 1, 2, · · · , M

2: set $\alpha^{(j)}$ to $\alpha_u^{t-\Delta t}$

3: Draw K sample pairs from the training space P_u^t

4: for each sample pair p = (l, l') do the following

5: update $\alpha^{(j)}$ with p and $\alpha^{(j)}$ using gradient descent)

6: end of the loop

7: set α_u^t as final updated value

$$\boldsymbol{\alpha}^{(j)} \leftarrow (1-\gamma)\cdot\boldsymbol{\alpha}^{(j)} + \gamma\cdot\tau\cdot\left(\frac{\nabla_{\boldsymbol{\alpha}^{(j)}} d_u^{\ell,\ell'}\left(\boldsymbol{\alpha}^{(j)}\right)}{1+e^{d_u^{\ell,\ell'}}}\right)$$

$$\nabla_{\boldsymbol{\alpha}^{(j)}} d_u^{\ell,\ell'} = \begin{bmatrix} \varphi\left(R_1^{t-\Delta t}(u,\ell)\right) - \varphi\left(R_1^{t-\Delta t}(u,\ell')\right) \\ \varphi\left(R_2^{t-\Delta t}(u,\ell)\right) - \varphi\left(R_2^{t-\Delta t}(u,\ell')\right) \\ \vdots \\ \varphi\left(R_n^{t-\Delta t}(u,\ell)\right) - \varphi\left(R_n^{t-\Delta t}(u,\ell')\right) \end{bmatrix}$$

Following are sampling methods and aggregation methods that can be used for good and more accurate results.

Random sampling (RS): Locations in L_-^t are selected uniformly at random.

Static sampling (SS): This strategy favors popular locations, i.e., locations with many visitors have higher chances of selected.

Adaptive sampling (AS): This strategy gives higher chances to those locations with higher scores (i.e., locations considered as promising by the recommender).

Recommendation Aggregation:

Score-based aggregation (SA): This strategy is to use the scaled score of the user-location pair (u, l) estimated by each recommender (at time t).

$$\varphi\left(R_i^t(u,\ell)\right) = \frac{R_i^t(u,\ell)}{\max_{\ell' \in L_-^t} R_i^t(u,\ell')}, \quad i = 1, 2, \cdots, n.$$

Rank-based aggregation (RA). This strategy considers the ranked position of a location. Given a ranked list of N locations, l_1, l_2, · · · , l_N , this strategy assigns higher scores to top locations.

$$\varphi\left(R_i^t(u,\ell)\right) = 1 - \frac{1}{N}\left(\operatorname{rank}_i(u,\ell) - 1\right), \quad i = 1, 2, \cdots, n,$$

3.1 Work done before mid-term project review

Data collection and preprocessing were completed, followed by the search for recommenders to combine. A literature survey on recommenders was conducted, and works done on combining recommenders were reviewed in the literature.

3.2 Work done after mid-term project review

We filtered the data for the top hundred users and extracted information for those who had at least 10 check-in records, focusing on locations that were visited at least twice. Following this, we divided the data into training and testing sets based on the dates recorded.

Next, we implemented the code for combining two recommenders as mentioned in the research article that utilized the LURA framework. By combining the two recommenders and utilizing monthly data, we were able to obtain accurate recommendations. Based on the correct recommendations generated, we assigned weights to Recommender 1 and Recommender 2 for each specific user.

This methodology is significant as it provides precise results when compared with the individual recommenders.

4 Data set Details

We took Gowalla dataset from SNAP: Network datasets: Gowalla (stanford.edu).It is .txt file. It contains five columns viz. user, timestamp, latitude, longitude, location-id. It contains total 6,442,890 records. It also contains the friendship network which is undirected and was collected using their public API, and consists of 196,591 nodes and 950,327 edges. It was collected between over the period of Feb. 2009 - Oct. 2010.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 192855 entries, 237 to 5049476
Data columns (total 5 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   user         192855 non-null  int64
 1   timestamp    192855 non-null  object
 2   latitude     192855 non-null  float64
 3   longitude    192855 non-null  float64
 4   location_id  192855 non-null  int64
dtypes: float64(2), int64(2), object(1)
memory usage: 8.8+ MB
```

5 Experiments

Data Processing: We filtered data for the top hundred users and again in that data we took data for the users having user-id less than 100. We took data for the users having at least 10 check-in records and took locations which are at least visited twice. We divided the data into training and testing sets based on date. We set training start date as 01-11-2019, training end date as 31-05-2010, testing start date as 01-06-2010, testing end date as 31-10-2010.

We have used two recommenders. Recommender one is UCF based, which suggests locations based on cosine similarity between users. Recommender2 is geo-distance based. The basic principle is those who are near to us affects more than at long distance. We initialized both the weights as 0.5 and set the learning rate as 0.01. With the help of correctly predicted locations by the recommenders we updated the weights. From the recently obtained weights, we took some locations from recommender 1 and some from recommender 2. We checked for the number of correctly predicted locations.

6 Results

We calculated weights for recommenders for every user. We calculated weights month by month, and based on correct predictions we increased weights. We coded for two recommenders those are recomender1 which is UCF based, and Recommender2 which based on geo-distance. We used the recommendations obtained from the two recommenders which are shown below. With the help of those recommenders and actual places visited by that particular user we calculated weights for R1 and R2.

```
[29] aa=recommender_2Sp(10,data_train1)

[30] aa

     {22: [72530, 21134, 32159, 17208, 32282, 15655, 65576, 17716, 10475, 10416],
```

Figure 2 Location recommendations for user 22 by Geo-distance based CF Recommender

```
[34] bb=recommender_UCF(22,data_train1)

[35] bb

     [17208, 27376, 18680, 12595, 19409, 9568, 33025, 14483, 8964, 18925]
```

Figure 3 Location recommendations for user 22 by User based CF Recommender

```
user 401 is date 2009-12-01 00:00:00 jj
actual places visited
{1183749, 86022, 702472, 681993, 106508, 215056, 671762, 10259, 690196, 65565, 1148958, 49183, 34848, 1290272, 565283,
tpr1=0 Accuracy= 0.0 tp_r2= 0 accuracy-0.0 for user401 LL
wtr1=0.5 wtr2=0.5 for user401,user 401 is date 2010-01-01 00:00:00 jj
actual places visited
{1183749, 702472, 681993, 106508, 671762, 10259, 690196, 65565, 1148958, 49183, 34848, 1290272, 565283, 10279, 2482216,
tpr1=0 Accuracy= 0.0 tp_r2= 1 accuracy-0.1 for user401 LL
```

Figure 4 Example of user 401 whose updated during training

```
r1 4 and r2 6 and user22
precision is 0.3 for user 2
r1 0 and r2 10 and user401
precision is 0.0 for user 4
r1 4 and r2 6 and user472
precision is 0.0 for user 4
r1 3 and r2 7 and user491
precision is 0.0 for user 4
```

Figure 5 Results on test data (eg, for user 22: 4 locations from R1 and 6 from R2 were taken)

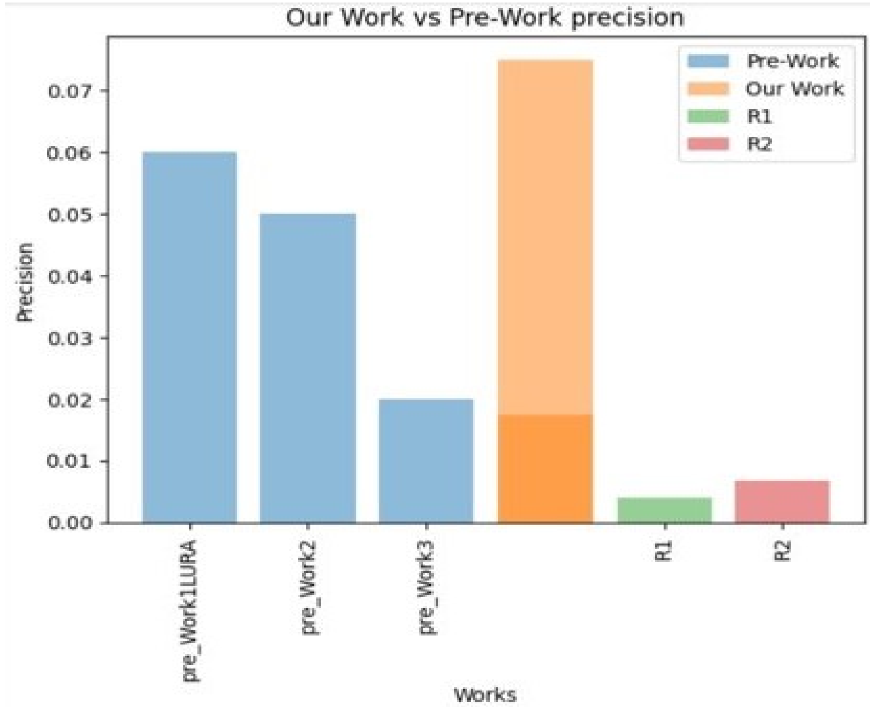

Figure 6 Precision

Fig.6 shows results for our precision and comparison with previous work. Here we can clearly see that precision is increases when we combine recommenders according user specific weights as compared to single recommenders. Also one thing to notice here that our precision is 0.075 which is greater than LURA authors work. Here we considered those users to whom R1 or R2 recommended at least one correct location during training phase. So we can say that for those users combing these two recommenders is good choice, but for other users we need to find other recommenders.

Similar results were obtained for recall, which is shown in fig.7 below.

Figure 7 Recall

7 Future Work

User feedback is an important source of information that can be used to improve the accuracy and relevance of recommendation results. We can also use **tags** for locations like café, sports, hiking, restaurants, we can ask user what type of location type you want to visit, and based on tag we recommend better recommendations.

8 Conclusion

As weights are calculated for each user for each recommender it becomes personalized. We used Lura framework and got significant results by combined the results. Even though our work's precision may look good than previous works we should know that we get higher precision when we take users whose weights are moved from initial values during training. So we can say that for those users combing these two recommenders is good choice, but for other users we need to find other recommenders. Also we filtered out many users data during preprocessing and took only top users. So saying our work is superior than previous is completely wrong. We recommend adding other recommenders also in it while calculating weights, so that precision will increase. Finally, we can say that by combining recommenders, we can increase precision of recommendations and make them personalized. We recommend that use good recommenders to combine, and do not use all recommenders based on same principle.

References

[1] Lu, Ziyu and Wang, Hao and Mamoulis, Nikos and Tu, Wenting and Cheung, David W. Personalized Location Recommendation by Aggregating Multiple Recommenders in Diversity. Kluwer Academic Publishers, oi = {10.1007/s10707-017-0298-x}, 2017

[2] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. TKDE, 17(6):734–749, 2005.

[3] J.-D. Zhang and C.-Y. Chow. iGSLR: Personalized geo-social location recommendation - a kernel density estimation approach. In SIGSPATIAL GIS, 2013

[4] Freund, Y., Iyer, R., Schapire, R. E., & Singer, Y. (2003). An efficient boosting algorithm for combining preferences. Journal of machine learning research, 4(Nov), 933-969

[5] Konstas, I., Stathopoulos, V., & Jose, J. M. (2009, July). On social networks and collaborative recommendation. In Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval (pp. 195-202).

[6] Cho, E., Myers, S. A., & Leskovec, J. (2011, August). 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 (pp. 1082-1090).